IMPROVING WORKLOAD ESTIMATION IN INEXPERIENCED TEAMS WITH HYBRID AGILE APPROACH

Fabiano S. Pires, Cícero A. L Pahins and Paulo Fonseca

Sidia R&D Institute, Manaus, AM, Brazil

ABSTRACT

SCRUM framework is an agile technique that is widely used by development teams in order deliver incremental value to customers and dynamically react to project needs. SCRUM framework might be adapted to conform the development team specificities. In the context of an industry project, we have found that an inexperienced development team frequently faced difficulties with estimating the time needed to complete tasks, which led to missed deadlines in most of the projects. Such problem hampers the risk management and degrades the relationship with the customer. Upon closer analysis, it was identified that the main reason to this issue was the team's inability to breaking down a larger task into smaller sub-tasks and associate a realistic workload to each part. Then, based on traditional techniques, a structured approach to workload estimation was introduced in the SCRUM planning meeting to leverage the team's estimation skill. This approach was implemented in two development projects and increased the accuracy in the estimate defined by the team, yielding realistic schedules and increasing technical visibility.

KEYWORDS

Agile, Hybrid Agile Approach, Management, Scrum, Inexperienced Teams, PMBOK, Workload Estimation.

1. INTRODUCTION

Novel business models, strategies, technologies, and global transformations have changed the way companies manage their businesses in an increasingly competitive market. This scenario thrives the companies to adopt agile-oriented attitudes in most aspects of their workflow, e.g., project and team management. The search for efficiency in management activities has assumed a fundamental role to overcome daily faced challenges and strengthen success chances. Nevertheless, off-the-shelf strategies frequently were unable to meet all requirements and internal policies of large companies, leading to novel management models.

In a scenario of sustained unpredictability, where projects suffer from constant scope changes, agile methods tend to reflect stakeholders' requirements along the project course. During the last few years, the Scrum framework becomes standard for most companies in the software industry, despite presenting challenges in the planning phase [1]. Another challenge is its dependency on empirical methods based on developers' experience to estimate project tasks accurately. Deriving and estimation tasks from backlog items are complex activities that are even more challenging to inexperienced teams, typically composed of recently graduated professionals with limited industry and software development experience. According to Cerpa et al. [2], Scrum's premises

David C. Wyld et al. (Eds): NLP, JSE, MLTEC, DMS, NeTIOT, ITCS, SIP, CST, ARIA - 2020
pp. 345-352, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101426

contribute to creating high-risk scenarios on projects developed by inexperienced teams since unrealistic schedules may result in unplanned costs that increase the probability of failures.

Another important factor of concern is the alignment with the client's expectations. Software clients typically expect developers to deliver high-quality projects, on the schedule, and at the lowest cost [3]. Most of Scrum's ceremonies are related to these expectations, mainly those designed to create a well-defined product and sprint backlogs. The product backlog is a document that contains all required features of a project that later was refined and collected into the sprint backlog. Inexperienced teams may have difficulty creating the backlogs since it also depends on the ability acquired on previous projects. A delay, or even poorly documented backlogs, can consequently impact the perceived quality associated with project deliveries and the result.

In order to improve workload estimation in inexperienced teams during the development of complex software projects, we designed and evaluated a hybrid agile approach that combines both (i) Scrum agile-oriented principles and (ii) PMBOK (Project Management Body of Knowledge) planning practices. This combination decreases the risk of missing deadlines due to inaccurate workload estimation. In addition to theoretical guidance in designing our approach, we took advantage of the lessons learned on two real-world projects from a large software company with well-defined policies and high client expectations about the development course deliveries.

Our paper is organized as follows: Section 2 presents the background information about traditional, agile, and hybrid project management approaches and discusses related work. Section 3 provides the context of our proposal. Section 4 describes our hybrid model and how it combines both PMBOK and Scrum aspects to produce a structured workload estimation that can be used by inexperienced developers. Section 5 discusses the lessons learned while implementing our model in real-world projects from a sizeable mobile-related software development company. We conclude the paper with a discussion of results and possible avenues for future work in Section 6.

2. RELATED WORK

In the last two decades, agile methods increased their popularity and were established as primary software development methodologies. A large number of efforts [4-7] were focused on studying their application under different circumstances and evaluate their advantages and shortcomings. Felker et al. [8] evaluated the implementation of the Scrum framework for both UX and software development in an undergraduate team without Scrum's previous experiences. Felker et al. describe multiple challenges the team faced, such as difficulty deciding Sprint length, balancing the amount of new functionality and a bug fix in each sprint, and estimating the time required to complete each task. Most of these problems are due to the team's lack of experience with the Scrum framework. To cope with these issues, the authors suggest allocating more time than the team initially suggested for some tasks and planning each task in a detailed fashion.

An approach to deal with estimating effort during planning sessions is to use computational models that automatically predict the required effort. Bilgaiyan et al. [9] review the most relevant works with this common goal and found that most of them use machine learning techniques, such as neural networks and support vector regression. These models can achieve accurate results. However, they require an extra effort of model training and fine-tuning its hyper-parameters. Besides, these approaches do not help an inexperienced team develop the team's skill in estimating task complexity and effort.

Another way to handle the challenges during Scrum implementation is to adapt and modify the standard process. This approach has the advantage of engaging team members in improving the team's deficiencies. Graphenthin et al. [1] facilitates task breakdown by using a meeting room with four whiteboards. Each board is used to analyze a single aspect, i.e., business, integration, data, and interaction) of a backlog item. Hayata et al.[10] propose a hybrid methodology that uses traditional software life cycle methodologies during the initial (e.g., requirements analysis, documentation) and final (e.g., testing phases of a project)t, whereas using Scrum for the design and implementation phases. These methods have the advantage of developing a team's skills; however, they add overhead to the usual process, e.g., an interaction room, detailed documentation, and still depend on expert's participation for success, e.g., waterfall practitioners.

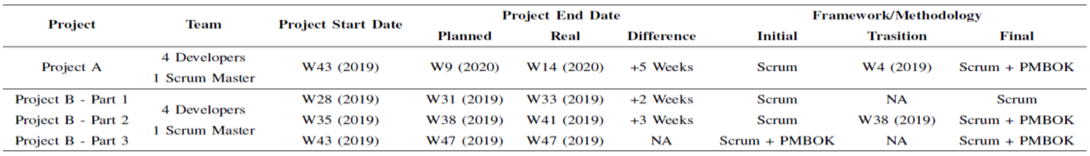
In [12], the authors assess review recommenders ensuring expertise during the review, reducing the core team's review workload, and the turnover risk. For this, they implement a recommender system that combines learning and retention; the aware recommenders effectively reduce the risk of turnover. In [13], the authors propose a hypothetical work commitment model to agile programming improvement groups. Utilizing auxiliary condition demonstrating, we found that agile practices lessen work requests (saw remaining task at hand and job vagueness) and backing position assets (saw importance and occupation self-governance). In [14], it is investigated how to improve the information on the best way to quantify productivity being developed groups where many inconstancies may exist because of the human factor. The primary spotlight is on disclosing the entire cycles and analyzing them as far as proficiency and adequacy. Like this, the authors uncover conceivably shrouded expenses and dangers to make remedial moves practically during the product venture life cycle. This work uses PMBOK's Work Breakdown Structure definition to provide a structured approach for workload estimation that can be used by inexperienced developers.

3. FROM SCRUM TO A HYBRID MODEL: TEAM PROFILE

The design of our hybrid model was based on projects that ran in a large mobile-related company that is mainly responsible for customizing the Android Operating System to Latin American countries with the requirements of its clients, e.g., device manufacturers and mobile carriers.

In this experience report, we focused on two projects that were performed during 2019 and were developed by two different developers' teams, as shown in Table 1. Each team was composed of four developers responsible for carrying out all technical activities, with an average of six months of experience in software development, and a member in the Scrum Master's role. As suggested in the Scrum framework, both teams were multidisciplinary and had all the necessary technical skills to carry out all the projects. Another premise was that both teams worked in a self-organized way: themselves defining, within the organization's context and the Scrum structure, how to perform the tasks and how to manage the progress towards the goals agreed with the Product Owners.

Table 1: Overview of different projects using (i) scrum only, (ii) transitioning from scrum to hybrid model, and (iii) hybrid model



Project	Team	Project Start Date	Project End Date			Framework/Methodology		
			Planned	Real	Difference	Initial	Trasition	Final
Project A	4 Developers 1 Scrum Master	W43 (2019)	W9 (2020)	W14 (2020)	+5 Weeks	Scrum	W4 (2019)	Scrum + PMBOK
Project B - Part 1	4 Developers 1 Scrum Master	W28 (2019)	W31 (2019)	W33 (2019)	+2 Weeks	Scrum	NA	Scrum
Project B - Part 2		W35 (2019)	W38 (2019)	W41 (2019)	+3 Weeks	Scrum	W38 (2019)	Scrum + PMBOK
Project B - Part 3		W43 (2019)	W47 (2019)	W47 (2019)	NA	Scrum + PMBOK	NA	Scrum + PMBOK

The first project, hereafter called Project A, consisted of 3 phases, where each one was associated with a release with well-defined functionalities. In July 2019, at the beginning of Project A, the development team defined an initial Sprint backlog that estimated 2 Sprints (4 weeks) to deliver the first project phase. However, during Sprint 1, unforeseen tasks were added to the planned stories, which increased their complexity. This led to the Sprint failure and increased the number of stories that needed to be completed in Sprint 2 to meet the defined deadline. During the Sprint retrospective, the team could not detect the leading cause of the deadline missing and assumed that increasing working hours would be enough to complete the Sprint 1 remaining tasks and Sprint 2 planned stories. Nevertheless, Sprint 2 occurred, and the did not planed tasks were added to the Sprint, causing another failure. Such behavior led to an increasing of 2 weeks in the first phase and three weeks in the second phase of the project, representing a 50% error in the development team's estimation. In the first two phases, it is possible to observe sufficient technical knowledge to meet the project's development demand. Scrum ceremonies performed according to the recommended guideline, the correct scope definition, task duration, and effort estimates were insufficient.

In this context, we designed a hybrid model to mitigate the workload estimation errors that were typically committed by an inexperienced team of developers. We detail the design of our hybrid model in Section 4. We also considered the lessons learned through the transition of Scrum only to our hybrid project management approach, as illustrated in Table I. Note that, in every transition to the hybrid model, both the scope and schedule of the projects were revisited. The revised scopes consisted of product backlog reviews and WBS (Work Breakdown Structure) creations. In comparison, the revised schedules were concerned about the definition, sequencing, and estimation of the tasks.

4. DESIGNING A HYBRID MODEL

Scrum is a framework based on empirical theories of process control. The knowledge comes from both experience and evidence manners, and their progress is formed on observations of reality, thus generating a solid premise that the team needs to be multidisciplinary and formed by experienced people.

Chow et al. [11] surveyed multiple software projects and concluded that agile techniques are one of the critical factors for project success. Thus, our goal designing a hybrid model was two-fold: i) Minimize the changes to the Scrum guidelines that were necessary to cope with the team's inexperience in order to retain Scrum's advantages and ii) improve the team's estimation skill so that, eventually, the team was capable of returning to Scrum original guidelines.

Therefore, our proposal consists of using concepts from the PMBOK, a globally recognized guide to the best practices in project management, and applying it to the agile-oriented principles and ceremonies of the Scrum Framework to design a hybrid project management model offers a structured workload estimation to inexperienced developers. By combining both PMBOK and Scrum, we aimed to provide a more elaborated project scope by using SCRUM best practices during initial phases, e.g., *Initiation* and *Planning*, and *Scrum ceremonies* during execution, as shown in Figure 1.

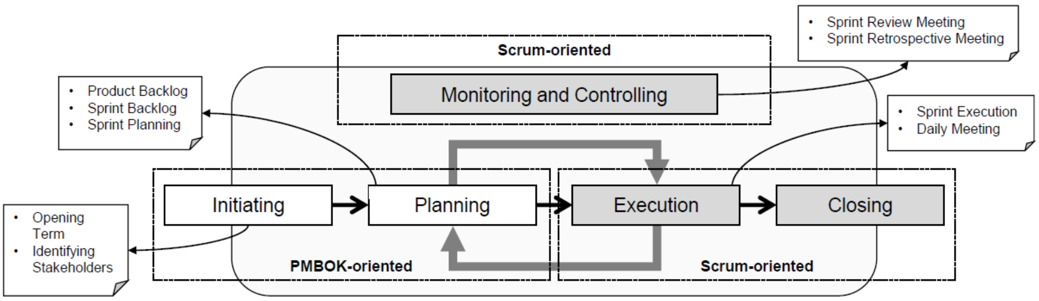


Figure 1: Model Lifecycle with PMBOK and SCRUM.

Our hybrid model combines both (i) PMBOK best practices of project management during initial phases with (ii) Scrum agile-oriented principles and ceremonies of execution to provide a structured approach for workload estimation that can effectively assist teams of inexperienced developers. An essential aspect of our proposal is that it prevents the definition of workload without ensuring the correct task prioritization and knowledge of critical paths that are not under the development team's responsibility, which is difficult to promise on empirical methods of estimates when conducted by inexperienced teams. A concern while designing the hybrid model was to maintain Scrum's premises, such as:

1. not bureaucratizing the process.
2. not excessively and unnecessarily documenting, and
3. not creating or using unnecessary processes to avoid adding slowness to the project's execution is a commonly criticized aspect of traditional methodologies.

In this context, our hybrid model attempts to avoid traditional project management's main disadvantages that are often so bureaucratic that a project's requirements may change even before development begins [12]. To refine our approach, we also considered the lessons learned through the transition of Scrum only to our hybrid project management model, as illustrated in Table 1. Note that, in every transition to the hybrid approach, both the project's scope and schedule were revisited. The revised scopes consisted of product backlog reviews and WBS creations. Simultaneously, the revised schedules were concerned about the definition, sequencing, and estimation of the tasks.

As depicted in Figure 2, our hybrid model implements a set of tools and techniques suggested by PMBOK during the Planning Phase to ensure a more reliable project scope definition by structuring the (i) scoping and (ii) schedule generation processes:

1. **Work Breakdown Structure (WBS) Creation:** developed to establish a common understanding of the project scope by creating a decomposition of the work into easily manageable parts called work packages, estimated with further exactness.
2. **Activities Definition:** creates the activities list (the work packages broken down into the activities needed to produce them) and the list of activity attributes (information related to the activities). These activities can be considered as actions that need to be performed to execute each work package
3. **Activities Sequencing:** Studies of the relationship between project activities determine the logical Sequence that serves as the basis for the project schedule.
4. **Activities Duration Estimation:** estimates the duration of each activity based on a three-point method (PERT), i.e., the best-case, most likely, and worst-case estimates.

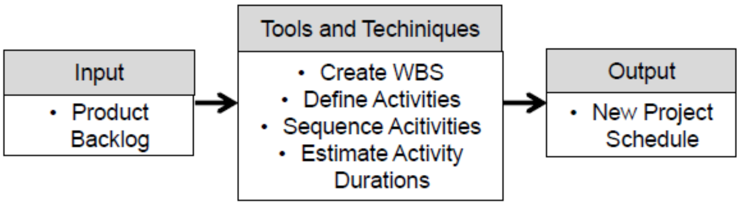


Figure 2: Set of tools and techniques suggests by PMBOK that are implemented in our hybrid model to provide a structured approach for workload estimation.

The designed model provides a structured approach to planning sessions while the remaining Scrum ceremonies (daily meeting, sprint review, and sprint retrospective) are performed without changes. The team does not need to have an extensive knowledge of PMBOK to apply these changes. Besides, given that PMBOK is a widely known process management guide, many companies may use existing collaborator knowledge to leverage such a hybrid model.

5. LESSON LEARNED

We applied the hybrid model in the phase 3 of the Project A. During the planning meeting, the *WBS creation activities definition* helped the team in obtaining a more detailed vision of required tasks, whereas *activities sequencing* helped them visualize inter-dependency among tasks that were not obvious at first sight. The PERT approach to effort estimation also assisted the team by leading them into thinking of all possible scenarios.

Using this model, the team estimated this project phase would require 5 weeks (3 sprints) to be completed. The team was able to meet the planned deadline and no additional tasks were observed during this period. In contrast to the observed in previous phases of the project, we noted that the designed model helped to leverage the team's estimation skill and yield planning sessions with improved accuracy.

The same pattern of task underestimation was observed in another development project, hereafter called Project A. After consecutive sprint failures and unforeseen tasks being added to the sprint, when there were 5 weeks remaining until the next release, it was clear that the project would miss its initial deadline. The hybrid model was applied to generate a new deadline and improve the detailing of product backlog and estimates. During the planning session, the team anticipated unforeseen tasks and interdependence, and estimated that it would require 10 weeks to complete the remaining stories. The team was able to meet the new deadline with only a few tasks being added to the sprints.

By combining (i) the best practices suggested by the PMBOK Guide for task definition and estimation and (ii) the Scrum agile-oriented principles and ceremonies, we notice that our hybrid model was able to effectively provide a more elaborate project scope during the initial phases of *Initiation* and *Planning*.

We also observed that our hybrid model helps to verify that companies with agile-oriented environments can benefit from the use of more traditional concepts brought by the PMBOK Guide without having to necessarily ignore the SCRUM's dynamism, thus being able to add a layer of maturity to the planning stage of the projects by eliminating the necessity to have self-managed and experienced teams.

Another important gain was to provide a *win-win* model for the team, in which the technical members can improve their task estimation skills in a structured approach, with no need of an external tool for effort estimation such as those discussed in Section 2. On the other hand, the *Project Manager* can benefit from mature and precise planning that guides the development team into producing within an agile structure.

6. CONCLUSIONS

This work addressed the challenge of accurately estimate effort during the Scrum planning meeting when there are no experts in the development team. We presented a hybrid project management model that assists inexperienced teams in workload estimation and minimizes the risk of missing deadlines. This model was implemented in two real-world software projects, and it was effective in improving the accuracy of effort estimation. We evaluated that traditional software development models can be useful in tackling Scrum challenges in a structured way and might be reproduced by other teams with similar issues. Besides improved accuracy, we also observed that this hybrid model retains Scrum qualities, such as dynamism and teams' autonomy, whereas developing teams' estimation skills leads to increasingly accurate planning meetings. Finally, another contribution was the understanding that new hybrid approaches to project management can be built and used to meet the most diverse demands, without needing to be limited to a single software development model.

In the future, we intend (i) to study how our model copes with developing a team of inexperienced developers in the long-term; (ii) to combine the PMNOK with other agile methodologies, for example, Extreme Programming-XP; (iii) validate our model with others contexts of applications and companies.

ACKNOWLEDGEMENTS

This work was partially supported by Samsung Eletrônica da Amazonia Ltda, under the auspice of the informatic law no 8.387/91. This work was supported by the Sidia: Institute of Science and Technology.

REFERENCES

[1] S. Grapenthin, S. Poggel, M. Book, and V. Gruhn, "Facilitating task breakdown in sprint planning meeting 2 with an interaction room: An experience report," in 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, Aug 2014, pp. 1–8.

[2] N. Cerpa and J. M. Verner, "Why did your project fail?" Commun. ACM, vol. 52, no. 12, p. 130–134, Dec. 2009. [Online]. Available: https://doi.org/10.1145/1610252.1610286.

[3] A. Tonini, M. Silva, and M. Spinola, "Software expectation management by means of service level agreements in software maintenance," in 19th International Conference on Production Research. Available: http://www. icpr19. cl/mswl/Papers/252. pdf. Citeseer, 2007.

[4] I. Zada and S. Shahzad, "Issues and implications of Scrum on global software development," Bahria University Journal of Information & Communication Technologies (BUJICT), vol. 8, no. 1, 2015.

[5] L. Silva, C. Santana, F. Rocha, M. Paschoalino, G. Falconieri, L. Ribeiro, R. Medeiros, S. Soares, and C. Gusm˜ao, "Applying xp to an agile–inexperienced software development team," in International Conference on Agile Processes and Extreme Programming in Software Engineering. Springer, 2008, pp. 114–126.

[6] E. Hossain, M. A. Babar, and H.-y. Paik, "Using Scrum in global software development: a systematic literature review," in 2009 Fourth IEEE International Conference on Global Software Engineering. IEEE. 2009, pp. 175–184.

[7] J. L´opez-Mart´ınez, R. Ju´arez-Ram´ırez, C. Huertas, S. Jim´enez, and C. Guerra-Garc´ıa, "Problems in the adoption of agile-Scrum methodologies: A systematic literature review," in 2016 4th international conference in software engineering research and innovation (conisoft). IEEE, 2016, pp. 141–148.

[8] C. Felker, R. Slamova, and J. Davis, "Integrating ux with Scrum in an undergraduate software development project," in Proceedings of the 43rd ACM technical symposium on Computer Science Education, 2012, pp.301–306.

[9] S. Bilgaiyan, S. Mishra, and M. Das, "A review of software cost estimation in agile software development using soft computing techniques," in 2016 2nd International Conference on Computational Intelligence and Networks (CINE), Jan 2016, pp. 112–117.

[10] T. Hayata and J. Han, "A hybrid model for it project with Scrum," in Proceedings of 2011 IEEE International Conference on Service Operations, Logistics, and Informatics. IEEE, 2011, pp. 285–290.

[11] H. F. Cervone, "Understanding agile project management methods using Scrum," OCLC Systems & Services: International digital library perspectives, vol. 27, no. 1, pp. 18–22, Feb. 2011. [Online]. Available: https://doi.org/10.1108/10650751111106528.

[12] Ehsan Mirsaeedi and Peter C. Rigby. 2020. Mitigating turnover with code review recommendation: balancing expertise, workload, and knowledge distribution. In <i>Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering</i> (<i>ICSE '20</i>). Association for Computing Machinery, New York, NY, USA, 1183–1195. DOI:https://doi.org/10.1145/3377811.3380335

[13] HUCK-FRIES, Veronika et al. The Role of Work Engagement in Agile Software Development: Investigating Job Demands and Job Resources. In: Proceedings of the 52nd Hawaii International Conference on System Sciences. 2019.

[14] CALDEIRA, João et al. Assessing Software Development Teams' Efficiency using Process Mining. In: 2019 International Conference on Process Mining (ICPM). IEEE, 2019. p. 65-72.