# A Technique to Classify and Compare Agile Methods

João M. Fernandes and Mauro Almeida

Departamento de Informática / CCTC, Universidade do Minho, Braga, Portugal

**Abstract.** This manuscript describes a technique to perform comparisons on agile methods, based on a set of relevant features and attributes. This set includes attributes related to four SWEBOK Knowledge Areas (KAs) and to the Agile Manifesto principles. With this set of attributes, by analyzing the practices proposed by each method, we are able to assess (1) the coverage degree for the considered KAs and (2) the agility degree.

## 1 Introduction

This manuscript presents a technique to compare and classify agile methods, using as criteria a set of selected attributes. The proposed technique is exemplified by comparing two agile methods: XP and Scrum. The technique intends to be a contribution for the creation of a guide to help developers on the selection of the software development method (with a specific focus on agile methods) that best fits a given development context. The attributes chosen for this study were selected (1) to assess the coverage degree of each method to four SWEBOK KAs which are transversal to all development methods, and (2) to assess the agility degree for each method.

## 2 Proposed Technique

We have chosen to distil a set of important features inductively from several methods and compare each method agains it [1]. Regarding the attributes used in our technique, we selected four of the eleven knowledge areas (KAs) defined in the SWEBOK [2]: (1) Software Requirements, (2) Software Construction, (3) Software Testing, and (4) Software Engineering Management. With this subset of attributes, we intend to assess the coverage degree of each method with respect to the selected KAs. A fifth attribute, which relates the Agile Manifesto principles and the practices advocated by a given agile method, was selected to assess the agility degree of the methods.

To classify the existence of practices, advocated by the agile methods, that support the selected KAs, and to characterise the coverage of the principles of the Agile Manifesto by each method, there criteria were considered: (1) **NS – Not Satisfied** (the proposed practices/concepts of the method do not support the sub-attribute or principle); (2) **PS – Partially Satisfied** (the proposed practices/concepts support the sub-attribute or principle, but some of its aspects are not considered); (3) **FS – Fully Satisfied** (the proposed practices/concepts entirely support the sub-attribute or principle).

The challenge in quantifying the coverage of a given sub-attribute or principle, by a set of practices proposed by each analysed method, has lead to the choice of a qualitative

A. Sillitti et al. (Eds.): XP 2010, LNBIP 48, pp. 385–386, 2010.

classification system. Although simple, a qualitative classification system satisfies the objective of our technique.

Due to space limitations, we just consider the results for the Software Requirements attribute, but similar exercises are available for the other attributes. Concerning this attribute, each agile method was analysed against all the following sub-attributes: (1) Software Requirements Fundamentals, (2) Requirements Process, (3) Requirements Elicitation, (4) Requirements Analysis, and (5) Requirements Validation.

## 3 Comparing XP and Scrum

A summary of the classification of XP and Scrum under this attribute is presented in the next table.

| *Sub-attribute* | *XP* | *Aspects of XP* | *Scrum* | *Aspects of Scrum* |
|---|---|---|---|---|
| Software Requirements Fundamentals | **Partially Satisfied** | – Definition of the concepts related to software requirements;<br>– No distinction between different types of requirements. | **Fully Satisfied** | – Definition of software requirements related concepts;<br>– Distinction among different types of requirements. |
| Requirements Process | **Fully Satisfied** | – Definition of a process to collect, specify, analyze and validate requirements, explicitly defining the actors and activities to be undertaken. | **Fully Satisfied** | – Pre-game phase;<br>– Sprint planning meeting;<br>– Definition of a process to collect, specify, analyse and validate requirements, explicitly identifying the actors and the activities to be undertaken. |
| Requirements Elicitation | **Fully Satisfied** | – On-site client;<br>– User Stories. | **Fully Satisfied** | – Close interaction between the client and project team in early stages;<br>– Product backlog;<br>– Sprint backlog;<br>– Possibility for any of the evolved entities to add a new element to the Product Backlog. |
| Requirements Analysis | **Partially Satisfied** | – Classification of requirements by setting priorities (business value criteria);<br>– No techniques for detection and resolution of conflicts between requirements. | **Partially Satisfied** | – Classification of requirements by setting priorities (business value criteria);<br>– No techniques for detection and resolution of conflicts between requirements. |
| Requirements Validation | **Fully Satisfied** | – Functional and acceptance tests written by the client. | **Not Satisfied** | |

Based on this type of classification of the agile methods, a worksheet, available at http://www.di.uminho.pt/~jmf/AgMethComp.xls, allows the generation of a report that helps on the quantitative evaluation of the agile methods considered in a particular context. The user just needs to decide the weights to assign to the sub-attributes and principles that are part of the technique. With the results for all the methods under comparison, a decision can be made regarding the one to use.

## References

1. Sol, H.G.: A feature analysis of information systems design methodologies: Methodological considerations. In: IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, pp. 1–8 (1983)
2. Abran, A., Bourque, P., Dupuis, R., Moore, J.W. (eds.): Guide to the Software Engineering Body of Knowledge - SWEBOK. IEEE Press, Los Alamitos (2004)