A Risk Management Tool for Agile Software Development

Breno Gontijo Tavares , Mark Keil , Carlos Eduardo Sanches da Silva & Adler Diniz de Souza

JOURNAL OF COMPUTER INFORMATION SYSTEMS
https://doi.org/10.1080/08874417.2020.1839813

A Risk Management Tool for Agile Software Development

Breno Gontijo Tavares [a], Mark Keil[b], Carlos Eduardo Sanches da Silva[c], and Adler Diniz de Souza[c]

[a]INATEL, Nacional Institute of Telecommunications, Santa Rita Do Sapucaí, Brazil; [b]Georgia State University, Atlanta, Georgia, USA; [c]UNIFEI, Federal University of Itajuba, Itajuba, Brazil

ABSTRACT
Lack of risk management or its inadequate application is one reason that software development projects fail. Agile methods, which have become increasingly popular, do not offer specific activities to manage risks. We develop and evaluate a tool for managing risks in software development projects that use agile methods. The proposed tool provides a collection of risk management practices and an iterative lifecycle. To analyze the effectiveness of the proposed tool we conducted an experiment with the participation of experts in agile methods. Our tool increased the effectiveness of risk response planning without increasing the time invested in this process.

KEYWORDS
Software development; risk management; agile methods; success of agile projects

Introduction

More than 11% of software projects are canceled before even delivering the first expected result to the customer.[1] One reason for this is lack of effective risk management.[2] This is not surprising as risk management is the least practiced knowledge area of project management in software projects.[3]

In recent years, agile methods have gained increased attention[4,5] as they may improve the chances of success on software development projects.[6] Agile methods are believed to have certain advantages over traditional software development methods such as accelerating the time to market, increasing quality and productivity, and improving alignment of information technology with the business.[7–10]

However, agile methods do not prescribe specific activities to manage risks on agile projects.[6,11–16] Therefore, risk management in agile methods tends to be conducted in an ad hoc manner.[17–20]

This is unfortunate because a systematic approach to risk management can potentially reduce uncertainty and increase the chances of success on software projects.[21–26]

Without a systematic approach to risk management, important risks are likely to be ignored.[27] Ineffective risk management has been associated with agile project failures.[28]

Further, several scholars have suggested the need for risk management practices to improve the chances of success of agile projects[14,20,25,27,29,30] and some have noted that it may be desirable to integrate such practices from traditional projects to ensure effective risk management in agile methods.[18,20] However, risk management processes need to stay true to the spirit of agility and the use of traditional practices may overload them.[20] Thus, our research aim was to develop a risk management tool that would be uniquely suited for agile projects.

In the remainder of the paper, we describe the risk management tool that was developed and grounded in risk management practices suggested in the agile methods literature. We evaluated the risk management tool to determine its effectiveness.

Background

Agile methods are a set of processes applied to software development that are iterative, incremental, self-organized, and emergent.[31] Among the more popular agile methods are Scrum,[32] Kanban,[33] Extreme Programming (XP)[29], Dynamic Software Development Method (DSDM),[34] Test Drive Development (TDD),[35] and Disciplined Agile Delivery (DAD).[36] None of the popular agile methods, however, incorporate specific activities to manage risks. As a group, agile methods do not tend to use any kind of intentional risk management approach.[11,19] However, agile methods are said to reduce risks through multiple iterations or sprints.[18] The feedback mechanisms of agile methods, such as the opinion of the customer about the increment delivered at the end of each sprint, help the project team to respond to constant changes in requirements, needs, new opportunities, and risks.[37] However, there are few guidelines on how these mechanisms can be most effectively implemented.[38] Also, there are project-level threats that may not be identified by iteration,[18] creating a void in the capacity of these approaches to tackle risks.[13] Hence, there is a need to incorporate risk management concepts into agile methods in order to address this issue.[20]

Siddique and Hussein[18] suggest that practitioners of agile methods should deal with risk in the same manner as would be done in traditional projects. Miller and Grski[39] agree, arguing that we should integrate traditional risk management processes into agile methods as risk management in agile methods does not differ significantly from traditional projects.

Nevertheless, agile approaches and the conventional risk management approach appear to be somewhat at odds with each other. Risk management typically implements a high overhead approach, while agile methods support a lightweight one. According to Tavares *et al.*[20], the application of risk management practices in agile projects should be in line with the Agile Manifesto principles.[40]

CONTACT Breno Gontijo Tavares breno.tavares@inatel.br INATEL, Nacional Institute of Telecommunications, Santa Rita Do Sapucaí, Brazil

Some published papers suggest practices to manage risks when using agile methods. Shrivastava and Rathod[41] propose a list of 28 risk management practices specific for distributed agile development. Moran[38] proposes a standardized risk management process in agile processes that integrates aspects of traditional risk management. Khatri, Bahri, and Johri[42] performed a study to propose a selection of risk management practices that could apply to Scrum. Each of the three aforementioned papers proposes approaches to risk management that are concentrated on agile methods or projects with unique characteristics. One of the studies proposes the use of traditional risk management practices, which may be too "heavy" to work well in agile project environments.

Thus, further research is needed in order to develop a systematic and comprehensive approach for risk management that is not tied to a specific agile method and which is lightweight enough to not violate the principle of agility. In the following section, we describe how we identified and ranked the risk management practices that could form the basis for a risk management tool that would be appropriate for agile software development projects and how we evaluated the tool that was developed.

Methodology

Building upon the work of Tavares *et al.*,[43] we created a risk management tool for agile methods. Specifically, we adapted the list of practices proposed by Tavares *et al.*,[43] which contains a ranked list of 127 unique risk management practices for agile methods that were classified into 5 components and 48 subcomponents. Artifacts, Features, Events, Roles and Techniques and methods were the components used as categories. In prior work by Tavares *et al.*[43] these components and subcomponents were ranked in order of importance by a panel of ten experts in agile methods using the Analytic Hierarchy Process (AHP) method (More detail on the AHP process and how it was used to rank components and subcomponents can be found in Tavares *et al.*[43]).

In this research study, we took this as a starting point to develop what we call the Rm4Am tool which stands for Risk Management for Agile Methods. In order to do this we invited the same ten experts in risk management and agile projects to assess the ranked list of risk management practices developed by Tavares *et al.*[43] The experts were asked to classify the 48 subcomponents according to the following risk management processes: (P) Risk planning; (I) Risk identification; (A) Risk analysis; (PR) Risk response planning; (MC) Monitoring and control of risks. We performed the classification in two phases: (1) the experts classified each subcomponent according to the risk management processes; (2) we conducted a meeting with the experts to reach consensus on any subcomponents where there was disagreement regarding its classification. Classifying the ranked list of risk management practices was important as it provides practitioners with some guidance for selecting the most appropriate risk management practices for each project.

In order to evaluate the risk management tool that we developed, we conducted an experiment. According to Falessi *et al.*[44] experimental research is helpful in validating new proposals for the area of software engineering. This method allows the researcher to make strong statements of causality. No other scientific research method provides the same certainty when determining that a variable causes a specific effect.[45]

Specifically, in order to evaluate the Rm4Am tool, we designed and executed an experiment involving 18 professionals who work with agile methods. We used a within subject pretest posttest design in which the 18 professionals were first asked to perform risk response planning on two projects without the Rm4Am tool and then fifteen days later they were asked to perform risk response planning on the same two projects with the Rm4Am tool.

We performed the Rm4Am evaluation in relation to risk response planning, because this process has the following characteristics: (1) it is important for risk management success, since it allows for prevention and control of risks by implementing appropriate strategies to deal with them,[46] (2) it has critical impacts on project success,[47] (3) it is a neglected process in projects,[48] and (4) there are few tools in the literature that assist with effective risk response planning.[49]

Participants

Since experience with agile methods and risk management, as well as individual risk tolerance, could be important factors in this context, we designed the experiment with this in mind. We defined these thresholds to ensure that subjects had at least some experience with agile methods and risk management. Furthermore, we purposely wanted to include people with different risk tolerance in order to cover a range of different profiles that could affect risk management behavior.

Table 1 presents the number of people by experience and risk tolerance profile who were recruited as participants.

We developed the online experiment based upon two actual agile software development projects that had been recently completed. Both projects involved more than 7,000 hours of effort, lasted 6–12 months, and involved 9–12 people. The two Scrum masters for projects, who were well acquainted with the risks associated with his/her project, served as judges in our experiment.

We defined the following criteria to select project risks for the experiment (based on what actually occurred in the two projects): (1) the risk occurred, but the response plan was effective in mitigating its impact; (2) the risk did not occur because the response plan was effective in mitigating its probability; (3) the risk did not occur because the response plan was effective in its elimination. These criteria allow identifying

Table 1. Experience and risk tolerance profile of participants.

Participants	Experience [a]	Risk tolerance profile[b]
2	Little	Averse
2	Little	Neutral
2	Little	Seeker
2	Average	Averse
2	Average	Neutral
2	Average	Seeker
2	Much	Averse
2	Much	Neutral
2	Much	Seeker

[a]Experience: Little (Less than 1 year), Average (From 1 to 3 years), Much (Beyond 3 years)
[b]Risk tolerance profile: Averse (Want to avoid or reduce risk), Neutral (Between averse and seeker), Seeker (Comfortable to take risks)

the successful risk responses. This information is important to judge responses provided by participants of the experiment.

Participants received information on eight risks, three from project A and five from project B. With this information, participants were asked to develop the best response plans for each risk. The Scrum master of each project evaluated the risk response plans for his/her project. The evaluation process compared the responses of the participants with the actual risk response plans that were used in these two projects and deemed to be effective. However, Scrum masters also used their judgment in the evaluation because some response plans developed by the participants could be even more effective than those used in the actual projects. We used a 5-point scale to score the participants risk response plans (with 0 points awarded for plans that were judged to be poor and 4 points awarded for plans that were judged to be excellent).

Pilot test

Initially, we pilot tested the materials to ensure that the research instrument was well structured and to identify any difficulties participants had in interpreting the instructions or the questions. Three professionals with at least 6 months of experience in agile methods and risk management participated in the pilot test. We did not invite participants of the pilot test to participate in the actual experiment. Participants of the pilot test recommended changing the description of two risks, as well as improving information in the video that introduced them to Rm4Am. After implementing these recommendations, we conducted the actual experiment.

Results and discussion

Table 2 presents the ranking of the 48 subcomponents to manage risks in agile projects and how these were mapped to the various risk management processes (Risk planning (P); Risk identification (I); Risk analysis (A); Risk response planning (PR); Monitoring and control of risks (MC)). Appendix A contains the full list of ranked risk management practices.

Based on the ranking of agile subcomponents and the classification of risk management practices, we developed

Table 2. Subcomponent ranking and level of recommendation (from Tavares *et al.*[43]) with mapping to risk management process.

Rank	Subcomponent (Component code*)	Number of Practices	Recommendation	Risk management processes
1	Daily meeting (E)	6	Fully recommended	I; MC
2	Increment (A)	6	Fully recommended	I; A; PR; MC
3	Prototype (A)	2	Fully recommended	I; A; PR; MC
4	Product backlog (A)	7	Fully recommended	P; I; A; PR; MC
5	Sprint planning (E)	2	Fully recommended	I; A; PR; MC
6	Sprint backlog (A)	4	Fully recommended	I; A; PR; MC
7	Product backlog refinement meeting (E)	2	Fully recommended	I; A; PR
8	Weekly risk meeting (E)	1	Fully recommended	P; I; A; PR; MC
9	Technical specification (A)	4	Fully recommended	I; A; PR; MC
10	Sprint (E)	11	Strongly recommended	P; I; A; PR; MC
11	Risk repository (A)	3	Strongly recommended	I; A; PR; MC
12	Continuous communication and collaboration (F)	6	Strongly recommended	I; A; PR; MC
13	Sprint review (E)	2	Strongly recommended	I; A; PR; MC
14	Business alignment (F)	1	Strongly recommended	P; I; A; PR; MC
15	Contingency plan (A)	1	Strongly recommended	P; I; A; PR; MC
16	Product owner (R)	12	Strongly recommended	I; A; PR; MC
17	Sprint retrospective (E)	1	Strongly recommended	I; A; PR; MC
18	Project update (F)	2	Strongly recommended	I; C
19	Development team (R)	25	Strongly recommended	P; I; A; PR; MC
20	Software code library (A)	2	Moderately recommended	I
21	Wiki (A)	1	Moderately recommended	I
22	Cost feasibility analysis (TM)	1	Moderately recommended	I; A
23	Qualitative and quantitative analysis (TM)	1	Moderately recommended	A
24	Multifunctional team (F)	2	Moderately recommended	P; I; A; PR; MC
25	Self-organizing team (F)	1	Moderately recommended	P; I; A; PR; MC
26	Continuous integration (TM)	1	Moderately recommended	I; C
27	Test driven development TDD (TM)	1	Moderately recommended	I; A; PR; MC
28	Stakeholders	6	Moderately recommended	P; I; A; PR; MC
29	Resource allocation (F)	1	Moderately recommended	PR
30	Strengths, weaknesses, opportunities, threats SWOT (TM)	1	Moderately recommended	I; A; PR; MC
31	Risk-driven approach (TM)	1	Moderately recommended	I; A; PR; MC
32	Artifact transparency (F)	4	Recommended	I; A; PR; MC
33	Architecture owner (R)	4	Recommended	I; A; PR; MC
34	Kanban (TM)	1	Recommended	I; A; PR; MC
35	Adaptation (F)	1	Recommended	I; A; PR; MC
36	Business analyst (R)	2	Recommended	I; A; PR; MC
37	Project manager (R)	2	Recommended	P; I; A; PR; MC
38	Scrum master (R)	15	Recommended	P; I; A; PR; MC
39	Technical coordinator (R)	2	Recommended	P; I; A; PR; MC
40	Team leader (R)	1	Recommended	P; I; A; PR; MC
41	Value-driven approach (TM)	1	Recommended	I; A; PR; MC
42	Pair programming (TM)	1	Recommended	I; PR; MC
43	Business visionary (R)	4	Recommended	P; I; A
44	Project milestone (TM)	3	Recommended	I; MC
45	Time scale (TM)	1	Recommended	I; A; PR; MC
46	Dynamic systems development method DSDM (TM)	1	Recommended	P; I; A; PR; MC
47	Must have, should have, could have, and won't have (MoSCow) prioritization (TM)	1	Recommended	A
48	Tests (TM)	3	Recommended	I; A; PR; MC

*Risk management components: Artifacts (A); Events (E); Features (F); Roles (R); Techniques and Methods (TM) a flexible risk management tool for agile projects called Rm4Am (Risk management for Agile methods). The Rm4Am tool contains both a lifecycle figure (shown in Figure 1) and a ranked list of subcomponents and associated risk management practices for agile software development projects (see Appendix A). An important element of the tool is the degree to which each subcomponent is recommended for risk management (i.e., fully recommended, strongly recommended, etc).

As shown in Figure 1, the Rm4Am lifecycle is an iterative process that begins with a business vision of the client and ends with delivery of specific functionality that provides value to the business (i.e., an increment) or allows the team to further refine requirements (i.e., a prototype). The process typically involves an open number of sprints or iterations of a fixed duration (typically ranging from two to four weeks and held consistent throughout the project). The overall process involves five components: Events, Artifacts, Features, Roles, Techniques and methods as described earlier. Events and Artifacts encompass subcomponents that are associated with specific points in the Rm4Am lifecycle, while Features, Roles, Techniques and methods can be chosen throughout the lifecycle based upon the specific needs of each project or sprint. In the remainder of this section, we discuss those aspects of the Rm4Am lifecycle that involve the nine subcomponents that are fully recommended for risk management on agile projects.

Table 3 presents the nine most important subcomponents that were previously identified by Tavares *et al.*[43] and how they map to their associated risk management practices. We focus on these nine subcomponents because they are the ones that are fully recommended under the Rm4Am tool. The full ranked list of 127 risk management practices for agile methods can be found in Appendix A.

The literature used to identify risk management practices revealed that certain practices could be related to more than one subcomponent. One example presented in Table 3 is the practice "Identify, analyze and respond to risks", which is related to the daily meeting, sprint planning, product backlog refinement meeting, and risk weekly meeting.

According to Rm4Am, the daily meeting is the most important subcomponent to manage risks in agile projects. Daily meetings occur during the sprint to monitor the progress of development. The team identifies and monitors risks during these daily meetings. The second and third most important subcomponents to manage risks in agile projects are the increment and the prototype. An increment is a piece of software that is put into production to generate value for the business, while a prototype is designed to demonstrate some functionality to the customer for purpose of validating or refining product requirements. The fourth most important subcomponent to manage risks in agile projects is the product backlog. The project team analyzes the product backlog to clarify the product requirements needed to meet the customer's vision.

The fifth and sixth most important subcomponents to manage risks in agile projects are sprint planning and sprint backlog. Sprint planning involves a meeting in which the team creates the sprint backlog from the product backlog. During this process, the team identifies risks that may affect the sprint or the entire project. The seventh most important subcomponent to manage risks in agile projects is product backlog refinement meeting in which the team clarifies the product requirements needed to meet the customer's vision. During this meeting, the team also engages in risk identification and risk planning. The eighth most important subcomponent to manage risks in agile projects is a risk weekly meeting to manage project risks. This meeting is an important part of Rm4Am and is necessary because daily meetings are short (typically no more than 15 minutes) and do not provide adequate time to deal with risks properly. This subcomponent represents a new addition to agile practices that is not

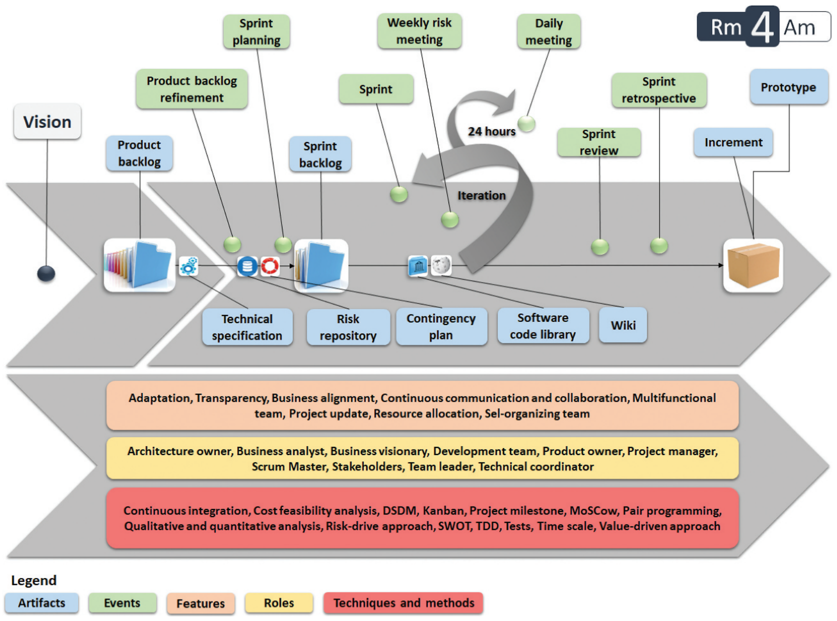
Rm 4 Am
Sprint planning
Weekly risk meeting
Daily meeting
Product backlog refinement
Sprint
Sprint retrospective
Prototype
Vision
24 hours
Product backlog
Sprint backlog
Sprint review
Increment
Iteration
Technical specification
Risk repository
Contingency plan
Software code library
Wiki
Adaptation, Transparency, Business alignment, Continuous communication and collaboration, Multifunctional team, Project update, Resource allocation, Sel-organizing team
Architecture owner, Business analyst, Business visionary, Development team, Product owner, Project manager, Scrum Master, Stakeholders, Team leader, Technical coordinator
Continuous integration, Cost feasibility analysis, DSDM, Kanban, Project milestone, MoSCow, Pair programming, Qualitative and quantitative analysis, Risk-drive approach, SWOT, TDD, Tests, Time scale, Value-driven approach
Legend
Artifacts
Events
Features
Roles
Techniques and methods

Figure 1. Rm4Am lifecycle.

Table 3. The nine most important agile methods subcomponents and associated risk management practices.

Rank	Subcomponent	Risk management processes	Practices to Manage Risk
1	Daily meeting	I; MC	Ask and answer the following questions: 1) What did I do yesterday? 2) What am I planning to do today? 3) Do I expect any impediment? Create a backlog of impediments Attendees must stand-up in the meetings Identify, analyze and respond to risks Make meetings brief and successful Manage a risk repository and share it with the team
2	Increment	I; A; PR; MC	Identify, analyze and treat the risks to project deliveries as soon as possible Concentrate on clearly defined deliverables Identify and disregard non-relevant risk information for the increment Implement a countermeasure in the increment when needed Make frequent deliveries to obtain feedback Deliver a piece of the solution at the end of all iterations to generate value for the business frequently
3	Prototype	I; A; PR; MC	Evaluate the prototype to obtain feedback Validate original estimates to improve planning and estimation processes
4	Product backlog	P; I; A; PR; MC	Address only the highest priority tasks Encourage team members to work together on only a few backlog items to completion before starting new work Identify the dependency between the requirements Implement the riskiest stories early in the project Investigate, refine and consolidate the requirements/user stories Optimize scheduling speed and risk Prioritize requirements, establish the scenario, govern risk management activities, and ensure that the highest value resources for the customer take precedence
5	Sprint planning	I; A; PR; MC	Carry out the initial risk assessment Identify, analyze and respond to risks
6	Sprint backlog	I; A; PR; MC	Analyze technical risks Encourage team members to work together on only a few backlog items to completion before starting new work Identify and respond to the risks Identify the dependency between the requirements
7	Product backlog refinement meeting	I; A; PR	Identify, analyze and respond to risks Investigate, refine and consolidate the requirements/user stories
8	Risk weekly meeting	P; I; A; PR; MC	Identify, analyze and respond to risks
9	Technical specification	I; A; PR; MC	Document important technical specifications Document key technical decisions Evaluate product architecture Refine the specification of the project continuously during development

Risk management processes: Artifacts (A); Events (E); Features (F); Roles (R); Techniques and Methods (TM)

currently incorporated in any of the mainstream agile methods. The ninth most important subcomponent to manage risks in agile projects is the technical specification which complements the product backlog with more detailed technical information that can be used by the project team to identify risks. This subcomponent also represents a new addition to agile practices that is not currently incorporated in any of the mainstream agile methods.

Evaluation of Rm4Am

The experiment we conducted to evaluate Rm4Am consisted of 3 phases as presented in the Figure 2.

Phase 1 involved identifying participants with a range of risk tolerance and experience in agile methods and risk management. In Phase 2, participants were asked to evaluate two projects and formulate risk response plans without using Rm4Am. Fifteen days later, in Phase 3, the same participants were asked to evaluate the same two projects and formulate risk response plans using the Rm4Am tool. For each subject, we computed a summative pretest score across the two projects and a summative posttest score across the two projects. Comparing these summative pretest and posttest scores allowed us to analyze the effectiveness of the Rm4Am in the context of risk response planning. Using IBM SPSS Statistics 25® software we first generated descriptive statistics to examine the mean risk response score with and without Rm4Am across the two projects. According to Table 4, the use of Rm4Am increased the mean risk response score by 49%. This result suggests that the Rm4Am tool helped participants formulate more effective risk response plans.

Table 5 presents the descriptive statistics of Rm4Am for each project. The use of Rm4Am increased the mean risk response score by 63% for Project A and 40% for Project B.

Using IBM SPSS we performed paired-samples t-tests across the two projects and for each project individually to determine if the mean difference between paired observations on a particular outcome was significantly different from zero. Table 6 presents the results of the paired-samples t-tests, which revealed significant differences.

Individuals' scores across the two projects (A and B), were 4.78 points lower without Rm4Am than they were with Rm4Am. For Project A alone, scores were 2.39 points lower without Rm4Am than they were with Rm4Am. For Project B alone, scores were 4.78 points lower without Rm4Am than they were with Rm4Am. Due to the relatively small number of participants, we also ran robust paired-samples t-tests using the WRS2 package for R which is a collection of robust statistical methods. Specifically, we used the yuend function which performs

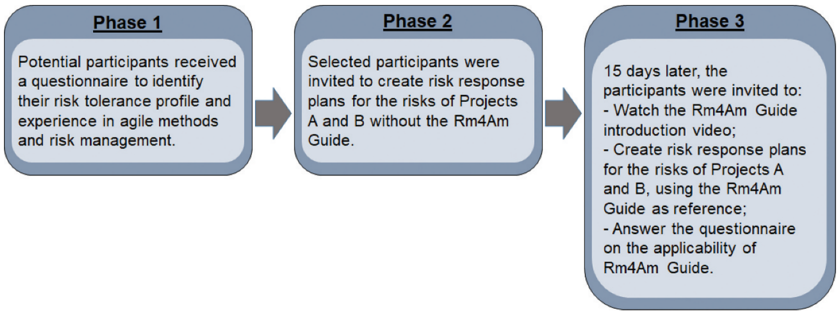
Phase 1
Potential participants received a questionnaire to identify their risk tolerance profile and experience in agile methods and risk management.
Phase 2
Selected participants were invited to create risk response plans for the risks of Projects A and B without the Rm4Am Guide.
Phase 3
15 days later, the participants were invited to:
- Watch the Rm4Am Guide introduction video;
- Create risk response plans for the risks of Projects A and B, using the Rm4Am Guide as reference;
- Answer the questionnaire on the applicability of Rm4Am Guide.

Figure 2. Phases of the experiment.

Table 4. Descriptive statistics of Rm4Am factor.

	Mean	Minimum	Maximum
Without Rm4Am	9.72	2.00	17.00
With Rm4Am	14.50	5.00	26.00

Table 5. Descriptive statistics of Rm4Am factor in each project.

	Project A			Project B		
	Mean	Minimum	Maximum	Mean	Minimum	Maximum
Without Rm4Am	3.78	2.78	4.78	5.94	4.72	7.06
With Rm4Am	6.17	4.89	7.39	8.33	7.00	9.67

Yuen's test on trimmed means for dependent samples. This approach uses bootstrapping and is robust to deviations from normality that can result in small samples. The results of this analysis were consistent with the findings obtained through IBM SPSS, thus giving us further confidence in the findings.

We recorded the time invested by the participants to plan risk responses, to determine if the Rm4Am tool added significant overhead to this process, since one of the prime motivations of agile methods is to prioritize agility and reduce activities that leave them heavy. Our results indicated that the use of the Rm4Am in the third phase did not significantly influence the time invested in risk response planning, compared to the time invested in the second phase without the tool. This result indicates that Rm4Am adheres to the principle of agility.

Limitations and conclusions

In our evaluation of Rm4Am, we conducted an experiment in which the same participants were asked to develop risk response plans for two agile projects, first without the Rm4Am tool and then with the Rm4Am tool. While this approach had certain strengths such as controlling for differences across participants and ensuring that the projects evaluated with the tool involved the same degree of complexity in risk response planning, it also presents a weakness. Specifically, with an experimental design such as ours, we cannot rule out the possibility of learning or carryover effects which could pose a threat to validity. To minimize this threat, we separated the two data collection points by fifteen days. In other words, after participants developed risk response plans without the Rm4Am tool, we waited for more than two weeks before asking them to develop risk response plans with the Rm4Am tool. This was done in order to minimize the risk of carryover effects. We have reason to believe that this measure was effective because when we interviewed participants at the end of the study, they indicated that they did not remember their risk response plans created without the Rm4Am tool when they were working on their risk response plans with the Rm4Am tool. Still, to rule out the possibility of carryover effects, we suggest that further testing of the Rm4Am tool be conducted in a way that controls for possible order effects. In spite of the limitations, we believe that our findings hold important implications for both research and practice.

In summary, the experiment showed that Rm4Am increased the effectiveness of risk responses. Specifically, the use of Rm4Am increased the mean risk response score by 49%, implying that Rm4Am increased the effectiveness of risk response planning. At the same time, the use of Rm4Am did not significantly influence the time required to do risk response planning, thus indicating that Rm4Am adheres to the principle of agility. These results provide support for incorporating risk management tools into agile method with aim of improving the chances of project success.

In terms of directions for future research, we suggest that it may be useful to create different models for each of the four Rm4Am recommendation levels (recommended, moderately recommended, strongly recommended, fully recommended) and perform experiments to evaluate their effectiveness in risk management. As a result, we would identify improvements in the recommendation levels of the subcomponents and, consequently, make Rm4Am more effective.

Table 6. Paired-samples t-tests.

					95% Confidence interval of the difference		
		Mean	Std. deviation	Std. error mean	Lower	Upper	t
Pair 1	Without_Rm4Am – With_Rm4Am	−4.78	1.63	.384	−5.59	−3.97	−12.44
Pair 2	ProjectA_Without_Rm4Am – ProjectA_With_Rm4Am	−2.39	1.82	.429	−3.29	−1.48	−5.57
Pair 3	ProjectB_Without_Rm4Am – ProjectB_With_Rm4Am	−2.39	1.85	.436	−3.31	−1.47	−5.47

ORCID

Breno Gontijo Tavares http://orcid.org/0000-0002-4158-7173

References

1. Jørgensen M. Failure factors of small software projects at a global outsourcing marketplace. J Syst Softw. 2014;92(1):157–69. doi:10.1016/j.jss.2014.01.034.
2. De WB, Visser JK. An evaluation of software project risk management in South Africa. South African J Ind Eng. 2013;24(1):14–28. doi:10.7166/24-1-497.
3. Niazi M, Mahmood S, Alshayeb M, Qureshi AM, Faisal K, Cerpa N. Toward successful project management in global software development. Int J Proj Manag. 2016;34(8):1553–67. doi:10.1016/j.ijproman.2016.08.008.
4. Stettina CJ, Hörz J. Agile portfolio management: an empirical perspective on the practice in use. Int J Proj Manag. 2015;33 (1):140–52. doi:10.1016/j.ijproman.2014.03.008.
5. López-Nores M, Pazos-Arias JJ, García-Duque J, Blanco-Fernández Y, Díaz-Redondo RP, Fernández-Vilas A, Gil-Solla A, Ramos-Cabrer M. Bringing the agile philosophy to formal specification settings. Int J Softw Eng Knowl Eng. 2006;16(6):951–86. doi:10.1142/s0218194006003075.
6. Serrador P, Pinto JK. Does Agile work? - A quantitative analysis of agile project success. Int J Proj Manag. 2015;33(5):1040–51. doi:10.1016/j.ijproman.2015.01.006.
7. Cockburn A, Highsmith J. Agile software development, the people factor. Computer (Long Beach Calif). 2001;34:131–33.
8. Qumer A, Henderson-Sellers B. A framework to support the evaluation, adoption and improvement of agile methods in practice. J Syst Softw. 2008;81(11):1899–919. doi:10.1016/j.jss.2007.12.806.
9. Jyothi VE, Rao KN. Effective implementation of agile practices ingenious and organized theoretical framework. Int J Adv Comput Sci Appl. 2011;2(3):41–48. doi:10.14569/ijacsa.2011.020308.
10. Nishijima RT, Dos Santos PDJG. The challenge of implementing scrum agile methodology in a traditional development environment. Int J Comput Technol. 2013;5(2):98–108. doi:10.24297/ijct.v5i2.3529.
11. Talal Alharbi E, Jameel Qureshi MR. Implementation of risk management with SCRUM to achieve CMMI requirements. Int J Comput Netw Inf Secur. 2014;6(11):20–25. doi:10.5815/ijcnis.2014.11.03.
12. Tomanek M, Juricek J. Project risk management model based on PRINCE2 and scrum frameworks. Int J Softw Eng Appl. 2015;6 (1):81–88. doi:10.5121/ijsea.2015.6107.
13. Bumbary KM Using velocity, acceleration, and jerk to manage agile schedule risk. Proc - 2016 Int Conf Inf Syst Eng ICISE 2016; 2016:73–80, Los Angeles, CA. Published online. doi:10.1109/ICISE.2016.21
14. Hammad M, Inayat I Integrating risk management in scrum framework. Proc - 2018 Int Conf Front Inf Technol FIT 2018; 2019:158–63, Islamabad, Pakistan. Published online. doi:10.1109/FIT.2018.00035
15. Mousaei M, Gandomani TJ. A new project risk management model based on Scrum framework and Prince2 methodology. Int J Adv Comput Sci Appl. 2018;9(4):442–49. doi:10.14569/IJACSA.2018.090461.
16. Buganová K, Šimíčková J. Risk management in traditional and agile project management. Transp Res Procedia. 2019;40 (1):986–93. doi:10.1016/j.trpro.2019.07.138.
17. Walczak W, Kuchta D. Risks characteristic to Agile project management methodologies and responses to them. Oper Res Decis. 2013;4(1):75–95. doi:10.5277/ord130406.
18. Siddique L, Hussein BA Practical insight about risk management process in agile software projects in Norway. In: 2014 {IEEE} International Technology Management Conference. Chicago, IL: IEEE; 2014. doi:10.1109/itmc.2014.6918616
19. Albadarneh A, Albadarneh I, Qusef A Risk management in Agile software development: A comparative study. 2015 IEEE Jordan Conf Appl Electr Eng Comput Technol AEECT 2015; Amman, Jordan, 2015. Published online. doi:10.1109/AEECT.2015.7360573
20. Tavares BG, da Silva CES, de Souza AD. Risk management analysis in Scrum software projects. Int Trans Oper Res. 2019;26 (5):1884–905. doi:10.1111/itor.12401.
21. Lobato LL, Bittar TJ, Neto PA, Machado IC, Almeida ES, Meira SR. Risk management in software product line engineering: a mapping study. Int J Softw Eng Knowl Eng. 2013;23(4):523–58. doi:10.1142/S0218194013500150.
22. Elzamly A, Hussin BA. Comparison of fuzzy and stepwise multiple regression analysis techniques for managing software project risks: implementation phase. J Comput Sci. 2014;10(10):1725–42. doi:10.3844/jcssp.2014.1725.1742.
23. Taherdoost H, Keshavarzsaleh A. How to lead to sustainable and successful IT project management? Propose 5Ps guideline. Int J Adv Comput Sci Inf Technol. 2015;4(1):14–37. doi:10.2139/ssrn.3224225.
24. Samantra C, Datta S, Mahapatra SS, Debata BR. Interpretive structural modelling of critical risk factors in software engineering project. Benchmarking An Int J. 2016;23(1):2–24. doi:10.1108/BIJ-07-2013-0071.
25. Rafeek MA, Arbain AF, Sudarmila E. Risk mitigation techniques in agile development processes. Int J Supply Chain Manag. 2019;8:1123–29.
26. Vujovic V, Denic N, Stevanovic V, Stevanovic M, Stojanovic J, Cao Y, Alhammadi Y, Jermsittiparsert K, Le HV, Wakil K, Radojkovic I. Project planning and risk management as a success factor for IT projects in agricultural schools in Serbia. Technol Soc. 2020;63(1):1–5. doi:10.1016/j.techsoc.2020.101371.
27. Teller J, Kock A, Gemünden HG. Risk management in project portfolios is more than managing project risks: A contingency perspective on risk management. Proj Manag J. 2014;45(4):67–80. doi:10.1002/pmj.21431.
28. Rai AK, Agrawal S, Khaliq M. Agile software quality of adaptability risk measurement using fuzzy inference system. Int J Comput Sci Eng. 2018;6(9):755–59. doi:10.26438/ijcse/v6i9.755759.
29. Beck K, Andres C. Extreme programming explained, second edition. Boston, MA: Addison-Wesley; 2004.
30. Gold B, Vassell C. Using risk management to balance agile methods: A study of the Scrum process. Int J Mechatronics, Electr Comput Technol. 2016;6(21):2943–50. doi:10.1109/KBEI.2015.7436020.
31. Lindvall M, Basili V, Boehm B, Costa P, Dangle K, Shull F, Tesoriero R, Williams L, Zelkowitz M. Empirical findings in agile methods. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2002;2418:197–207. doi:10.1007/3-540-45672-4_19.
32. Schwaber K, Beedle M. Agile software development with scrum. Vol. 18. Pearson Education, Upper Saddle River, NJ: Prentice Hall; 2001. http://www.amazon.co.uk/Agile-Software-Development-SCRUM-Schwaber/dp/0130676349
33. Kniberg H, Skarin M Kanban and scrum - making the most of both. *Washingt C4media*. Published online 2009.
34. Consortium D, Stapleton J DSDM: business focused development. *Second Ed Pearson Educ*. Published online 2003.
35. Beck K. Test-driven development: by example. Boston, MA: Addison-Wesley Professional; 2003.
36. Ambler SW, Lines M. Disciplined agile delivery. Upper Saddle River, NJ: IBM Press; 2012. Published online.
37. Conforto EC, Salum F, Amaral DC, Silva SL, Almeida LF. Can agile project management be adopted by industries other than software development? Proj Manag J. 2014;45(3):21–34. doi:10.1002/pmj.21410.
38. Moran A. Agile risk management. SpringerBriefs Comput Sci. 2014;9783319050072:33–60. doi:10.1007/978-3-319-05008-9_3.
39. Miler J, Górski J. Risk identification patterns for software projects. Found Comput Decis Sci. 2004;29:115–31.
40. Beck K, Beedle M, Bennekum AV, Cockburn A, Cunningham W, Fowler M, Thomas D. Manifesto for agile software development. www.agilemanifesto.org.
41. Shrivastava SV, Rathod U. A risk management framework for distributed agile projects. Inf Softw Technol. 2017;85(1):1–15. doi:10.1016/j.infsof.2016.12.005.

42. Khatri SK, Bahri K, Johri P Best practices for managing risk in adaptive agile process. Proc - 2014 3rd Int Conf Reliab Infocom Technol Optim Trends Futur Dir ICRITO 2014; 2015, Noida, India. Published online. doi:10.1109/ICRITO.2014.7014759
43. Tavares BG, Da Silva CES, De Souza AD. Practices to improve risk management in agile projects. Int J Softw Eng Knowl Eng. 2019;29(3):381–99. doi:10.1142/S0218194019500165.
44. Falessi D, Juristo N, Wohlin C, Turhan B, Münch J, Jedlitschka A, Oivo M. Empirical software engineering experts on the use of students and professionals in experiments. Empir Softw Eng. 2018;23(1):452–89. doi:10.1007/s10664-017-9523-3.
45. Bryman A. Research methods and organization studies (contemporary social research) 1st Ed. London, River Thames, England: Routledge; 1989. Published online.
46. Fan Z-P, Li Y-H, Zhang Y. Generating project risk response strategies based on CBR: A case study. Expert Syst Appl. 2015;42(1):2870–83. doi:10.1016/j.eswa.2014.11.034.
47. Mousavi SM, Raissi S, Vahdani B, Mojtahedi SM. Fuzzy decision-making methodology for risk response planning in large-scale projects. J Optim Ind Eng. 2011;7:57–70.
48. Syedhosini SM, Noori S, Hatefi MA. An integrated methodology for assessment and selection of the project risk response actions. Risk Anal. 2009;29(5):752–63. doi:10.1111/j.1539-6924.2008.01187.x.
49. Saari HL Risk management in drug development projects. A Rep by Helsinki Univ Technol Lab Ind Manag Helsinki, Finl. Published online 2004.

Appendix A. Risk Management Practices

Ranked list of agile methods subcomponents and associated risk management practices

Rank	Subcomponent	Risk management process	Practices
1	Daily meeting	I; MC	Ask and answer the following questions: 1) What did I do yesterday? 2) What am I planning to do today? 3) Do I expect any impediment? Create a backlog of impediments during the meetings Attendees must be stand-up in the meetings Identify, analyze and respond to risks Make meetings brief and successful Manage a risk repository and share it with the team
2	Increment	I; A; PR; MC	Identify, analyze and treat the risks to projects deliveries as soon as possible Concentrate on clearly defined deliverables Identify and disregard non-relevant risk information for the increment Implement a countermeasure in the increment when needed Make frequent deliveries to obtain feedback Deliver a piece of the solution at the end of all iterations to generate value for the business frequently
3	Prototype	I; A; PR; MC	Evaluate the prototype to obtain feedback Validate original estimates to improve planning and estimation processes
4	Product backlog	P; I; A; PR; MC	Address only the highest priority tasks Encourage team members to work together on only a few backlog items to completion before starting new work Identify the dependency between the requirements Implement the riskiest stories early in the project Investigate, refine and consolidate the requirements/user stories Optimize scheduling speed and risk Prioritize requirements, establish the scenario, govern risk management activities, and ensure that the highest value resources for the customer take precedence
5	Sprint planning	I; A; PR; MC	Carry out the initial risk assessment Identify, analyze and respond to risks
6	Sprint backlog	I; A; PR; MC	Analyze technical risks Encourage team members to work together on only a few backlog items to completion before starting new work Identify and respond to the risks Identify the dependency between the requirements
7	Product backlog refinement meeting	I; A; PR	Identify, analyze and respond to risks Investigate, refine and consolidate the requirements/user stories
8	Risk weekly meeting	P; I; A; PR; MC	Identify, analyze and respond to risks
9	Technical specification	I; A; PR; MC	Document important technical specifications Document key technical decisions Evaluate product architecture Refine the specification of the project continuously during development
10	Sprint	P; I; A; PR; MC	Define one-week iterations of customer-requested features Define short release cycles, a few months at most Help carefully design and restructure the process Identify risks early in Sprint Make frequent deliveries to obtain feedback Monitor risks across Sprint Perform shorter sprints Perform some initial sprints prior to distributed development Deliver a piece of the solution at the end of all iterations to generate value for the business frequently Promote frequent feedback Manage a risk repository and share it with the team

(*Continued*)

(Continued).

Rank	Subcomponent	Risk management process	Practices
11	Risk repository	I; A; PR; MC	Create an electronic repository of risks Review the risk repository periodically Manage a risk repository and share it with the team
12	Continuous communication and collaboration	I; A; PR; MC	Encourage and support collaboration between the team and customer Establish communication between the team and the customer to improve the requirements identification. Support the team with tools to improve communication. Promote the team collaboration by using effective communication media like video conferencing, web conferencing and other collaboration tools. Provide training for the team to develop communication and language skills. Use of Community-of-practice (CoP), which are groupings linked by a mutual interest in addressing project issues.
13	Sprint review	I; A; PR; MC	Identify, analyze and respond to risks Periodically present what the team built to their key stakeholders
14	Business alignment	P; I; A; PR; MC	Align agile methodologies to business objectives by using engagement management as a complementary activity
15	Contingency plan	P; I; A; PR; MC	Agree contingency plans in advance
16	Product owner	I; A; PR; MC	Take an active part in the development effort Select the release that has lesser effort, and that makes the most business sense Appoint a person for deciding on product requirements Identify the business-oriented people and define them as the first-class members of the team Define the "Done" concept and transmit it to the team Refine the product backlog items with the product owner Ensure that the team has a clear picture of the stakeholders Establish communication between the team and the customer to improve the requirements identification. Focus on the minimal amount of features required to meet the business needs Actively engage in the iteration Launch the first product versions and use the market to decide where to move next
17	Sprint retrospective	I; A; PR; MC	Identify, analyze and respond to risks
18	Project update	I; C	Update the project as lessons are learned Update the project as new risks and opportunities are detected
19	Development team	P; I; A; PR; MC	Accept responsibility for estimating and completing your own work Audit the team with regard to Project Management and the use of software testing tools Define the "Done" concept and transmit it to the team Define the development team with a maximum of 9 members Discover and address any deficiencies in the architecture early in the project Discuss the standardization of processes with the team Encourage human contact among the team Encourage team members to gradually accept more and more responsibility Encourage team members to work together on only a few backlog items to completion before starting new work Encourage the use of social media among development team members and teams Encouraging team members to do knowledge sharing Ensure that the team has a clear picture of the stakeholders Establish communication between the team and the customer to improve the requirements identification. Improve the estimates through the feedback about the actual time taken in the tasks Make enough architecture to ensure the project produces a system that meets its quality attribute requirements and mitigate the risks Periodically present what the team built to their key stakeholders Recognize that the key to XP is to keep relevant knowledge while keeping good members in the development team Recruit members to the development team who have expertise in XP or offer training to those who are inexperienced Scoring risks to identify critical risks Test from the perspective of both programmers writing tests function-by-function Test the deliveries as much as possible Train the development team in TDD Provide training for the team to develop communication and language skills. Use experienced teams in the project Write the identified risks on cards and pin them to the wall for everyone to see
20	Software code library	I	Create and use software code libraries Encourage code reuse
21	Wiki	I	Use Wiki to document project progress and explain key decisions
22	Cost feasibility analysis	I; A	Focus on product requirements justifiable by cost
23	Qualitative and quantitative analysis	A	Use qualitative and quantitative risk analysis
24	Cross functional team	P; I; A; PR; MC	Change the pair programmers periodically Rotate team members in different functional areas
25	Self-managed team		Balance self-organization adding appropriate lean governance
26	Continuous integration	I; C	Use continuous integration
27	Test driven development (TDD)	I; A; PR; MC	Use TDD

(*Continued*)

(Continued).

Rank	Subcomponent	Risk management process	Practices
28	Stakeholders	P; I; A; PR; MC	Define a milestone to ensure that the project stakeholders come to a reasonable consensus as to the vision of the release Engage actively in the development effort Ensure that the team has a clear picture of the stakeholders Actively engage in the iteration Periodically present what the team built to their key stakeholders Use stakeholder opinion
29	Resource allocation	PR	Allocate resources in a flexible way
30	Strengths, weaknesses, opportunities, threats (SWOT)	I; A; PR; MC	Conduct SWOT analysis
31	Risk driven approach	I; A; PR; MC	Add risk-driven viewpoint to the value-driven approach
32	Artifact transparency	I; A; PR; MC	Make decisions based on the perceived state of the artifacts Make it obvious which stories are riskiest Make risk-related activities and artifacts visible to the entire team at all times Make the list of risks visible to the team at all times
33	Architecture owner	I; A; PR; MC	Create an experienced group to decide on problems with the architecture Create flexible projects that allow the team to incorporate architectural changes Discover and address any deficiencies in the architecture early in the project Make architectural decisions and technical priorities
34	Kanban	I; A; PR; MC	Explicitly present the most valuable tasks that need more attention and energy
35	Adaptation	I; A; PR; MC	Force a change-tolerant incremental approach
36	Business analyst	I; A; PR; MC	Model the organization's current and future state in the area of the solution and identifying opportunities, risks and impacts Own a dedicated business analyst for the team
37	Project manager	P; I; A; PR; MC	Manage risk and any issues as they arise, collaborating with senior business or technical roles as required to resolve them Select the top 5 team-scored risks as critical and determine the product backlog items that should be in the next Sprint
38	Scrum Master	P; I; A; PR; MC	Audit the team with regard to Project Management and the use of software testing tools Define the "Done" concept and transmit it to the team Discuss the standardization of processes with the team Encouraging team members to do knowledge sharing Ensure that the team has a clear picture of the stakeholders Ensure that minimum team disruption happens during project development Establish communication between the team and the customer to improve the requirements identification. Proactively guide the product owner to focus on the minimal amount of functionality that is needed to meet the market's expected needs Recognize that the key to XP is to keep relevant knowledge while keeping good Scrum Masters Recruit a Scrum Master who has expertise in XP or provide a training for those who are inexperienced Scoring risks to identify critical risks Provide training for the team to develop communication and language skills. Use experienced teams in the project Work as a facilitator in regular meetings Write the identified risks on the cards and attach them to the wall for everyone to see
39	Technical coordinator	P; I; A; PR; MC	Engage to ensure that solution and test design across teams is compatible Identify and own architectural and other technically based risks
40	Team leader	P; I; A; PR; MC	Manage risks and issues at the timebox level, escalating to the Project Manager, Business Visionary or Technical Coordinator as required
41	Value driven approach	I; A; PR; MC	Use value-driven approach
42	Pair programming	I; PR; MC	Use pair programming to implement a GUI
43	Business visionary	P; I; A	Arrange a session for the Business Visionary to share his/her vision and answer any questions Contributing to key requirements, design and review sessions Identify and own business-based risk Make the business roles ready, willing and able to engage in face-to-face conversation with the Solution Development roles immediately and whenever their guidance is needed.
44	Milestone	I; MC	Adopt explicit and lightweight milestones Define a milestone to ensure that the project stakeholders come to a reasonable consensus as to the vision of the release Use risk-based milestones
45	Timescale	I; A; PR; MC	Define a timescale for a cycle of evolution
46	Dynamic systems development method (DSDM)	P; I; A; PR; MC	Identify risks early and consider how to mitigate them
47	Must have, should have, could have, and won't have (MoSCow) prioritization	A	Use MoSCow prioritization and timeboxing
48	Tests	I; A; PR; MC	Create and maintain a comprehensive suite of automated tests, which are run and rerun after every change Ensure testing is fully embedded as part of the iterative and incremental development approach Test everything as early as possible

Risk management processes: Artifacts (A); Events (E); Features (F); Roles (R); Techniques and Methods (TM)
Adapted from Tavares *et al.*