A Reference Model for Software Agility Assessment: AgilityMod

Ozden Ozcan-Top(✉) and Onur Demirörs

Infomatics Institute, Middle East Technical University, 06800, Ankara, Turkey
ozdentop@gmail.com, demirors@metu.edu.tr

Abstract. In this paper, we present AgilityMod that we developed with the purpose of identifying agility levels of software development projects, indicating agility gaps and providing roadmaps to organizations in adopting agile principles/practices. AgilityMod shares the meta-model structure of ISO/IEC 15504, software process assessment model, however, it differentiates from ISO/IEC 15504 in terms of its process architecture, process descriptions and description of other model elements. In this paper, we focus on the structure of the Model and describe the development stages of the Model. In addition, we briefly present a multiple case study that included eight cases, which was conducted to identify applicability and suitability of the Model.

Keywords: Agility assessment · Agile software development · Agile · ISO/IEC 15504 · AgilityMod

1 Introduction and Motivation

Agile software development approaches are developed as a reaction to traditional methods that are characterized with extensive planning, heavyweight processes and bureaucracy [1]. They are characterized by delivering a working software to the customer through short, time-boxed iterations, and encouraging people to minimize bureaucracy, collaborating, self-organizing, embracing variability, balancing up-front work and just-in-time work, favoring adaptive and exploratory approaches and providing fast-feedback [2, 3]. Agile software development methods are frequently adopted in the recent years by the software community as they are seen as a complete solution for problems like missing deadlines, exceeding budgets, delivering final products that do not meet the needs of the customer [4]. In the state-of-agile survey, VersionOne presents that 52% of the projects are managed with agile techniques in software organizations [5].

On the other hand, Ambler [4] states that there are an increasing numbers of project failures associated with agile strategies. In the 2013 IT Project Success Rates Survey, 30% of the participants reported that they had experienced challenges in an agile project, and 6% of the participants reported failure [6].

These failure stories indicate that organizations do not get a full benefit from agile software development techniques. What we also observed from our personal experiences is that the organizations that are new at agile software development techniques

T. Rout et al. (Eds.): SPICE 2015, CCIS 526, pp. 145–158, 2015.
DOI: 10.1007/978-3-319-19860-6_12

start by selecting a few agile practices, adapt them in the way they prefer and convince themselves as doing agile software development, until they see no improvement or even get worse results than their traditional techniques. In addition "agile" is being used as an excuse for being undisciplined by some of the organizations.

Because of these reasons there is a fundamental need to assist organizations in adopting agile methods/practices and to guide them for improving their agile capability [7].

In the current state, there are about forty models related to agile maturity, including both academic publications and Internet publications [8, 9]. These models are grouped into three based on the classification by Schweigert et al.: ones that are influenced by the structure of CMMI, ones that have a specific leveling structure and ones that do not use explicit leveling structure [8]. They argue that these models do not measure the real agility and support guidance. Instead, they check for the implementation of some specific agile practices.

In one of our previous studies [10], five of the most frequently referenced agile maturity models were applied in an organization, and evaluated. The evaluation was based on six quality criteria: fitness for purpose, completeness, definition of agile levels, objectivity, correctness and consistency. The results of this study indicated that none of these models satisfies all the expected criteria, and need to be improved in terms of scope, definitions of agility levels and objectivity. The most obvious deficiency of the models is that they do not support an agile process architecture holistically. Each model focus on different parts of the software development life cycle. None of the models has a well-defined structure with process inputs, practices and outputs forms.

Among this model quagmire, there is no commonly accepted agile maturity/ assessment model.

In order to cover this need, we developed a well-structured Software Agility Assessment Reference Model (AgilityMod) to be utilized for the agility assessment of software projects. AgilityMod is designed fully compatible with the agile process architecture (the structural design of the processes). The Model provides a complete guidance to organizations so that they could observe their weaknesses and problematic areas and implement the agile processes and practices correctly and in consistency with agile manifest. The model also provides means for helping project teams avoid incorrect tailoring.

In this paper, we present the structure of the Agility Assessment Reference Model_v3.0 in detail, and briefly describe the development stages of the Model. We performed a multiple case study including eight cases with the purpose of observing the applicability of AgilityMod in real projects. Although our focus in this paper is the description of the Model, we briefly present a multiple case study.

2 A Software Agility Assessment Reference Model

Software Agility Assessment Reference Model has been developed in two years time. The first version was published in July 2014. Since then, various works have been published wih the previous versions of the Model [11-13]. The version that is introduced in this paper is the latest and the third one.

2.1 Relation of AgilityMod with ISO/IEC 15504

Existing agile maturity models do not have sound structures [10]. These models provide agile level descriptions, basic key characteristics and assessment questions. They are insufficient in defining outcomes and performance indicators such as practices and work products [10].

Although there are different ways to describe a model, we selected to use ISO/IEC 15504 (SPICE) [14, 15] as a basis for the meta model structure of AgilityMod. Major reason of selecting ISO/IEC 15504 as a basis is, its well-defined and is a commonly accepted meta-model structure. On the other hand, SPICE which has been developed based on 12207 Software Life Cycle proceeses [16], has not been extended to be compatible with agile practices and processes. The current process structure of ISO/IEC 15504 does not comply with agile processes and principles.

ISO/IEC 15504 provides a structured assessment framework for software processes. It facilitates process assessment, provides a basis for use in process improvement and capability determination, and provides process rating, which represents an objective image of current state of a process. The structure of ISO/IEC 15504 allows evaluation and improvement of processes separately. This property brings a significant level of flexibility to process improvement endeavors. There is no need to classify a numbers of processes for a maturity level and define the rationale behind that classification.

On the other hand, although studies show that SPICE can be effectively used in agile contexts [17, 18], existing process structure of ISO/IEC 15504 does not comply with agile processes and principles. ISO/IEC 15504 provides a two dimentional approach for assessment: the capability dimension and the process dimension. The process dimension includes 48 processes defined in conformance to ISO/IEC 12207 Software life cycle processes AMD1 [16] and AMD2 [19]. Capability dimension defines software capability in five levels. We do not go into further details of ISO/IEC 15504 here but we need to mention that, in order to achieve compatibility with agile process architecture, we had to change its components and component descriptions.

2.2 Components of AgilityMod

AgilityMod consists of the following concepts and components:

Dimensions: AgilityMod has two dimensions: the aspect dimension, and the agility dimension. We define agility levels and aspect attributes at the agility dimension, and define aspects, aspect practices, outcomes and example work products at the aspect dimension.

Aspects: Formal process layers of traditional software development are intertwined to each other in agile software development. It is difficult to specify boundaries of agile processes. Aspects which are new modularization of agile processes and practices are integrated under meaningful and agile compatible abstract definitions. They are sets of interrelated and interacting activities. From this point of view, we defined four aspects in AgilityMod, fully covering a software development life cycle: Exploration, Construction, Transition, and Management. Aspects belong to the aspect dimension.

Aspect Practices: Aspect practices are activities or activity groups that contributes to achievement of an aspect purpose and outcomes. Aspect practices also include agile elaborations which describe how plain software development practices can be applied from an agility perspective.

Aspect Attributes: An aspect attribute is an indicator of the aspect performance. It defines the characteristic of an aspect. AgilityMod defines 5 aspect practices, all of which are applicable to all aspect practices. Aspect attributes are represented in the agility dimension, are listed below:

The Aspect Attribute of Level 1 is "Perform Aspect Practices". "Performing Aspect Practices" attribute is a measure of the extent to which purposes and goals of the aspects are achieved by implementing the related practices described in aspect dimension.

The 1st Aspect Attribute of Level 2 is "Iterative". "Iterative" attribute is a measure of the extent to which the work products are delivered in an iterative and incremental way to achieve the specific outcomes.

The 2nd Aspect Attribute of Level 2 is "Simple". This attribute is a measure of the extent to which the aspect practices are arranged and performed by focusing on delivering business value. The purposes of "simple" attribute are to support aspects to eliminate any kind of activity that does not add value and cause waste in software development process, to achieve the balance between the just-in-time works and up-front works and to manage the incoming and outgoing workflows.

The 1st Aspect Attribute of Level 3 "Technically Excellent". This attribute is a measure of the extent to which the agile engineering methods and tools are integrated into aspects to improve productivity and lower defects. Agile engineering practices such as test-driven development, continuous integration, and pair programming and integration of agile tools bring technical excellence to aspects. When technical excellence and other attributes from second level are brought together, teams gain the Agility to manage technical debt, improve team productivity and decrease defects.

The 2nd Aspect Attribute of Level 3 is "Learning". "Learning" attribute is a measure of the extent to which from a broader point of view aspects serve for the purpose of organizational learning and improvement

All these attributes are derived from the agile manifesto and twelve agile principles [20] by combining the related principles together. We cover each principle in one of the aspect attributes.

Example Work Product: Example work products are outputs that are produced at the end of the successful achievement of an aspect or agility attribute.

Fallacy: Fallacies describe the wrong implementations which are assumed to be true.

Generic Agility Practice: Generic agility practices are activities or activity groups that contributes to achievement of an aspect attribute. Descriptions given after each generic practice specify the outcomes after a successful achievement of a practice.

Generic Resource: A kind of resource that is utilized in the conduct of an aspect or agility attribute.

Outcome: Outcomes are observable results of aspects.

After the description of the components, we provide the mapping of AgilityMod components and ISO/IEC 15504 components in Table 1.

Table 1. Mapping Between AgilityMod Components and ISO/IEC 15504 Components

ISO/IEC 15504 Component	AgilityMod Component
Process	Aspect
Base Practice	Aspect Practice
Process Attribute	Aspect Attribute
Generic Practice	Generic Agility Practice
Generic Resource	Generic Resource
Work Product	Example Work Product
Purpose Statement	Purpose Statement
Outcome	Outcome

2.3 Dimensions in Detail

The Figure below shows the dimensions of the Model, the aspects, and the agility levels:

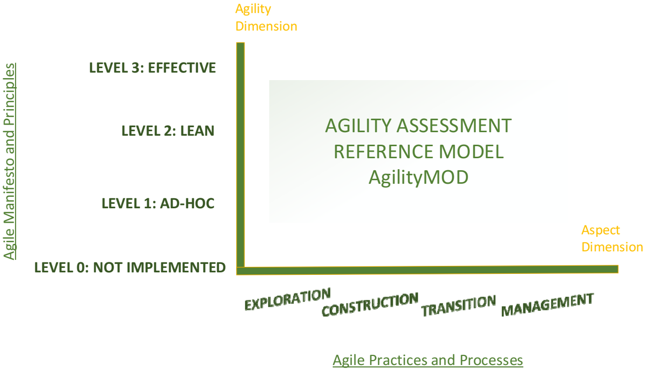


Fig. 1. Major components of AgilityMod

Agility Dimension

Each aspect might be at one of the 4 levels of agility that are “Not Implemented”, “Ad-Hoc”, “Lean” and “Effective”. When an aspect’s agility progresses from the bottom level: “Not Implemented” to the top level: “Effective”, its conformance to agile values and principles increases.

At Level 0, aspect practices are either partially achieved or not achieved. At Level 1, organizations are capable of performing fundamental development processes such as requirements development, design, coding, integration, testing, and deployment consistently. There are transition attempts towards the agility by exploring best fitting agile practices or approaches. Aspect practices are implemented and aspect purposes are achieved; however agile values and principles are not fully incorporated into aspect practices. At Level 2, work products are developed iteratively and incrementally, non-value added activities are eliminated from the aspect practices, balance is achieved between adaptive and predictive works. At Level 3, each aspect is performed to achieve delivering value with high productivity and low defects by employing agile engineering practices and using agile tools to support a continuously improving environment.

In Table 2, we provide aspect attributes and generic agility practices related to each agility level.

Table 2. Generic Agility Practices and Aspect Attributes of Agility Levels

Agility Level	Aspect Attribute	Generic Agility Practices
Level 1: Ad-Hoc	1.1 Performing Aspect Practices	GP 1.1.1 Perform aspect practices
Level 2: Lean	2.1 Iterative	GP 2.1.1 Develop work products in an iterative and incremental way
		GP 2.1.2 Communicate effectively
	2.2 Simple	GP 2.2.1 Balance the predictive work and adaptive work
		GP 2.2.2 Employ minimally sufficient ceremony
Level 3: Effective	3.1 Technically Excellent	GP 3.1.1 Incorporate agile engineering methods/practices to the aspect practices
		GP 3.1.2 Integrate tools to aspects to improve the productivity
	3.2 Learning	GP 3.2.1 Support collaborative work and shared responsibility
		GP 3.2.2 Adopt agile leadership styles and adjust the behaviors towards mistakes of people
		GP 3.2.3 Encourage people in the organization to participate in learning, teaching and improvement
		GP 3.2.4 Collect measures to support learning and improvement

Aspect Dimension

Aspect dimension is characterized with 4 aspects: Exploration, Construction, Transition and Management. In this dimension, we describe the aspect purposes, the outcomes, the aspect practices, the relation of the aspect practices with outcomes, the example work products and the fallacies which needs to be avoided. We provide the aspect practices of each aspect in Table 3.

Table 3. Aspect Practices based on each Aspect

Aspects	Aspect Practices
Exploration Aspect	E.AP1: Capture the customer and user needs
	E.AP2: Elaborate requirements artifacts
	E.AP3: Detect and resolve conflicts of requirements arti-
	E.AP4: Specify dependencies among requirements arti-
	E.AP5: Manage the requirement artifacts
	E.AP6: Make the artifacts visible to everyone
Construction Aspect	CN.AP1: Elaborate the work items
	CN.AP2: Explore the design
	CN.AP3: Develop the solution
	CN.AP4: Ensure the correctness of software at developer
Transition Aspect	T.AP1: Create and Manage the Workspace
	T.AP2: Integrate the Code
	T.AP3: Deploy the solution
	T.AP4: Test the integrated solution
	T.AP5: Make the progress visible
	T.AP6: Create the supporting documentation
Management Aspect	M.AP1: Initiate the project
	M. AP2: Form the team
	M.AP3: Align and adopt the environment
	M. AP4: Establish the physical work space
	M.AP5: Plan the progress
	M.AP6: Estimate the work items
	M.AP7: Monitor the progress
	M.AP8: Manage and mitigate the risks

3 Development Stages of AgilityMod

In this sub-section, we describe the progress of development of AgilityMod in time. All of the development stages, the findings and the actions that was taken to improve the model, worth further explanation. However, because of the page limitation, we just mention them here and leave the detailed discusions to further studies.

We developed the first version of the Model after the research on agile software development, agile adoption and agile transition concepts. We also evaluated existing agile maturity models through a multiple case study [10]. By exploring agile models [21-25], we understood the agile values in practice and the reasoning behind the practices, and developed *the aspect dimension* of the model. By exploring how organizations mature in agile environments [26-30], we developed *the agility dimension* of the model. Following that, we performed an exploratory case study to observe the applicability of AgilityMod in a real case and to discover improvement opportunities related the Model. The details of the case study can be found at [11]. We updated and published the second version of the Model based on our observations, and the case study findings [13].

The second version of the Model were reviewed by three agile software development and process improvement experts. Three experts reviewed the Model. Expert A has 10 year-experience in software process improvement consultancy. He is an internal and external process assessor since 2001. He has knowledge on agile processes. Expert B: a SEI (Software Engineering Institute) authorized CMMI lead appraiser who has 4 years hands on practices on Scrum and ISO/IEC 15504. He is from India. Expert C is a hands on agile practitioner and trainer. She is a scrum master since 2006. She is a consultant in agile adoption and co- author of a book in French on Agile. The book was awarded "Best French Informatics Book of 2012 by the Association Française d'Ingénierie de Systèmes d'Information (AFISI), whose members are voting the best book annually for over 20 years. She is also a CMMI consultant. She is from Canada.

We asked experts to review the model based on a set of criteria (fitness for purpose, completeness, definitions of agile levels, objectivity, correctness and consistency) that were set in one of our previous studies [10]. Two of the experts mentioned that the component descriptions are clear enough to perform agility assessment and the model is capable of providing directions for improvement on agility and can be used as a roadmap by organizations for getting better at agility. Expert A expressed his ideas in these topics as follows:

> "*The model aims to bring a maturity view on the agile principles, and I believe it is a successful model. Using ISO 15504 as a reference model supports the validity of the model and increases the possibility of usage among organizations. The model perfectly fits the need of providing roadmap by organizations for getting better at agility*"

In the Model, we described the agility in an abstract way to cover various agile methods and approaches. Therefore it is very important the Model components' and component descriptions' both cover all agile principles in an abstract way and be independent of any agile method. Experts evaluated the Model from these perspectives and rated as fully and largely achieved ratings. Expert A found the level of abstraction appropriate when the audience of the model is considered as daily agile practitioners. The more you keep the abstraction at a reasonable level, the more the experience and knowledge of the assessor becomes important. The target group that is expected to use AgilityMod for assessment are experts who have specific knowledge and experience in the agile domain.

Expert C gave specified the descriptions of components that is too specific or valid for a particular agile method. The Model is updated considering the expert comments and 3rd version of it is published.

In terms of "consistency", experts specified minor inconsistencies and concluded that the Model is internally consistent and does not include any logical conflicts.

All experts thought that the Model is "correct" such that all component descriptions are compatible with agile values and principles.

One of the requirements of an assessment model is to achieve a required level of "objectivity" in order to guarantee the repeatability of the assessment results. AgilityMod aims to achieve "the objectivity" through clear description of aspect purpose and outcomes, and aspect and agility practices. AgilityMod uses a common rating scale with ISO/IEC 15504 [15] that clearly specifies the ranges for rating. In terms of objectivity, Experts A mentions that clarifying the normative and informative features of the Model would increase the objectivity. Expert C calls attention to the need for a rating scheme for assessing multiple agile projects and specifying agility of an organization rather than project basis. We are going to define the rules for assessing agility of organizations, however, this improvement is not in the scope of this study. Therefore we consider that these comments of experts do not violate "objectivity" characteristic of AgilityMod.

We performed required changes on the Model based on experts' feedback and published the third version [31].

4 Case Studies

Following the second update of the Model, we performed a multiple case study including eight software companies. The domains of companies ranges from technical media to home appliances, from ERP solutions to multimedia solutions and e-governance solutions. The team sizes of the assessed eight projects change between 6 employees to 45 employees.

The purpose of this case study is to answer the following research questions:

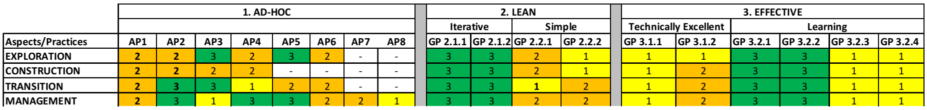
RQ1: How suitable is the third version of Software Assessment Agility Reference Model to be used with the purpose of identifying aspects' agility, identifying the agility gaps and providing roadmaps for improving agility in a software project?

RQ2: What are the strengths and weaknesses of the third version of AgilityMod?

For assessment performance, we met groups of people who belong to project teams, asked them to answer a set of questions and evaluated the projects' direct evidences. Following the assessment process, we prepared detailed assessment reports and discussed the findings with assessment team members, or in some cases with all project team members, managers and CEOs. We obtained feedback from them about the following issues:

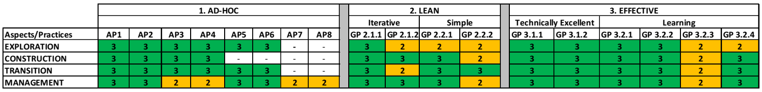
- if there is a misunderstood concept or practice presented in the report or presentation
- if the report or presentation covers all the improvement areas that are noticed about project's agile processes
- if the findings presented to them are beneficial for getting better at agility
- if they follow the same improvement path suggested in the report and presentation
- which of the suggested practices are new to them or noticed previously
- and to what extent the presented findings and improvement opportunities in their projects overlap with reality

We used a four-level scale to express the achievement of the aspect attributes: “not achieved (0-red), partially achieved (1-yellow), largely achieved (2-orange) and fully achieved (3-green) and not applicable (NA)”. For an agility level to be reached, all practices should be largely or fully achieved. Below, we provide the colored schemas of the assessment ratings for Case #1 and Case #2 as samples which enable capturing detailed results at a glance. For the other assessment results, the technical report [32] can be requested.



	1. AD-HOC								2. LEAN				3. EFFECTIVE					
									Iterative		Simple		Technically Excellent		Learning			
Aspects/Practices	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	GP 2.1.1	GP 2.1.2	GP 2.2.1	GP 2.2.2	GP 3.1.1	GP 3.1.2	GP 3.2.1	GP 3.2.2	GP 3.2.3	GP 3.2.4
EXPLORATION	2	2	3	2	3	2	-	-	3	3	2	1	1	1	3	3	1	1
CONSTRUCTION	2	2	2	2	-	-	-	-	3	3	2	1	1	2	3	3	1	1
TRANSITION	2	3	3	1	2	2	-	-	3	3	1	2	1	2	3	3	1	1
MANAGEMENT	2	3	1	3	3	2	2	1	3	3	2	2	1	2	3	3	1	1

Fig. 2. Rating of Each Practice of Case 1



	1. AD-HOC								2. LEAN				3. EFFECTIVE					
									Iterative		Simple		Technically Excellent		Learning			
Aspects/Practices	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	GP 2.1.1	GP 2.1.2	GP 2.2.1	GP 2.2.2	GP 3.1.1	GP 3.1.2	GP 3.2.1	GP 3.2.2	GP 3.2.3	GP 3.2.4
EXPLORATION	3	3	3	3	3	3	-	-	3	2	2	2	3	3	3	3	2	2
CONSTRUCTION	3	3	3	3	-	-	-	-	3	3	3	2	3	3	3	3	2	3
TRANSITION	3	3	3	3	3	3	-	-	3	2	3	3	3	3	3	3	2	3
MANAGEMENT	3	3	2	2	3	3	2	2	3	3	3	2	3	3	3	3	2	3

Fig. 3. Rating of Each Practice of Case 2

Figure 4 below shows the achieved agility levels of Case 1 and Case 2 in a bar chart view.

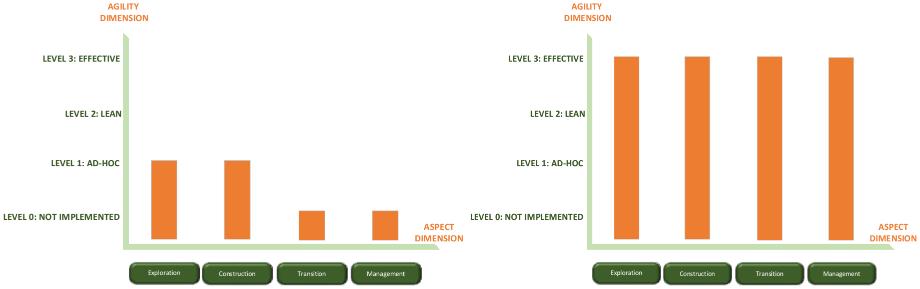


Fig. 4. Achieved Agility Levels of Aspects for Case 1 and Case 2

Figure 5 below, shows the gap between the ideal case (outer trapezoid) and the current situation of Case 2 (trapezoid in the middle) and Case 1 (inner trapezoid). The data to draw this radar chart is obtained by adding the rating values of each aspect given on Fig. 2 and Fig. 3. Here, we can see from radar chart display that, even if Case 2 reached Level 3 agility levels for all of the aspects, there are still some space for improvement for Case 2, especially for the exploration aspect.

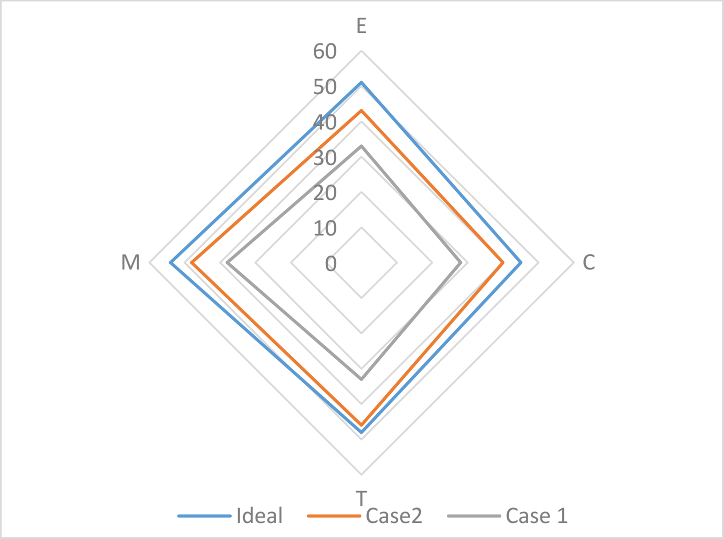


Fig. 5. Display of the Gap among Ideal Case, Case 1 and Case 2

Because of the page limitation, we will not go into the details of the case study discussion part. Interested readers may request the resources in [32, 33] for further information. The feedback that we obtained from people about the findings and the model are provided in [33] in detail. In Table 4, we present the feedback results. Each person, who evaluated the accuracy of our findings about the aspects, gave a rating in a range from "Not Achieved (NA)" to "Fully Achieved (FA)".

In order to construct the table, we calculated the median of the ratings if the assessment findings were rated more than one person. In the overall, 84.4 % percent of the evaluation indicated that the findings and improvement suggestions fully overlap with current problems in the projects. The remaining 15.6 % thinks that aspect findings largely overlaps with current problems. Achieving such high ratios for finding the gaps in the projects is an indicator of how successful the Model in revealing agility improvement opportunities and the potential of the Model for the usage of agility assessment.

Table 4. Rating of the Findings' Accuracy by Aspect Owners

Aspects	**C1**	**C 2**	**C 3**	**C 4**	**C 5**	**C 6**	**C 7**	**C 8**
Exploration	FA	FA	LA	FA	FA	FA	FA	FA
Construction	FA	FA	FA	FA	FA	FA	LA	FA
Transition	FA	LA	FA	FA	FA	FA	LA	FA
Management	FA	FA	FA	LA	FA	FA	FA	FA

5 Conclusion

In this study, we presented the structure of Agility Assessment Reference Model in detail. We briefly described the development stages that had continued until the Model reached a maturity level to be published. AgilityMod's meta-structure relies on the meta-structure of ISO/IEC 15504. Therefore, we explained the reasons of this choice and the relation between the components of two models.

AgilityMod presents dimensions, aspects, aspect attributes, aspect practices and generic agility practices, which are very specific to agile software development phenomenon. On the other hand, the Model has been developed independent of any specific agile software development method. Its holistic structure allows the assessment of software projects developed with different type of agile software development approaches.

Considering the multiple case study results, the opinions of the interviewees on the results and the feedbacks of experts, we conclude that we could use AgilityMod to identify the agility gaps in projects, to specify agility levels of aspects and to provide roadmaps to projects for agility improvement.

The Model allows agility assessment of projects in terms of four aspects instead of checking compatibility of processes to some agile practices. In AgilityMod, we do not only evaluate the existence of some specific agile practices such as performing daily stand-up meetings or pair programming or collective code ownership. Instead, we evaluate the aspects from a holistic approach and understand if the teams are capable of keeping the design structure sound while responding to the changes quickly, are disciplined, and serving for organizational learning or not.

The level of abstraction used in the Model, objectivity, accuracy, completeness and consistency issues were evaluated and approved by the experts. The multiple case study that we conducted with the 3rd version of the Model including eight cases, indicated that the Model can be applied with a reasonable effort in software companies by agile experts. Improvement suggestions given based on the Model can be utilized as a roadmap for improving organizations' agility.

AgilityMod is a model that was developed to assess software projects' agility. We consider updates to extend the Model's coverage for organizational agility assessment. In addition, more case studies will be valuable that is to be conducted by other researchers to observe the applicability of the Model.

Acknowledgement. This study was supported by Turkish Scientific and Technological Research Council of Turkey (TUBITAK), Project 113E528.

References

1. Dingsøyr, T., Dybå, T., Moe, N.B.: Agile Software Development: Current Research and Future Directions, vol. 1. Springer, Heidelberg (2010). ISBN 978-3-642-12574-4
2. Rubin, K.S.: Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional (2012)
3. Cockburn, A.: Agile software development: the cooperative game (agile software development series). Addison-Wesley Professional (2006)

4. Ambler, S.W., Lines, M.: Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press (2012)
5. VersionOne, 8th Annual State of Agile. http://stateofagile.com/8th-annual-state-of-agile-form/2013
6. Ambler, S.: IT Project Success Rates Survey Results (2013). http://www.ambysoft.com/surveys/success2013.html
7. Sidky, A.: A structured approach to adopting agile practices: The agile adoption framework. Virginia Polytechnic Institute and State University (2007)
8. Schweigert, T., Vohwinkel, D., Korsaa, M., Nevalainen, R., Biro, M.: Agile maturity model: a synopsis as a first step to synthesis. In: McCaffery, F., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2013. CCIS, vol. 364, pp. 214–227. Springer, Heidelberg (2013)
9. Schweigert, T., Vohwinkel, D., Korsaa, M., Nevalainen, R., Biro, M.: Agile maturity model: analysing agile maturity characteristics from the SPICE perspective. Journal Of Software: Evolution And Process (2013)
10. Özcan-Top, Ö., Demirörs, O.: Assessment of Agile Maturity Models: A Multiple Case Study. In: Woronowicz, T., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2013. CCIS, vol. 349, pp. 130–141. Springer, Heidelberg (2013)
11. Özcan-Top, Ö., Demirörs, O.: Assessing Software Agility: An Exploratory Case Study. In: Mitasiunas, A., Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2014. CCIS, vol. 477, pp. 202–213. Springer, Heidelberg (2014)
12. Özcan-Top, Ö.: AgilityMod: Agility Assessment Model v1.0, Informatics Institute, METU/II-TR-2014-37
13. Özcan-Top, Ö.: AgilityMod: Agility Assessment Model v2.0, Informatics Institute METU/II-TR-2014-38
14. I. O. f. Standardization and I. E. Commission, ISO/IEC 15504-2:2003 Information technology – Process assessment – Part 2: Performing an assessment, ed (2003)
15. ISO/IEC 15504-5:2012 Information technology – Process assessment – Part 5: An exemplar software life cycle process assessment model, ed (2012)
16. ISO/IEC 12207:1995/Amd.1:2002, Information technology — Software life cycle processes, ed (2002)
17. Lami, G., Falcini, F.: Is ISO/IEC 15504 applicable to agile methods? In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) Agile Processes in Software Engineering and Extreme Programming. LNBIP, vol. 31, pp. 130–135. Springer, Heidelberg (2009)
18. Bianco, C.: Agile and SPICE capability levels. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 181–185. Springer, Heidelberg (2011)
19. ISO/IEC 12207:1995/Amd.2:2004, ed (2004)
20. Agile Manifesto (2001) www.agilemanifesto.org
21. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice-Hall (2002)
22. Beck, K.: Embracing change with extreme programming. Computer **32**, 70–77 (1999)
23. Poppendieck, M., Poppendieck, T.: Lean software development: An agile toolkit. Addison-Wesley Professional (2003)
24. Middleton, P., Joyce, D.: Lean Software Management: BBC Worldwide Case Study. IEEE Transactions on Engineering Management, **59**, 20–32 (2012)
25. Ambler, S.: Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons (2002)
26. Elssamadisy, A.: Agile adoption patterns: a roadmap to organizational success. Addison-Wesley Professional (2008)

27. Tomasini, A., Kearns, M.: Agile Transition: What you need to know before starting. InfoQueue Enterprise Software Development Series (2012)
28. Adkins, L.: Coaching agile teams: a companion for ScrumMasters, agile coaches, and project managers in transition. Addison-Wesley Professional (2010)
29. Williams, L., Brown, G., Meltzer, A., Nagappan, N.: Scrum+ engineering practices: experiences of three microsoft teams. In: 2011 International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 463-471 (2011)
30. Benefield, G.: Rolling out agile in a large enterprise. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, p. 461 (2008)
31. Özcan-Top, Ö.: AgilityMod: Software Agility Assessment Reference Model v3.0, Informatics Institute, METU/II-TR-2014-392014
32. Özcan-Top, Ö.: AgilityMod: Software Agility Assessment Reference Model v3.0 Application: Case Study Results, Informatics Institute, METU/II-TR-2014-40 (2014)
33. Özcan-Top, Ö.: Agilitymod: A Software Agility Reference Model for Agility Assessment, PhD, Information Systems. Middle East Technical University, Ankara (2014)