ISSN No. 0976-5697

Volume 3, No. 4, July- August 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Approach to Measure the Quality of Software using Lightweight Processes

N. Veeranjaneyulu*
Associate Professor, School of Computing
Vignan University, Vadlamudi
Guntur, India
veeru2006n@gmail.com

B. Jyostna Devi
Assistant Professor, School of Computing
Vignan University, Vadlamudi
Guntur, India
jyostna.bodapati@rediffmail.com

Dr. O. Nagaraju
Assistant Professor, Department of CSE
Acharya Nagarjuna University
Guntur, India
onrajunrt@gmail.com

Abstract: A software product is said to have high quality if it is having high customer satisfaction. The same is said to have lack of quality if it is not satisfying all the customer requirements. In conventional development methods series of life cycle activities are followed very strictly. Conventional methods are process oriented approaches. In conventional approaches, to deal with complex software products many quality measures and models like ISO, CMMI, SPICE, SIX SIGMA etc; are introduced. Agile development totally changed the way software is developed. Agile is a family of development methods also called as Lightweight Processes. Change can be introduced at any point of time by using agile methods. Pair programming and Test-Driven Development (TDD) are some of the key practices of agile development to achieve quality in software products. In Agile methods quality activities are done as part of the development not as a separate activity. The role of QA practices in agile projects needs to be redefined to achieve high quality. The method proposed is use of an extra layer of QA in agile development projects. The proposed approach is developer must not test his own written code, but in agile development most of the testing is done by developer that must be kept on minimum level in order to enhance quality.

Keywords: Agile Methodologies, Software Quality, Software Quality Assurance, Pair Programming, TDD;

I. INTRODUCTION

In recent years agile development methods became popular in software development. Software quality is difficult to define exactly. It is defined in many different ways. Quality is sometimes defined as the conformance to some standards like ISO. It is also defined as conformance to user requirements. Quality is a non-functional requirement. It is not required by the customer explicitly. Some obvious software quality factors are understandability, completeness, reliability and usability. Traditional development models are process oriented models.

a. ***The Waterfall Model:*** This model consists of sequence of activities that are to be followed. So it is also referred to as the “Linear sequential model”. Phases in water fall include: System feasibility, Requirements gathering, Design, Code, Implementation and Maintenance. Advantage of using waterfall model is: easy to adopt; the drawbacks are: Customer cannot express all the requirements at the beginning itself; Customer has to wait till the end of the product development.

b. ***The V-Model:*** The V-Model is an extension of waterfall model. In V model each phase of water fall is improved by a verification and validation activity. The V-Model has almost the sequence of phases as the waterfall model. However, instead of going down in a liner way, for each requirement and design phase, a testing phase will take place to ensure the system validation and verification. The criticism about the V-Model is that it divides system development with firm boundaries between them; this is the same problem with the waterfall. More importantly, it discourages people from carrying testing information across the different phases as some tests are done earlier than required and others are executed to be done later. [1].

c. ***The Spiral Model:*** It provided the potential for rapid development of incremental versions of the software. In this model, the software is developed in a series of incremental versions. This model starts with exploring and determining the risks of the current portion of the product. In the next step, evaluate the risks and try to find a strategy to resolve them. The most important advantage of the spiral model is risk assessment. The spiral model suggests a customer communication where the customer will address all the requirements. In Win-Win spiral model, a set of negotiation activities will be defined at the beginning of each pass around the spiral.

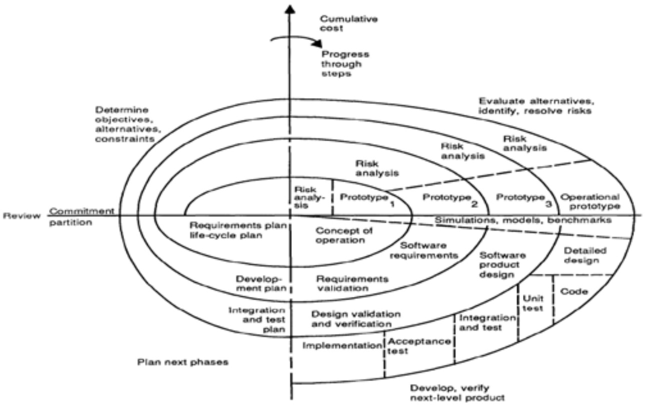

Figure 1: Spiral Model

d. ***The Rational Unified Process:*** The Rational Unified Process (RUP) is a result of merging the rational approach with the Objectory Process. In RUP there are two dimensions: a horizontal dimension to represent time and to show the dynamic aspect of the process which describes phases and iterations; and a vertical dimension to represent the workflows of the process, and to show the static aspect of the process which describes activities and roles.

Each phase in the RUP can be enacted in an iterative way, and the result will develop incrementally.

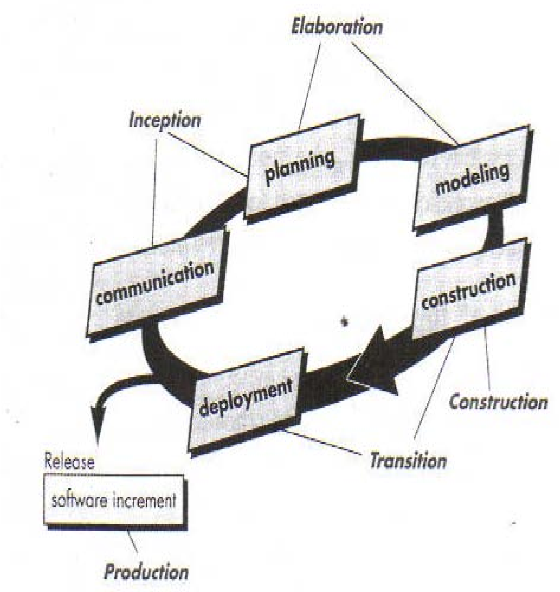

Figure 2: Phases in Rational Unified Process

e. ***Iterative, Incremental and Evolutionary:*** Iterative and incremental concepts formed the basics for these iterative approaches. In addition, the Extreme Programming practice of test-first development was applied as tests were planned and written and then the code was written to pass the tests. Furthermore, the project used continuous integration as each mini-iteration required integration of all code. [2]

Iterative and evolutionary development approaches are almost the same as both emphasise developing a piece of the system, reviewing it for improvements and feedback either by the customer or by the team. Thus after each iteration the customer will see an improved system. In this case, we prioritise quality over features. In incremental development, after each increment, the customer will see the system growing.

II. AGILE METHODS

The appearance of agile methods has been the most noticeable change to software process. The term “agile” (also known as light weight process) Agility, with regard to software development, can be expressed as the flexible, ready to change and quick-responsive nature of software development process. Agile is a family of methodologies. This family includes: Extreme Programming (XP), Scrum, Dynamics Systems Development Methods (DSDM), Adaptive Software Development (ASD), Crystal Methods, Feature-Driven Development (FDD), Pragmatic Programming etc;

In agile methodologies importance is given to Individuals and interactions over Processes and tool, working software over Comprehensive documentation, Customer collaboration over Contract negotiation, responding to change over following a plan.

Agile Vs traditional Based on the following views agile is different from traditional approaches.

a. ***Process:*** Traditional approaches aim to make software development predictable. Agilists believe that developing software is an unpredictable activity. Therefore, they try to focus on an adaptive process. An adaptive process is a process that can give control over unpredictability.

b. ***People and Communication:*** An agile process requires talented and skilled people and moulds the process to specific people and teams. In agile development, all team members are involved in a variety of different activities during the development process such as design, coding, and testing, where traditional methodologies try to give people separate tasks according to the lifecycle phases.

c. ***Measure of Success:*** The traditional criteria for a successful project are being the one on time and within budget. However from an agile point of view, the measurement is the business value

d. ***Requirements:*** Traditional approaches prefer formal, stable, and complete requirements in advance, where in agile methods the requirements are adjustable, informal stories.

e. ***Customer Involvement:*** Traditional approaches use documentation, contracts and review boards to communicate with the customer, where agile methods strongly emphasise having a dedicated customer who is involved in the development

f. ***Management:*** In traditional methods most people must be controlled, directed and threatened with punishment. Agile methods believe that self-direction and self-control are more important and can be directed to achieve the organisation’s objectives. In agile development, all team members will have a go in leading the team.

g. ***Quality:*** Traditional approaches to quality is measured by conformance to plan, CMM defines quality assurance as specification and process compliance. In agile methods quality.is customer satisfaction. Scott Ambler stated “*quality is an inherent aspect of true agile software development*” and this is as a result of practices such as iterative development, test driven development, and refactoring. [3]

III. EXISTING AGILE METHODS

A. *Extreme Programming (XP):*

XP is the most widely recognized agile method. XP practices were originally intended for small, collocated teams. XP is based on four values, communication, simplicity, feedback, and courage. XP practices include pair programming, continuous integration, refactoring, test-first programming, and user stories.

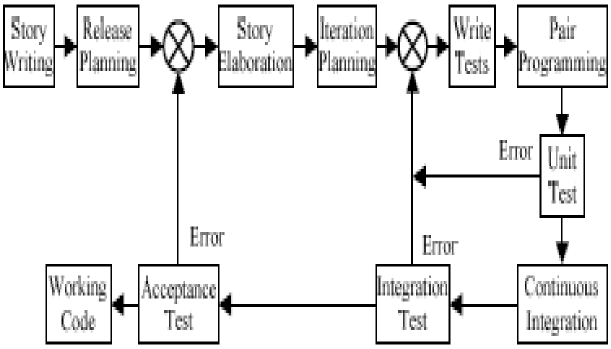

Figure 3. System Representation of XP

B. *Scrum:*

Scrum, along with XP is one of the most widely used agile methods. This method defines a project management framework, managed by the Scrum master. Scrum is one of the few agile methods that has been scaled up to medium projects. In Scrum, the iteration length is 30 days and it is called a "sprint". The sprint will be preceded by pre-sprint planning and will be followed by a post-sprint meeting. Scrum practices include the daily scrum meeting and product backlog.

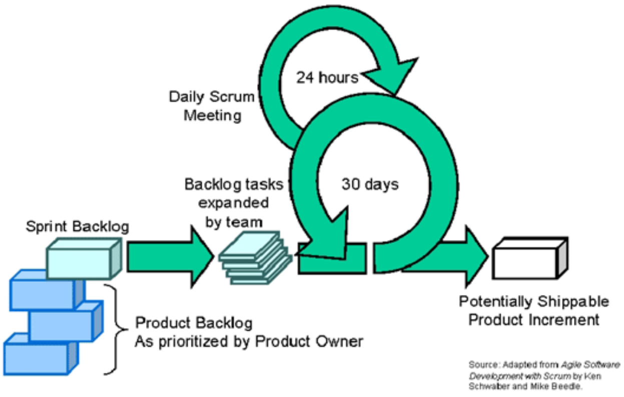
24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
30 days
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Potentially Shippable Product Increment
Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Figure 4. Scrum Model

C. *Crystal Methods:*

Crystal is a family of methodologies. There is no one Crystal methodology but different Crystal methodologies for different types of projects. The factors that influence the methodology selection are staff size, system criticality, and project priorities. Crystal Clear is an optimization of Crystal family and it is targeted at projects where the team consists of two to eight people sitting in the same room or in adjacent offices.

IV. SOFTWARE QUALITY REVIEW

A. *Software Quality:*

Software quality is defined as:

a. The degree of which a system, component, or process meets specified requirements

b. The degree to which a system, component or process meets customer or user needs or expectations.

Quality means different things to different people such as users, customers, and managers. The user will see quality as "what I want", "fast response" or "cheap to run" where from the designer point of view it can mean "good specification" "technical correct" or "well documented [4]. Following is the criteria for quality

a) ***Correctness:*** The extent to which a program fulfills its specifications

b) ***Reliability:*** The extent to which a system perform its intended function without failure

c) ***Efficiency:*** The computing resources required by a system to perform a function

d) ***Integrity:*** The extent to which data and software are consistent and accurate across systems

e) ***Usability:*** The ease of use of the software

f) ***Maintainability:*** The effort required to locate and fix a fault in the program within its operation environment

g) ***Flexibility:*** The effort required to modify a system

h) ***Testability:*** The ease of testing the program, to ensure that it is error-free and meets its specification

i) ***Portability:*** The effort required to transfer a program from one hardware configuration and/or software environment to another or to extend the user base.

j) ***Reusability:*** The ease of reusing software in different context

k) ***Interoperability:*** The effort required to couple the system to another system.

B. *Software Quality Assurance (SQA):*

The IEEE Glossary (IEEE 1990) definition of SQA is:

A planned systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established functional technical requirements

A set of activities designed to evaluate the process by which products are developed or manufactured. Following are the most common activities.

a. ***Management reviews:*** To monitor progress, determine the status of plans and schedules, confirm requirements and their system's allocation, or evaluate the effectiveness of management approaches used to achieve fitness for purpose

b. ***Technical reviews:*** To evaluate a software product to determine its suitability for its intended use. The result should provide the management with evidence that either confirm (or does not confirm) that the product meets the specification and adheres to standards, and that changes are controlled

c. ***Inspections:*** A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Types include code inspection, design inspection

Purpose: To detect and identify software product anomalies

d. ***Walk-throughs:*** A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code and the participants ask questions and make comments about possible errors, violation of development standards and other problems

Purpose: to evaluate a software product

Less formal than inspection

e. ***Audits:*** To provide an independent evaluation of the conformance of software products and processes to applicable regulations standards, guidelines plans and procedures.

V. TESTING CONVENTIONAL AND AGILE PROJECTS

To make a software product free from errors and assess its efficiency, testing is conducted. Where software development is moving towards agile software development, testing approaches also gets changed in projects.

In agile software development testing and development of software runs parallel in order to achieve higher level of quality. Testing can be valued as the backbone of QA activities and a vital step to attain quality in software product. In agile development process an error or bug is fixed as soon as it is known and no matter who fixes it. Adopting this manner, bugs and errors are detected and fixed in early stages. Fixing bugs and errors sooner, implies to save time, cost, resources and maximization to quality.

A. Quality of Process:

“In an agile project, it’s particularly important to use simple approaches, because they are easier to change. Agile development process are claimed to be more flexible toward change [5]. So by nature agile approach responds to change efficiently and extra time and effort is not needed. And being iterative the resources of the organization are not locked in linear processes. Several studies have found and many practitioners have claimed that errors and bugs found after the release cost more time and resources to organization. Agile practices like Pair Programming and Test Driven Development targets to detect and fix flaws of software before final release and ultimately this development nature benefits an organization in terms of time and resources.

B. Agile Testing:

Pair programming and Test-Driven Development (TDD) are some of the key practices of agile development to achieve quality in software products. Using these approaches, agile development makes testing an integral part of project. In agile methodologies the testing responsibility is shifted from QA tester to developer. In agile, developers are supposed to write tests and test their code or each other’s while doing Pair Programming. Customer is supposed to be highly involved throughout the project in agile development. Acceptance testing is the responsibility of customer who is participating in project.

C. Test-Driven Development (TDD):

TDD focuses on writing tests before coding and frequently integrating the new code. In TDD new code denotes to either a piece of code or a piece of code that already exists and integrated after few changes; in order to emerge changed requirements (according to customer’s feedback). TDD is influenced by Test-First Development (TFD) approach where developers are to write all test cases and tests before starting actual programming.

TDD = Refactoring + TFD [6].

To visualize the concept of TFD now, let us present the UML diagram of TFD by Scott W. Ambler.

TDD is not only a testing approach but also an approach to design the software gradually and incrementally. As shown in the above diagram that adopting TFD, each piece of code is supposed to pass all test cases before it is written.

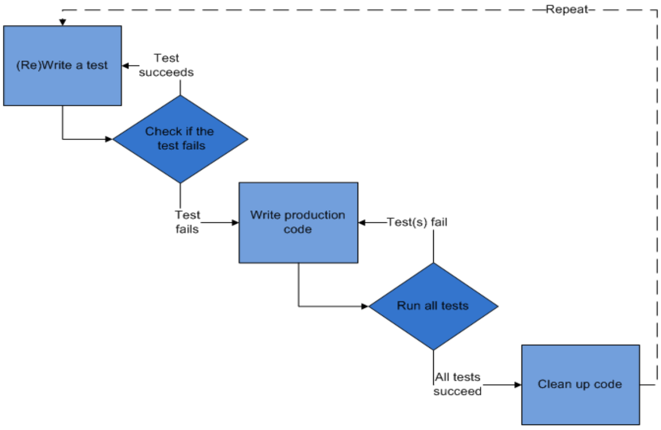

Figure 5: Test Driven Development

D. Pair Programming:

Pair Programming (PP) means code is written by two people at one screen/keyboard/mouse. PP is one of the core activities of XP. Purpose of adopting PP is to monitor and learn from each other continuously, while writing the code. Monitoring of software development and process is the responsibility of QA staff but PP imposes this responsibility on developers by sticking them to work together. PP approach is adopted to achieve high quality. Pair programming can improve design quality and reduce defects. But the question is how to trust developers over professional QA testers. Agile project personnel must be competent and experienced.” Here, competence and experience is meant by experience of building similar systems, prior knowledge of technology domain and having sufficient interpersonal and communication skills.

E. Refactoring:

Refactoring is the process of changing software systems without changing the external behaviours of the code yet improves its internal structure. During refactoring, we change the design of the code after it has been written. This can be done by moving one field from one class to another, pulling some code out of a method to create a new method, or pushing some code up or down a hierarchy.

There is no claim that refactoring will solve all software problems, yet it is a valuable tool.

Advantages of refactor are:

a. Improves the design of software
b. Makes software easier to understand
c. Helps you find bugs
d. Helps you program faster

The most common time to refactor is when adding a new function.

F. SPI and Agile Methodology:

Software Process Improvement (SPI) is improving software development process. Ultimately SPI develops quality software products. Commonly used standards and models to improve software development process, are ISO standards and Capability Maturity Model Integration (CMMI) [7]. Chasing these standards and models SQA plays the most active role. QA personnel are responsible to implement, monitor and guide selected model or standards for SPI on an organization level as well as on project level.

G. Problems Identified:

Unlikely conventional development methodologies agile development integrates QA practices in development activities, rather than practicing them independently and separately. And for standard conformance generation of documentation is essential where as agile principle is to develop working software providing only comprehensive documentation. The problem is that make CMMI and agile methodologies are incompatible. Although people involved in project are participants of SPI but QA personnel are responsible of SPI activities. SQA personnel conduct surveys, evaluation, meetings and internal audits. They are considered as experts within an organization as they educate and implement SPI in it. Besides, merging QA activities in software development, agile methodologies cut short the organizational role of QA. Developers may be aware of testing and designing but they might be less aware of SPI on organizational level. Literature shows that there is need to redefine the role of SQA in agile development projects in order to increase organization knowledge and maturity for maximum output.

VI. PROPOSED SOLUTION

It has been observed that good quality is achieved by using agile process models in software models. But in reality agile process models are not able follow standards. In proposed method by integrating QA activities with agile methods good quality is achieved. The QA practices in agile projects are redefined in proposed method. Most of the QA activities in agile projects are performed by the developer. In conventional models QA personnel are supposed to perform QA activities. There is no doubt that agile methodologies came up with higher quality due to their incremental and test-driven nature. Production of higher quality, and absence of systematic, organized and well defined procedures and standards, show that there is much space of improvement in agile SQA activities. To improve quality of the products it is proposed that redefine SQA activities.

A. Extra Layer of QA:

Agile methodologies are incremental, adaptive to change and self-organized. But in any situation, leadership is required because self-organization cannot lead further when requirements and system get complex. Agile integrates QA activities in software development by requiring some QA personnel to play the role of developer in projects.

In convention approach SQA activities target not only quality of product but also the quality of process over organization level[8]. SPI is the vital part to achieve in conventional practice of SQA. It is proposed that agile development approach needs to redefine their SQA role and activities in development projects. By enhancing the role of SQA may result in making agile methodologies effective over organization level. In proposed system an extra layer of QA is proposed in agile development projects.

In our proposed solution, applying extra layer of QA means to place QA experts with development team. Adopting this approach, organization might be able to maintain quality standards. All typical QA activities like testing, requirement gathering and evaluation must be the responsibilities of QA expert rather than developer. To maintain the agility, communication among development teams and within team members must be encouraged.

B. The Need of Extra Layer of QA:

Being virtually separated an extra layer of QA may perform conventional QA activities without disturbing agility of project. Focus of QA is to monitor the process and product development to ensure the quality of the product, delivered. Throughout the development it is the responsibility of QA to provide the organization with process assurance and product assurance [9]. In process assurance QA conducts meetings and audits to assure that standards, procedures and plans are being followed. In product assurance QA activities are to assure the quality of requirements, design and code. To follow a defect prevention technique and conduct testing of developed software is typically a QA activity and responsibility. By imparting testing culture in agile projects may increase the value to product quality but process quality cannot be compromised to make good business. According to agile practitioners are less attracted towards CMMI certification or other commercial standards of product and process quality. Agile focuses more on testing that may cause to neglect other QA activities. If the responsibility of requirement analysis and testing is shifted on developer, it may overload the work of developer causing lack of quality in process and product. There is no question on the role of QA professionals in agile software development projects.

QA professionals' role, in agile development projects, needs to be redefined in order to get higher quality rather than putting their responsibilities on developers. In the proposed method an extra Layer of QA is proposed in agile projects. It is proposed that to allocate QA professionals in development team as member. And all QA activities within the team must be monitored by that QA professional and mutual collaboration of those QA professionals must be encouraged and maintained to confirm agility of project.

In agile methodologies either developer works as tester or vice versa, this makes a dramatic shift in responsibilities and may be takes as a challenge of expertise. By adopting proposed approach an organization will be able to get quality of work as all resources will be working in their expertise. According to theory developer must not test his own written code, but in agile development most of the testing is done by developer that must be kept on minimum level in order to enhance quality.

Agile methodologies are lacking standards to access their capabilities and maturity unlike conventional development [10]. This may be because all experts are involved in development and do not consider to monitor and measure the project on organizational level. By adopting proposed approach organization may develop and adopt standards and may handle the quality of complex projects efficiently.

Instead of educating the customer it is better to utilize expertise of those people who have abilities to interpret customer need to be implemented technically. But continuous interaction with customer is also a positive value to product quality and agility. In proposed approach QA personnel rather than developer must interact with customer on regular basis to ensure the quality of product.

Many developers claim that maintenance of detailed documentation is required. Documentation is not only required by customer, but also essential to maintain the quality of product. Developer cannot follow all of the quality concerns and procedures without the assistance of QA personnel. If QA activities are imparted in agile project through extra layer of QA, it may lead to maximization of product quality.

VII. CONCLUSION

QA in Agile methodologies will not succeed if addressed as it has been during traditional Waterfall processes, i.e., as a separate activity during the development process. QA is something that must be integrated into the process from the start, rather than be added as an afterthought. It is concluded that by adopting the proposed approach an organization will be able to get quality of work as all resources will be working in their expertise. According to the proposed method developer must not test his own written code, but in agile development most of the testing is done by developer that must be kept on minimum level in order to enhance quality.

VIII. REFERENCES

[1] Agilemanifesto, Available at http://www agilemanifesto.org , last checked: 05-01-2009

[2] Miller, G. G. The Characteristics of Agile Software Processes. The 39th International Conference of Object-Oriented languages and Systems 2001.

[3] Sajid Ibrahim Hashmi, Jongmoon Baik, “Software Quality Assurance in XP and Spiral - A Comparative Study”, Fifth International Conference on Computational Science and Applications.

[4] E. Mnkandla, .B. Dwolatzky,” Defining Agile Software Quality Assurance”

[5] Coram, M., and Bohner, S., “The Impact of Agile Methods on Software Project Management”, In proceedings of the 12th IEEE International Conference and Workshops on the Engineering of `Computer-Based Systems (ECBS), April 2005, pp. 363-370.

[6] “NASA Software Quality Assurance Center”, Available at http://satc.gsfc.nasa.gov/assure/ assurepage. html,last checked: 04-01-2009

[7] V. Esther Jyothi and K. NageswaraRao, “Effective implementation of agile practices – A collaborative and innovative framework”, published in CIIT International Journal of Software Engineering and Technology, Vol 2, No 9, September 2010.

[8] Maurer, F. and Martel, S. (2002a). Extreme programming: Rapid development for Web-based applications. IEEE Internet Computing 6(1): 86-90.

[9] Highsmith, J. and Cockburn, A. (2001). Agile Software Development: The Business of Innovation. Computer 34(9): 120-122.

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