Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 219 (2023) 1634–1641

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022

Agile teams' assignment model for Scaling Agile

Vitor Anes[a,b,c*], António Abreu[a,d], Ana Dias[a,b], João Calado[a,b,c]

[a]Instituto Superior de Engenharia de Lisboa, R. Conselheiro Emídio Navarro 1, 1959-007 Lisboa, Portugal
[b]Instituto Politécnico de Lisboa, Estr. de Benfica 529, 1549-020 Lisboa, Portugal
[c]Instituto de Engenharia Mecânica (IDMEC), Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[d]Centro de Tecnologias e Sistemas (CTS)–Uninova, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2825-149 Caparica, Portugal

Abstract

Adherence to the schedule for completion of a project is one of the basic requirements for the success of a management team. Failure to meet this schedule can have a negative impact on both the business side and the reputation of the company, this issue is a constant challenge in project management. This challenge becomes even greater with agile project management. Agile methodology has increasingly become the method of choice for project delivery. However, in this type of project, uncertainty increases in terms of execution time due to the uncertainty that arises from the agile characteristics that are so valued. Optimized assignment of agile teams based on their performance to project tasks can reduce this uncertainty and increase the probability of success. However, it appears that the tools available in the literature have very limited applicability to this problem.
In this paper, we present a model for agile team assignment that uses the Program Evaluation Review Technique (PERT) along with the effective risk model and the Monte Carlo model. An illustrative case study was used to illustrate the application of the proposed model. The results show that the project execution time is highly dependent on the performance of the agile teams and their respective assignment to the project tasks. In practice, the proposed model enables the determination of the execution time of a given agile project considering the performance of the respective teams.

* Corresponding author. Tel.: +351-218-317-000 ; fax: +0-000-000-0000 .
E-mail address: vitor.anes@isel.pt

Keywords: Agile; Scaling Agile; Agile teams assigment;PERT; Effective Risk; Monte Carlo; Project management

1877-0509
10.1016/j.procs.2023.01.456

1. Introduction

Agile at scale is a current trend in project management. Agile principles efficiently promote project management, increase delivery quality and customer satisfaction. With this in mind, companies have adopted this management method at the team level and are applying agile principles and practices to the entire organization. The benefits are many: shortening the product delivery cycle and increasing the corresponding quality, increasing engagement and collaboration between the customer and the company's work teams, identifying problems, risks and processing errors in all phases of the project [1–3]. Despite the undeniable benefits, implementing agile methodology at the organizational level is still a challenge. In particular, shortening the duration of agile sprints and creating a strategy for collaboration between teams are challenges that have not yet been overcome and have a negative impact on the duration of projects, which in turn negatively affects the profitability of the company and customer satisfaction [4].

1.1. Problem definition

Projects involving multiple agile teams must be planned so that project tasks can be distributed among the various teams. Within each task, the teams apply agile methods to achieve the goals set for completing the task. However, in projects involving multiple teams, there are tasks that take precedence over others. When the next task is performed by another team, the problem arises that one team has to wait until the other is finished to start working on its assigned task. This waiting time can cause the project duration to exceed the expected deadline, resulting in penalties and wasted resources [5]. In projects developed by a single agile team, this problem does not occur because the team depends only on itself to develop the work in each sprint and is able to compensate for any delays during the various sprints of the project. In large agile projects, each task can be considered as a project developed by a specific team. In this sense, and depending on the history of each agile team, there may be teams that are better prepared for a particular task than others. Therefore, properly assigning each agile team to each project task can lead to easier and more efficient project management with better results.

K. Dikert et al. [6] conducted a literature review on the challenges and success factors in large agile projects and identified five main factors that negatively impact the management of large agile projects, namely: 1) lack of best practices and low capacity in implementing agile methods, 2) poor coordination, collaboration, and management of dependencies between agile teams, 3) setting sprints without looking at the big picture of the project, 4) lack of management commitment and alignment of the entire organization to the large agile project, and 5) unrealistic delivery velocity expectations.

When we analyze these five factors, we conclude that they all have a direct impact on the timeliness of the project. The lack of best practices, poor implementation of agile methods, and poor communication between teams can lead to rework that causes delays in achieving the goals of each task. Lack of a general framework for tasks can lead to poor communication between teams and also result in rework. Insufficient commitment from both the agile teams and the organization can negatively impact creativity and problem-solving performance and increase the time required to complete tasks. Unrealistic expectations regarding the time needed to complete each task can put pressure on the team, which can affect the quality of the results achieved on each task. We therefore conclude that for large agile projects, the execution time for each task is an extremely important factor in managing these scaling agile projects. Properly allocating agile teams according to their performance on a given task can be a fundamental strategy to prevent the effects of delays in project tasks. In this sense, a decision support tool that can systematically and easily optimize the allocation of agile teams considering their performance is beneficial both for managing large agile projects and for analyzing the performance of individual teams, which in turn enables improvement actions regarding team performance.

2. Description of the proposed model

In this section, we propose a model for assigning agile teams to scaling agile projects that considers the capacity of each team for each task assigned to it. Teams are assigned using Monte Carlo modeling [7] of the method PERT [8], which can be used to determine the critical paths in a statistical way. The novelty of the proposed model is that the beta distributions traditionally used in the model PERT are modified according to the performance of each team. In this modification, the beta functions are replaced by normal distributions and the mean value of these distributions is updated according to the performance of the team. Then, based on the PERT paths and the combinatorial analysis, different scenarios are created for the assignment of the teams. These scenarios are then evaluated using the Monte Carlo model and the assignment of teams that minimizes the execution time of the project is determined. Fig. 1 shows the proposed conceptual model.

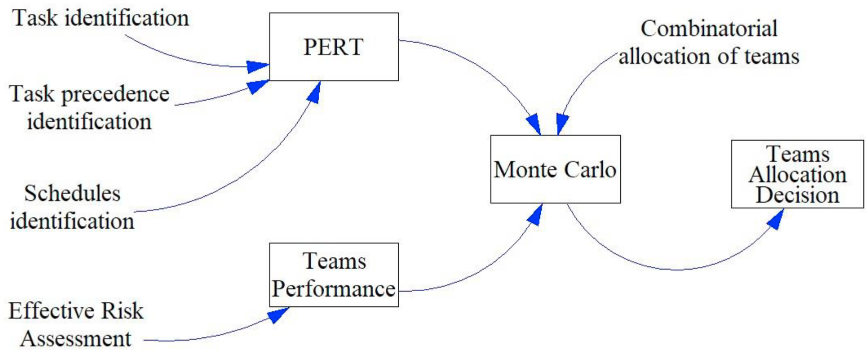

Fig. 1 - Conceptual model of agile team allocation in large projects.

2.1 Description of the model steps

Step 1 - Identification of the project tasks. The first step is to identify the project tasks and define the parameters of the corresponding beta distributions, namely the minimum, the most likely and the maximum execution time. These parameters can be set based on the experience of the agile project manager or based on commonly accepted standard times for the identified type of tasks.

Step 2 - Determine the priority of the project tasks. In this step, the priority between tasks is determined based on the dependency that one task starts when another task finishes.

Step 3 - Identify schedules. Based on the result of Step 2, the various project schedules are identified. For projects with teams assigned to tasks without precedence, they may be executed simultaneously, resulting in different schedules.

Step 4 - Evaluate the performance of each agile team. The performance of each agile team is evaluated using the effective risk model proposed by V. Anes et al. [9].

The effective risk model was developed with the goal of correcting two limitations associated with the risk priority number (RPN) function developed in the original failure mode and effects analysis approach, better known by the acronym FMEA. This function aims to prioritize failure modes according to the values assigned to the variables of risk, severity, occurrence, and detectability on a scale of 1 to 10. This risk model is well known and used in practice but has some limitations that may bias the results. These limitations are non-injectivity and non-surjectivity, i.e., the RPN function used to prioritize failure modes in FMEA is not continuous, and the same FMEA score can be obtained for different values of the risk variables [10].

To overcome these limitations, V. Anes et al. developed the risk isosurface function. This function is as easy to use as the RPN but has the advantage of not having the problems of the RPN. In addition, weights can be assigned to the risk variables, which is not possible with the RPN function. Another limitation of the RPN function is the use of the detectability variable as a means to quantify the mitigation capacity in a given failure mode. In fact, this variable must be split into two parts, one quantifying the ability to detect a failure mode before it occurs and the other quantifying the ability to mitigate or completely avoid the failure mode in question.

In this sense, the qualitative variables of reliability, availability, resilience, and robustness were considered as a way to quantify the ability to mitigate a particular failure mode. Combining the risk isosurface function with the 4 performance variables yielded the effective risk model used to quantify the risk level of failure modes. Fig. 2 shows the effective risk model in a schematic diagram.

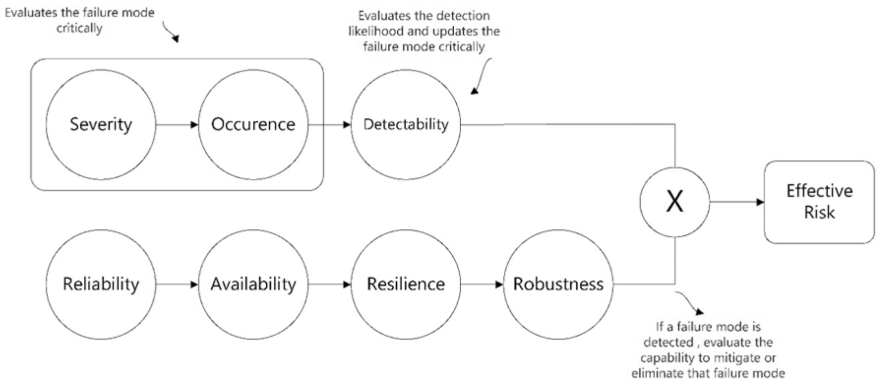

Fig. 2 – Effective Risk conceptual diagram.

Eq. (1) represents the effective risk expression used to prioritize failure modes by risk. The weighting of each qualitative risk variable is based on its position in Eq. (1), i.e., the position of each variable in Eq. (1) determines its order of importance in evaluating the risk associated with each failure mode.

$$RI\left(x_1, x_2, ..., x_n\right)_{x_1 > x_2 > ... > x_n} = \sum_{i=1}^{n} \left(x_i - 1\right) \cdot \mu^{n-i} + 1 \tag{1}$$

where $x_1, x_2, ..., x_n$ are risk variables, μ is the scale rating maximum value, and $x_1 > x_2 > ... > x_n$ is the order of importance between risk variables, and n is the total number of risk variables. Eq. (2) shows an example of Eq. (1) set for a four variables risk space.

$$RI(x_1, x_2, x_3, x_4)_{x_1 > x_2 > x_3 > x_4} = \left(x_1 - 1\right) \cdot \mu^3 + \left(x_2 - 1\right) \cdot \mu^2 + \left(x_3 - 1\right) \cdot \mu + \left(x_4 - 1\right) + 1 \tag{2}$$

The model proposed in this article considers the failure modes presented by K. Dikert et al. (summarized in section 1.1) together with the effective risk model. In this perspective, these factors that have a negative impact on project execution time are considered as risk factors that allow evaluating the performance of each agile team on a given task.

Step 5 - Introduce the Program Evaluation Review Technique (PERT). In this step, the PERT model is implemented in its traditional form with the beta distributions identified in Step 1.

Step 6 - Introduction of the Monte Carlo Model. In the Monte Carlo method, a large sample of stochastic events is run to obtain the appropriate statistical distribution based on that sample, in this sense, the Monte Carlo method is applied to the PERT model implemented in the previous step. In this step, the beta distributions are replaced by normal distributions whose mean and standard deviation are determined by the parameters of the beta distributions. Normal distributions follow the events resulting from human interaction very well and fit the pattern of agile task execution very well. However, normal distributions are difficult to identify without statistical data. In contrast, identifying the minimum, most likely, and maximum values of beta distributions is intuitive and easy. For this reason, Beta distributions have been retained in the model proposed here to facilitate the identification of normal distributions in Monte Carlo modeling.

The mean of the normal distribution corresponds to the most probable value obtained for the beta distribution, and the standard deviation is obtained based on the concept of central moments, as shown in Eq. (3).

$$sd = \sqrt{\frac{\min^2 + \max^2 + \mathrm{mod}^2 - \min\cdot\max - \min\cdot\mathrm{mod} - \max\cdot\mathrm{mod}}{18}} \tag{3}$$

Where *min*, *mod*, and *max* stand for the optimistic execution time (the lowest expected time for executing a task), the most probable (the most likely execution time for executing a particular task), and the pessimistic execution time for a particular task (the highest time for executing a particular task), respectively, and *sd* stands for the standard deviation. Based on the results of Step 4, the mean values of the normal distributions are then updated according to the team's performance. If the team's performance on a particular task falls short of expectations, the mean of the respective normal distribution shifts toward the maximum value set for that task in Step 1. If the team's performance is above average, the mean value of the normal distribution shifts towards the minimum value set in step 1 for the task. Eq. (4) shows the correction of the mean of the normal distribution according to the effective risk (represented in Eq. (4) by x).

$$\mu = \begin{cases} Likely + (Optimistic - Likely)\cdot(2x-1)\ldots x \geq 0.5 \\ Likely - (Likely - Pessimist)\cdot(1-2x)\ldots x \leq 0.5 \end{cases} \tag{4}$$

Step 7 - Assigning teams to schedules. In this step, the teams are assigned to the schedules determined in step 3 according to the number of teams available for the project. This assignment is done through a combinatorial analysis without repetitions, resulting in a number of analysis scenarios equal to the number of possible combinations.

Step 8 - Run the Monte Carlo model for each assignment scenario. In this step, apply the Monte Carlo method to each of the assignment scenarios identified in the previous step by randomly determining the execution time for each task from its respective normal distribution. The result is the time and corresponding probability for each of the simulated scenarios for each critical PERT path.

Step 9 - Analyze the results and make a decision. Based on the results of the previous step and for each of the allocation scenarios, the critical paths of each scenario are determined. Later, from these results, the critical path with the shortest time and the corresponding allocation scenario are determined. In this way, the optimal allocation of the agile teams is made taking into account their performance.

3. Illustrative Case Study

The application of the model to a hypothetical case study is presented below. This project consists of 10 tasks from A to I. The times estimates of each task and respective precedence are shown in Fig. 3 by a CPM diagram. In addition to this information, Fig. 3 also shows 5 possible critical paths, namely Path 1: ACFI, Path 2: ADHJ, Path 3: ADGJ, Path 4: AEHJ, and Path 5: BFI. The analysis of the precedence between tasks and tasks that may occur simultaneously revealed that three teams are required. Table 1 shows the schedules established for each of these teams. The breakdown of the three teams spans 22 weeks in three schedules, Schedule 1 (S1), Schedule 2 (S2), Schedule 3 (S3). Table 2 shows the parameters of the beta distribution for each project task, namely the most optimistic time (shortest time estimate), the most likely time, and the most pessimistic time (longest time estimate). These estimates are determined without considering the performance of the team performing the task.

Columns 7 through 9 then report the effective risk of each team for each task. These results were obtained using Eq. (1). In the following columns of Table 2, the corrected mean of the normal distribution and the corresponding variance are calculated using the effective risk of each team together with Eq.(s) 3 and 4. Finally, the corresponding time is calculated using the Monte Carlo method. This time is then used to determine the critical path and the total time of the project.

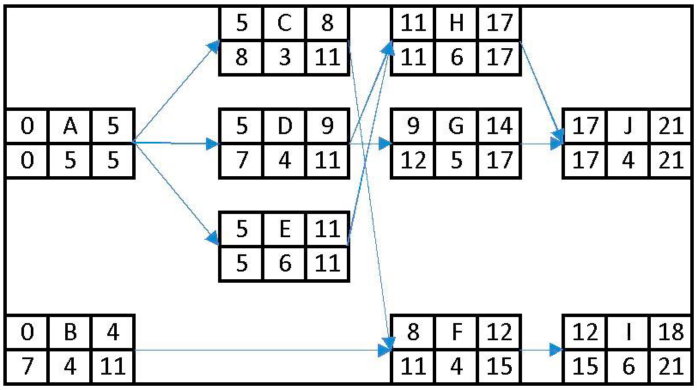

Fig. 3 – Illustrative case study CPM diagram.

Table 1. Project scheduling for teams' allocation.

		weeks																					
	Team	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Schedule 1	T_i	A	A	A	A	A	A	C	C	C		H	H	H	H	H	H	H	J	J	J	J	J
Schedule 2	T_j	B	B	B	B			E	E	E	F	F	F	F	I	I	I	I	I	I			
Schedule 3	T_k							D	D	D	D	G	G	G	G	G							

Table 2. Effective risk and normal distribution results for each team according to the project task and Monte Carlo iteration.

	Beta parameters (weeks)					Effective Risk			Team 1			Team 2			Team 3		
Activity	a - optimistic	m - M Likely	b - pessimist	Mean	Variance	Team 1	Team 2	Team 3	Mean corrected	new Variance	Time (weeks)	Mean corrected	new Variance	Time (weeks)	Mean corrected	new Variance	Time (weeks)
A	1	6	7	6	1.31	0.20	0.25	0.21	6.6	1.37	6.6	6.5	1.36	6.4	6.58	1.37	7.69
B	2	4	8	4	1.25	0.64	0.46	0.39	2.87	1.32	2	4.34	1.23	5.78	4.86	1.23	4.12
C	1	3	6	3	1.03	0.52	0.97	0.08	2.91	1.03	3.4	1.11	1.17	1	5.51	1.13	5.45
D	2	4	8	4	1.25	0.38	0.34	0.90	5	1.22	7.5	5.25	1.23	3.87	2.39	1.37	4.51
E	1	3	7	3	1.25	0.86	0.82	0.89	1.57	1.35	1.5	1.73	1.34	0.27	1.44	1.37	0.22
F	1	4	5	4	0.85	0.09	0.15	0.03	4.82	0.92	3.7	4.69	0.91	4.24	4.93	0.93	5.3
G	2	5	6	5	0.85	0.53	0.97	0.92	4.81	0.84	5.7	2.2	0.92	3.9	2.49	0.89	1.47
H	1	7	8	7	1.55	0.92	0.81	0.89	2.02	1.54	3.6	3.32	1.46	6.66	2.26	1.52	2.55
I	2	6	9	6	1.43	0.06	0.93	0.95	8.61	1.61	8.8	2.58	1.59	4.51	2.42	1.60	2.19
J	2	5	7	5	1.03	0.09	0.41	0.27	6.62	1.14	7.9	5.37	1.04	4.07	5.9	1.07	6.71

Table 3 shows the results obtained by the proposed model for six permutations without repeating the assignment of teams to the three schedules presented in Table 1. Based on these permutations, the respective assignments, and the parameters of the normal distributions presented in Table 2, six Monte Carlo simulations were performed, each with 10000 iterations.Using the traditional method CPM, without statistical variation of the execution time of the tasks as well as without considering the performance of the teams, we obtain a total project time of 19.67 weeks with a variation of 2 weeks, in this case the critical path AEHJ was determined. The analysis of the results in Table 3 shows that the critical path varies between path 1 and path 3 depending on the permutation, with critical path number 3 occurring in 4 permutations and critical path number 1 occurring in only 2 permutations. It should be noted that the critical path AEHJ (critical path number 4), which was determined by the traditional method CPM, is not one of the critical paths determined by the proposed method. From these results, it can be concluded that the stochastic variation of task execution times and the performance of the project team must be taken into account when determining the critical path. Permutation number 1, where team 1 was assigned to schedule 1, team 2 was assigned to schedule 2, and team 3 was assigned to schedule 3, is the assignment with the best results, i.e., with the least project duration. In this case, the estimated project duration was 19.05 weeks with a variance of 0.04 weeks. The probability that the project will be completed before 21 weeks is 84%. The worst result is for permutation number 5 with a duration of 25.57 weeks and a probability of 1.7% that the project will last less than 21 weeks. From these results, it can be concluded that the performance of the individual teams as well as their respective allocations have an influence on the project duration. When we compare this result with the result of the traditional CPM analysis, we find that the proposed method not only considers the performance of each team in determining the project duration, but also significantly reduces the uncertainty in estimating the project duration, i.e., in the traditional analysis, the estimated time variance for the project is 2 weeks, while in the proposed method it is 0.04 weeks. Moreover, it is still possible to determine the probability that the project will be completed at the end of a given period.

Table 3. Monte Carlo results for the six permutations.

					Permutation 1 : S1-T1; S2-T2; S3-T3			**Permutation 2 : S1-T1; S2-T3; S3-T2**		
Path	Tasks				Frequency	Rel. Freq.	Project time	Frequency	Rel. Freq.	Project time
1	A	C	F	I	2218	22.18%	19.05±0.04	463	4.63%	21.26±0.04
2	A	D	H	J	2013	20.13%		4362	43.62%	
3	A	D	G	J	4171	**41.71%**	P(T<21)	5134	**51.34%**	P(T<21)
4	A	E	H	J	1593	15.93%	0.841±0.007	41	0.41%	0.456±0.01
5	B	F	I		5	0.05%		0	0.00%	

					Permutation 3 : S1-T2; S2-T1; S3-T3			**Permutation 4 : S1-T2; S2-T3; S3-T1**		
Path	Tasks				Frequency	Rel. Freq.	Project time	Frequency	Rel. Freq.	Project time
1	A	C	F	I	8850	**88.50%**	21.33±0.04	43	0.43%	21.81±0.04
2	A	D	H	J	674	6.74%		1529	15.29%	
3	A	D	G	J	197	1.97%	P(T<21)	8392	**83.92%**	P(T<21)
4	A	E	H	J	243	2.43%	0.441±0.01	35	0.35%	0.335±0.01
5	B	F	I		36	0.36%		1	0.01%	

					Permutation 5 : S1-T3; S2-T1; S3-T2			**Permutation 6 : S1-T3; S2-T2; S3-T1**		
Path	Tasks				Frequency	Rel. Freq.	Project time	Frequency	Rel. Freq.	Project time
1	A	C	F	I	9697	**96.97%**	25.57±0.04	1306	13.06%	22.51±0.04
2	A	D	H	J	180	1.80%		409	4.09%	

3	A	D	G	J	123	1.23%	P(T<21)	8269	**82.69%**	P(T<21)
4	A	E	H	J	0	0.00%	0.017±0.003	16	0.16%	0.228±0.008
5	B	F	I		0	0.00%		0	0.00%	

4. Conclusion

This paper presents a new model for assigning agile teams to scalable agile projects. The basic goal was to incorporate team performance into the estimation of total project execution time and to develop a methodology for optimizing the assignment of teams to different project schedules. Three tools were used for this purpose, namely the PERT method, a well-known tool in project management; the effective risk model, a qualitative model for risk analysis and management; and the Monte Carlo method. By combining these three models, it was possible to create the allocation model proposed in this paper. In practice, this model supports decision making in the allocation of agile teams. The output of the model is the team allocation configuration that minimizes project execution time. In addition to this output, the model also provides the probability of the project execution time and the corresponding statistical confidence level. As a limitation, we can consider the fact that the agile team performance evaluation is performed before the work starts, which is a static evaluation. For future work, it is foreseen to further develop the proposed model to include a dynamic assessment of the agile team's performance.

Acknowledgements

This work was supported by the Polytechnic Institute of Lisbon through the Projects for Research, Development, Innovation and Artistic Creation (IDI&CA), within the framework of the project ReEdIA — Risk Assessment and Management in Open Innovation, IPL/2021/ ReEdIA/ISEL.

References

[1] Carl Marnewick and Annlizé L. Marnewick. 2022. Benefits realisation in an agile environment. *International Journal of Project Management* (2022).
[2] Abheeshta Putta, Maria Paasivaara, and Casper Lassenius. 2018. Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In *International Conference on Product-Focused Software Process Improvement*, Springer, 334–351.
[3] Abheeshta Putta, Ömer Uludağ, Shun-Long Hong, Maria Paasivaara, and Casper Lassenius. 2021. Why Do Organizations Adopt Agile Scaling Frameworks? A Survey of Practitioners. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 1–12.
[4] Martin Kalenda, Petr Hyna, and Bruno Rossi. 2018. Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process* 30, 10 (2018), e1954.
[5] Mike Cohn. 2005. *Agile estimating and planning*. Pearson Education.
[6] Kim Dikert, Maria Paasivaara, and Casper Lassenius. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software* 119, (2016), 87–108.
[7] Nick T. Thomopoulos. 2014. *Essentials of Monte Carlo simulation: Statistical methods for building simulation models*. Springer.
[8] S. C. Sharma. 2006. *Operation research: Pert, Cpm & cost analysis*. Discovery Publishing House.
[9] Vitor Anes, Elsa Henriques, M. Freitas, and Luís Reis. 2018. A new risk prioritization model for failure mode and effects analysis. *Quality and Reliability Engineering International* 34, 4 (2018), 516–528.
[10] Hu-Chen Liu, Long Liu, and Nan Liu. 2013. Risk evaluation approaches in failure mode and effects analysis: A literature review. *Expert systems with applications* 40, 2 (2013), 828–838.