



Agile Software Development of Embedded Systems

Version : 1.0
Date : 2005.06.13
Pages : 39

Authors

Minna Pikkarainen

Status

Final

Confidentiality

Public

Agile Deliverable D.5.2.3

Nokia Agile Assessment Experience Report

Abstract

The purpose of this document is to present experiences and results of Agile Assessment at Nokia.



I T E A

INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

CHANGE LOG

Vers.	Date	Author	Description
0.1	3.3.05	Minna Pikkarainen	First Full Draft Created
1.0	13.6.05	Minna Pikkarainen	Comments from Tuomo Kähkönen

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

TABLE OF CONTENTS

CHANGE LOG	2
1 Introduction	4
2 Background Information	4
2.1 Description of the Company	4
2.2 Agile Assessment Environment	4
2.3 Agile Assessment Background Theory	5
3 Piloted Agile Assessment Method	8
4 Agile Assessment Data Collection	8
5 Trial Implementation	10
5.1 Focus Definition	11
5.2 Data Collection Planning.....	11
5.3 Interviews	12
5.4 Analysis	12
5.5 Workshop and learning.....	12
6 Results and Analysis of the Agile Assessment.....	13
6.1 Current State Analysis	13
6.1.1 Initializing	13
6.1.2 0-iteration	20
6.1.3 Iteration planning	21
6.1.4 Working	25
6.1.5 Integration	29
6.1.6 Finalizing	30
6.2 Agile Practice Utility evaluation.....	31
6.3 Empirical Evaluation.....	33
6.4 Analysis	35
6.5 Agile Assessment suitability in current project context	36
6.6 Future work.....	36
7 Summary	37
References	38

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

1 Introduction

The purpose of this document is to describe the results of the Nokia Agile SW development project evaluation.

The report is composed as follows, the section 2 presents background and theoretical assumptions for the agile assessment. Third section provides piloted agile Assessment method. Section 4 illustrates the data collection methods for the agile assessment. In Section 5 the main steps for the agile assessment is described. Section 6 includes the main results and experiences of the agile assessment analysis. The last section concludes the paper with final remarks and outlines future actions.

2 Background Information

2.1 Description of the Company

This report includes the Agile Assessment results of the Nokia Agile Assessment which was made as a co-operation between the Nokia product platforms and VTT Electronics in the project called agile ITEA.

The Purpose of the agile assessment was to get the objective evaluation of the agile software development project efficiency and to find out strengths, improvements and possible new agile methods that would be suitable also in the Nokia future development.

2.2 Agile Assessment Environment

The aim of the evaluated project was to develop a new generation error management system for Nokia Product projects. Early versions of the error management system were developed during the past years using the excel scripts. Now the purpose was to transfer error data in a new database and redevelop new software to support the metrics collection and management in product projects.

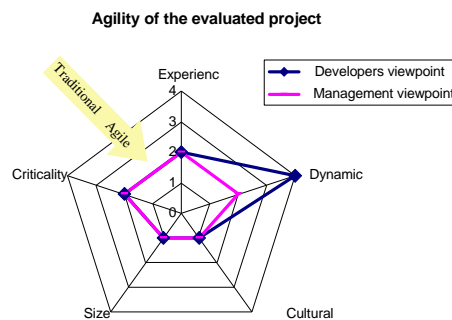
Agility degree of the evaluated project was very high based on the Boehm and Turner agility dimensions (Size, Dynamism, Personnel, Culture and Experiences). Average experience of the project developers was about three years. Application developed in project was not business critical but its result data validity is important for the product project management. External dynamic in project was high. According to the project manager and customer the evaluated project had many changes in the project scope and technical implementation during the working iterations. However, because of the working agile practices the project developers felt that project was quite stabile. Requirements which were selected in planning game were not changed between the working iterations. Size of the project was quite small. It had four developers, project manager and the on-site customer. Project culture also supported well the agile based development. Project team had much freedom to make the technical solutions. Figure 1 provides an example of using the polar chart with the Nokia case study.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Fig.1. Agility of the evaluated project



In figure 3 each axis is labelled based on the Boehm and Turner dimension definitions. When a project's data points for each factor are joined and the resulting shape is located directly toward the center, this suggests using agile method. Shapes that gather distinctly toward the periphery suggest using plan-driven methodology. More varied shapes suggest the use of a hybrid method including both agile and plan-driven practices. According to the empirical case analysis the evaluated project environment actually supported the use of the agile software development practices. The context evaluation leads to the assumption that the agile practices would be suitable and efficiency in the evaluated project. This evaluation does not, however, give any details of the hybrid practices actual status in the evaluated project.

2.3 Agile Assessment Background Theory

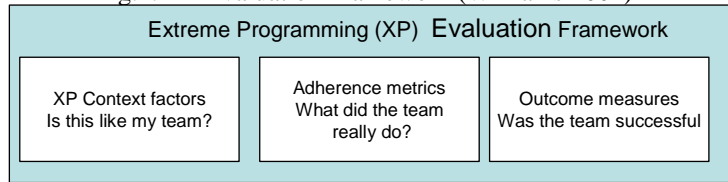
Even if, some agile practices (e.g. the functional increments and the faster development cycle) are seen as a one solution for more efficient product development (e.g. improved customer satisfaction, quality, productivity and cost) (Lindvall, Muthig et al. 2004). Both agile and plan driven development practices have situation dependent shortcomings that, if left unaddressed can lead to project failure. Performance efficiency especially in the situations where the project uses some new technology (e.g. agile practices) also concerns the project developers and managers in organizations (Abrahamsson 2002). Especially, before deploying the new agile practices, in larger scale, the organizations need an objective assessment and evidences of the used agile project as well as agile practices efficiency and utility (Lindvall, Muthig et al. 2004) including the benefits and risks analysis. Williams has recently presented a framework for the XP evaluation (Williams 2004) based on these needs (in Figure 2).

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Fig.2. XP Evaluation Framework (Williams 2004)



In the XP Evaluation Framework Williams evaluates the team agility (context factors) using the Boehm and Turner agility dimensions which affects on the agile and plan-driven method selection (Boehm and Turner 2003). The five factors are following (Boehm and Turner 2003):

- 1) **Size:** The Number of people of the team,
- 2) **Critically:** Products safety criticality (in agile development the potential difficulties can cause of the simple design and lack of documentation),
- 3) **Dynamism:** Degree of requirements and change in technology,
- 4) **Personnel:** The skill and experience of the tea, and
- 5) **Culture:** The agile project culture should support the developers' freedom to make the technical solutions for achieving the high motivation for the development team.

The context factor analysis gives a starting point for a more deep agile project assessment. It gives an appraisal if the agile methods would be suitable and effective in the evaluated projects. According to Williams the purpose of the Context factor analysis is to answer to the question: is this team like my team? Context information helps the comparison between the different agile and plan-driven teams and explains the variations in results when different teams use the similar agile practices.

In the second part of the framework the Adherence Metrics is presented. The aim of this step is to analyze the agile project current state.

The last part of the XP Evaluation framework is the Outcome measures s. In this phase, the current (agile) and previous (plan-driven) projects measurement results are compared. (Williams 2004). (Figure 1). The purpose of the Outcome Measure part is to answer following questions: 1.) was the team successful, 2.) was the team more or less productive, 3.) was the quality of software better or worse, 4.) was the team morale high or low and 5.) were customer satisfied with the product. The question of the team successful can be implemented using the XP practices risk and utility assumptions (in Table 1) (Williams 2004):

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Table 1. Assumptions of the XP practice utility (Williams 2004)

Utility	Agile Practices	Reasons
High Utility, Low Risk	Stand-up meetings	Utility: Communication, sharing, quick help
	continuous integration	Utility: Improvements in build and configuration management activities
	Short Releases	Utility: Frequent Customer Feedback, focusing
High Utility, Medium Risk	Test Driven Development	Utility: Code Quality improvements, Risk: Technical challenges
	Pair Programming	Utility: Increased Quality of Code, Increased team communication, Risk: Resistance among the programmers
	On-site customer	Utility: Customer gets what he wants not what he asked, Risk: High Costs
	Simple Design	Utility: Possibility on focusing on the product rework instead of creating documentation, Risks: Maintainability of the end product without documentation
High Utility, Low Risk	Coding standard	Utility: Quick understanding how the code works (useful in pair programming)
Medium Utility, Medium Risk	Sustainable Pace	Utility: Happy employees and higher quality code in the long run, Risk: Can be difficult if the programmers are aware of the schedule pressures, demands the culture change
	Refactoring	Utility: Support the implementation of simple design, improve the code maintainability Risk: Increased maintenance costs
	Planning Game	Utility: Beneficial project management strategy, Risk: Requirements that has not been chosen for current iteration should be ignored when design decisions are made
	Collective Code Ownership	Utility: Knowledge sharing, Risk: Nobody is responsible of the code
Low Utility, Low Risk	Methaphor	Utility: Most teams can be successful without methaphor

Data collection in XP Evaluation framework is done mainly using the metric based data but also using the interviews and surveys. The purpose of the qualitative data collection is to clarify the information behind the numbers.

The Williams framework do not, however, give much specifics for the agile project efficiency assessment in the situation where the project uses the combination of the several agile practices (Boehm and Turner 2003) such as XP Planning Game (Schwaber and Beedle 2002; Greene 2004; Salo, Kolehmainen et al. 2004), Scrum sprint goal definition (Schwaber 1995) and post-iteration workshops (Salo, Kolehmainen et al. 2004). Also, only little empirical knowledge of the XP evaluation framework exists.

Before deploying new practices the software development project teams need to understand their effects and implications. For example there is need for an evaluation which agile practices would increase productivity and cycle time in the projects which still produce the high quality products. (Lindvall, Muthig et al. 2004). While efficient plan driven software development project means that the software development project stays on schedule and products high quality products which are easy to further develop and demands the requirements. (Abrahamsson 2002). Efficient agile project also characterizes by good teamwork between the stakeholders, high motivation between the developers and high technology freedom to make new innovations and concentrate to the technical solutions instead of the documentation (Beck, Beedle et al. 2001). Even if, there has been arguments that the agile development would faster the development times, the aim to stay in schedule is not anymore the most critical evaluation driver. More important is to answer to the rapid change flow with the frequent releases (Beck, Beedle et al. 2001). Definition of suggested efficiency assumptions for the used agile practices in Table 2 helps the agile project evaluation.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Table 2. Efficiency assumptions for the agile practices

Efficiency assumptions	Agile Practice
High quality releases (Williams 2004)	TDD (Williams 2004)
Maintainability (Williams 2004)	Refactoring (Williams 2004)
Requirements understanding, Customer satisfaction, high visibility	Product Backlogs, Iteration planning and goal definition, requirement analyzing and selection in the iteration planning (Williams 2004) , on-site customer, Information Radiators
Improved teamwork, communication, High motivation between developers, High freedom	Methaphora, Dynamic architecture, On-site customer, Iteration planning, Stand up meetings, Pair Programming (Williams 2004)
Purpose to become more efficient (Salo, Kolehmainen et al. 2004)	Post-iteration workshops (Salo, Kolehmainen et al. 2004)

A goal in itself either in organizations or projects is not to use agile practices. For example the pair programming is a good way to solve the problematic programming solutions more efficiency utilizing teamwork. Even if, it is not so efficient when developing simple software modules. (Padberg and .Muller 2003). Therefore, the aim is to find the most suitable integration of the agile and plan driven practices and tailor them so that the utility goals are achieved.

3 Piloted Agile Assessment Method

The various agile methods and experiences of their use have been described in several publications in literature e.g. (Grenning 2001; Greene 2004) (Paetch, Eberlein et al. 2003) (Macias, Holcombe et al. 2003). However, those publications do not provide a sufficient and adequate answer for the three main challenges, which, according to our experience are:

1. How to evaluate the agility of the software product development?
2. How to tailor agile practices, methods and tools to fit the needs of the projects?
3. How to tailor agile practices, methods and tools to fit the needs of the organization?

Agile Assessment is a method which is developed to provide answers to these questions above. Purpose of the Nokia Agile Assessment was to pilot the developed agile assessment method in the project which has a high agility level.

4 Agile Assessment Data Collection

Williams presents the metric based *data collection* method for the agile project assessment in the Outcome Measures parts in her agile efficiency framework Adherence Metrics and (Williams 2004). In our case, the purpose was to get the reliable information of the developers, project managers and customers' opinions of the agile practice current use, utility and efficiency in different project phases. The empirical data for the agile practice evaluation was qualitative. It was collected both using the focused interviewing technique (Merton, Fiske et al. 1956) and the collaborative workshop. The focused interviewing means the situation where the interview subject matter are known but the structured detailed questions do not exist (Kerlinger and Lee 2002). According to the empirical case study experiences, the focused interviewing technique is suitable in the agile project assessment situation. This because the structured detailed questions for the agile projects evaluation would be difficult to define without existent standard reference model. The aim of the collaborative workshop was to analyze the developers and project managers' opinions of the agile practice utility and efficiency in different

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

development iterations and secondly to evaluate the objective results got from the interviews.

Many authors have concluded that CMM and agile approaches are compatible (Paulk 2002; Turner and Jain 2002; Vriens 2003) (Reifer 2003), (Glass 2001) (Glazer 2001) (Martinsson 2003) (Kähkönen and Abrahamsson 2004) and it has been suggested that CMMI could be used as a tool when building up methods that combine agile and traditional elements (Kähkönen and Abrahamsson 2004). As a basis for the agile assessment a list of most commonly used agile practices (from XP, Scrum and Mobile-D (Abrahamsson, Hanhineva et al. 2004)) and their efficiency assumptions in Table 2 were collected. The practices were classified under CMMI level 2 and 3 process areas. Another classification dimension was the incremental project phases where the practice can be used. The use of CMMI as a framework helps to ensure that all relevant software development and management aspects are covered (Kähkönen and Abrahamsson 2004). The purpose of this categorized practice list was to give focused but unstructured questions for the assessment.

Table 2. Agile practice mapping with CMMI level 2 and 3 goals

CMMI Level 2 and 3 Goals and agile practices	Phase in incremental software development (Graham 1989)
SG Manage requirements	
Product backlogs (Scrum), short iterations	Initial investigation
On-site customer (Paulk 2001)	Initial investigation, Iterative requirements definition
User story definitions (Paulk 2001) in iteration planning, Last iteration user story status checking in the iteration planning	initial investigation
Continuous integration (Paulk 2001)	Implementation
Continuous verification for developed releases	Implementation
Establish estimates	
Agile development model selection (Kettunen and Laanti 2004)	Initial investigation
Iteration goal definition (Scrum)	Iterative requirements definition
Tasks estimations for one-two week releases (Paulk 2001), Effort estimations for one-two week releases (Paulk 2001)	Iterative requirements definition Implementation
Develop a project plan	
Suitable experienced team selection and training, Stakeholder participation in iteration planning, On-site customer	Initial investigation
Information Radiators as a tool in the project planning	Initial investigation, Implementation
Task definitions and effort estimations (Paulk 2001) and technical risk identification and analysis in planning meetings	Iterative requirements definition
Metrics, collection e.g. in daily meetings, Simple documentation	Implementation
Plan the milestones in the agile development	Initial investigation
Monitor project against the plan	
Use of metrics in iteration planning and post-iteration workshops	Iterative requirements definition
On-site customer, Developers freedom, concentrating to the technical solutions instead of the documentation	Implementation
Project velocity and commitments for small releases (Paulk 2001)	Iterative requirements definition, Implementation
Risk status analyzing in the iteration planning meetings	Iterative requirements definition
Keep the milestone meetings in the agile development	Implementation, maintenance
Quality assurance	
Post-iteration workshops	Increment planning
Refactoring, Pair-programming	Implementation, maintenance
Configuration management	
Release planning in agile environment	Initial Investigation,
Product and iteration backlogs (Scrum)	Initial investigation, Iterative requirement definition, Implementation
Small releases, Collective code ownership, Continuous integration (Paulk 2001)	Implementation
Select product component solutions	
Methaphor (Paulk 2001)	Initial Investigation
Technical trials, Dynamic Architecture, Developers freedom	Implementation
Develop the design	
simple design (Paulk 2001), Teamwork (e.g. Rapid 7) in specification, Implement the design, Coding standards (Paulk 2001)	Implementation
Refactoring (Paulk 2001), TDD, Pair programming, Simple documentation	Implementation, Maintenance

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

The collaborative workshop were developed using an adaptation of process workshop technique which uses the prepared worksheets in the process analysis (Dingsoyr and Moe 2004) and the Focus Group Interviewing technique where one moderator interviews the two or more persons at the same time (Kerlinger and Lee 2002). The worksheet for each agile development phase and iteration was prepared beforehand (Figure 3).

Fig. 3. Worksheet for the agile practice suitability evaluation

Iteration 1	
Agile Practices	Benefits
Traditional practices	New potential agile practices

During the workshop, each of the development phase and iteration were analysed together with the workshop team using the agile assessment worksheet. The worksheet included the evaluation of the agile practices, achieved benefits, used supporting traditional practices and new potential agile practices. (Figure 2). The collaborative workshop in our case included the three parts:

1. The developers and project mangers defined agile practices, benefits, problems, traditional practices and new potential practices for the worksheets without discussion.
2. The results were analysed together with the workshop team.
3. Interview based results of the agile practice suitability (improvements and strengths) were presented to the project team and discussed together with the developers, management and agile assessor utilizing the previous partitions' analysis results.

5 Trial Implementation

The Agile Assessment method includes the following steps: 1.) Focus Definition; 2.) Agility Evaluation; 3.) Data Collection Planning; 4.) Interviews; 5.) Analysis and 6.) Workshops and learning phases (Figure 4).

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

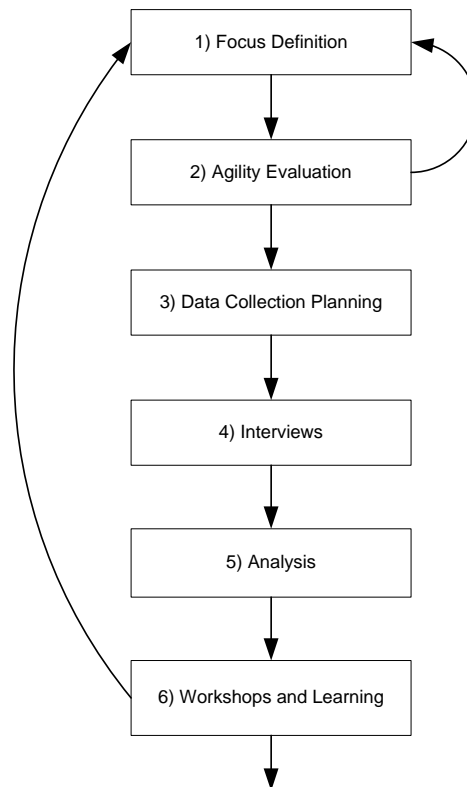


Figure. 4. An Agile Assessment Framework

Focus definition in Nokia was done in one workshop with the evaluated project manager and developers.

5.1 Focus Definition

The goal of the Nokia Agile Assessment was:

1. To clarify how the agile practices has worked
2. Have the planned benefits achieved using the agile software development methods
3. What were the main problems with agile practices and what would be the best (existent and new potential) agile practices in the next Nokia software development projects.

Evaluation was focused in the CMMI level three engineering and level two process areas. Evaluated project used the agile practices which were the integration of the XP, Scrum and Mobile-D practices in each agile SW development phases.

5.2 Data Collection Planning

Data Collection planning was done by the Agile Assessor. It was very complex task in the situation where the idealized list of the agile practices has not been proven to work. The assessment itself is, still, is possible, if the assessment the focus is rather to improve the working than for evaluating the maturity of the agile practices.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Data Collection Planning procedure is described in section 4 objectives and Metrics of the Trial.

5.3 Interviews

Analysis was based on four interviews which were made for the project manager, two project developers and “virtual” customer. Interviews took 1-2 hours. Analysis was based on the agile practices that were selected based on the evaluation of different agile methods, agile literature, Mobile-D experiences and CMMI analysis. Agile practices were the integration of the XP, Scrum and Mobile-D best practices in each agile SW development phases.

5.4 Analysis

The goals of the XP evaluation framework Adherence metrics and Outcome Measures parts are to define what the team really do and was the team successful (Williams 2004). In this case study, the current state and efficiency of the agile project was evaluated for each incremental development phase utilizing the defined utility and efficiency assumptions. Results were based on the interview and workshop results.

5.5 Workshop and learning

The main results of Agile Assessment were defined in the *workshop* where the agile practices were analyzed together with the project manager and development team.

The Workshop was implemented in the three main steps which were 1) Data collection, where the developers and project managers defines agile practices, benefits, problems, traditional practices and new potential practices for the worksheets without discussion. 2) Result analysing, which is done together with the workshop team and 3) An interview based evaluation of the agile practice suitability (improvements and strengths) which are presented by the agile assessor and discussed together with the utilizing the previous partitions’ analysis results. An Example of the workshop results is illustrated in Figure 5.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

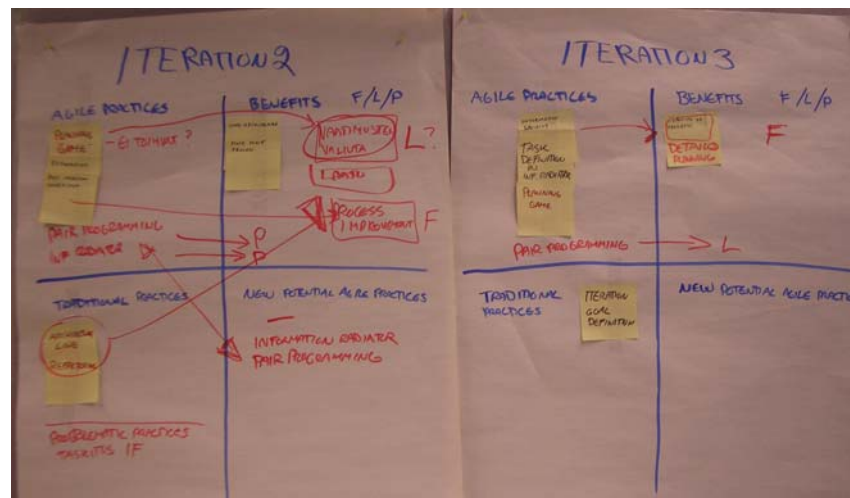


Figure 5. Agile assessment results of iterations 2 and 3

After Agile Assessment, the improvements and defined agile practices should be prioritised and further analysed in the organization internal meetings. Before taking the agile practices, methods, or tools into large-scale use within an organization, they must be evaluated by testing them in a pilot project or projects. Suitable practices should be piloted in projects the selection of which could be based on projects' agility. The piloting is carried out according to the improvement plan. Besides piloting, the use of agile practices, methods and tools are evaluated in feedback sessions.

6 Results and Analysis of the Agile Assessment

6.1 Current State Analysis

The purpose of the Williams Adherence Metrics step is to define the agile project current state {Williams, 2004 #327}. The purpose of this section is to describe the results of the Nokia Agile projects current state analysis.

6.1.1 Initializing

Agile software development starts with project initializing (Figure 1) where the initial requirements and overall description of the project future release iterations are planned together with the project management and customer. Purposes of the agile SW development initializing are:

1. To prepare the agile SW development by selecting the experienced team, preparing the needed workspace, guidelines, process instructions, offer trainings etc.
2. To define the initial requirements together with the customer and project management
3. Report the initial requirements in the product backlog
4. Plan the milestone / steering group meetings in Agile SW development
5. Deal the product initial requirements in the short iterations
6. Plan the initial schedule and budget for the project

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

SP.1.1, SP 1.2 Requirements management, Obtain understanding and commitment of initial requirements

Initial requirement definition during the project preparation

Current state

Requirements were mainly based on the previous error reporting system which was redeveloped during the project. Additional requirements were collected from steering group which had participators from different product development projects at Nokia. Steering group participators were the product end users. Initial requirement sources were the product owners, metric reports and the previous error management tools. Initial requirements were collected and analysed in a general level in the first steering group meeting.

In the initial phase the system functionality, non-functionality and database requirements were evaluated and reported in the general level user stories by the “virtual” customer and project manager. One user story was created for each result report.

Acceptance criteria for the project were that the result reports of the new error management tool would include the same data than the old reports using the new regenerated database and functionality. Therefore, the system verification was done comparing the results of the old and new data. Iterations were also planned during the initialization. User stories were initially dealt with the iterations and the steering group accepted the budget as well as the initial evaluation of the project schedule.

Analysis of the Used agile practices

“Virtual” customer defined the user stories together with the project manager before the first development iteration. Project members saw that the requirements were at too general level.

Strengths:

- + Milestone meetings and initial requirements analysis in steering group
- + Requirements reporting and communicating to the developers using the User Stories
- + User stories were initially dealt with the iterations and the steering group accepted the budget as well as the initial evaluation of the project schedule.

Improvements:

- User stories were in too general level

Analysis compared to CMMI

-

New potential agile practices

- Rapid 7 or other workshop techniques could be used in requirements analyzing together with the customer and project management
- Product backlog could be more visible in project

SP.1.3 Project planning, Define project life cycle

Short releases, incremental development

Goal in agile SW development was an incremental development with short iterations according the agile principles.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Current state

Evaluated project was developed based on several predefined agile practices (e.g. Short iterations, Planning Game, Pair Programming etc.). Main results of the project were the metric data based reports that the developed system will produce. These reports content could not, however, specified at very detailed level at the beginning of the project. Therefore, the Agile, short iteration based, development model was seen as a one solution for the project verification needs.

In the initial phase the project manager and project “virtual” customer generated the list of the used agile practices in project and planned the project deliverables. Agile based SW development process model were evaluated and inserted in the wall of the future Open Workspace using the “yellow labels”.

Iteration size was two weeks and it included the planning, working, continuous integration steps. Verification of the previous iteration results was made during the next iteration.

Analysis of the Used agile practices

Strengths:

- + Short iterations gave the project members a peace to develop the software.
- + Short iterations and user story selection in planning game gave possible that even if the project organization and budget changed radically middle of the project, the project members could continue their work.
- + Milestones were defined

Improvements:

- Even if the work was based on short iterations, it was not incremental. Previous iteration results were verified during the next iteration when the increment development was not implemented in the same cycle as the iterations.

Analysis compared to CMMI

Strengths:

- + Project lifecycle were selected and defined based on agile life cycle model.

New potential agile practices

SP.2.4 and 2.5, Project planning and monitoring, Plan for project resources and their knowledge and skills

Resource definition / project agile SW development preparation

Current state

Project had a successful team selection. Selected four project developers had a high motivation and the needed technical excellence. The team had a much freedom for example to define the working tasks, architecture and find the best solutions and techniques to implement the product. Project did not had much turnover.

At the beginning of the project there was also the kick-off where the project purposes and used agile practices were discussed together with the project management, “virtual” customer and developers.

Project also had the VTT agile SW development trainings.

Analysis of the Used agile practices

Strengths:

- + Project had a successful team selection.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

- + Selected four project developers had a high motivation and the needed technical excellence.
- + The team had a much freedom for example to define the working tasks, architecture and find the best solutions and techniques to implement the product.
- + Project did not had much turnover
- + At the beginning of the project there was also the kick-off where the project purposes and used agile practices were discussed together with the project management, “virtual” customer and developers.
- + Project also had the VTT agile SW development trainings.
- + Process requirements for the developed project were defined in the information radiators. before 0-iteration. Project manager had a list of the used agile practices which were selected based on the literature and organization SW development comparison and analysis.
- + Project had the needed technical excellence and open workspace and the project members get the needed knowledge and trainings for the project.

Analysis compared to CMMI

-

New potential agile practices

-

SP. 2.6, SP. 1.2 and 1.5 Project planning and monitoring, Plan and monitor stakeholder involvement

Communication planning with customer (on-site customer)

Current state

Project had a named “virtual” customer who had the responsibility to verify the produced reports and to be available to the developers when needed.

Project also had a named steering group/ milestone meetings where the project status were evaluated together with the project management

Virtual customer and project manager participated in project planning Game

Project developers defined also some internal roles in the project, but these roles did not worked in practice.

Project did not use any defect reporting systems. Number of change of virtual customer was minimal and the defects from “virtual” customer were discussed during the face-to-face conversations and reported to tasks to the information radiators in the open workspace walls.

Project had a defined budget and it had decided finalizing day which were agreed in steering group. Developed user stories were, however, selected and implemented for each iteration together with the “virtual” customer and project team. Purpose was that the most important reports would be developed and verified when project ends.

Analysis of the Used agile practices

Strengths:

- + Named “virtual” customer who had the responsibility to verify the produced reports and to be available to the developers when needed.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

- + Named steering group/ milestone meetings where the project status were evaluated
- + Virtual customer and project manager participated in project planning Game
- + Project budget was discussed and agreed in steering group
- + Agile methods supported the organizational changes. Even if, the steering group and project budget was changed the project developers had a peace to continue the development. Project manager could focus the project work in daily meetings and planning game based on his best knowledge of the current situation. Project developers were informed of the change when all the needed agreements were accepted.
- + Project status were reviewed bi-weekly with the end user

Improvements:

- At the beginning of the project the project kept the regular daily meetings with customer where the project status was analysed together and new tasks were created to the information radiators if needed. In some part of the project the daily meetings were not, however, seen anymore necessary and the project “virtual” customer could not participate in the project. Therefore, the project developers had difficulties to focus their work. When the release was ready for the verification, the “virtual” customer as well as the project manager had again new interests for the project work.
- Project original steering group was changed during the project.

Analysis compared to CMMI

Strengths:

- + Stakeholders were defined and roles and responsibilities were clear for the project members. Steering groups made sure the external stakeholders involvement.

Improvements:

- Project developers however criticized that the scope of the project was not clear in the middle of the project.
- Milestone planning and scheduling in agile development

New potential agile practices

- Project should have regular meetings between the Planning Games (not necessary daily meetings) and on-site customer also in the middle part of the project.

SP. 1.3 Configuration management, Create Release Baseline

Incremental release planning (for whole product)/ Small releases/ sprints

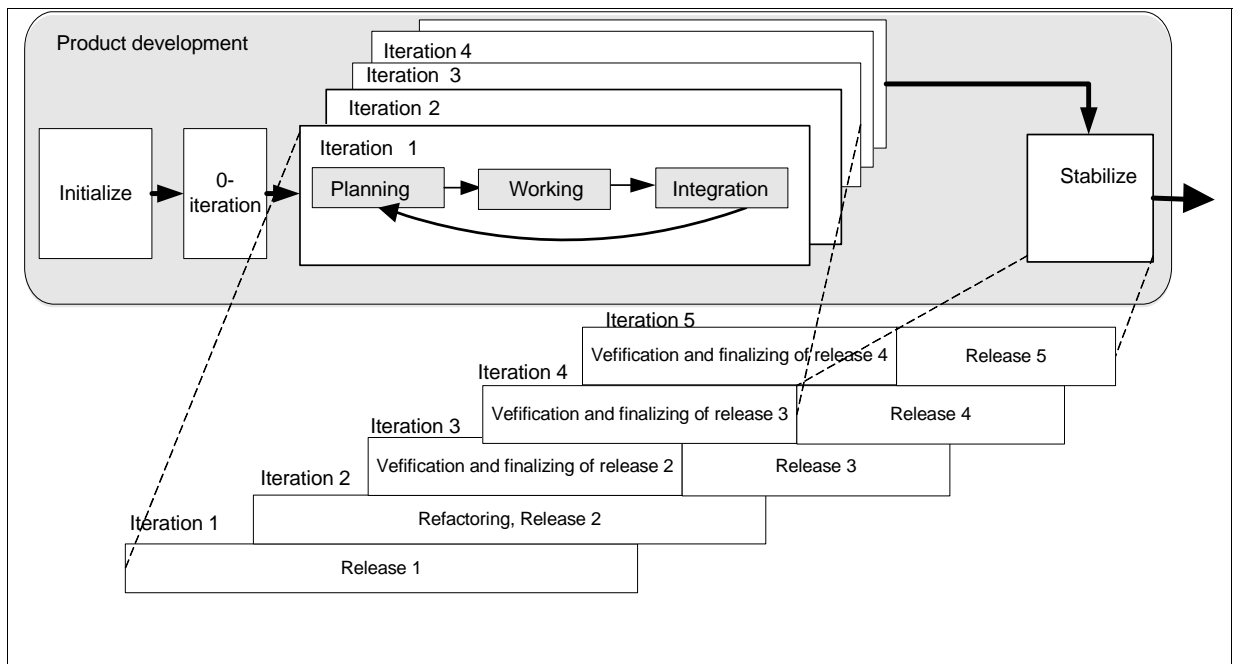
Current state

Release planning in the short iterative SW development is one challenge for the agile SW projects. In the evaluated project the first pilot release was created during the 0 and 1 iterations. When the project developers experience of the SW domain was increased the need for the SW, architecture and database refactoring was noticed. This was implemented during the second iteration. Third iteration included the verification activities for the release 2. This was done by the “virtual” customer and project manager while the developers implemented the next release. (Figure).

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05



Analysis of the Used agile practices

Strengths:

Continuous release development gave the possibility to have a release all the time during the project.

Improvements

- Identifying the different development iterations
- Release development took one and half iterations, sometimes more or less. This kind of approach was complex to plan and therefore it was not very clear when the release should be ready and accepted.
- Release planning in agile development

Analysis compared to CMMI

Improvements:

Release contents

New potential agile practices

Release planning in agile development

SP.1.6 and 1.7 Project monitoring, Conduct Progress and milestone reviews

Project Milestones in Agile development

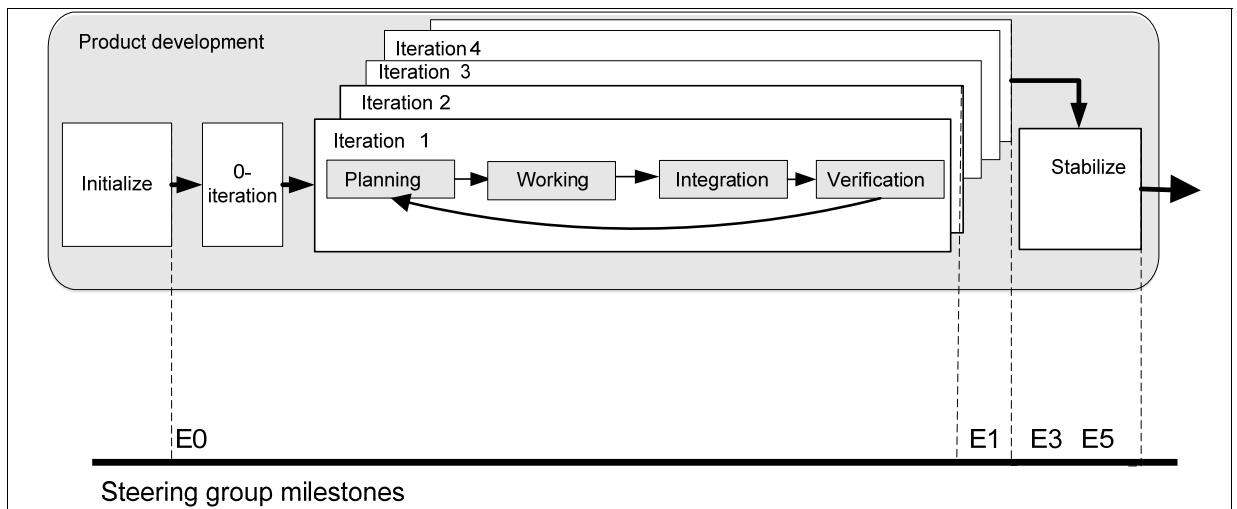
Current state

Agile SW development project were implemented using the short iterations, planning Game, working, integration and verification steps (Figure).

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05



E1 requirements demands that the project scope is well known in project. In agile project this was actually known after the two development iterations. In other parts the agile SW development steps are easily inputted in the Nokia milestone practices (Figure).

Milestones can be difficult to join in the short iteration based development. It can be difficult to predict when the project has been enough ready for the E1 and E2-4 milestones.

Purpose of the milestones E1 was to get the clear knowledge of the project scope. It was planned to be at the end of the first iteration. However, in practice the project scope was clear after the second iteration.

Analysis of the Used agile practices

Improvements:

- Milestone planning and scheduling in agile SW development

Analysis compared to CMMI

Strengths:

- + Project status was monitored in steering group using one Power Point presentation which included the information of the project scope, organization, milestones, risks, pudget, business case, tasks for each milestone and requirements.

New potential agile practices

- Milestone planning in agile environment

SP.1.1 and 2.2 Configuration management, Identify and control the configuration items

Documentation needs

Current state

All of the planned documentation were not implemented during the project. Developed documentation was simple and mainly based on Power Point presentation and information radiators in the wall of the open workspace.

Analysis of the Used agile practices

Strengths:

- Documentation did not take much time from the project and the project developers had more time to concentrate in the technical SW development.
- Information Radiators were efficient way to inform about project status to the project

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

management and “virtual” customer.

New potential agile practices

-

6.1.2 0-iteration

SP.1.3 Technical selection, Select product component solutions

Agile Trials in 0-iteration

Current state

Before the first release development, the project had a 0-iteration where the project developers set up the needed tool environments and made trials of the used development technologies.

Analysis of the Used agile practices

Strengths:

- + 0-iteration worked well and the project members get quickly the most suitable technologies for the future development.
- + The project developers had freedom to make their own decision of the planning and development languages.

Analysis compared to CMMI

Strengths:

- + Alternative development solutions were evaluated and the best solutions were selected for the further development.

New potential agile practices

-

Improvements compared to CMMI

-

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

6.1.3 Iteration planning

Iteration starts with iteration planning and continues with working, integration and releasing. In the iteration planning the iteration goals, requirements and tasks are defined for the subsequent iteration. Working means the software design, implementation and continuous integration which are done by the self organizing team using the agile techniques (e.g. simple design, pair programming, Test first development). In releasing phase the software (iteration results) is built and tested.

SP. 1.1 Project planning, Estimate the scope for project

Iteration goal definition (sprint goals)
<p>Current state Iteration scope was defined based on result reports and the prioritized user stories. In practice, the project needed two iterations to clarify the overall project scope.</p> <p>Developers, project manager and “virtual” customer participated in the iteration scope definition and user story prioritization</p>
<p>Analysis of the Used agile practices <i>Strengths:</i></p> <ul style="list-style-type: none"> + Scope definition in Planning Game based on result reports and user story prioritisation + Planning Game worked well in the project. <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Because of the general level requirements project needed two iterations to scope the actual activities <p>Analysis compared to CMMI More analysis in “Task and schedule definition in Planning Game”</p> <p>New potential agile practices</p> <ul style="list-style-type: none"> - <p>Improvements compared to CMMI</p> <ul style="list-style-type: none"> -

SP.1.1 and 1.2 Requirements management, Obtain understanding and commitment of requirements

User Story definition and analysis in Planning Game
<p>Current state Developers saw that the user stories were at too general level at the beginning. Therefore, the project needed two iterations for the actual scope definition.</p> <p>Requirements were well analysed and prioritised for the subsequent iteration. Project manager, “virtual” customer and developers participated in the requirements analysis and selection.</p> <p>Detailed requirements were not saved in any database or program during or after the</p>

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

planning Game
<p>Analysis of the Used agile practices</p> <p><i>Strengts:</i></p> <ul style="list-style-type: none"> - Well analysed and prioritized requirements <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Too general level User Story definition - Iteration level requirements reporting <p>Analysis compared to CMMI</p> <p><i>Improvements:</i></p> <p>Criteria for requirements acceptance was not defined</p> <p>New potential agile practices</p> <p>Effective use of iteration backlog</p>

SP. 1.2 Configuration management, Establish Configuration Management System

Product and Sprint backlogs
<p>Current state</p> <p>Product backlog included the old tasks from the previous iteration. It was not, however, systematically utilized in project</p>
<p>Analysis of the Used agile practices</p> <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Product backlog list could be more systematically used - Contents of the backlog lists (could be requirements, features, functions, technologies, enhancements, bug fixes, Issues) <p>Analysis compared to CMMI</p> <p>Not available in CMMI</p> <p>New potential agile practices</p> <ul style="list-style-type: none"> - Use of the product backlog list - Use of the iteration backlog list - Contents of the backlog lists (could be requirements, features, functions, technologies, enhancements, bug fixes, Issues)

SP.1.2 Project planning and monitoring, Establish estimates for work products and task attributes

Task and schedule definition in Planning Game and monitoring during the project
<p>Current state</p> <p>Developers defined the project tasks based on prioritised requirements. Tasks were not, however, systematically reviewed by the project management and “virtual” customer</p> <p>Tasks were defined in the information radiator. At the beginning of the project they were, however, in too general level for the project monitoring. Situation were noticed in the post-iteration workshops and improved during the project.</p> <p>Acceptance tests for the system were not systematically planned. Automatic system for the acceptance test were planned and developed, but it was not used during the project.</p>
<p>Analysis of the Used agile practices</p> <p><i>Strengths:</i></p>

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

- + Developers defined the tasks based on prioritised requirements in the information radiator
- + Task definition was improved during the project

Improvements

- Tasks were not systematically reviewed
- Task validation criteria were missing
- Tasks were at too general level at the beginning
- Acceptance tests for the each release were not systematically planned

Analysis compared to CMMI

- Too high level working tasks
- Task reviewing with planned validation criteria

New potential agile practices

-

SP.1.4 Project planning and monitoring, Determine Estimates of Effort and Costs

Effort Estimation in Planning Game

Current state

Project had the general budget estimation which was regularly presented and discussed in steering group.

Effort estimations were tried to do for the project working tasks at the beginning. In the implementation phase it was not needed.

Analysis of the Used agile practices

Strengths:

Budget estimations were regularly presented in milestone meetings

Improvements:

Effort estimations for working tasks were not done after the second iteration.

Analysis compared to CMMI

-

New potential agile practices

Effort estimation for each working tasks in Information Radiator

SP.2.2, SP. 1.3, Project planning and monitoring, Identify and monitor Project Risks

Risk identification in Planning Game

Current state

Risks were identified and analysed for the whole project and presented in steering group.

Technical risks were not, however, identified and analysed in planning Game together with Customer, project manager and developers.

Analysis of the Used agile practices

Improvements:

Technical risks were not identified and analysed in planning Game

Analysis compared to CMMI

Strengths:

Risks were analysed in milestone meetings

New potential agile practices

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Technical project risk identification and analysis in planning Game

Improvements compared to CMMI

Risk prioritisation

Project planning and monitoring, SP.1.2 Establish estimates of work product and Task Attributes, SP. 1.4 Monitor and analyse the metric data

Metrics definition for Agile context

Current state

Metrics were defined and metric data were collected but all of the collected metrics were not systematically analysed or utilized during the project.

Analysis of the Used agile practices

Strengths:

<i>Metrics were systematically collected and monitored using the information radiator technique during the project</i>
--

Analysis compared to CMMI

<i>Improvements:</i>

Metrics analysis and utilizing against the defined plan

New potential agile practices

-

Project planning and monitoring, SP.2.7 Establish the project Plan

Project plan results reporting

Current state

Project plan was reported in the information radiators and two different level Power Point presentations. Information Radiators were efficient way to describe the project status and monitor the situation.
--

-

Analysis of the Used agile practices

Improvements

<i>Strengths:</i>

+ Information Radiators were efficient way to describe the project status and monitor the situation.
--

+ Post-iteration workshops improved the quality of the tasks in the information radiators

<i>Improvements:</i>

- Tasks were at the beginning at too general level in information radiators

Analysis compared to CMMI

<i>Improvements</i>

- Effort estimations were missing at the end of the project

New potential agile practices

-

Previous iteration including the requirements, tasks, schedule, effort and risks are evaluated and reviewed as a part of the next iteration planning.

SP.1.4 Maintain Bidirectional Traceability of Requirements

Requirement status checking in Sprint review or Planning Game
--

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

<p>Current state</p> <p>Requirements from previous iteration were transferred to the subsequent iteration. Requirements status was not, however, analysed in the Planning Game</p> <p>Edellisestä iteraatiosta toteutumatta jääneet tulokset siirtyivät seuraavaan iteraatioon, mutta ei analysoitu erikseen. Uudet vaatimukset kuitenkin käytiin läpi ja priorisoitiin seuraavaa iteraatiota varten</p> <p>Vaatimusten toteutuminen läpikäytiin verifiointin kautta, mutta ei muuten tarkistettu</p> <ul style="list-style-type: none"> - How often are you keeping the sprint review meetings or checking the iteration status in Planning Game? - Are you checking the requirements/ feature status/ traceability in these meetings? - Are the change requests appeared during the last iteration discussed, prioritized and selected in the Planning Game? - Use of the Product Backlog for requirement status checking - Participants - How are you making sure that the components fill the customer requirements (acceptance testing, communication etc.)
<p>Analysis of the Used agile practices</p> <p><i>Strengths:</i></p> <ul style="list-style-type: none"> + Requirements from previous iteration were transferred to the subsequent iteration. + Requirement status analysing using the iteration backlog list in Planning Game <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Requirements status was not, however, analysed in the Planning Game <p>Analysis compared to CMMI</p> <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Requirements traceability evaluation <p>New potential agile practices</p> <p>-</p>

6.1.4 Working

SP.1.2 Technical solutions, Evolve operational concepts and scenarios

<p>Methaphora</p>
<p>Current state</p> <p>Methaphora was useful when the project communicated of the project development and results to the external customers and partners and at the beginning of the project when the project developers tried to clarify the project purposes. However, when the project developers created the first version of the product architecture, meaning of the methaphora was reduced.</p>
<p>Analysis of the Used agile practices</p> <p><i>Strengths:</i></p> <p>Methaphora was useful when the project communicated of the project development and results to the external customers and partners</p> <p>Analysis compared to CMMI</p> <p>Not available in CMMI</p>

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

New potential agile practices

-

Architecture line**Current state**

Developers created the first version of architecture after the first planning game. After that there were many changes for the architecture during the project. Project developers were responsible to make the needed updates for the developed architecture. “Virtual” customer or the project management did not give any requirements for the architecture solutions. Even if, the technical problems were discussed together with the customer, management and developers during the project. Architecture solution was developed simple using Power Point.

Analysis of the Used agile practices*Strengths:*

- Project developers had much freedom to make their own architecture solutions. However, they get the needed support of the customer and developers.
- Developed architecture solution was simple and fast to develop and change.

Analysis compared to CMMI*Improvements*

- Alternative architecture solutions were not implemented
- Project did not have any acceptance criteria or inspection for the developed architecture solution.

New potential agile practices

-

Select product component solutions**Teamwork for specifications and documentation****Current state**

Developers made the specification and architecture using the Power Point, face-to-face communication and team work.

Analysis of the Used agile practices*Strengths:*

- + Teamwork for the specification and architecture definition worked well in project
- + Developers had much freedom for the best architecture and specification solution definition
- + Developers discussed together of the “Virtual” customer and project manager of the technical problems

New potential agile practices

- Rapid7 based specification workshop method

Analysis compared to CMMI*Improvements*

Defined acceptance criteria for the specification and architecture

Develop the design, SP.2.1 Design the product or product component**Simple design**

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Current state

Design simplify was not refactored during the project. If the project had some technical problem it, however, tried to solve using the simplify principle even if it had take more effort than the other solution

Analysis of the Used agile practices

Strengths:

- + Technical solutions were based on the simplify principles

Improvements

- Design refactoring using the simplify principle

Analysis compared to CMMI

Improvements:

- Criteria for the design
- Ensuring that the design adheres the defined standards and criteria
- Document the design

New potential agile practices

- Planned Design refactoring using the simple principles: SW runs all tests, has no duplicated logic, have fewest possible methods and classes

Implement the product design, SP.3.1 Implement the design

Pair Programming

Current state

Complex work was done using the pair programming, smaller work and refactoring was done using the single programming. At the beginning, the pair programming was not worked, but it was improved during the project using the post-iteration workshops.

Analysis of the Used agile practices

Strengths:

- + Pair programming worked well at the end of the project and it was improved during the project using post-iteration workshops

Improvements:

- Pair programming was not worked at the beginning

Analysis compared to CMMI

-

New potential agile practices

-

SP.1.2 Objectively evaluate work products

Refactoring

Current state

Second iteration included the database, architecture and implementation refactoring. In the later phases the refactoring was focused mostly in the code. It improved the product quality.

Refactoring was not, however, systematically planned or implemented for each iteration.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Analysis of the Used agile practices*Strengths:*

- + Refactoring improved the product quality
- + Refactoring was implemented for the database, architecture and code

Improvements:

- Refactoring could be systematically planned and implemented for each iteration results

Analysis compared to CMMI

-

New potential agile practices

Refactoring for design

SP.3.1 Perform verification**Test First Development****Current state**

Unit testing was automated and worked well in project. Use cases were written before or after the testing depending of the situation. Developers implemented the unit testing.

Analysis of the Used agile practices*Strengths:*

- Unit testing was automated and worked well in project improving the product quality
- Use cases were developed before or after the testing depending on the situation

Analysis compared to CMMI

-

New potential agile practices

-

Monitoring and communication**Daily meetings/ Daily wrap-ups/ Scrum stand-up meetings****Current state**

Daily meetings were kept systematically at the beginning of the project. But not at the end of the project. Daily meetings were felt as a laborious practice when the project was continued for a while. They would need much effort from the development team, customer and management.

Analysis of the Used agile practices*Improvements:*

- Daily meetings were felt as a laborious practice when the project was continued for a while
- Some kind of focus meetings (not necessary daily) would be needed when the daily meetings end.

Analysis compared to CMMI

Not available in CMMI

New potential agile practices

-

Requirement change management**Current state**

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Requirements prioritisation in planning Game was answering well to the change management problems. Project team was responsible of the change task creation. Verification change requests were discussed with the “virtual” customer and developers and the developers created the tasks in the information radiator based on the discussion. This was not, however, systematic.

Analysis of the Used agile practices

Strengths:

- + Requirements prioritisation in planning Game was answering well to the change management problems
- + Verification change requests were discussed with the “virtual” customer and developers and the developers created the tasks in the information radiator based on the discussion.

Improvements:

- Task creation based on the change requests was not systematic

Analysis compared to CMMI

Improvements

- Requirements in database
- Requirements status monitoring
- Requirement decision database

New potential agile practices

- Requirements including their status in iteration backlog

Information radiators

Current state

Information radiators were used as training, communication, project planning and monitoring. It was good way to manage the project. However, the task size was too big at the beginning and the effort estimations were partially missing in the tasks.

Analysis of the Used agile practices

Strengths:

- + Information radiators were efficiency used as training, communication, project planning and monitoring.

Improvements:

- Task creation based on the change requests was not systematic
- Validation were missing

Analysis compared to CMMI

Not available in CMMI

New potential agile practices

-

6.1.5 Integration

Continuous Integration

Current state

Continuous integration was done nightly and it generates some defects that were daily corrected. Practice worked well in project.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Analysis of the Used agile practices*Strengths:*

Nightly integration of the developed SW worked well in project

Analysis compared to CMMI

-

New potential agile practices

-

SP. 1.1 Objectively Evaluate Processes**Post-Iteration Workshops****Current state**

Post-iteration workshops were regularly kept after the each iteration. Workshops were seen as very useful for the project and during the workshops many important improvements like pair programming, working tasks in information radiators, documentation and code comments were discussed in post-iteration workshops and improved during the project.

Analysis of the Used agile practices*Strengths:*

- + Post-iteration workshops was useful and working practice in project
- + Project members defined the project strengths, improvements and the main improvement ideas for the project
- + Results of the post-iteration workshops were documented in the wall papers.
- + Many important practices like pair programming, working tasks in information radiators, documentation and code comments were improved based on post-iteration workshop analysis

Analysis compared to CMMI

Not available in CMMI

New potential agile practices

-

6.1.6 Finalizing**Verification****Current state**

Results verification was done concurrent with the release development. “Virtual” customer was not available all the time and therefore, also, the project manager implemented the verification activities. Defects and change requests were reported face-to-face to the developers, who generated the working tasks and implemented the activity.

Acceptance tests were not, however, systematically planned in project.

Results were also reported by the end users who participated in the project steering groups. Only one change requests was generated during this end user evaluation.

Some initial user requirements were not implemented during the project, but they were not necessary for the SW functioning. Agile methods supported the requirements prioritisation so that the important functions could be implemented during the project

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

<p>Analysis of the Used agile practices</p> <p><i>Strengths:</i></p> <ul style="list-style-type: none"> + Concurrent verification with implementation worked well in project + Agile methods supported the requirements prioritisation so that all important functions were implemented during the project and there was not much change requests in the end of user verification. <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Validation was not systematically planned - Planned automatic test environment did not worked in project <p>Analysis compared to CMMI</p> <p><i>Improvements:</i></p> <ul style="list-style-type: none"> - Defects reporting was not systematic - Acceptance tests were not systematically planned <p>New potential agile practices</p> <ul style="list-style-type: none"> -
--

6.2 Agile Practice Utility evaluation

The last step in XP Evaluation framework is Outcome measures. In this phase, the current (agile) and previous (plan-driven) projects measurement results are compared. (Williams 2004). The purpose of the Outcome Measure part is to answer following questions: 1.) was the team successful, 2.) was the team more or less productive, 3.) was the quality of software better or worse, 4.) was the team morale high or low and 5.) were customer satisfied with the product. The questions of the team successful and productivity can be answered using the utility assumptions (Williams 2004). In Table 3 the utility of the used agile practices in Nokia project is analysed and compared with the Williams assumptions

Table 3. Utility of the agile practices in project

XP Practices	Manager viewpoint	Developers Viewpoint	Customer Viewpoint	Williams analysis (Williams 2004).
Stand-up meetings	Practice was not seen necessary because of the open workspace and schedule problems	Stand up meetings were not seen very efficient because of the open workspace	Customer was not participated in daily meetings.	Communication, sharing, quick help, essential for project
Continuous integration	-	-	Release was always available for validation	Improvements in build and configuration management activities
Short Releases	-	Improved the verification, always available releases	Information of the system increased step by step to the customer, helped the validation	Frequent Customer Feedback, focusing
Test Driven Development	High Utility, Increased product reliability	Useful practice which improved the release quality	Too heavy technique at the beginning. But when the number of code was increased was seen very important.	Code Quality improvements, Risk: Technical challenges
Pair Programming	High Utility, Knowledge increasing between the developers, new ideas for development	Knowledge of the overall system is increased, positive experiences. May cause resist of the	Worked well in project, harmonize the code and help the product maintenance,	Increased Quality of Code, Increased team communication, Risk: Resistance among the

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

		developers in organizations.	problematic beginning because of the technical problem solving	programmers
On-site customer	-	-	Helped the requirements definition, analysed in more detailed level if needed	Customer gets what he wants not what he asked, Risk: High Costs
Simple Design	Project team had purpose to answer the technical development problems as simple as possible	Developers have not done anything additional work	Not participated in development work, only discussion if the team had some problems	
Coding standard	-	-	-	Quick understanding how the code works (useful in pair programming)
Sustainable Pace	Not much overwork in project.	Not overwork, expect some times in last iterations during the maintenance phase		Happy employees and higher quality code in the long run, Risk: Can be difficult if the programmers are aware of the schedule pressures, demands the culture change
Refactoring	Not participated in development work, only discussion if the team had some problems	Improved the code quality	Not participated in development work, only discussion if the team had some problems	Support the implementation of simple design, improve the code maintainability Risk: Increased maintenance costs
Planning Game	Efficient mechanism for requirements definition, analysis and prioritisation	Efficient mechanism for requirements definition, analysis and prioritisation	Improved the visibility to the project, knowledge of the next iteration activities	Beneficial project management strategy, Risk: Requirements that has not been chosen for current iteration should be ignored when design decisions are made
Collective Code Ownership	-	Increased knowledge of the system	Not participated in development work, only discussion if the team had some problems	Knowledge sharing, Risk: Nobody is responsible of the code
Methaphora	Improved external communication in project	Useful at the beginning, Improved external communication in project	Improved external communication in project	Most teams can be successful without methaphor

Project team, however, used also the other agile and plan-driven practices. For example the technical trials before first iteration, information radiators and post-iteration workshops were seen as very efficient practices in the agile software development project. (Table 4).

Table 4. Utility of the Other Practices

	Manager Viewpoint	Developers Viewpoint	Customer Viewpoint
Technical trials before first iteration	-	Improved motivation and teamwork between the developers	Worked well in project, increased the team knowledge of the technical difficulties, if bigger team it would be even more efficient
Information Radiators	High Utility, Increased communication between developers and customer	Important technique for information sharing in project	Good Visibility to project
Task Definition	Problematic for the development team at the beginning, too large task descriptions, improved during the project	Too general level at the beginning, improved during the project.	Problematic for the development team at the beginning, too large task descriptions, improved during the project
Release Validation	Could be more systematic	Could be more systematic	Not systematic, could be planned better
Post-iteration workshops	Efficient way to improve used agile practices	Discussion of the used Practices, negative and positive findings, Improved	Improved developers work when knowledge of the problems in the used agile practices increased

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

		the use of agile practices in project	
--	--	---------------------------------------	--

6.3 Empirical Evaluation

The purpose of this chapter is to describe main results of the Nokia Agile Assessment results empirical evaluation. Results presented in this section are based from both interviews and workshop.

In Initiating phase an initial product backlog was created together with end user. Selected development team had a kick-off meeting and workshop where the agile practices, used technology solution proposals and implemented product were discussed and analysed together with the project manager and project team. Used agile practices were described on using information radiators on the wall labels of open workspace. According to the developers the kick-off as well as the workshop in the beginning improved the team commitment to the agile development approach. Resource, knowledge and skills and project life cycle selection worked well in agile project. This was because of the successful team selection, efficient communication and requirements discussion with end users in steering group meetings.

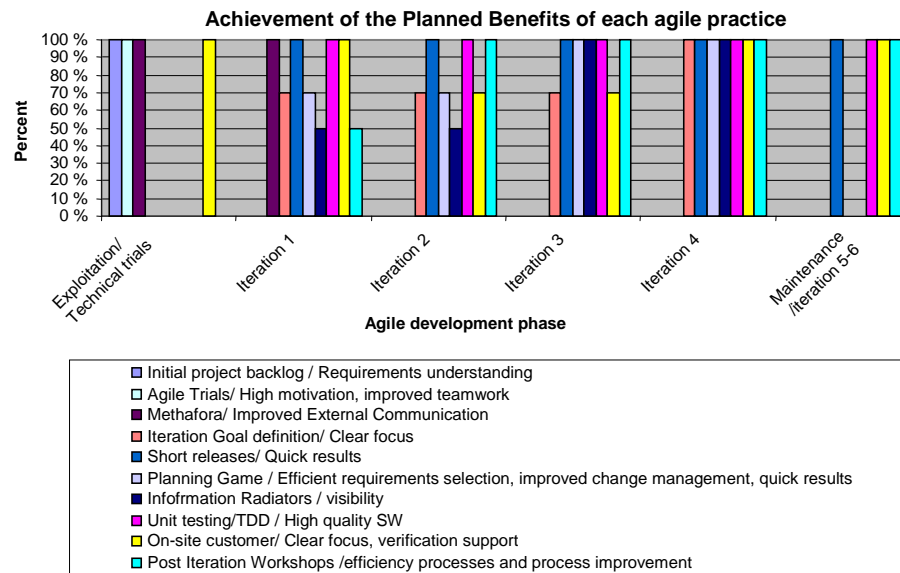
The first technology step in project was to make technology trials using pair programming and simple design (Figure 4). According to the developers and manager the first technology trials was a great success. The first working version of the product was developed during the first two days. Most important plan-driven practices used during the exploitation were milestone approval, use of configuration management system and process planning.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Fig. 5 Achievement of the planned benefits of the agile practice use in different iterations



During the first iteration the project used test driven development, pair programming, agile modelling and reflection workshops. The purpose was to use information radiators in the project task status management. According to the project developers and managers opinions, one of the main problems in the first iteration was too high level task size in the information radiators. This was, however, discussed and improved in the first post-iteration workshop. (Figure 5).

Release planning in agile development project caused a new problem field for the project management. Agile development should be incremental. Therefore release development rhythm should be systematically planned concurrent with the short iterations. This kind of planning was not systematically in the evaluated project.

When the project developers experience of the SW domain was increased the need for the software architecture and database refactoring was noticed. The architecture refactoring was mainly implemented during the second iteration. When discussing of the second iteration the project developers' criticized most too general level user stories in the iteration one and two planning games (Figure 5). The stories were, however, improved in the iteration three when the project developers' knowledge of the developed product was increased. (Figure 5). The third iteration included the verification activities for the release two. The verification was done by the "virtual" customer and project manager while the developers implemented the next release. Information radiators and planning games worked best in the iterations three and four. This was because of the improved working tasks and increased knowledge of the project demands. Efficient information radiator and planning game techniques increased also the project visibility and requirements definition in these iterations.

According to the interview results simple documentation worked well in the evaluated project. The aim in the use of simple documentation was to improve the release quality and maintainability (Figure 4). Because of the light documentation project developers had possibility to concentrate in the technical problem solution and development instead

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

of the documentation. According to the developers, this maintained a high motivation among the developers during the product development. Post-iteration workshops were most efficient in project iterations 2-4 and offered the information of needed improvements in the used agile practices (Figure 5).

Iterations five and six included mainly the maintenance activities including the product acceptance testing and milestone approval. Because, the new requirements were not selected, the Planning Games were not seen necessity anymore (Figure 5). On-site customer and short releases were seen as the most important practices in the maintenance phase.

Compared to the plan driven development project goals the evaluated agile project should have for example defined acceptance criteria that are needed for the successful agile software development, regular technical risks analysing (e.g. in planning Games) and requirement status and traceability checking (e.g. in Planning Games). Potential improvements and practices for the agile based development were also analysed in the collaborative workshop. According to analysis the main improvements/ new potential practices for the future development projects would be for instance the use of iteration backlogs, Improved use of the information radiator task status definition and more detailed level user requirements definition in the iterations one and two planning games.

6.4 Analysis

Utility of the used agile practices were in agile assessment analysed from project manager, developers and customer viewpoint. Results from Nokia Agile Project evaluation supports the Williams assumptions of the XP practices risks and utility. For example the test driven development improved the release quality and the pair programming increased the knowledge between developers in both cases. Also, Planning Game was the beneficial practice for project management and requirements analysis, prioritisation and selection. Instead Williams has seen the stand-up meetings as an essential practice for the agile software development project. However, anyone of the developers, manager or customer in Nokia agile development project did not keep the stand up meetings necessary for development project. This was because of the project team working in the open workspace and increased costs. Williams has also argued that the use of methaphora in agile development project will bring the low utility. In the evaluated project, however, the methapora was actually seen useful practice for the external communication.

According to the Customers opinion:

“Agile practices works well if the project has a clear starting point and well defined requirements in Planning Games. Management could be difficult if the content of the project is complex at the beginning. “

According to the empirical findings, the use of agile practices increase the motivation , technical excellence of development team. It also improves the knowledge and communication between the management, customer and developers. Simple and light documentation gives to the developers' possibility to concentrate to technical development work instead of the documentation.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

6.5 Agile Assessment suitability in current project context

Agile Assessment is useful method when the organization wants to define the most utility agile practices in their current project and organization context.

The evaluation of the agile based development was not easy. It demanded much knowledge of the agile practices, their use possibilities and suggested benefits in the evaluated organization. The assessments itself was light including only one-two days interviewing and one three hours workshop together with the development team. Definition of the most efficiency agile practices will help the future Nokia projects to balance between agile and plan-driven methods and, therefore, to achieve best motivation of developers and best results answering to the challenge of the rapid change demands.

Even though the empirical agile project evaluation provided excellent results, new theory background creation for assessment also decreases the results reliability compared to the plan-driven (e.g. CMMI or SPICE) assessments. The first problem is the content validity. It is difficult to define what means the efficient agile practices or agile project in software development. The question is: would the use of agile practices really increase the development productivity if it increases the developers' motivation? And what risks cause of the simple documentation in the later in the dynamic software development environment? The agile project efficiency definition in this assessment case study was based on the current literature of the agile principles and reports of the most efficient agile practices in different software development situations.

Second reliability problem is that the maturity of the hybrid agile and plan driven project is impossible to define because the idealized list of the agile practices has not been proven to work. The assessment itself is, still, is possible, if the assessment the focus is rather to improve the working than for evaluating the maturity of the agile practices.

The agile assessment results have, also, many limitations. For example, project developers do not necessary remember the each iteration agile practices after the several months of the evaluated situation. Thus, in future, the Agile Assessment result validity could be improved utilizing the metric and post-iteration workshop data as a part of the agile assessment results evaluation.

6.6 Future work

There is no experience available yet about validating the developed project evaluation and analysis approach in the other organizations. The Agile Project Efficiency Assessment approach should, also, be researched in the organizational level. In the future assessments the different types of software development projects should be compared using more deep literature analysis of agile methods view of agile practice suitability and efficiency in different product development contexts as well as the more validated metric data.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

7 Summary

The purpose of this document was to describe the experiences and empirical results of the Nokia Agile Assessment. According to the findings the agile practices improve motivation and technical excellence of the development team. Simple and light documentation gives the developers the possibility to concentrate on technical development work. Developers' opinion was that pair programming, test first development and refactoring improved the release quality in the evaluated project. This was mainly because the number of defects found in the evaluated project end product was minimal. However, the daily meetings were seen as a laborious agile practice which took too much effort from the project stakeholders.

The empirical data from the case study as well as other agile assessments shows that release planning in the agile development environment causes new challenges for project management. Incremental development and releasing rhythm can be different than the one-two week iterations in agile development. This worked in the evaluated project but demands much planning. Milestone meetings (in agile development) can be included with the regular iteration planning meetings, but the planned milestone meetings are, however, needed at least during the project exploitation and maintenance phases. According to the agile assessment workshop results, the use of information radiators was very efficient for agile project planning and monitoring. It increased the project visibility in all development phases. It however demands enough detailed level task definitions and task acceptance together with project management. The planning game supported well the requirement selection and analysis. Therefore it held also the internal stability during the project.

The efficiency of the agile practice can be successfully analysed from the developers, management and customer viewpoint and the achieved results support directly the project learning. Even though the agile assessment provided excellent results, there were however many challenges in the agile projects evaluation and results validation. The main problem was the content validity of the results, mainly because the idealized list of agile practices has not been proven to work. The assessment itself is, still, possible, if the assessment's focus is rather to improve the working than for evaluating the maturity of the agile practices. Also, the evaluation of agile based development is not especially easy. It demands that the agile assessor has much knowledge of the assessments, agile practices.

There is no experience available yet about validating the developed project evaluation and analysis approach in other organizations. The Agile Project Efficiency Assessment approach should, also, be researched at the organizational level. In future assessments, the different types of software development projects should be compared using more deep literature analysis of agile methods, view of agile practice suitability and efficiency in different product development contexts as well as the more validated metric data.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

References

- Abrahamsson, P. (2002). "Commitment nets in software process improvement." Annals of Software Engineering, in press.
- Abrahamsson, P., A. Hanhineva, et al. (2004). Mobile-D: An Agile Approach for Mobile Application Development. 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'04), Vancouver, British Columbia, Canada.
- Beck, K., M. Beedle, et al. (2001). Manifesto for Agile Software Development. **2002**.
- Boehm, B. and D. Turner (2003). Observations on Balancing Discipline and Agility. Agile Development.
- Boehm, B. and R. Turner (2003). Balancing Agility and Discipline. Balancing Agility and Discipline -A Guide for the Perplexed, Addison Wesley.
- Dingsoyr, T. and N. B. Moe (2004). The Process Workshop: A Tool to define Electronic Process Guides in small software companies. Software Engineering Conference, 2004, Australian, IEEE MCF.
- Glass, R. L. (2001). "Agile Versus Traditional: Make Love, Not War!" Cutter IT Journal **14**(12): 12-18.
- Glazer, H. (2001). "Dispelling the Process Myth: Having a Process Does Not Mean Sacrificing Agility or Creativity." CrossTalk The Journal of Defense Software Engineering: pp. 27-30.
- Graham, D. R. (1989). "Incremental Development: Review of Nonmonolithic life-cycle development models." Information Software Technology: 7-20.
- Greene, B. (2004). Agile Methods Applied to Embedded Firmware Development. Agile Development Conference, Salt-Lake city.
- Grenning, J. (2001). "Launching XP at a Process-Intensive Company." IEEE Software **18**: 3-9.
- Kerlinger, F. N. and H. B. Lee (2002). Foundations of Behavioral Research, Harcourt College Publishers.
- Kettunen, P. and M. Laanti (2004). "How to steer an embedded software project: tactics for selecting the software process model." Information and Software Technology.
- Kähkönen, T. and P. Abrahamsson (2004). Achieving CMMI Level 2 with Enhanced Extreme Programming Approach. Profes, Japan, Springer.
- Lindvall, M., D. Muthig, et al. (2004). "Agile Software Development in Large Organizations." Computing Practices: 38-46.
- Macias, F., M. Holcombe, et al. (2003). A Formal Experiment Comparing Extreme Programming with Traditional Software Construction. The Fourth Mexican International Conference on Computer Science, Mexican.
- Martinsson, J. (2003). Maturing XP Through the CMM. XP2003, Genova, Italy.
- Merton, R. K., M. Fiske, et al. (1956). The Focused Interview, The Free Press.
- Padberg, F. and M. M. Muller (2003). "Analyzing the cost and benefit of pair programming." Software Metrics Symposium: 166 - 177.
- Paetch, F., A. Eberlein, et al. (2003). Requirements Engineering and Agile Software Development. Twelfth IEEE international workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Computer Society.
- Paulk, M. (2002). "Agile Methodologies and Process Discipline." CrossTalk The Journal of Defense Software Engineering: pp. 15-18.
- Paulk, M. C. (2001). "Extreme Programming from a CMM Perspective." Software **18**(6): 19-26.
- Reifer, D. (2003). "XP and the CMM." IEEE Software: pp. 14-15.
- Salo, O., K. Kolehmainen, et al. (2004). Self-Adaptability of Agile Software Processes: A Case Study on Post-Iteration Workshops. 5th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2004), Garmisch-Partenkirchen, Germany, Springer.

Author: Minna Pikkarainen

Version: 1.0

Date: 13.6.05

Schwaber, K. (1995). Scrum Development Process. OOPSLA'95 Workshop on Business Object Design and Implementation, Springer-Verlag.

Schwaber, K. and M. Beedle (2002). Agile Software Development With Scrum. Upper Saddle River, NJ, Prentice-Hall.

Turner, R. and A. Jain (2002). Agile Meets CMMI: Culture Clash or Common Cause. 1st Agile Universe Conference, Chicago,.

Williams, L. A. (2004). "Extreme Programming Practices: What's on Top." Agile Project Management Advisory Service 5(12).

Vriens, C. (2003). Certifying for CMM Level 2 and ISO9001 with XP@Scrum. Agile Development Conference, IEEE Computer Society.