



Agile Software Development of Embedded Systems

Version : 1.0
Date : 2006.01.09
Pages : 12

Authors

Minna Pikkarainen

Status

Final

Confidentiality

Public

Agile Deliverable D.2.9

Mapping Agile Software Development onto ISO 12207

Abstract

This document describes the agile software development mapping with the Industry Implementation of International Standard ISO/IEC 12207.



ITEA

INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT



TABLE OF CONTENTS

CHANGE LOG	2
1. Introduction	3
2. Research Context	3
2.1 Research Goals	3
2.2 Data Collection for the Analysis.....	3
2.3 Description of the Case Projects	4
3. Mapping Between Agile Software Development and ISO 12206	5
4. Conclusions and Future Work	10
5. References	10



AGILE document template
Deliverable ID: N.N.X-N


Page : 2 of 12

Version: 1.0
Date : 20.02.06

Status : Final
Confid : Public

CHANGE LOG

Vers.	Date	Author	Description
0.1	26.5.05	Minna Pikkarainen	First full draft created
0.2	13.2.2006	Minna Pikkarainen	Comments from Tuomas Ihme
1.0	15.2.3006	Minna Pikkarainen	Final Version based on comments from Outi Salo

	AGILE document template Deliverable ID: N.N.X-N	Page : 3 of 12
		Version: 1.0 Date : 20.02.06
		Status : Final Confid : Public

1. INTRODUCTION

This document reports the results of the agile assessment results mapping (based on the experience of four projects) with the traditional software development standard (Industry Implementation of International Standard ISO/IEC 12207). The purpose of this document is to support the standardization of agile software development.

The document is composed as follows: section 2 presents the research goals and a short description of the case projects, section 3 and 4 illustrate the agile software development mapping results between the ISO 12207 standard. The last section concludes the document with final remarks and outlines future actions.

2. RESEARCH CONTEXT

2.1 RESEARCH GOALS


The goal of this research is to define how the ISO 12207 key activities were addressed in an agile development. The data for the analysis is collected in agile assessments which were performed for four different agile software development projects. All of these assessments were focused on the actual project work (e.g. requirements management, project management and development) excluding the preparation and marketing phases and the project maintenance. Therefore, these problem areas are outside the scope of this mapping.

2.2 DATA COLLECTION FOR THE ANALYSIS

Agile practices (such as XP Planning Game (Beck 2000), Test Driven Development (TDD) (Beck 2000) and post-iteration workshops (Salo, Kolehmainen, et al. 2004)) have been offered as a promising way to meet the needs of rapid external changes in the dynamic market situations, lower the defect rates and make the development times faster (Beck 2000; Boehm and Turner 2003; Highsmith 2004). Agile methods have also been claimed to increase customer satisfaction (Boehm and Turner 2003) and the motivation of the software developers (Beck 2000; Greene 2004). Only few organizations or projects can, however, take a selected agile method in use as proposed (Lindvall, Muthig, et al. 2004; Williams 2004). In many cases, the best way is to use a hybrid method where the most suitable practices from the agile and plan-driven (Boehm and Turner 2003) methods are integrated based on the organization needs (Boehm and Turner 2003; Lindvall, Muthig, et al. 2004; Williams 2004)(Boehm and Turner 2003; Lindvall, Muthig, et al. 2004; Williams 2004)[2, 6, 7]. Both agile and plan-driven (Boehm and Turner 2003)(Boehm and Turner 2003) approaches, however, have the situation dependent shortcomings that, if left unaddressed, can lead to project failure (Boehm and Turner 2003). The organizations need to see compelling evidence on the hybrid projects' efficiency before the agile practice is deployed in a larger scale (Lindvall, Muthig, et al. 2004). Therefore, when an organization is building up its own method this way, a systematic agile assessment approach is useful.

Agile Assessment in all cases was based on Agile Assessment method including the following steps: 1) focus definition, 2) agility evaluation, 3) data collection planning, 4) data collection, 5) analysis and 6) final workshops. In the first step, the goals are set for adopting agile methods. The second step provides a better understanding on how suitable and effective the various agile methods would be in the specific projects.

The agile assessment data was collected using interviews, agile assessment workshops, and from the recorded iterative SPI actions (Post-Iteration Workshop data) and improvement opportunities (from project postmortems). In addition, various metrics data was utilized in the analysis. The agile assessment workshops were conducted in order to identify the strengths and weaknesses of the software development process and to discuss the possibilities of increasing the agility of the development process together with the project stakeholders. The assessment workshops thus supported the project and organizational learning between different projects and also the development of a agile software development model on the organizational level. The agile assessor was from VTT and well aware of the available agile methods, as well as of the agile assessment method. (Pikkarainen, Salo, et al. 2005).

	<p style="text-align: center;">AGILE document template Deliverable ID: N.N.X-N</p>	Page : 4 of 12
		Version: 1.0 Date : 20.02.06
		Status : Final Confid : Public

2.3 DESCRIPTION OF THE CASE PROJECTS

Scrum Projects

The first evaluated project used the Scrum (Schwaber 1995) method The project team consists of six persons including four software engineers, a Scrum master/project manager. The core of project 1 (developers) works in an open office space. The project had been going on for over a year conducting three Sprints. The Scrum method has been used in the project for about five months until now within the following principles:

1. Every use case is planned
2. Documenting only based on the external needs
3. Reflection workshop after each Sprint
4. Demo presentation and feedback after each Sprint together with the customer
5. Design as simple as possible including the unit testing

The second evaluated project also used the Scrum method for their software development. The idea of piloting the Scrum method was based on the excellent experiences of project 1 on the Scrum method use. The project team of project 2 consists of four persons. One of them works as a Scrum master as well as a project manager, and the other three project members are software engineers who are focused on development work as well as unit testing activities. The project team has been operated for about four months. The Scrum methodology had been successfully utilized in the project for about three months until now.

Mobile-D Project

The third evaluated project used the Mobile-D (Abrahamsson, Hanhineva, et al. 2004) method. The core of the case project consisted of four software developers and one tester who worked in an open office space. The team conducted five software development iterations in all (1x1 week, 3x2 weeks, and 1x1 week). The team leader of the project was an expert of the Mobile-D process and was provided by the research organization. Thus, the team had constant support and coaching available on adopting the new agile process model and its practices and tools.

Integration of XP, Scrum and Mobile-D Project

The fourth evaluated project used the selected set of agile practices of Extreme Programming (XP) (Beck 2000), Scrum (Schwaber and Beedle 2002) and Mobile-D (Abrahamsson, Hanhineva, et al. 2004) methods (e.g. planning games, user stories, test driven development, pair programming and reflection workshops). The team included four developers, a project manager and an on-site customer. All developers worked in the open office space.



3. MAPPING OF AGILE SOFTWARE DEVELOPMENT AND ISO 12206

In this section, the results of the mapping of the Agile Performance Assessment experience (from four case projects) and ISO 12206 is described. NA in the following analysis means information that is not collected or analysed related to the evaluated project.

Phase in ISO 12206	Key Activities	Scrum projects	XP, Mobile-D and Scrum integration project	Mobile-D project
Acquisition		<i>Requirements were defined in the product backlog (i.e. list of product level features that is gleaned from vision document and gathered from brainstorming session). Product owner collected the requirements from the several customers.</i>	Not possible to answer with the current knowledge	<i>Requirements were defined in product backlog by customer and product manager.</i>
Supply		NA	NA	NA
Development	Process Implementation	Selection of the agile life cycle model based on Scrum literature. Developers and management read Scrum book	Used Agile Practices were described in the Information Radiators on the walls of the Open Workspace	Mobile-D was selected because of its good use experiences in mobile SW development and the possibility of getting process model deployment support (from VTT)
	System Requirement Analysis	<i>Product Backlog</i> Requirements for the whole product were defined and listed in the <i>product backlog</i> . <i>They were analysed and prioritized by the product owner and Scrum master before the Sprint planning</i>	<i>Product Backlog</i> Requirements for the whole product were defined for using the traditional methods, Requirements were documented as user stories	<i>Product Backlog</i> Product level Requirements were analysed and prioritized by the customer, product manager and the technical architect before the planning day. Priorities were documented in the product backlog
	System Architectural Design	<i>Traditional</i> <i>Architecture design</i> was implemented by the developers using the standard layer architecture solutions.	System architecture was developed during the project. It was updated and <i>refactored during the second</i>	<i>Traditional</i> <i>Architecture</i> was implemented by the product architect who made an <i>architecture baseline (prototype)</i>



AGILE document template
Deliverable ID: N.N.X-N

		Architecture was quite stabile, it did not include many updates during the project.	<i>iteration</i>	<i>before the poject starting. The baseline was updated during the project by the developers.</i>
Software Requirement Analysis	<i>Sprint Planning</i>	The defined features were selected, analysed for the next iteration in <i>the Sprint planning meetings</i> together with the Scrum master, product owner, development team and quality engineering. As a result of the sprint planning, project had a sprint backlog where the development team was committed.	<i>Planning Games</i>	<i>Planning Days</i>
Software Architectural design		Architecture was based on common standards. It was developed by the development team	Light architecture description was done in the first and <i>refactored</i> in the second working iteration	Architectural baseline (prototype) was defined before the development. Baseline was updated during the project.
Software detailed design		Principle of the Self-organizing team gave much freedom to the developers Technical issues were discussed and decided face to face in daily meetings. Open Office space helped decision making	<i>As simple as possible</i> to give much freedom to the developers. Daily technology risk management and problem solution	<i>As simple as possible</i> , lack of design documentation. (Not recommendation of the Mobile-D method) Technical issues were discussed and decided in face to face discussion.
software coding and testing	<i>Pair programming</i>	in complex situations only. One comment from the developers was that pair programming was good when creating new features.	<i>Pair Programming</i> in complex situations.	<i>Pair Programming</i> in Using pair programming was very controversial in the project: few developers of the team were very reluctant to adopt of PP where as the others would have liked to increase its usage
		<i>Unit testing automation</i> in many cases. Automation was improved all the time during the project. Test Driven development was used in some situations.	Not automated but <i>Test Driven Development</i> in suitable situations	<i>Test Driven Development</i> was not used in project. <i>Creation of extensive set of Unit tests</i> was found important in automating the testing with J-Unit tool. (Not recommendation of



AGILE document template
Deliverable ID: N.N.X-N

Page : 7 of 12

Version: 1.0
Date : 20.02.06

Status : Final
Confid : Public

				Mobile-D method)
	Software Integration	<i>Continuous, hourly Integration environment was implemented during the project. Hour level integration helped the iterative testing work.</i>	<i>Continuous nightly integration</i>	<i>Continuous integration, release days at the end of each iteration</i>
	Software Qualification Testing	<i>Quality Engineer made the integration tests after the each sprint</i>	<i>Refactoring when needed</i>	<i>Refactoring when needed, overall system was tested after the development phase . Project had the separated external quality engineering team which purpose was to do iterative integration testing</i>
	System Integration	<i>Automated Continuous Integration</i>	<i>Automated Continuous Integration</i>	<i>System testing was implemented when the first version of the overall product was ready.</i>
	System Qualification Testing	<i>Traditional</i> Testing was done feature by feature, roles and responsibilities of the quality engineering team was unclear during the project	<i>Traditional</i> Customer and project manager made the tests for the product parts developed by the previous iteration	<i>Traditional</i> System testing was started after the eight weeks development period. The management purpose was to to implement it concurrently with the project
	Software Installation	NA	NA	NA
	Software Acceptance support	NA	NA	NA
Operation		NA		
Maintenance		NA	<i>Traditional</i>	NA
Documentation		As simple as possible, only based on external needs, documentation was taken as a part of prioritization in the planning day. The architecture documentation was implemented by the external team based on the workshop with the developers	<i>As simple as possible, simple architecture description, tasks and project status in information radiators.</i> <i>Any criteria or needs for the documentation in agile project, was not defined</i>	Product Backlog, user stories, tasks cards which were both in electronic format and in the information radiator
Configuration management		NA	<i>Traditional version management system was used. Everybody had</i>	NA

