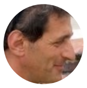
Supported approach for agile methods adaptation: An adoption study

Article · June 2014

DOI: 10.1145/2593812.2593820

Hajer Ayed
University of Namur

Benoît Vanderose
University of Namur



Naji Habra
University of Namur

Supported Approach for Agile Methods Adaptation: An Adoption Study

Hajer Ayed
PReCISE Research Center
University of Namur, Belgium
hajer.ayed@unamur.be

Benoît Vanderose
PReCISE Research Center
University of Namur, Belgium
benoit.vanderose@unamur.be

Naji Habra
PReCISE Research Center
University of Namur, Belgium
naji.habra@unamur.be

ABSTRACT

Adopting agile software development methods is a wide and complex organisational change that usually impacts several aspects of the organisation (e.g., its structure, culture, management practices, produced artefacts, technologies in use, etc). In order to successfully handle the several key challenges, it's crucial to understand the organisation context and carefully study the transformation strategies.

This paper presents an agile transformation experience that has been undertaken in a public organisation in Belgium and during which Scrum was applied in two pilot projects. The first project retrospective shows that the change cannot be accomplished only at the team-level without taking into account the overall structure of the organisation and that we must carefully evolve toward a context-specific adapted method. In the second pilot project, we defined structured and repeatable steps to assist the adoption of agile practices. The experience shows the usefulness of such an approach but suggests that automation efforts should be addressed.

The last section of the paper summarizes the issues encountered and presents the AM-QuICK framework [4] which aims at providing a supported approach to guide the agile adoption, adaptation and assessment.

Categories and Subject Descriptors

K.6.3 [**Management of computer and information systems**]: Software Management; D.2.9 [**Software Engineering**]: Management

General Terms

Management, Experimentation

Keywords

Agile Software Development, Agile Process Assessment, Software Process Improvement, Agile adoption, Software Methods Customisation / Adaptation,

RCoSE '14, June 3, 2014, Hyderabad, India

1. INTRODUCTION

While no longer a new phenomenon and while many software companies claim its undeniable benefits, agile software development is still controversial in some circles such as the public IT sector [14]. The main reason for this scepticism is that the public sector reality is perceived as hardly suitable for agile management structures and culture. For instance, the project budgeting in public organisations is managed very tightly (i.e., the government customers want to know up-front how much a system will cost). This may seem to be in contradiction with the "responding to change" agile value. Actually, there is no contradiction with the agile principles and values but with the common agile practice referred to as "fixed time-material contracting"(i.e., paying for work as it gets done).

More generally, regarding the flexibility of the agile values, most of the practitioners state that agile software development methods and practices can be stretched to a broad set of contexts [2] (e.g., by scaling them to distributed teams, larger projects, etc.) insofar the agile adoption strategies are carefully studied.

However, while several agile adoption success stories exist in the literature [10][11][7][9], many of them are too narrowly focused at a specific organisation and cannot be generalised to other organisations with different needs. Understanding the organisation specific context is therefore a key challenge for agile methods adoption (i.e., how to capture the organisational and project's context and how to adapt accordingly?). Assessing the readiness of the organisation to accept cultural change is also crucial to minimize the adoption risks and avoid failure.

Furthermore, organisations aspiring for agility are commonly confronted to the lack of guidance and assistance approaches. The experiences and contributions the agile community practitioners and researchers have reported are valuable but are more often based on teams' intrinsic non-quantified knowledge instead of neutral quantitative elements that would assist the adoption decisions.

This paper aims to understand the several key challenges of agile software adoption through formalised investigations. It presents an agile transformation experience that has been undertaken in a public organisation in Belgium and during which Scrum was applied in two pilot projects.

In view of the numerous issues that face organisations pursuing agility and the lack of guiding approaches (particularly in the public domain) [16], we propose to generalize the experience and to investigate a supported approach to assist software methodologists in adopting context specific agile methods.

The remainder of this paper is organised as follows : Section 2 describes the two pilot projects and analyse the methodologies that have been applied in both cases. Section 3 examines the results of the case study. Section 4 presents the AM-QuICK framework as an alternative to support agile methods adoption. Finally, Section 5 presents the future venues to investigate in order to improve this support.

2. CASE STUDY AND KEY FINDINGS

2.1 Scope of the Study

The study has been conducted in an IT service that develops business applications for the Walloon ministry of environment in Belgium. For ease of reading, we call this service the Walloon IT Service (WIS). WIS employs 84 people and is part of a middle-sized structure[1] of 2,300 employees. The scope of the experience is the following business units : Project Management (PM), solutions ARchitecture (AR), Quality Assurance (QA), DEVelopment (DEV), Functional Analysis (FA) and Software Maintenance (SM).

This adoption case study started 6 months after the start of the first agile experience in the company. It aimed at analysing the encountered issues and defining a structured approach that would henceforth help project managers to go progressively through agile software development.

For this purpose, we first conducted a project retrospective and a survey-based study to identify the potential improvement opportunities and the key issues that the development team encountered in this first experience (Section 2.2). Based on the results of this experience analysis, we proposed a structured approach to guide the adoption effort in the organisation. The proposed approach is currently being deployed within a second pilot project (Section 2.3).

2.2 First Pilot Project (IDEES)

The desire to adopt an agile approach in the WIS came from a development team working on the IDEES project. The team members, seeking to improve their own working environment, introduced the Scrum framework in their group comprised of one technical project manager and 8 developers. Regarding the organisational structure (i.e., separate units per business roles), the other essential members for the project (i.e., one business project manager, two business analysts, one architect and two testers) were considered external stakeholders to the Scrum development Team. They therefore did not contribute to the decision to go through Scrum.

It is worth noting that prior to Scrum implementation, the team members had no special skills in agile software development. Only the project manager received a Scrum training. More importantly, the decision to go through agile software development was not motivated by impartial studies (no risk evaluation and no assessment whether scrum is the adequate process or not).

Table 1: Retrospective results

Category	Did not worked	Improvement opportunities
Initiation	- Late delivery of the high-level architecture - Lack of time for technology learning - Project plan template : not adapted to agile	- Shorter architecture envisioning
Analysis	- Matching between UML use cases and user stories - Too fine-grained requirements : existence of hypothetical ones	- Do requirements envisioning first then iterate
Meetings	- Too long daily Scrum - No sprint retrospective	- Two daily meetings (one per theme): the project can be divided in two separate sub-projects, two scrum teams
Visibility	- Lack of visibility tools - No clear and visible definition of priorities	- Implement task board and burndown charts, track velocity
Planning	-Self-work assignment, specialised team members : this imply priorities changes - Work estimation : the development effort is not considered when estimating - No team estimation of the work to do : no planning poker or other	- Knowledge sharing, pair-programming ... but may affect productivity
Sprint	- Out-of-scope tasks during the sprint : team disruption - Development sometimes continuous after the end of the sprint - Bug correction tasks resulting from usability testing are not quickly treated	- Team protection from disturbing tasks : Scrum master role - Reinforce the agile culture and clearly define the notion of timeboxed iterations
Scrum team	- Unclear role definition (Everybody is responsible for everything !) : No Scrum master, PO low responsibilities : not directly responsible for prioritisation, few maintenance of the product backlog, etc. - Weak collaboration between the PO and business stakeholders : the team is therefore unprotected	- Communicate the agile culture to the stakeholders

[1] The IT service is part of the General Operational Direction for Agriculture, natural Resources and Environment (DGO3) in Wallonia. The service is developing business applications according to the Common Agricultural Policy of the European Union.

2.2.1 Methodology Analysis

The case study begun after 6 months of Scrum appliance. To understand the methodology applied, we organised a retrospective meeting and asked the team members the following three questions : "What worked well?" "What can be improved?" "What did not worked?". We then organised the generated ideas using the *affinity diagram*[2] business tool.

The retrospective results were highly mitigated : the participants reported an important list of issues but also undeniable benefits, especially at the team-level (e.g., better team collaboration, better visibility of the work progress, regular releases and product delivery within deadlines, quick detection of bugs, etc).

Tab.1 synthesizes the major problems encountered by the development team. One major problem was the lack of involvement of the business units, as their lifecycle is not aligned with the team iterations. For example, the FA unit is stand alone and its resources are not exclusively allocated to IDEES. Due to these circumstances, the analyst cannot always be present at the daily agile meetings and can find difficulties to deliver detailed analysis on time (i.e., at the sprint beginning). Other obvious issues like the lack of guiding tools and indicators and the unclear definition of Scrum roles can also be noticed. It is also worth noting that the team reported no clear definition of roles.

2.2.2 Methodology Assessment

Based on their culture, their values, and the technologies they use, the IDEES project-team selected only the practices that they judged convenient. This process (i.e., selecting and tailoring only suitable practices and techniques) is called methods adaptation or customisation [5].

The customisation process resulted in a hybrid approach that combines Scrum and the old waterfall methodology, more commonly known as "water-Scrum-fall" approach. In fact, the development team used whatever suitable Scrum practices and techniques in their day-to-day product development while other units were still employing the traditional waterfall process for planning, budgeting, testing, etc. However, the FA unit attempted to adapt its process by transforming the project UML Use Cases (UCs) to User Stories(USs). This adaptation of the analysis process was particularly problematic, since the relationship between UCs and USs is not straightforward (there's no real equivalence) and since it charged the analysts with duplicated work. Moreover, the written user stories were found to be of too fine granularity (few hours). This is probably due to the lack of agile skills.

To confirm the extent to which the customised method respects the agile values, we chose to apply the 4-DAT framework proposed in [3]. We only considered the second dimension of 4-DAT, since our purpose was to check the existence of agility in the applied method. Another reason is that it is the only dimension of the four that is quantitative which will allow comparison with the typical Scrum process.

The degree of agility of the customised methodology phases and practices has been measured in terms of five features (see Tab.2) relating to the second dimension of 4-DAT : flexibility (FY), speed(SD), leanness(LS), learning (LG) and responsiveness (RS). If any phase or practice supports a particular agility feature, then 1 point is allocated, otherwise 0 [3].

Table 2: Degree of agility in the applied method (4-DAT - Dimension 2)

	Agility features					
Custom meth.	FY	SD	LS	LG	RS	*Total*
Phases						
Business study (BA)	0	0	0	0	0	0
Pr. initiation (DEV)	0	0	0	0	0	0
Analysis (FA)	1	0	0	0	0	1
Design (AR)	1	0	0	0	1	2
Dev. sprints (DEV)	1	0	0	1	1	3
Testing (QA)	0	0	0	0	0	0
Final QA & release	0	0	0	0	0	0
Maintenance (SM)	1	0	0	0	0	1
Total	4	0	0	1	2	7
Agility degree	4/8	0/8	0/8	1/8	2/8	7/(8*5)
DEV practices						
Iterative incremental	1	1	0	1	1	4
Sustainable pace	1	1	0	1	1	4
Regular builds	1	1	0	1	1	4
Work specialisation	0	1	0	0	0	1
Sprint planning	1	1	1	1	0	4
Daily Scrum	1	0	0	1	1	3
Sprint review	1	1	0	1	1	4
Stories splitting	1	1	0	0	1	3
Total	7	7	1	6	6	27
Agility degree	7/8	7/8	1/8	6/8	6/8	27/(8*5)
FA and AR practices						
UC to US	0	0	0	0	0	0
Fine-grained stories	1	1	0	0	0	2
Arch. envisioning	1	0	0	1	1	3
Arch. iterations	1	1	0	1	1	4
Total	2	1	0	3	2	9
Agility degree	2/5	1/5	0/5	3/5	2/5	9/(5*5)

Table 3: Degree of agility in Scrum and the customised method

Process	Scrum	Customised method
Phases	0.6	0.17
Practices	0.8	*Dev. unit :* 0.67 *Other units :* 0.36

The assessment of the customised method phases shows that the degree of agility of the development sprints is satisfactory (0.6) while it is still very low in the other phases (0, 0.2 or 0.4).

Tab.3 shows a comparison between Scrum and the customised method that the team applied. The agility degree values corresponding to Scrum come from [15]. Here again, we observe a low agility degree in the units surrounding the development team (0.36). While not satisfactory, this value still shows agility. This can be explained by the existence of some Iterative and Incremental practices (e.g., architectural envisioning, requirements iterative refinement, user focused requirements elicitation, etc.).

[2] Also named KJ diagram. The affinity diagram process consists of the following steps : Record each idea on cards or notes, look for ideas that seem to be related and finally sort cards into groups until all cards have been used.[13]

2.3 Second Pilot Project (CONTROLE)

Once the interest in Agile raised in the service, a natural desire to spread agile values among all the teams has followed. However, considering the mitigated results of the first experiment and the lack of relevant surveys to assess teams readiness and analyse risks, the management advocated a soft and cautious transformation strategy, i.e., a step-by-step transformation.

This transformation mode means that agile is applied in a progressive way. First, agile values are applied in a pilot project which is not too risky but is fairly representative of the organisation reality (CONTROLE was selected as the second pilot project for these reasons). Then, the experience is capitalised and spread to other projects transversally or progressively.

In this sense, we took the decision to formalise the adoption approach, by providing structured and repeatable steps to guide and assist agile adoption efforts (Sect.2.3.1).

2.3.1 Structured Approach Proposal

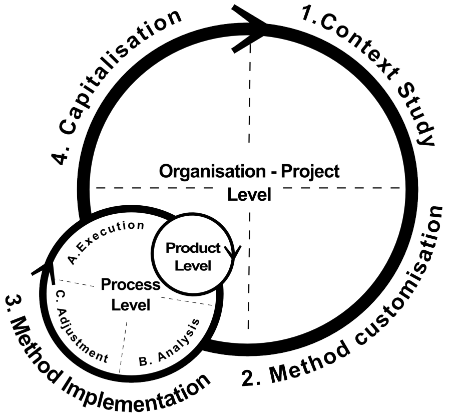


Figure 1: Proposed steps for agile adoption guidance

Fig.1 shows the proposed structured approach that has been adapted from [12] and which is based on the Quality Improvement Paradigm (QIP) [6]. The main cycle consists of the following steps :

1. **Context Study** : This step aims at understanding the context at the organisation level. Interviews, GQM-based diagnosis, and risk-assessment tools can be used. The step also concerns the previous experiences or current processes evaluation in terms of agility and suitability.

2. **Customised Method Construction** : After the context analysis, the suitable agile practices (and eventually training needs) should be identified. At this step, the relevant software components, including both product and process elements and quality assessment metrics, should be identified.

3. **Implementation** : This step is about the execution (instantiation) of the customised method, the analysis of feedbacks and process adjustment.

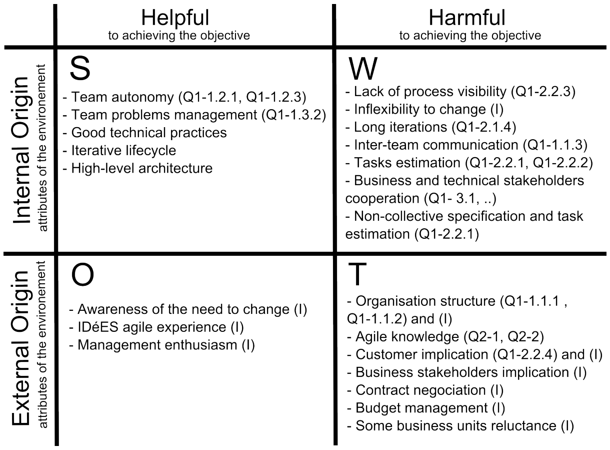
4. **Capitalisation** : Future incoming projects have to be able to profit from the gained experience of the current project. Agile Process Capability Assessment tools such as patterns check-lists, agility degree evaluation and so forth, can be implemented by the organisation in-order to assess their agile maturity.

2.3.2 Context Study

A formalised study was conducted in order to assess the organisation readiness to adopt agility. To this end, we first interviewed the stakeholders from the different business units regarding their actual development process (64 person from all the units), their motivation or reluctance towards agile software development adoption and the units' culture.

We complemented these interviews with two questionnaires adapted from [1]. 47 respond to the survey, which gives a participation rate of 74%. More details on the designed questionnaires can be found in [4].

Based on the collected data, we performed a SWOT risk analysis (Strengths, Weaknesses, Opportunities, Threats) (See Fig. 2). The main intention of this analysis is to specify the organisation's internal and external factors that are favourable or unfavourable to achieve particular business objective. Strength and weaknesses are included within the internal factors, while opportunities and threats are part of external factors.



	Helpful to achieving the objective	Harmful to achieving the objective
Internal Origin attributes of the environement	S - Team autonomy (Q1-1.2.1, Q1-1.2.3) - Team problems management (Q1-1.3.2) - Good technical practices - Iterative lifecycle - High-level architecture	W - Lack of process visibility (Q1-2.2.3) - Inflexibility to change (I) - Long iterations (Q1-2.1.4) - Inter-team communication (Q1-1.1.3) - Tasks estimation (Q1-2.2.1, Q1-2.2.2) - Business and technical stakeholders cooperation (Q1- 3.1, ..) - Non-collective specification and task estimation (Q1-2.2.1)
External Origin attributes of the environement	O - Awareness of the need to change (I) - IDéES agile experience (I) - Management enthusiasm (I)	T - Organisation structure (Q1-1.1.1 , Q1-1.1.2) and (I) - Agile knowledge (Q2-1, Q2-2) - Customer implication (Q1-2.2.4) and (I) - Business stakeholders implication (I) - Contract negociation (I) - Budget management (I) - Some business units reluctance (I)

Figure 2: SWOT analysis

The analysis reveals that the main strengths are team autonomy, transparency in problem management and good technical practices. In fact, 57% of the respondents think that the team members are sufficiently autonomic and 70% think there's enough transparency in problem management.

The main weaknesses identified are the lack of process visibility, the inflexibility to change, and the long iterations (38% agree). Moreover, 50% of the respondents think that the inter-team communication is inefficient. This factor can also be considered as a threat : the organisation structure is business-oriented, the agile development team should therefore communicate efficiently with the business units (e.g., AR unit for example).

The study also reveals the main threat factor as the lack of knowledge of the agile practices : 65% have no agile experience, 18% have a little experience and only 2% have good experience. Based on the results of the questionnaires, we identified training needs and planned two half-days training sessions. Another major threat is the lack of implication of the customer and business roles. 60% of the respondents believe that the customer does not track the work progress frequently and 72% would like a better implication. 70% think that the business stakeholders don't collaborate enough with the technical team members.

We also noticed that, despite the commitment at the management level to adopt agility, there still some reluctance towards the agile philosophy, especially among business units that are working on many projects at the same time or in which the task realisation is not collaborative (e.g., SM unit in which each member works on the maintenance of a specific project).

For better visualisation of the precedent results and to objectively bring out the most critical risk factors, we used the risk assessment methodology proposed in [8] which was primarily designed to help balancing agile and plan-driven methodologies.

The assessment methodology results are pictured in Fig.3. It brings a high level overview of environmental, agile and plan-driven risks and give insights about the resolution strategies to undertake. The results show high risk ratings which strengthens the idea of careful agility adoption.

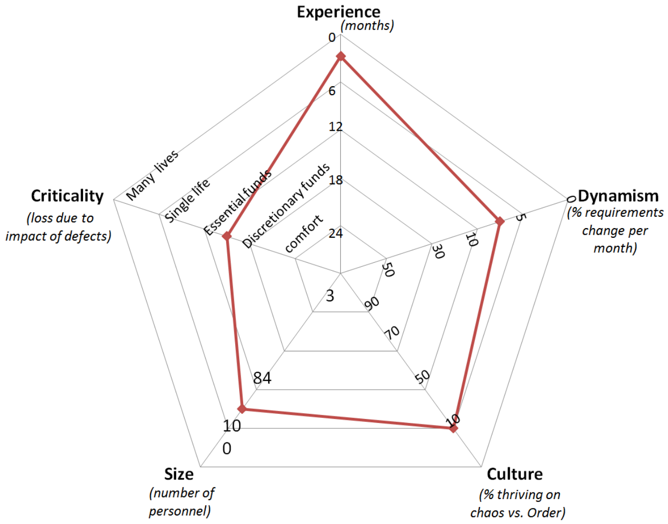


Figure 3: Organisation agility according to [8]

3. KEY FINDINGS

The first pilot project revealed that adopting agile software development methods only at the team level is not sufficient to provide satisfactory level of agility. If agile methods are only adopted at the team level, the team still have to face collaboration issues with surrounding units. Therefore, it is necessary to think agile transversely or at least select the agile practices that do not require high coupling with the other units.

It is noticeable that the agile methods that are not adequately implemented tend to be discarded overtime. In fact, in the first project, we observed that agile practices were selected based on the team preferences and therefore were not systematically well suited to their needs. This is due to the fact that adequate agile methods should originate from the organisation and project context and not from the ability or preferences of the team members.

These issues reinforce the motivation of creating supported methodologies that are structured and context-oriented

In the second pilot project, we introduced a structured approach to guide agile adoption. This approach tends to be repeatable for future projects and also more context-oriented. Indeed, with this approach we have the opportunity to consider which practice would be more suited to the team and to the organisation or how the practice should modify the way actions are performed. To this end, the implementation of a project-wide methodological backlog allowed the prioritization of actions to be undertaken in order to guarantee that specific agile practices would be applicable and efficient. For instance, in the context of the second project, the use cases were automatically discarded and replaced by user stories during the requirements elicitation phase. This replacement helps implements the "user-focused requirements elicitation" practice in an efficient way. It guarantees that no duplication of requirements elicitation techniques arise (i.e., that the team would write user stories based on already defined use cases). This also helps preparing the context to a specific practice. In the project, Entreprise Architect templates were defined early in the development in order to support requirements elicitation based on user stories.

However, the steps performed so far in the second project also show that the result of the context study are still difficult to exploit in a systematic manner. Indeed, much of the information collected during the context study is mainly qualitative and not quantitative. This makes the decision-making process regarding which agile practice to choose based on the context-related findings slightly less objective than one would expect. At this point, this process of relating the context elements with the right practices remains mainly a matter of expert knowledge and intuition. Automation techniques would therefore be beneficial.

4. AM-QUICK FRAMEWORK

In view of the numerous challenges identified in the WIS case study and the absence of guidance to organisations pursuing agility, we propose a supported approach to assist software methodologists in adopting context-specific agile methods: the Agile Methods Quality Integrated Customisation framework (AM-QuICK).

AM-QuICK aims to continuously assist agile methodologists (i.e., during the initial organisation adoption, the customisation phase and the methodology assessment). Starting from initial context analysis, the framework allows the agile methodologists to identify the context attributes and to construct a customised agile method accordingly.

To do so, the framework defines Situational Method Engineering (SME) techniques in order to organise the methodology practices, techniques and artefacts based on context attributes. A metamodel has been designed to structure and classify process elements which offers a way to systematically link context-related elements to relevant practice. It relies on measurement to provide guidance to agile methodologists during the method construction phase and throughout the development process.

The core notion of this approach is to use measurement results from the assessment of the various deliverables produced during the process to reflect on the methodology that is currently followed. Based on this quantitative information, the agile methodologists may be oriented towards a specific process element that is expected to correct the problems detected through measurement.

More details on the approach can be found in [5] and [4].

5. CONCLUSION AND FUTURE WORK

So far, the case study performed in the WIS shows that Agile methods adoption should be carefully planned and follow a disciplined repeatable methodology in order to be fully satisfactory. Although the notion of adaptation could be viewed as an opportunity for the development team to choose the practices that they prefer or are familiarised with, the two pilot project undertaken in the WIS revealed that this approach could in fact lead to inadequate and therefore useless agile practices. On the contrary, the adoption or transformation of an agile software development process should be based on the context-related , i.e., that only practice that may be *beneficial* and *applicable* in a specific organisational context should be selected even though they require small adaptation to the way tasks are performed (e.g., the migration from UML use cases to user stories). Besides, this process of context-oriented adaptation should be performed in a automated and objective manner as much as possible.

The AM-QuICK framework provides the conceptual elements required to support such a context-oriented adaptation. The use of a metamodel to structure and classify process elements offers a way to systematically link context-related elements to relevant practice. In the meantime, the integration of measurement-based adaptation provides an opportunity to make the adaptation process as objective and automated as possible.

Future studies should focus on the objective assessment of the advantages of the proposed approach. To this end, a first step will be to apply the 4-DAT framework to the CONTROLE project during latter stages of its course. The next step will be to apply the complete methodology to other project in order to refine the approach.

6. ACKNOWLEDGMENTS

This research has been co-funded by the European Regional Development Fund (ERDF) and Wallonia, Belgium. The research is also partially supported by the e-Government Chair of the University of Namur.

7. REFERENCES

[1] Comparative agility survey@ONLINE, 2010.

[2] S. W. Ambler. The agile scaling model (asm) : Adapting agile methods for complex environments. Technical report, IBM, December 2009.

[3] A.Qumer and B. Henderson-Sellers. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50:280–295, 2007.

[4] H. Ayed, N. Habra, and B. Vanderose. Am-quick: a measurement-based framework for agile methods customisation. In *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*, pages 71–80. IEEE, 2013.

[5] H. Ayed, B. Vanderose, and N. Habra. A metamodel-based approach for customizing and assessing agile methods. In *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the*, pages 66–74. IEEE, 2012.

[6] V. R. Basili and G. Caldiera. Improve software quality by reusing knowledge and experience. *Sloan Management Review*, 37, 1995.

[7] J. M. Bass. Influences on agile practice tailoring in enterprise software development. In *AGILE India (AGILE INDIA), 2012*, pages 1–9. IEEE, 2012.

[8] B. Boehm and R. Turner. *Balancing agility and discipline: A guide for the perplexed.* Addison-Wesley Professional, 2003.

[9] L. Cao, K. Mohan, P. Xu, and B. Ramesh. How extreme does extreme programming have to be? adapting xp practices to large-scale projects. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2004.

[10] B. Fitzgerald, G. Hartnett, and K. Conboy. Customising agile methods to software practices at intel shannon. *European Journal of Information Systems*, 15(2):200–213, 2006.

[11] B. Fitzgerald, N. Russo, and T. O'Kane. Software development method tailoring at motorola. *Communications of the ACM*, 46(4):64–70, 2003.

[12] A. A. Janes and G. Succi. The dark side of agile software development. In *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software*, pages 215–228. ACM, 2012.

[13] J. Kawakita. The kj method–a scientific approach to problem solving. Technical report, Technical report, Kawakita Research Institute, Tokyo, 1975.

[14] T. Mackinnon. Xp: Have you got the discipline? *TickIt International magazine*, 2004.

[15] A. Qumer and B. Henderson-Sellers. Comparative evaluation of xp and scrum using the 4d analytical tool (4-dat). In *Proceedings of the European and Mediterranean Conference on Information Systems*, 2006.

[16] A. Sidky, J. Arthur, and S. Bohner. A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in systems and software engineering*, 3(3):203–216, 2007.