IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2022, accepted 18 November 2022, date of publication 24 November 2022,
date of current version 30 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3224447

APPLIED RESEARCH

Maturing the Scrum Framework for Software Projects Portfolio Management: A Case Study-Oriented Methodology

SAMIRA VAZIFEH-NOSHAFAGH[1], **VAHID HAJIPOUR**[1,2], **SAJJAD JALALI**[1], **DEBORA DI CAPRIO**[3], **AND FRANCISCO JAVIER SANTOS-ARTEAGA**[4]

[1]Research Center, FANAP Company, Tehran 1654120840, Iran
[2]Department of Industrial Engineering, Faculty of Engineering and Architecture, Altinbas University, 34128 İstanbul, Turkey
[3]Department of Economics and Management, University of Trento, 38122 Trento, Italy
[4]Departamento de Análisis Económico y Economía Cuantitativa, Universidad Complutense de Madrid, 28223 Pozuelo de Alarcón, Spain

Corresponding author: Debora Di Caprio (debora.dicaprio@unitn.it)

The work of Debora Di Caprio was supported by the University of Trento (NEOASSUNTI 40202312). The work of Santos-Arteaga was supported by the María Zambrano Contract of the Universidad Complutense de Madrid financed by the Ministerio de Universidades with Funding from the European Union Next Generation Program.

ABSTRACT In the modern era, information technology-based solution providers are encountering a growing request for satisfying the versatile requirements of their customers in terms of software applications. To this end, specific approaches have been designed to streamline the way of accomplishing software projects in an efficient manner, i.e., agile-oriented frameworks. Even though previous studies have highlighted variations of such a framework, the literature has not addressed the adaptations required in response to the gradual maturity of a wide-ranging case study dealing with software applications. Following a case study-oriented methodology, this paper focuses on elaborating a set of workable maneuvers to mature the Scrum framework when applied to portfolio management. Particularly, we highlight how Scrum should be adapted from its basic setting to a vision and goal-oriented configuration or Scrumban under certain conditions. As a maturing practice, we propose a heuristic scoring technique to determine the sprint length of subprojects with different characteristics in the context of a portfolio. The study also introduces a multi-level refinement structure to enhance the monitoring of the teams' performance under the proposed mature framework. The results obtained display a considerable spike in the realization rate of release planning in light of the actual performance.

INDEX TERMS Agile methodology, project management maturity, smart cities, sprint length, adaptive project portfolio management.

I. INTRODUCTION

The increasing trend of using software-based utilities beyond the distinguishing nature of the software development process has surged active participants to tailor the prevalent methodologies for managing such projects ([1]). The flexible domain of the software projects does not align with the rigid structure of the classical waterfall methodology. In the waterfall methodology, "the project is broken down into a sequence of tasks, with the highest-level grouping referred to as phases" (Chapter 18 of [2], pp. 449-492). Hence, a formal process takes place where the single phases comprise a list of detailed tasks accompanied by documentation and exit criteria. In contrast, the concept of agile methodology entails the application of an iterative and incremental approach for building up the desirable scope.

The agile methodology has been recently extended into several branches. The most popular extension can be attributed to the Scrum framework, which breakdowns the whole list of requirements into smaller batches. The list is entitled product backlog and is formed by product backlog items. Following special instructions, the product is

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyan Zhang.

incrementally developed without the necessity of sticking to a pre-defined set of sequences while respecting inevitable changes in customers' requirements. However, there is not a single variation of the Scrum framework that could guarantee the projects' successes under any circumstances and during all periods of the lifecycle.

As a project is matured, it must be synchronized with the ongoing condition. In other words, prescribing the accurate version of a framework like Scrum that fits well the projects' requirements at different stages of the product lifecycle is of paramount importance. This is exactly the topic of the current study, which deals with proposing applied practices for maturing not only a single software project but also a set of projects at the portfolio level.

This study is centered on a specific aspect of the broad concept of portfolio management, namely, framework-oriented maturity. Portfolio management encompasses a variety of aspects such as the selection of the subprojects and the related processes for managing and handling them. We focus on illustrating how the Scrum framework has been adapted to cover a growing number of subprojects with different characteristics in the context of the portfolio management of a real case study. Other facets such as the subprojects selection and relative functional processes are out of the scope of the current paper.

For the sake of completeness, we add that in the case study, the selection of the subprojects was performed based on the strategic plans. Also, the communications between subprojects, such as risk and cost management, were carried out based on the associated knowledge areas of the PMBOK framework. However, these aspects are peripheral to the main focus of this study and are not considered further.

Additionally, the meaning of "maturity" in this paper is different from that of agile maturity assessment. In a model of agile maturity assessment, there are some factors that measure the maturity of each unit (e.g., team/organization). Based on such an evaluation, the maturity level of each unit is determined, and the required maneuvers are prescribed for enhancing the corresponding preparedness level ([3]). In this article, the term "maturity" refers to how the underlying framework is matured and evolved. The present study does not aim to measure the capabilities of the units' agility towards releasing software. Its objective is to show the maturity and evolution of the Scrum framework of the proposed real case study within a given time interval.

The rest of the paper is structured as follows. Section II reviews the relevant literature in order to highlight the contribution of this paper compared to previously published studies. Section III describes the case study and related maturity framework in detail. Section IV summarizes the empirical results and implications that follow from implementing the maturity procedure. Section V discusses the advantages and managerial implications of the outputs of this study as well as its limitations. Possible extensions and future research directions are also highlighted. Section VI concludes.

II. LITERATURE REVIEW

This section is composed of three subsections. The first subsection provided the background for the agile-based terminology adopted through the paper. The second subsection reviews the previous related studies. The third subsection highlights the contribution of the proposed work.

A. AGILE-BASED TERMINOLOGY

This section concerns the terms "approach", "model", "method", "methodology", and "framework" within our study and the well-foundedness of their use based on the existing agile literature. In particular, it is worth to clarify the reasoning behind attributing methodology and framework to the agile and Scrum concepts, respectively.

Following [4] and based on the context, agile and its variations can be associated with approach, method, framework, methodology, practice, and similar terms.

Herein, agile is referred to as methodology since it is embedded in the body of project management methodology. Regarding this specific context and in the same vein as [5] and [6], the term agile is interpreted as the agile project management methodology or agile methodology. Furthermore, Reference [6] implies that methodology is used when it comes to focusing on the policies related to framing the elements of a framework. In this paper, the term agile is used exactly with the purpose to address the policy and procedure of managing a project with respect to the iterative concept.

On the other hand, Scrum is referred to as framework since its application to the current paper is very much in line with the definition of framework provided by [7]. According to this definition, Scrum is a framework as it addresses principles and guidelines for accomplishing the project. More precisely, quoting [8], "Scrum is not a prescriptive process; it doesn't describe what to do in every circumstance. Scrum is used for complex work in which it is impossible to predict everything that will occur. Accordingly, Scrum simply offers a framework and set of practices that keep everything visible." The community of Home of Scrum also has considered Scrum by the term framework: "Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems" ([9]).

Despite the above justification for adopting the terminology "agile methodology" and "Scrum framework", it remains the fact that both the academic and professional communities are not unanimous about how to exactly characterize agile and its byproducts. One can easily find multiple studies in the agile literature where the authors do not distinguish among "agile approach", "agile framework" and "agile methodology", or between "Scrum approach", "Scum framework" and "Scum methodology". Terms like "approach", "model", "method", "methodology", "framework", are mostly used as synonyms and often interchanged several times within a given paper/book ([10], [11], [12], [13], [14]).

B. RELEVANT STUDIES

Despite the recent attention of researchers to software development methodologies, the literature suffers from empirically exploring the maturity procedure of a specific agile methodology like Scrum. This is particularly the case when dealing with the modifications that should be enacted in response to managing a portfolio of information technology (IT) projects that are expanded over time ([15]).

Reference [16] studied the conformity of the agile project portfolio management with the standard one in the context of 30 organizations varying from 5 to 15 teams. Focusing on the telecommunications sector, they tried to provide a clear understanding of the special characteristics related to the implications of the project portfolio management in an agile manner. The software development methods of the surveyed companies were Scrum, Kanban, and Rational unified process. The evidence showed that the standard project portfolio management might not be applicable due to the high level of interdependencies between agile teams. Furthermore, for the sake of adjusting the traditional method, efficient remedies for enabling the indispensable features related to routines and structure of the agile methodology were pointed out.

By underlining the scarceness of works related to managing agile projects at the portfolio level, [17] enumerated the corresponding steps for tailoring the agile portfolio management. They included adjusting the organizational structure beyond content and procedure of the ongoing portfolio to the agile-based one throughout the project lifecycle. To better elaborate on the issues pertaining to the agile project management portfolio in practice, he also addressed the case studies of the previous studies.

The case study of [18] illustrated the difficulties to achieve an agile portfolio level for small and medium sized enterprises. The authors concluded that Ocuco Ltd could not successfully implement the scaled agile framework methodology at the portfolio level while effectively surmounting the team level. The empirical study proposed by [19] focused on triggering the processes enhancing collaboration with customers into the traditional portfolio management. They concentrated on the Kanban version of the agile in the project portfolio context. Reference [20] injected the agile practice, instead of the waterfall one, into the portfolio management of the research and development projects of a matured manufacturing company. The author highlighted the limitations arising in the sequencing problem of accomplishing the tasks, tracking the work items from team to portfolio levels, and so on.

Reference [21] also considered the transformation steps and challenges derived from dealing with the agile structure. Apart from building cross-functional teams and institutionalizing the common elements of the framework, significant attention was paid to the continuous integration of the different components defining the working scope. The results obtained suggested employing more than one agile methodology instead of having a common one for managing all the projects of the portfolio. The prime challenge of the Paf.com case study was to select the set of tasks covered in the releases triggered by the lack of visibility over the priorities of the business ([22]). The problem was resolved by designing a hybrid approach consisting of the project management body of knowledge and the tools of stage-gate models.

Reference [23] categorized the contributions of previous studies in the context of quantitative maturity models related to software engineering. Their study illustrated the increasing trend exhibited by project management maturity using agile development practices. Reference [24] considered critical factors to successfully mature the IT-based projects drawing upon processes, outcomes, and roles of project managers. The authors noted that the IT projects could not be successful by fulfilling either the expectations of a single stakeholder or focusing on outcomes while neglecting the resultant cost.

Reference [25] explored the use of Scrum framework in a telecommunications company using a survey-based questionnaire. Leveling the maturity degree of the Scrum framework from 1 to 5, a significant number of maneuvers were devised to maximize the accurate utilization of such a practice in the company. The authors only provided recommendations without describing the details. For instance, they mentioned the necessity of creating key performance indicators (KPIs) to evaluate the performance of Scrum teams. Reference [26] argued that since the popular software project management methodologies have been developed by West/European countries, they should evolve now focusing on their application to developing countries. The authors addressed the necessity of providing an evolutionary software project management methodology that considered the lack of professional human resources, presence of financial problems, and rigid organizational structures.

Reference [27] measured the maturity level of Malaysian IT companies. A grading instruction was adopted when considering small and medium-sized companies to justify whether the maturity level was at a satisfactory status. Reference [28] adapted the Scrum framework to carry out research-type projects rather than software development ones, concluding that in practice the configuration of the Scrum framework entails being continuously used by the employees. Reference [29] compared agile practices under the Scrum and Kanban frameworks. They explored the factors conditioning the preference for either framework under varying circumstances. For instance, Kanban outperformed Scrum when considering constant changes in the project scope or the presence of inexperienced teams. Scrum was recommended when dealing with a high level of team engagement or a project-based environment.

Reference [30] concluded that one of the main inputs of the agile software product management process is product vision, which should be updated gradually. According to their proposed terminology, the expectations of the stakeholders constitute the vision, which is converted to themes used for highlighting the essential outputs of the product. Each theme is then segmented into smaller elements called

TABLE 1. Literature review in a nutshell.

References	Maturing/Improvement Focus	Single vs. multiple agile-oriented frameworks	Portfolio vs. single project management	Survey, review, application-based study	IT-based orientation (Yes$^+$, No$^-$)	Involving Case study (Yes$^+$, No$^-$)
[16]	Fastening on teams' interconnections	Multiple	Portfolio	Survey	+	+
[18]	Measuring maturity at program, team, and portfolio levels	Single	Portfolio	Survey	+	+
[19]	Customer collaboration	Single	Portfolio	Survey	+	+
[20]	Working culture	Single	Portfolio	Application	-	+
[21]	Team making and integration	Single	Portfolio	Survey	+	+
[22]	Scheduling management	Single	Portfolio	Application	+	+
[23]	Quantitative assessment of maturity models	Single	Single	Review	+	-
[24]	Hard and soft skills of project managers	Single	Single	Review	+	+
[25]	Oral narration of improvement steps	Single	Single	Survey	+	+
[26]	Introducing economic, cultural, social, and political key process areas	Single	Single	Survey	+	+
[27]	Evaluation of maturity	Single	Single	Survey	+	+
[28]	Flexibility of the methodology	Multiple	Single	Survey	-	+
[29]	Recommending the preference methods under different projects' configuration	Multiple	Single	Survey	+	+
[30]	Goal-oriented, alternate sprint cycles, detailing complex requirements	Single	Single	Application	+	+
[31]	Goal-oriented	Multiple	Single	Application	+	-
[32]	Story point	Single	Single	Application	+	-
[33]	Sprint length	Single	Single	Application	+	-
Current study	Goal-oriented, sprint length, multi-level refinement, scheduling management, story point, handling Scrum of Scrums by a customized system	Multiple	Portfolio	Application	+	+

9

Concepts. Product requirements are extracted as proposed solution stories pertaining to the Concepts. Reference [31] introduced the goal-oriented agile practice for minimizing risks and vulnerabilities during software development. It is aimed at detecting underlying interdependencies between the goals while considering the predecessor/successor criterion within the planning.

The story point is a specific tool to measure the complexity of tasks during planning in the Scrum framework. Reference [32] deemed the concept of story point as one of the embodiments of agile maturity over time. The study analyzed the improvement of the estimation accuracy of story points by imposing notifications of both quality assurance engineers as well as product owners. An alternative maturing practice consists in tuning the settings of the Scrum framework. In this regard, [33] used a multi criteria decision making-based approach to optimize the sprint length of agile software development in terms of the underlying cost and work intensity of the planned work. However, their approach requires a cumulative distribution function related to the pattern of performing user stories, which makes it intractable for projects with lacking data.

C. CONTRIBUTION

Table 1 reviews the related literature, illustrating how most of the relevant works either conducted a survey or are review-based studies. These papers focus on assessing the validity of specific hypotheses and eliciting future research developments regarding maturity and improvement facets in agile methodology. Studies rarely propose applied frameworks for maturing agile portfolio project management in the presence of more than one practice. The current paper is readily distinguishable from the previous studies by laying a strong and applied emphasis on the maturity of the Scrum-based software portfolio project management methodology in three ways.

I. To begin with, the proposed combination considering multiple drivers of maturity has been absent from the literature. The main consequences from introducing such a combination are reflected in the realization rate of planning, whose increasing trend is illustrated empirically.

II. Furthermore, our methodology simultaneously employs Scrumban and classical and goal-oriented Scrum frameworks under flexible settings.

This is a non-trivial adaptive remedy to fulfill the requirements of the portfolio management being inclusive of subprojects with an inherently distinctive scope and, at the same time, correlated characteristics. In this regard, a basic setting of the Scrum framework such as the subprojects' sprint length is heuristically determined through the designed scoring method. This method consists of a weighting method in which the importance levels of the underlying factors can be evaluated based on a scale of three values, namely, 1 (low), 3 (medium), and 5 (high). For each subproject, the sprint length is determined by the total points assigned to the subproject. The scores of the subprojects are then mapped into representation intervals defined according to the experimental setting.

III. Lastly, the study provides a comprehensive set of metrics for constantly monitoring the project lifecycle throughout all the steps of the Scrum framework. These metrics can also be applied to inject the necessary insights into the key participants of the project.

III. SOFTWARE PROJECT MANAGEMENT MATURITY

The current paper provides a basis for answering the research question of how variations of the Scrum framework can be used to yield maximum efficiency when managing a wide-ranging portfolio of software projects. To devise the practical maneuvers required to adapt and mature the Scrum framework for software projects, the research methodology of this paper is laid out in the context of a real case study. The scale, structure, and special characteristics of the case are introduced first. Then, the proposed maturing and refinement body associated with the case study is set out in the presence of step-by-step enlargement of the case's scale and real work constraints.

A. CASE STUDY

The case study of the paper is shaped with respect to a project called Smart City Mega Project (SCMP), directed by the large-scale IT solution provider FANAP Company in Asia. Our case study includes a wide range of software applications intended to encompass a variety of platforms designed to facilitate the lifestyle of human beings in the context of smart technologies. Table 2 describes the typical areas of SCMP's subprojects that are gradually created owing to the orders of indigenous and endogenous stakeholders as well as request-based requirements.

Starting in 2017, the project was initially set out to institutionalize digital transformation in the banking industry in the form of fintech-led facilitators such as Neobank. Gradually and under the financial support of a mega-scale holding, different businesses have come across each other to inject a wide spectrum of smart solutions into the project. The project currently involves a portfolio of 32 subprojects distributed through the general categories of basic platforms, entertainment, tourism, healthcare system, education, and fast-moving consumer goods. For instance, the basic platforms category includes the development of core functionalities for running software applications such as Single Sign-On (SSO), digital wallet, Internet-of-Things (IoT) manifesto, cloud services, and messenger system.

As the size of the portfolio grows through time and different types of subprojects are defined, designing a flexible yet practical project management methodology becomes crucial. Herein, it will be shown how the software project management methodology can be matured to come up with both an increasing quantity and inherently different nature of subprojects. This feature will be illustrated in terms of the SCMP real case study.

TABLE 2. Typical working areas of SCMP's subprojects.

General category	Working area	Ownership type
Basic Platform	Providing SSO and digital wallet	Indigenous stakeholder
	Creating the platform for realizing Internet-of-Thing	Indigenous stakeholder
	Blocking cyber threats	Indigenous stakeholder
	Delivering cloud storage service	Indigenous stakeholder
	Designing user-interface of the systems	Request-based
	Developing a platform of artificial intelligence-based products	Indigenous stakeholder
	Presenting cloud-based solutions	Indigenous stakeholder
	Generating reporting Dashboard for subprojects	Request-based
Finance	Developing electronic payment platform	Indigenous stakeholder
Transportation	Providing real-time traffic data	Endogenous stakeholder
Legal	Providing legal services	Endogenous stakeholder
Entertainment	Kids' entertainment platform	Indigenous stakeholder
	Game streaming service provider	Indigenous stakeholder
FMCG	Messenger platform	Indigenous stakeholder
	Subscription streaming service provider	Indigenous stakeholder
Tourism	Ticketing for museums	Endogenous stakeholder
	Ticketing for amusement parks	Endogenous stakeholder
	Reserving management system	Endogenous stakeholder
	Providing hospitality solution	Endogenous stakeholder
Education	Streaming of educational content	Indigenous stakeholder
Electronic Health System	Providing remote healthcare refinement	Indigenous stakeholder

B. MATURITY PROCEDURE OF SCMP

At the inception of the project, the agile project management system was used to coordinate related activities by focusing on the Scrum framework. At that time, the multiplicity of subprojects was not as high as today. In fact, SCMP began with a single project in the general category of basic platforms. The trend of SCMP's exponential growth has been gradually broken by setting up new working areas associated with an IoT-based platform, ticketing system, social welfare application, and so on. These spikes in the portfolio size were accompanied by increasing interdependencies between subprojects.

Methodologically speaking, the Scrum framework has been utilized from the beginning and is still used to frame 90% of todays' ongoing subprojects. Scrumban has been used for some subprojects due to its inherent request-oriented structure. Other subprojects consist of Hybrid Scrum with extreme programming whose release planning is made up of the targets defined in the roadmap. In terms of the methodological timeline, even though the Scrum framework has been the cornerstone to managing subprojects, it has been gradually matured to fit well with the SCMP's dimensions and requirements.

Fig. 1 provides a graphical representation of the sequence of maneuvers adopted for maturing SCMP portfolio management. Each one of the maneuvers represented in the figure relates to one of the subsections among III.B.1 to III.B.7, where it is introduced and discussed with respect to its application to the case study. The first two maneuvers, involving agile goal-oriented roadmap and incorporating Scrumban are related to III.B.1. The third and fourth maneuvers, taking story point into account and defining retrospective KPIs, are associated with III.B.2 and III.B.3, respectively. The fifth and sixth maneuvers, facilitating SOS and providing a basis for the Scrum-based tools pertain to III.B. 4. The seventh maneuver, setting the sprint length, relates to III.B.5. The eighth and ninth maneuvers, considering the management of the peripheral activities and customizing the scheduling is the focus of III.B.6. The last maneuver, providing a multi-level refinement tool is related to III.B.7.

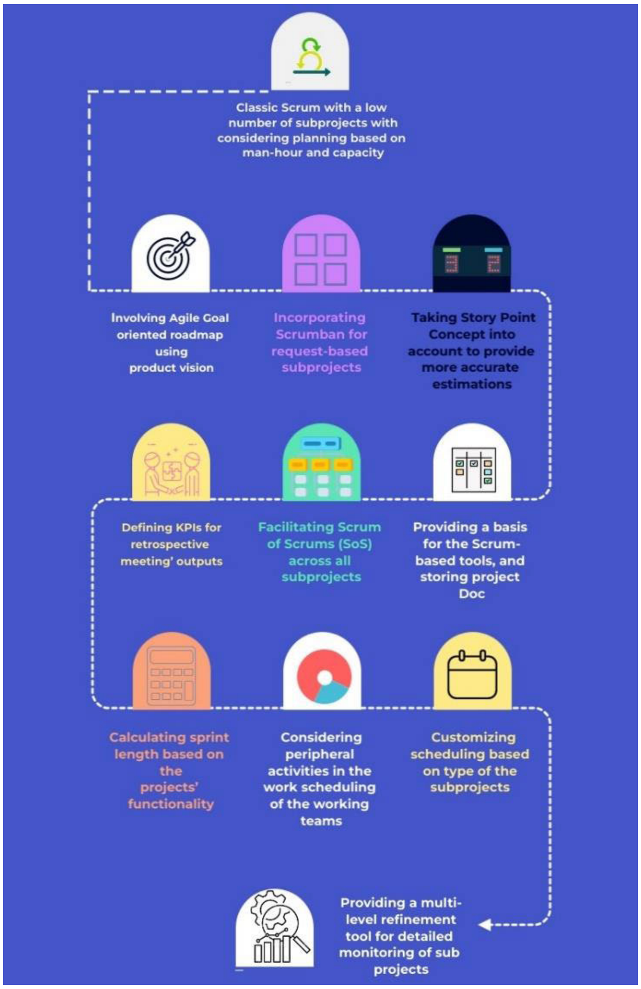

Classic Scrum with a low number of subprojects with considering planning based on man-hour and capacity
Involving Agile Goal oriented roadmap using product vision
Incorporating Scrumban for request-based subprojects
Taking Story Point Concept into account to provide more accurate estimations
Defining KPIs for retrospective meeting' outputs
Facilitating Scrum of Scrums (SoS) across all subprojects
Providing a basis for the Scrum-based tools, and storing project Doc
Calculating sprint length based on the projects' functionality
Considering peripheral activities in the work scheduling of the working teams
Customizing scheduling based on type of the subprojects
Providing a multi-level refinement tool for detailed monitoring of sub projects

FIGURE 1. **Maneuvers for maturing SCMP portfolio management.**

1) CONSIDERING ADAPTIVE FRAMEWORKS TO MANAGE THE PORTFOLIO

We start by demonstrating why the goal-oriented approach and Scrumban framework are used to deal with the incompatible nature of the underlying subprojects within the portfolio. There was no certain roadmap when preparing the initial release planning for a period of three months. Time has shown that devising a vision for products is a must. Although the main skeleton follows the agile methodology, the planning space could be isolated from the prime objectives and vision of the subproject in the absence of a roadmap. This roadmap should clarify the progressive process of the product.

The main drawback from implementing the standard Scrum in our case study involved consecutively advancing the features of subprojects absent future insights regarding the works' residuals. Consequently, the value assigned to the done features as a percentage of accomplished subprojects was not transparent. By conducting the necessary evaluations and reviewing the best practices, the agile roadmap, which is a goal-oriented method, was put into practice. In this regard, product owners were supposed to devise the objectives for a specified predictable timeline. Note that the agile terminology does not allow to plan for a long interval period due to the speed of technology advancement, underlying changes, and risks of software projects.

Taking the goal-oriented roadmap into account, product owners were asked to visualize the backdrop of a release plan and specify customers' requirements as user stories. Product owners were indeed supposed to draw up master features, objectives, and milestones according to the recent needs of the subprojects and the system analysis conducted. The release plan was extracted from said roadmap thereby transforming Scrum into extreme programming. By institutionalizing the roadmap, positive consequences were brought about, illuminating the trajectory of the next release.

The completion of the product is a key indicator to justify the functionality of the approaches adopted to promote the management methodology of the subproject. When subprojects were planned in a feature-based structure, a specific progress percentage (e.g., 70%) did not convey the level of accomplishment of the whole product. In other words, the roadmap-based methodology sheds light on the realization route of the product and avoids the short-sighted perspective of project management.

In this regard, releases are defined considering the objective and master features of the proposed roadmap. The matured methodology can facilitate computational progress by highlighting the contribution of each feature to product accomplishment. As a result, the evaluation of each team becomes more accurate, specifying whether a task fulfilled is critical to realizing the objectives of the roadmap or a refinement related to the previous features. The tractability of the subprojects is eased, and new types of features declared within the evolved methodology.

On the other hand, it has been observed that not all teams can be assigned periodic release plans and a subsequent roadmap. These teams must be deemed request-based subprojects. These subprojects are managed by Scrumban, through which they are planned and go through an execution phase. The most common request-based subprojects consist of developing reporting dashboards, generating content, and designing user interfaces and user experiences for different subprojects. Kanban is not appropriate for our type of request-based subprojects since its inherent terminology does not concentrate on capacity assignment. It mainly works by prioritizing and monitoring the work in process backlog items within its to-do, doing, and done columns. In the proposed Scrumban, the velocity and capacity of the corresponding working teams are assessed through the feedback received from daily and demo-planning meetings.

2) INCORPORATING STORY POINTS

Similarly, to neglecting product vision, story points had also been initially ignored and planning was done in terms of the capacity and man-hour of the team members. This avoided comparing the performance of the teams involved when facing tasks of different complexity, duration, and priority. Given the increasing growth in the number of subprojects, the concept of story point has been injected into the body of project management methodology.

3) MONITORING THE MEETINGS' OUTPUT

Meetings and events associated with Scrum, particularly retrospective gatherings, have become more purposeful. Initially, in the retrospective meetings, members stipulated the strengths and drawbacks of the corresponding team, which entailed pondering suggestions to improve the result and imposing decisive action plans. However, nowadays, a set of KPIs is extracted from the discussions taking place in these meetings and relative targets are set. For instance, if the deliverables' tardiness is inappropriate, an updated delivery time is considered for the next release.

4) DEVELOPING A CUSTOMIZED PROJECT MANAGEMENT SYSTEM

The next project management challenge pertains to the interdependencies existing between the increasing number of subprojects. Accordingly, the features of one release in a subproject are interconnected with the features defined for the release of another one. In response, several Scrum of Scrums (SOS) meetings were held. The need for this sort of meetings has been gradually covered by the development of a customized project management system. The interconnected works between subprojects are handled by this customized system, which leads to shortening the SOS meetings.

By issuing a specific request, the recipient may ask for clarification meetings where the upcoming requests of the interdependent teams are addressed. This sets the basis for drawing up the subsequent release planning of the master subprojects, e.g., platforming the category of basic platforms. In other words, the registration of requests should lead the influencers of the main subprojects to consider the tasks required in their upcoming roadmap, release planning, and so on. Complementarily, the scheduling of the subprojects is tunned based on the corresponding feature planning, which enables a high level of collaboration between teams.

In the beginning, there was no specific sort of Scrum tools except the physical board. Tasks are currently registered in a shared framework called Share Point, which brings about a platform for managing knowledge. In addition to preserving the documents generated, Share Point stores other routine companions, such as the minutes of the meetings and the mission history of the employees. The board of the customized system has been developed like a Team Foundation Server project management system ([34]). The customized system manages the requests of the teams involved and constitutes the main tool for handling the Scrum-based framework and conducting release and sprint planning.

5) ADJUSTING THE SPRINT LENGTH

Another change associated with the monitoring of the teams' tasks involved the duration of the sprints. One of the noticeable concerns in the SCMP's Scrum-oriented framework implied considering a prudent time interval for the sprint sessions. Previously, a similar two-week period was assigned to the sprint time of every subproject.

Due to the high volume and wide range of differentiation among SCMP's products, it was essential to design a framework to analyze whether a given sprint time fitted well an ongoing subproject. It became clear that the sprint's duration of a specific subproject could differ from that another one (e.g., one, two, or three weeks) based on their distinctive characteristics. The key factors included agility, the number of team members, having endogenous or exogenous stakeholders, etc. After identifying and analyzing the factors, the project management office evaluated and ranked the subprojects and assigned them the relevant sprint time.

6) UPDATING SCHEDULING MANAGEMENT

The scheduling management practice of the portfolio of subprojects requires updating. The Project Management Office usually plans the total available time of the team members involved during a sprint while neglecting any requirements made to conduct peripheral actions. As a prime example, a technical leader must carry out certain responsibilities for supporting, reviewing, versioning, refactoring, and merging the outputs of his/her teammates. There are also unplanned tasks with the capacity of disrupting the baseline scheme.

These time-consuming activities had been ignored in the planning, leading to unmet accomplishments at the end of sprints/releases and hedging against the identification of the root cause of delay by the Project Management Office. This drawback was avoided by segmenting the full working time into the working time for conducting feature-oriented activities and that allocated to deal with peripheral affairs. The latter was not distinctively planned for the members involved and baseline planning was still far beyond the actual one. The work of the relative members must therefore be scheduled now based on the aforementioned peripheral activities. The remaining time capacity available was allocated to the activities required for developing the associated features.

After adopting the new task assignment procedure, the number of failing and incomplete features during the planning horizon had been drastically decreased and the performances of teams' members had been enhanced. As a technical leader devoted an explicit amount of time to support and guide the developers, the reworking and debugging activities were plunged leading to a higher performance. The working hours of other team participants such as developers and system analysts had not been fully allocated to the duties of the planning horizon. A particular percent of their capacities had been freed to fulfill unplanned tasks per release planning, as well as to meet the unplanned requirements of stakeholders and increase their ability to respond to changes.

The diversity of the portfolio forced the project management office to customize and adapt the scheduling terminology per type of subproject. For none-employer-based subprojects, there was no work breakdown structure or progress percent since the scope entails a higher level of variability in comparison with employer-based subprojects. When managing such subprojects, roadmap, release backlog, and sprint backlog were used to derive the work breakdown structure and progress report. The milestones of the scheduling were determined with respect to the objectives and master features. For employer-based subprojects, the work breakdown structure was prepared, the relevant activities defined, and the tasks scheduled in the specific releases and sprints. Afterward, daily/milestone-oriented progress and refinement reports were geared up for managerial decision-making.

7) PROPOSING MULTI-LEVEL REFINEMENT PATTERNS

To synchronize the reporting outputs of the SCMP's project management practice with its gradual maturity, a multi-level refinement framework was presented. In the first level, the daily progress was measured by scoring the burndown power and deviation from baseline planning. At the end of each sprint, performance, team power, and speed metrics were calculated. The former is the ratio of done hours from the completed tasks to the total planned hours whereas the latter compares the rate of done tasks with the power of the team. Similarly, the amounts of done story-points, new features/requirements, improved features/requirements, support, and bugs fixed were obtained per sprint. Furthermore, the realization percent, earliness, lateness, and changes of every release were all considered.

When evaluating the refinement statistics, a lower number of tasks with changed scope indicates a more accurate definition of the user stories and identification of customer needs. Exploring the quantity of resources involved while estimating the volume of remaining work constitutes another refining technique for a subproject. The evaluations required were performed through a detailed analysis of the product owners prior to the start of a release. Finally, the S-curve was generated combining user stories time, point, and count. It was used to illustrate the deviation between the baseline and actual planning in the context of a given period. Following the roadmap, the accomplished percent of the proposed master features and objectives were also evaluated.

IV. EMPIRICAL RESULTS

We describe below the tangible results and implications from adopting the above maturity steps. These results illustrate how the framework implemented contributes to the realization rate of planning, the scoring method for determining the sprint length works, and KPIs should be defined to assess the meetings' output. Furthermore, we illustrate both the working mechanism of the proposed multi-level refinement process together with the application of the developed project management system.

A. GOAL-ORIENTED PLANNING AND CAPACITY MANAGEMENT OF MEMBERS

In terms of remedies pertaining to the product roadmap and management of the working time of teams' members, focal symptoms can be monitored within the realization status of the releases. The realization rate is equal to the percentage of the done (delivered) story points divided by the planned ones. To analyze the consequences, we focus on a subproject that has gone through both classical and matured methodologies. The comparison of five consecutive releases, prior and subsequent to running the maturity's maneuvers, delivers a significant improvement of 23.5% in terms of the planning realization rate.

Fig. 2 illustrates how the new methodology outweighs the initial one, narrowing the gap between the baseline and actual planning by enhancing the realization rate of releases. Note that, the subprojects of the first five releases were managed applying the initial methodology. The new methodology was used for the remaining releases.

Counteracting deviations from the planned releases could not be isolated from the impact of managing the capacity of teams' members by involving peripheral activities.

For instance, 20% and 10% of the total release working time was assigned to the unplanned and refactor type of tasks, respectively.

These values were chosen based on the observation of the experimental setting of SCMP. Considering different release plans, it was observed that 20% and 10% of the actual work

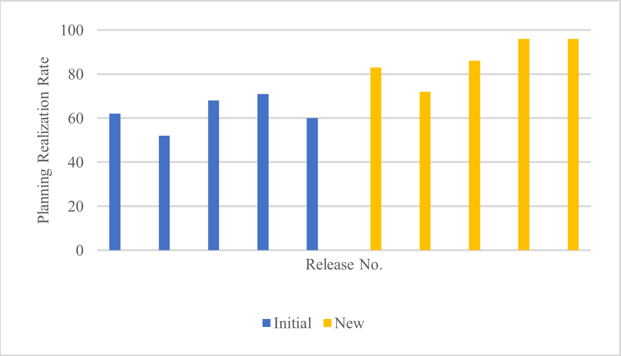


FIGURE 2. Planning realization trend of releases in terms of initial and new methodologies.

was related to the unplanned and refactor-oriented activities, respectively.

Although these two values, 20% and 10%, were not obtained through an optimization formulation, it is clear from the experimental results that such a time allocation has significantly contributed to the completion of the tasks of SCMP.

Following the same logic, the float was projected into the sprint planning together with the corresponding support, refactor, code review, and clarification-based user stories. Code review and clarification were specifically designed for the technical leaders while the other stories were also defined for the development team. Notably, the perceived impact of considering goal-oriented planning and capacity management of members is to considerably improve the releases' realization rate and offer flexibility to the planning horizon.

B. VARIABLE SPRINT TIME

As discussed before, finding the appropriate sprint interval for different types of subprojects was deemed one of the underlying maturity steps. A heuristic score-based method has been proposed in which the subprojects with higher scores involve larger sprint times and vice versa. The method estimates the sprint period of a subproject according to the summation of the points attributed to a number of factors, which include the scale of user stories, stakeholder feedback/delivery time, frequency of changes, Scrum culture and commitment, degree of refinement, and exposition level to unknown risks.

Regarding each subproject, low, medium, and high labels were assigned to the factors. These labels have been associated with a three-value scale of 1, 3, and 5 points, respectively. Six factors were considered. Table 3 describes the factors and the corresponding tendency patterns. Table 4 is used to aggregate the results. Note that the opinions of the experts were considered when determining the duration of the sprint period for the in-between scenarios.

Given six factors and the three-value scale, the lowest possible score that a subproject could acquire, was equal to 6. That is:

$$6 \text{ (number of factors)} \times 1 \text{ (minimum points of each factor)}$$
$$= 6 \text{ (lowest possible score of each subproject)}$$

TABLE 3. Factors used to estimate sprint periods.

Factor	Description	Tendency
Scale of user-stories	As the scale of user stories increases, the sprint period needs to be enlarged to increase the chance of completing a shippable output. For instance, consider user stories that entail an infrastructural-type task equal to 60 hours. In this case, a one-week sprint period does not yield any output.	The *High* label has the highest point.
Stakeholder feedback/ delivery time	How often is it required to convey feedback to the stakeholders based on the pre-defined delivery time? Clearly, as the delivery time is intensified, a shorter sprint time becomes more feasible. For instance, a subproject that has one month delivery time requires narrowing the sprint period in comparison with a counterpart subproject with three months of feedback time.	The *High* label has the highest point.
Change	Facing a high rate of changes in a subproject is not consistent with setting long sprint periods if one wants to avoid re-planning in the corresponding interval. Keeping planning fixed in a sprint is, indeed, the favorable axiom of the Scrum framework.	The *Low* label has the highest point.
Scrum culture and commitment	If a working team has not been matched with the continuous delivery of the Scrum, a long sprint period would probably decrease the done tasks during the given interval. As a result, the rejection rate should be expected to escalate since the accomplishment of the tasks would be postponed to the last hours, preventing the performance of the review and quality control checks.	The *High* label has the highest point.
Refinement	Subprojects needing to go through a significant number of refinement and review-type tasks require a longer interval to fulfill the sprint backlog.	The *High* label has the highest point.
Unknown risk	The possibility of being exposed to unknown risks hedges against defining a lengthy sprint period.	The *Low* label has the highest point.

TABLE 4. Rule determining the Sprit period.

Score (point)	Sprint duration (weeks)
6	1
[7,17]	1-2
[18,29]	2-3
30	3

Similarly, the highest possible score of a subproject equals 30. That is:

$$6 \text{(number of factors)} \times 5 \text{ (maximum points of each factor)}$$
$$= 30 \text{ (highest possible score of each subproject)}$$

Moreover, since the sprint intervals of the agile methodology are usually short, 1, 2, and 3 weeks were considered as the possible sprint periods. More precisely, the subproject with the lowest score (i.e., 6 points) was assigned one week sprint length. The sprint length of the subproject with the highest score (i.e., 30 points) was set to be three weeks. Additionally, for the scores that fell in the intervals [7], [17], [18], [29], the experts assigned a sprint period equal to one or two week(s) and two or three weeks, respectively.

Note that, the lengths of the intervals in Table 4 follow a specific and consistent rule. The first and last categories are extreme cases and include just a single value, that is, the lowest (i.e., 6) and highest (i.e., 30) obtainable scores, respectively. The in-between categories include almost the same amount of remaining integer numbers between 6 and 30. Indeed, the available 23 integer numbers were divided into two categories. In order to keep a consistent format, one category involved 11 values (i.e., all integers in [7] and [17]) and the other one contained 12 values (i.e., all integers in [18] and [29]).

The constant sprint period of three weeks considered for all the subprojects has been changed into the specific values resulting from the underlying method. For instance, the sprint length of a request-based subproject associated with designing the user-interface of the systems was turned into one week. Subprojects in the working areas of security issues and delivering a message to a set of recipients have been planned using sprints of two weeks. The sprint period of some subprojects has remained intact and equal to three weeks. This is the case for the subprojects in the working area of communication platforms.

To illustrate the reasoning behind these modifications, consider the request-based subprojects within Scrumban. Their sprint length has been shortened to one week since the frequency and changes of requests are high and deliveries should be accomplished in short periods. Thus, the preference is to have 10 outputs in one week instead of achieving 20 shippable products in two weeks, accelerating the requirements of the dependent subprojects. By equipping the subprojects with such preferences, the proposed practice could contribute to the efficient accomplishment of SCMP's portfolio.

C. DEFINING KPIS IN RETROSPECTIVE MEETINGS

Navigating the retrospective meeting in a meaningful manner is strictly dependent on the content and queries raised by the parties involved in the session. For instance, assume that the participants – based on the ongoing performance data of a team – detect either a high rate of bugs or a low number of outputs. Then, the associated improvement percent would be amongst the KPIs of the next release. Assume that tasks are suspended after a review check.

The minutes of the meeting should include the percent of user stories waiting to be validated for more than two days in the next planning period. Moreover, a comparative KPI can be set to assess the ratio of user stories that are moved to the QC column within the deadline while excluding refinement-type tasks. Tasks such as code cleaning, swaggering, updating docs, and preparing test cases are not counted since they do not need QC tests or require any commitment from the product owners to be delivered.

In a nutshell, the perceived impact of defining KPIs in retrospective meetings is grounded in making the meeting sessions more purposeful. Such a change provides an opportunity for taking the observations and comments of the members and participants of the project into account. It is indeed worth to understand whether a subproject has been improved in terms of a specific KPI based on observations and comments of the members. In this regard, the portfolio can be improved by using the wisdom of the members and their constructive hints over a particular facet of the project.

D. ACQUIRING SCRUM OF SCRUMS BY THE CUSTOMIZED PROJECT MANAGEMENT SYSTEM

In the portfolio management framework, the practice of accumulating requests from various subprojects is suitable to efficiently ease the planning of the platform layer that provides services for other products. In this respect, new requests should be discriminated if they were to compromise a release and intervention in the original planning should be minimized.

As described earlier, to preserve the flexibility of the proposed matured Scrum framework, 20% of the release working time is reserved for carrying unplanned user stories. If portfolio requests were not collected, the rate of unplanned activities would violate the limits imposed and achieving the Scrum of Scrums using the most suitable tool becomes a must. Aggregating the requests of all teams leads to the total backlog of SCMP. It must be emphasized that some of these requests are services that should be prepared by other teams to develop the output required by the stakeholders.

For example, a subproject that involves finding grocery applications needs a tool to search the marketplace for dealers. This request is sent to a team that is responsible for managing the system's content. This team assigns the request to the artificial intelligence team, which, in turn, engages another team that runs the service wrapper. Note that a total of four teams will be involved.

First, the artificial intelligence team has to implement the search tool while requesting the actions necessary to wrap the service. Then, the subproject calls the resultant service from the working team to manage the context of the system. Capturing these interconnected requests, specifically in large volumes, is only possible if an integrated information management system is used. In fact, SCMP is equipped with a customized project management system. Being able to continually update the SCMP backlog in a regular manner is the main purpose of creating Scrum of Scrum through the customized project management system.

E. MULTI-LEVEL REFINEMENT FRAMEWORK

Refinement at the release level is conducted by computing the corresponding deviation metrics, categorizing user stories, and analyzing the S-curve report. As shown in Table 5, deviations from the release plan are assessed in different ways.

TABLE 5. Deviation metrics to refine release planning.

Type	(I)		(II)		Deviation
	Factor	Formula	Factor	Formula	
General perspective	Done users stories	$Planned + Unplanned$	Capacity	$Total\ working\ days \times Daily\ working\ hours \times Team's\ members\ count$	$(\boldsymbol{II})-(\boldsymbol{I})$
	Initial Release plan	∀ Story item, Story point, Man/Hr: *Total amount of initial Release plan*	Final Release plan	∀ Story item, Story point, Man/Hr: *Total amount of final Release plan*	$\frac{(\boldsymbol{II})-(\boldsymbol{I})}{(\boldsymbol{I})}$
Progressive Perspective	Planned (Initial Release)	∀ Story item, Story point, Man/Hr: $\frac{Planned\ initial\ Release\ plan}{Initial\ Release\ plan}$	Actual (Initial Release)	∀ Story item, Story point, Man/Hr: $\frac{Done\ final\ Release\ plan}{Initial\ Release\ plan}$	$(\boldsymbol{I})-(\boldsymbol{II})$
	Planned (Final Release)	∀ Story item, Story point, Man/Hr: $\frac{Planned\ Final\ Release\ plan}{Final\ Release\ plan}$	Actual (Final Release)	∀ Story item, Story point, Man/Hr: $\frac{Done\ final\ Release\ plan}{Final\ Release\ plan}$	$(\boldsymbol{I})-(\boldsymbol{II})$

From a general perspective, the relative change between the initial and final release plans is obtained by taking the volume of releases into account. Computing this volume requires adding the amounts of story items, story points, and Man/Hr within a release plan.

For instance, assume that there were 89 items in the initial release, which turned into 122 items in the final release. This indicates a 37.08% increase in the story items' count. Another deviation metric explores the earliness and tardiness of subprojects by subtracting the Man/Hrs used to accomplish user stories from the total capacity of the members of the teams. Consider the case where 976 and 309.5 hours are spent to complete planned and unplanned user stories within a release plan, respectively. A total of 52 working days, 7 working hours per day, and 4 members available to develop user stories, leads to a capacity of 1456 Man/Hr. The resulting positive difference implies a tardiness of release equivalent to 170.5 hours or 6.09 Man/day behind schedule.

The progressive monitoring of the releases implies double-checking the deviations of story items, story points, and Man/Hr in terms of the initial and final plan, separately. The former computes deviations dividing the difference between the planned and done number of story items (as well as story points and Man/Hr) by the value planned in the context of the initial release. The latter calculates the same formula using the final release as a reference.

For instance, assume that 64 items out of a total of 89 user stories are planned in the initial release and 60 could be done from the initial list. Consequently, 94 items out of the total 122 user stories are planned, which involves 88 done items pertaining to the final release. As a result, the deviations of the release progress are 4.49% and 4.92% according to the initial and final plans, respectively. This comparison reveals that the extra lag in the actual planning stems from the existence of unplanned user stories rather than the performance of the corresponding teams.

At the end of a release, the status of the user stories is also explored in terms of items' count, used Man/Hr, and story points. The categorization encompasses the contribution of To Do, Rejected, Blocked, Doing, Duplicated, Out of order, and Done user stories relative to the corresponding factor. Table 6 describes how to compute the relevant ratios. For instance, the story point of To Do type stories is obtained dividing the cumulative count by the total amount of story points in the To Do list. Shift type statuses refer to the user stories that need to be transferred from the current release to the next one whereas out of order items are not moved to the Doing list due to lack of prioritization.

The final refining step consists in visualizing the comparative and cumulative differences in progress between the baseline and actual plans under the umbrella of the S-curve. Consider the example presented in Fig. 3. The cumulative expected progress, as well as the realized one, are calculated with respect to the count of items and Man/Hr of all user stories per sprint. It was estimated that 92.62% of the story items would have been completed in the fourth sprint while the actual data shows 20.49% less progress. In the same vein, the Man/Hr utilized falls behind the estimated one by 16.76%.

The categorization of user stories is addressed within the refinement level of sprints in a more detailed manner. It describes the status of done user stories in terms of the frequency of meeting, support, clarification, bug, versioning, unplanned, and planned ones (a typical example is presented in Fig. 4). Furthermore, the performance of teams' members at a sprint is justified in two ways. First, the state of accepted works per member is analyzed using a bar chart.

Fig. 5 depicts a "good performance" from the members of a team in terms of done user stories regarding capacity, used time, and volume of blocking stories. Note how the performance of A and D has been weakened due to blocking tasks related to their user stories. We compute the general performance of the different team members by dividing the

TABLE 6. User stories assessment metrics to refine release planning.

User story status	Metrics		
	Items' count	**Used Man/Hr**	**Story points**
To Do	$\frac{\textit{Number of To Do items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of To Do's Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of To Do's story points}}{\textit{Total sum of story points}}$
Reject	$\frac{\textit{Number of Rejected items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Rejected items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of Rejected items' story points}}{\textit{Total sum of story points}}$
Block	$\frac{\textit{Number of Blocked items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Blocked items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of Blocked items' story points}}{\textit{Total sum of story points}}$
Doing	$\frac{\textit{Number of Doing items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Doing items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of Doing items' story points}}{\textit{Total sum of story points}}$
Duplicate	$\frac{\textit{Number of Duplicated items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Duplicated items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of Duplicated items' story points}}{\textit{Total sum of story points}}$
Out of order	$\frac{\textit{Number of Out of order items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Out of order items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of out of order items' story points}}{\textit{Total sum of story points}}$
Done	$\frac{\textit{Number of Done items}}{\textit{Total stories' count}}$	$\frac{\textit{Sum of Done items' Man / Hr}}{\textit{Total sum of stories' Man / Hr}}$	$\frac{\textit{Sum of Done items' story points}}{\textit{Total sum of story points}}$

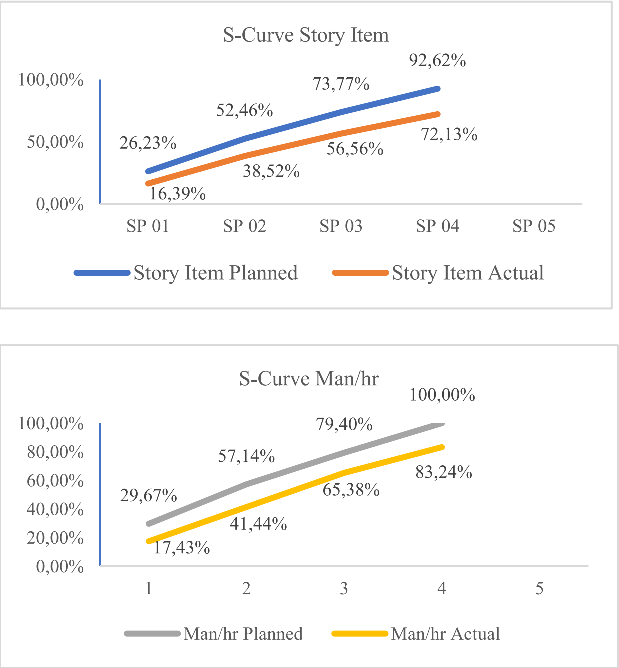


FIGURE 3. S-curve describing the refinement of the releases.

Man/Hr used to complete their planned and unplanned tasks by their corresponding capacities. In our example, the general performance coefficients of team members A through F are equal to 75%, 100%, 100%, 24%, 100%, and 100%, respectively. Note that, if the sum of planned and unplanned completed user stories exceeds the capacity, the coefficient of performance is set to the maximum value of 100%.

The refinement practice concludes by computing the coefficient of performance and speed of the sprint. The coefficient of performance is the result of dividing the Man/Hr required to complete the done user stories by the total time-based capacity of the team during the sprint. If the numerator of this ratio is divided by the power of the team, we obtain the performance speed of the sprint. Power is the product of capacity and the performance of the team in the context of the sprint. Both these values are critical metrics providing significant insights about the participants of a subproject that can be regularly reported as done in Fig. 6. As a result, any fluctuations in the performance of the sprints and speed over different releases, e.g., release 4 (R4) through 8, can be detected.

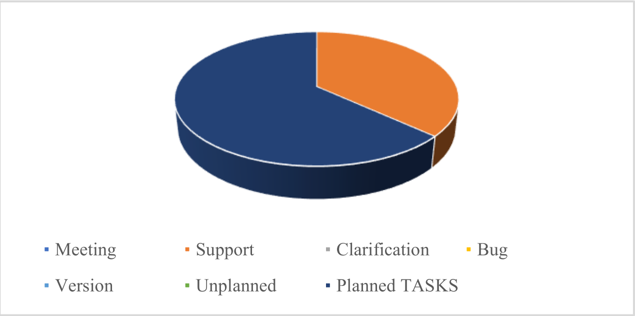


FIGURE 4. Frequency status of done user stories at sprint level.

Monitoring the daily performance is another step composing the proposed multi-level refinement procedure of the SCMP's matured portfolio management. The daily time devoted to user stories by each member relative to his or her working hours capacity is illustrated in Fig. 7. The patterns observed imply that member A has not registered his working time while B, C, and D have allocated equal, extra, and less time to focus on the tasks of the ongoing sprint, respectively.

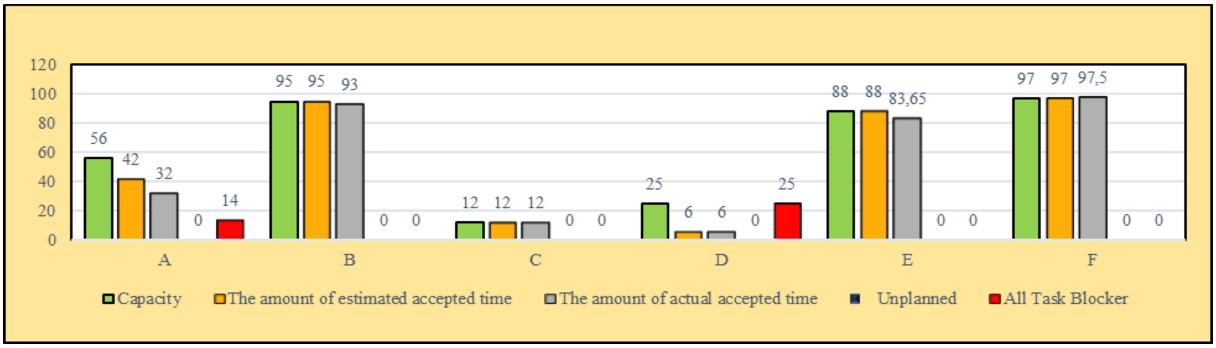


FIGURE 5. Performance of members over the accepted works of a sprint.

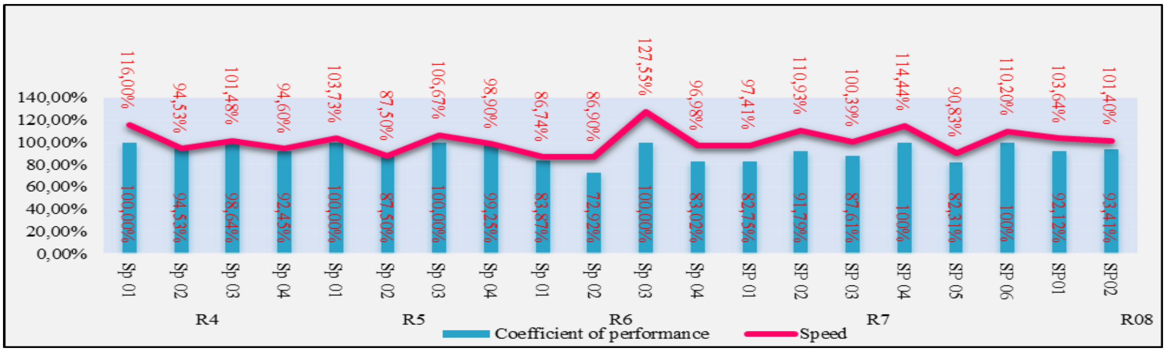


FIGURE 6. Coefficient of performance and speed of sprints.

Finally, the burndown graph is employed to double-check the daily performance of the contributors to the subproject.

This graph compares the planned and actual number of remaining tasks considering the Man/Hr used daily. The former subtracts the total Man/Hr capacity from the cumulative sum of the team's power whereas the latter computes the Man/Hr of the tasks available for Product owner-check until the deadline. Considering the user story items, the graph is converted into a descending count of the total items minus those transferred to the Product owner-check column.

Based on the content of Table 7, Fig. 8 schematically overviews the performance of a team until the fourth day of the planning horizon. It has been assumed that there are 30 story items, a total capacity of 473 Man/Hr, and the performance of the team equals 100%. Note that since this last parameter has been set to its maximum value, the daily power of a team is the same as the corresponding capacity.

Fig. 8 (a) illustrates how the burndown and remaining user stories remain unchanged through the first two days due to the zero capacity of members. When entering the third day, 49 hours of the capacity are used, and it is expected that the planned value of the remaining user stories gets to 424 (473 – 49). The Man/Hr of user stories moved to the PO-check column is equivalent to two hours, resulting in 471 remaining Man/Hr (473 – 2). The first relevant change in Fig. 8 (b) arises when, on the third day, two user stories are sent to PO-check resulting in 28 available units (30 – 2). Fig. 8 (c) summarizes the daily status by computing the fraction $^{460}/_{473}$, which defines a 3% progress at the end of the third day of our data sample.

TABLE 7. Sample daily data describing the capacity of a team and tasks available in the PO check column.

Day No.	Capacity	Tasks in PO check	
		Man/Hr	Story item
1	0	0	0
2	0	0	0
3	49	2	2
4	49	11	6
5	0	-	-
6	56	-	-
7	49	-	-
8	0	-	-
9	0	-	-
10	56	-	-
11	56	-	-
12	56	-	-
13	53	-	-
14	49	-	-

In the refinement framework, stakeholders are informed about the status of the subprojects through general and detailed progressive data. General data involve illustrating the S-curve, highlighting excusable and non-excusable delays, and prospective earliness as well as presenting the percent of on-schedule, pending, behind-schedule, and ahead-of-schedule activities.

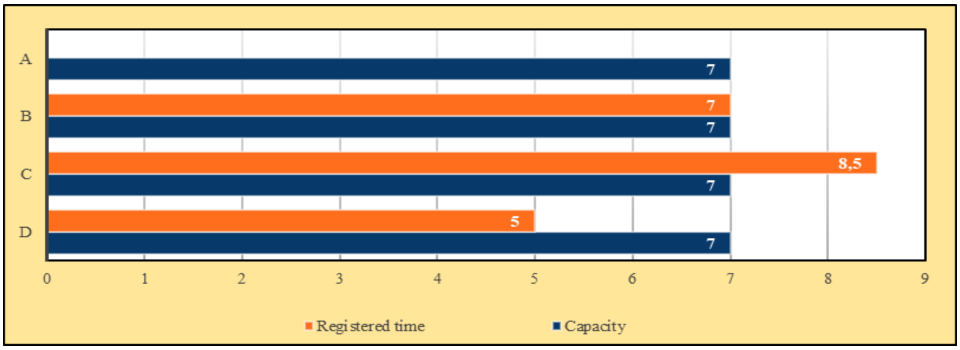


FIGURE 7. Typical daily work report registered.

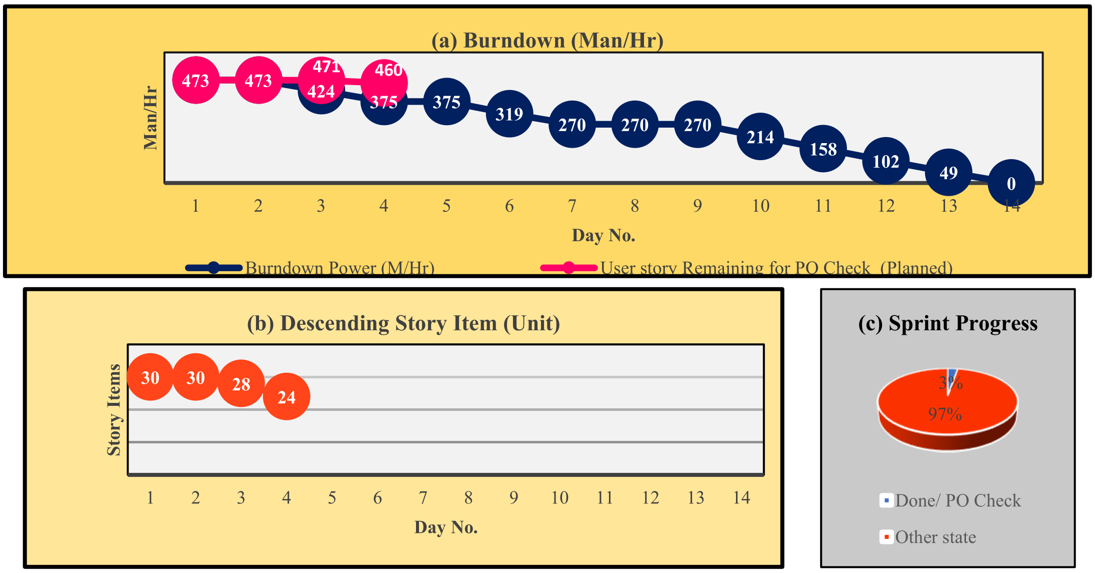


FIGURE 8. Graphs describing daily performance of teams.

The completed, in progress, and upcoming features are further reported to bring about exhaustive insights for the decision-makers. Table 8 presents a partial report delivered to the stakeholders associated with the subproject of a bank facility carried out by different responsible teams.

Regarding the report, the detailed status of in-progress and completed activities is shared with the corresponding stakeholder. This allows the stakeholder to grasp the progresses that have been made and those that are still to be made. In our specific case, as Table 8 shows, three activities are currently behind schedule by 5%, 5%, and 15%. Three activities have been completed within the given deadlines. The responsibility of performing each activity is also addressed and referred to a subproject. Therefore, the stakeholder can adopt the required follow-up for the delayed activities such as displaying, inquiring, and activating the credit cards. Using this type of reports, stakeholders are kept regularly informed about the detailed status of the project while their support is secured.

All in all, the discerned influence of the multi-level refinement framework pertains to providing the key participants of the project with significant information on the baseline versus actual planning comparison and progress reports. This information considerably helps the key participants to adopt the necessary actions for improving the status of the project at different accomplishment levels.

V. DISCUSSION

This section provides an in-depth interpretation of the results of the current research. First, a discussion of the advantages of the results obtained compared with the findings of previous studies is presented. Second, the managerial insights are outlined. Hence, the limitations of the performed analysis are presented. Finally, some future research directions are drawn.

A. ADVANTAGES

As mentioned in Contribution section (Subsection II.C), differently from the existing works in the literature, mostly consisting of surveys or reviews, we focus on assessing the validity of specific hypotheses for the design of an applied framework for maturity agile portfolio project management

TABLE 8. Detailed status of in-progress and completed activities at the stockholder level.

In-Progress Activities						Completed Activities					
WBS ID	**Activity**	**Responsible**	**State**	**Planned completed%**	**Actual completed%**	**WBS ID**	**Activity**	**Responsible**	**State**	**Planned start date**	**Actual finish date**
1.25.3	Displaying the list of credit cards	Basic platform-led subproject	Doing	100%	95%	1.5	Implementing score calculation service	Financial service provider	Done	1/16/2021	1/27/2021
1.21.3	Inquiring credit cards	Basic platform-led subproject	Doing	100%	95%	1.7	Implementing score display	Payment platform	Done	2/2/2021	2/1/2021
1.26.1	Activating credit cards	Basic platform-led subproject	Doing	100%	85%	1.18.3	Viewing user profile	Basic platform-led subproject	Done	1/16/2021	1/16/2021

in the presence of more than one practice. The strong emphasis placed on the maturity of the Scrum-based software portfolio project management methodology is the feature that makes the current paper readily distinguishable from the previous studies. The applied quality of our study includes: (I) a novel combination of multiple drivers of maturity; (II) the simultaneous implementation of Scrumban and classical and goal-oriented Scrum frameworks; (III) a comprehensive set of metrics for a constant monitoring of the project lifecycle throughout all the steps of the Scrum framework.

We follow a case study-oriented methodology and focus on elaborating a set of workable maneuvers to mature the Scrum framework when applied to portfolio management. In this sense, our study occupies a unique position in the literature since all the maneuvers for maturing the proposed portfolio project management were designed considering the experimental setting provided by the real case study of SCMP. That is, all the results of this paper were achieved under the umbrella of empirical experiments associated with SCMP case study. None of the relevant studies (Table 1) previously published could benefit from such an advantage.

Regarding, in particular, the proposed heuristic score-based method for computing the sprint length, its superiority in comparison with the previous studies is in the availability of data. In SCMP, where new subprojects are born with an increasing trend, the sprint length calculation methods based on the statistical inferences ([33]) are not workable. This is due to the fact that there is no previous history with rich data associated with the upcoming subprojects of SCMP. Thus, the proposed heuristic score-based method outperforms [33], since it does not depend on the presence of a large amount of data.

B. MANAGERIAL INSIGHTS

The managerial insights are better understood by focusing on the role played by the goal-oriented version of the agile methodology. Employing the goal-oriented version of agile methodology provides a result-driven basis for participants to perform their duties in line with the targets preset for the subprojects. It hedges the participants, subprojects, and thereby the whole SCMP against derailing from the predefined short and concise goals of the upcoming planning horiz

In fact, in any specific field, setting a goal encourages the involved parties to gear up their attempt and resources toward meeting the planned outcome as much as possible. Owing to the presence of such a synergy, this change could boost the conformity of the businesses' expectations and the delivery of products. Freeing a specific percentage of the participants' working capacity allows the planning to better predict what will happen in the actual performance. Thus, the gap between the baseline and actual planning is significantly dropped.

From a managerial viewpoint, defining KPIs for the retrospective meetings is the key to a correct interpretation of the results obtained. In SCMP, KPIs are mainly considered to secure the engagement of the project participants. They must be defined so as to allow capturing the comments at the employees' level in terms of technical and administrative aspects, concurrently.

At the managerial level, the successful development of a customized project management system represents one of the most suitable tools to aggregate the requirements characterizing the multiple subprojects of a portfolio and capture their interconnected dependencies.

In order to understand the usefulness of the adopted maneuvers, the subprojects must be constantly monitored through different periods. The comprehensiveness of the proposed multi-level reporting tools shapes an organized skeleton to achieve goals whose progress may be more critical, from the shortest planning horizon to the longest one.

The insights provided by these tools enable the managers and stakeholders to keep track of different subprojects and take appropriate decisions for surmounting possible bottlenecks and obstacles.

C. LIMITATIONS

First, the remote working condition made the planning and clarification of requirements complicated. The online implementation of the proposed method has suffered from the lack of eye contact with the subprojects' participants.

Second, there were intense interconnections among the services required for supporting the subprojects. Consequently, the planning of each subproject could not be done in isolation from the other ones.

Third, the novelty inherent to some subprojects, often accompanied by unclear requests and expectations of the employers, added uncertainties to the body of portfolio management. When this happens, the trouble is to precisely evaluate the employers' satisfaction with the progressing trend of the subproject.

Fourth, having to create an agile project management system concurrent with the development of the products pertaining to the subprojects constrained the monitoring and control of the execution of SCMP. Indeed, the project management office had to face the challenge of not having a customized system for coordinating the portfolio at the beginning of the analysis of the SCMP case study.

D. POSSIBLE EXTENSIONS AND FUTURE RESEARCH LINES

The performance of the teams within a portfolio could be compared in terms of the complexity rate of their done tasks. Since the strict or lenient attitude of product owners may influence their estimated story points at the time of planning, this latter concept could be normalized to disclose complexity.

We could also analyze the data from previous refinement reports to predict the performance of teams through the next releases and reduce the deviation between upcoming baseline and actual planning.

Finally, we could consider structured decision-making methods like the analytic hierarchy process (AHP) for computing the sprint length of subprojects. The proposed heuristic scoring method as well as the other maneuvers for maturing the proposed portfolio project management were designed considering the experimental setting provided by the real case study of SCMP. The use of AHP-oriented methods could represent an interesting alternative for the development of future extensions and research lines within other experimental settings.

VI. CONCLUSION

Coupling the concept of portfolio management with Scrum-based maturing frameworks constituted an important research gap in the literature. To fill the gap, this paper has applied a case study research-based methodology defining a set of procedures to bring about the maturity of a portfolio of a mega-scale software project. The case study has focused on the management of a project portfolio related to delivering social, entertainment, payment, and authentication-based platforms associated with a smart-based ecosystem.

The paper has illustrated how the case study evolved from a classical Scrum to an adaptive version encompassing measurable sprint lengths, multi-level refinements, and Scrum of Scrums capabilities. The adaptive version switched from goal-oriented Scrum subprojects to Scrumban request-based ones that developed tools for other subprojects.

The interaction of the goal-oriented framework with the systematic management capacity of teams' members led to a significant spike in the realization rate of releases and narrowed the gap between baseline and actual planning. A customized project management system was designed to map the requirements of one team that had to be developed by another team. This system provided the basis for conducting the Scrum of Scrums at the portfolio level of a wide range of subprojects.

The development of a workable scoring pattern to determine the sprint length of different subprojects is another significant contribution of our study. Managerial insights were provided to the key participants of the project through a complete set of refinement and reporting items at the different levels of the Scrum framework. In this regard, Scrum meetings incorporated a special quantitative feedback mechanism based on key performance indicators.

The methods described could significantly contribute to the literature on project portfolio management, while wider implications can also be defined as future research ideas. In this sense, a discussion has been provided regarding the advantages, managerial implications, and limitations of the performed analysis. Finally, a few future research directions have been drawn.

ACKNOWLEDGMENT

Samira Vazifeh-Noshafagh, Vahid Hajipour, and Sajjad Jalali would like to acknowledge the FANAP Studies and Research Center for scientifically supporting the present study and playing significant roles in shaping the project and subsequently the current paper.

REFERENCES

[1] P. Rosenberger and J. Tick, "Applicability of traditional project closeout approaches in agile developed IT projects," in *Proc. Design, Simulation, Manuf., Innov. Exchange*, in Lecture Notes in Mechanical Engineering, V. Ivanov, J. Trojanowska, I. Pavlenko, J. Zajac, and D. Peraković, Eds. Cham, Switzerland: Springer, May 2021, pp. 243–252, doi: 10.1007/978-3-030-77719-7_25.

[2] R. Sherman, *Business Intelligence Guidebook*. Boston, MA, USA: Elsevier, 2015.

[3] A. Sahid, Y. Maleh, and S. Mounir, "Towards an agile ITSM maturity framework for organizations: A case study," *EDPACS*, vol. 66, no. 6, pp. 1–21, Dec. 2022, doi: 10.1080/07366981.2022.2045541.

[4] A. Alliance, *Agile Practice Guide*. Newtown Square, PA, USA: Project Management Institute, 2017.

[5] A. Webb, *Agile Project Management Methodology for Beginners: Scrum Project Management for Beginners*. New York, NY, USA: Streetlib, 2017.

[6] D. Harned, *Hands-On Agile Software Development With JIRA: Design and Manage Software Projects Using the Agile Methodology*. Birmingham, U.K.: Packt Publishing, 2018.

[7] H. Kerzner, *Project Management Metrics, KPIs, and Dashboards: A Guide to Measuring and Monitoring Project Performance*. Hoboken, NJ, USA: Wiley, 2017.

[8] K. Schwaber, *Agile Project Management With Scrum*. London, U.K.: Pearson, 2004.

[9] K. Schwaber and J. Sutherland, "The scrum guide," Tech. Rep., Apr. 2020. [Online]. Available: https://www.scrumguides.org/scrum-guide.html

[10] A. Faber, A. Hernandez-Mendez, S.-V. Rehm, and F. Matthes, "An agile framework for modeling smart city business ecosystems," in *Proc. 20th Int. Conf. Enterprise Inf. Syst.*, vol. 2, Funchal, Portugal, Mar. 2018, pp. 39–50.

[11] T. J. Gandomani and M. Z. Nafchi, "An empirically-developed framework for agile transition and adoption: A grounded theory approach," *J. Syst. Softw.*, vol. 107, pp. 204–219, Sep. 2015.

[12] S. Beecham, T. Clear, R. Lal, and J. Noll, "Do scaling agile frameworks address global software development risks? An empirical study," *J. Syst. Softw.*, vol. 171, Jan. 2021, Art. no. 110823.

[13] J. Heimicke, G.-L. Ng, M. Krüger, and A. Albers, "A systematic for realizing agile principles in the process of mechatronic systems development through individual selection of suitable process models, methods and practices," *Proc. CIRP*, vol. 100, pp. 619–624, Jan. 2021.

[14] H. Lei, R. O'Connell, L. Ehwerhemuepha, S. Taraman, W. Feaster, and A. Chang, "Agile clinical research: A data science approach to Scrumban in clinical medicine," *Intell.-Based Med.*, vols. 3–4, Dec. 2020, Art. no. 100009.

[15] R. Sweetman and K. Conboy, "Portfolios of agile projects: A complex adaptive Systems' agent perspective," *Project Manage. J.*, vol. 49, no. 6, pp. 18–38, Dec. 2018, doi: 10.1177/8756972818802712.

[16] C. J. Stettina and J. Hörz, "Agile portfolio management: An empirical perspective on the practice in use," *Int. J. Project Manage.*, vol. 33, no. 1, pp. 140–152, Jan. 2015, doi: 10.1016/j.ijproman.2014.03.008.

[17] K. Nielsen, "Knowledge that sets you apart," in *Agile Portfolio Management: A Guide to the Methodology and Its Successful Implementation*. New York, NY, USA: Taylor & Francis, 2021.

[18] M. A. Razzak, I. Richardson, J. Noll, C. N. Canna, and S. Beecham, "Scaling agile across the global organization: An early stage industrial SAFe self-assessment," in *Proc. 13th Int. Conf. Global Softw. Eng.*, May 2018, pp. 116–125.

[19] M. O. Ahmad, L. E. Lwakatare, P. Kuvaja, M. Oivo, and J. Markkula, "An empirical study of portfolio management and Kanban in agile and lean software companies," *J. Softw., Evol. Process*, vol. 29, no. 6, p. e1834, Jun. 2017, doi: 10.1002/smr.1834.

[20] C. Kapic, "Managing the agile scalability to implement agile project portfolio management: A case study within the automotive industry," Student Thesis, School Ind. Eng. Manag., KTH Royal Inst. Technol., Stockholm, Sweden, 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-264080

[21] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen, "Large-scale agile transformation at Ericsson: A case study," *Empirical Softw. Eng.*, vol. 23, no. 5, pp. 2550–2596, Oct. 2018, doi: 10.1007/s10664-017-9555-8.

[22] K. Rautiainen, J. von Schantz, and J. Va, "Supporting scaling agile with portfolio management: Case Paf.com," in *Proc. 44th Hawaii Int. Conf. Syst. Sci.*, Jan. 2011, pp. 4–7.

[23] C. T. Cerdeiral and G. Santos, "Software project management in high maturity: A systematic literature mapping," *J. Syst. Softw.*, vol. 148, pp. 56–87, Feb. 2019, doi: 10.1016/j.jss.2018.10.002.

[24] C. Millhollan and M. Kaarst-Brown, "Lessons for IT project manager efficacy: A review of the literature associated with project success," *Project Manage. J.*, vol. 47, no. 5, pp. 89–106, Oct. 2016, doi: 10.1177/875697281604700507.

[25] J. Setiyawan, F. Gunawan, T. Raharjo, and B. Hardian, "Application of scrum maturity model: A case study in a telecommunication company," *J. Phys., Conf. Ser.*, vol. 1566, no. 1, Jun. 2020, Art. no. 012050, doi: 10.1088/1742-6596/1566/1/012050.

[26] A. Sukhoo, A. Barnard, M. M. Eloff, and J. A. V. D. Poll, "An evolutionary software project management maturity model for Mauritius," *Interdiscipl. J. Inf., Knowl. Manag.*, vol. 2, pp. 1–20, Jan. 2007.

[27] W. W. Yen, Y. K. Peng, and Y. S. Gee, "A case study assessment of project management maturity level in the Malaysia's IT industry," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, 2016, pp. 361–371.

[28] E. S. Hidalgo, "Adapting the scrum framework for agile project management in science: Case study of a distributed research initiative," *Heliyon*, vol. 5, no. 3, Mar. 2019, Art. no. e01447, doi: 10.1016/j.heliyon.2019.e01447.

[29] W. Zayat and O. Senvar, "Framework study for agile software development via scrum and Kanban," *Int. J. Innov. Technol. Manage.*, vol. 17, no. 4, Jun. 2020, Art. no. 2030002, doi: 10.1142/S0219877020300025.

[30] K. Vlaanderen, S. Jansen, S. Brinkkemper, and E. Jaspers, "The agile requirements refinery: Applying SCRUM principles to software product management," *Inf. Softw. Technol.*, vol. 53, no. 1, pp. 58–70, Jan. 2011, doi: 10.1016/j.infsof.2010.08.004.

[31] S. Kiv, S. Heng, Y. Wautelet, and M. Kolp, "Towards a goal-oriented framework for partial agile adoption," in *Software Technologies*, vol. 868, E. Cabello, J. Cardoso, L. Maciaszek, and M. van Sinderen, Eds. Cham, Switzerland: Springer, 2018, pp. 69–90, doi: 10.1007/978-3-319-93641-3_4.

[32] A. Raza, M. Tayyab, D. Shahid, D. Abdullah, and D. Muhammad, "Impact of story point estimation on product using metrics in scrum development process," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 4, pp. 1–7, 2017, doi: 10.14569/IJACSA.2017.080452.

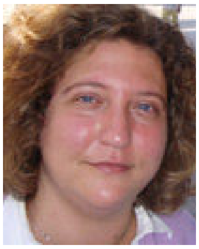
[33] A. Anand, J. Kaur, O. Singh, and O. H. Alhazmi, "Optimal sprint length determination for agile-based software development," *Comput., Mater. Continua*, vol. 68, no. 3, pp. 3693–3712, 2021, doi: 10.32604/cmc.2021.017461.

[34] J. Rossberg, "Agile project management in azure DevOps and TFS," in *Agile Project Management With Azure DevOps*, J. Rossberg Ed. New York, NY, USA: Apress, 2019, pp. 251–306, doi: 10.1007/978-1-4842-4483-8_8.

SAMIRA VAZIFEH-NOSHAFAGH received the master's degree in industrial engineering, in 2013. She professionally has focused on empowering her skills in the context of project management. In this regard, she has been the Head of PMO of a Mega-Scale Smart City Project, called POD, of FANAP Company, since 2016. She also coached more than 20 individuals to play a role as the Scrum Master to be capable of monitoring around 40 subprojects. During her working stint, she could successfully mature the POD's project management methodology from classical Scrum to the goal-oriented one through applied and heuristic techniques. Customized document and project management systems were developed under her supervision. Always welcoming putting theories into practice, she nowadays has delved deeply into the workable way of employing risk management in the agile life cycle using software agents.

VAHID HAJIPOUR received the B.S., M.S., and Ph.D. degrees in industrial engineering. He is currently an Assistant Professor of industrial and system engineering with Azad University, West Tehran Branch, and the Head of the FANAP Studies and Research Center, Iran. He also holds a visiting professorship position in industrial engineering at Altinbas University, Turkey. He has published two books and more than 60 papers in academic journals and conferences such as *Computers and Operations Research*, *Applied Mathematical Modelling*, *Computers and Industrial Engineering*, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, *Reliability Engineering and System Safety*, *Journal of Intelligent Manufacturing*, *Applied Soft Computing*, *International Journal of Systems Science*, *Engineering Optimization*, and *Measurement*. His research interests include supply chain network design, emerging technologies, reliability theory, computational intelligence, and enterprise solutions. He is a member of the Iranian National Elite Foundation and Young Researchers Club. He was a recipient of 2011, 2013, 2014, and 2015, Distinguished Young Researcher Awards; and the 2016 Distinguished Researcher Award at Azad University. He was inducted into Alpha Iota Delta (the International Honor Society for Decision Sciences and Information Systems), USA, in 2017.

SAJJAD JALALI is currently involved in FANAP Studies and Research Center endeavoring to spotlight the engineering knowledge and achievements of this large-scale Iranian Technology Leader through prolific publications. During his academic stint, he majored in industrial engineering. In 2018, he successfully defended his Ph.D. thesis at the Azad University of Qazvin concentrating on the risk-averse reliable design of the supply chain structures. He has also taught different Industrial Engineering courses at the Azad University of Shahrood. In addition to authoring several papers, he is the Elsevier Reviewer at scholarly journals such as *Transportation Research Part* E besides *Computers and Operations Research*. His research interests include enterprise solutions, game theory, crowdsourcing, and project management as well as decomposition and heuristic-based algorithms.



DEBORA DI CAPRIO received the Ph.D. degree in mathematics from York University, Canada. She is a Visiting Researcher with York University. She is currently an Assistant Professor of mathematical economics with the University of Trento, Italy. She is also a Researcher with the International Business and Markets Group, Universidad Complutense de Madrid (Spain); the Group on Big Data and Artificial Intelligence of the Spanish Society of Nephrology; and the ARES (ARtificial IntElligence for better tranSplant) Research Group, Hospital Clinic—IDIBAPS of Barcelona (Spain). She has published over 100 peer-reviewed articles in the areas of mathematics, decision theory, and operations research. She is a Department Editor (Descriptive Analytics) of *Healthcare Analytics* and an Associate Editor of *Expert Systems with Applications*, *Intelligent Systems with applications*, *Decision Analytics Journal*, *Space Mission Planning and Operations*, the *International Journal of Enterprise Information Systems*, and the *International Journal of Strategic Decision Sciences*. She is also an Editorial Board Member of the *International Journal of Applied Decision Sciences*, the *International Journal of Management and Decision Making*, and the *Journal of Applied Intelligent Systems and Information Sciences*.

FRANCISCO JAVIER SANTOS-ARTEAGA received the Ph.D. degree in mathematical economics from York University, Canada, and the Ph.D. degree in applied economics from the Universidad Complutense de Madrid, Spain. He is currently a Researcher with the Department of Análisis Económico y Economía Cuantitativa, Universidad Complutense de Madrid. He is also a Researcher with the International Business and Markets Group, Instituto Complutense de Estudios Internacionales; the Group on Big Data and Artificial Intelligence of the Spanish Society of Nephrology; and the ARES (ARtificial IntElligence for better tranSplant) Research Group, Hospital Clinic—IDIBAPS of Barcelona (Spain). He was awarded the Dean's Academic Excellence Award from York University. He has published over 120 peer-reviewed papers in the areas of decision theory and operations research. He is a Department Editor (Prescriptive Analytics) of *Healthcare Analytics*, an Associate editor of *Decision Analytics Journal*, *Space Mission Planning and Operations*, the *International Journal of Enterprise Information Systems*, the *International Journal of Strategic Decision Sciences*, and *Fuzzy Optimization and Modeling*, and also an Editorial Board Member of the *International Journal of Applied Decision Sciences*, the *International Journal of Management and Decision Making*, and the *Journal of Applied Intelligent Systems and Information Sciences*.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement