An Information System Audit Model for Project Quality Improvement by the Agile Methodology

Dong Hyup Kim, Dong Soo Kim, Chan Koh, and Hee Wan Kim

Abstract—**Software development has been applied with various automatized tools and techniques to increase the productivity in order to supply the requested products on time with the highest quality to customers. As a result, numerous software development methodologies have been tried continuously during the practical work. As a consequence of this effort, an agile methodology has recently been utilized in many projects in order to promptly deal with changing demands of customers. In the case of South Korea, project-applicable methodologies have been established and applied to different fields, and this tends to expand further.**

Information system audit evaluated product suitability throughout the whole development process including the previous step and contributed greatly to the quality enhancement. However, there are several differences between recently growing agile project and the previous one, such as management methods, construction phases, and key products. As a result, information system audit should understand the character of agile and review the comprehensive development process.

This paper proposes an audit model that is suitable to the agile project by comparative analysis between the current audit model and agile methodology. Agile methodology audit model will be advantageous to the Agile applied project audit. The model should be verified and supplemented by applying to the practice, and various Agile practical items should be applied continuously and studied further.

Index Terms—**Agile methodology, information system audit, audit model style, agile audit model.**

I. Introduction

There has been rapid growth and progress in numerous development methodologies to model information system. To draw or model the information system as a communication tool within a term or with the customer, the model will need the methodology or method that can be used as an explicit way to think and act [1]. For some writers, methodology is different from method [2]. A methodology or software engineering method consists of process, standard vocabulary, set of rules and guidelines which tell what to do, how to do, when to do and why it is done [1], [3]. Agile methodology as an evolutionary development will more effectively meet the needs of the customer quickly, adaptively and focus on the software itself. In the beginning, agile methodology was suitable for small- or medium-sized business and personal computer products [3], but soon this has been implemented in large-sized business as well [4]. A primary methodology, which is similar to "coding, modification, re-coding," became structured like a waterfall model. It was developed into standardized methodology, and many methodologies that control and administer software production were created and applied constantly [5]. An organization established methodologies that are suitable to their character, determined tools and outputs, and standardized them. Based on these, efforts were put on providing customer-requested products at the right time with high qualities. During the 21st century, a new paradigm called the agile development methodology appeared and achieved good results by putting value on interaction among software developers, operating software, cooperation with customers, and coping with the change [6]. Most of development process goes through tailoring and is modified to be suitable for the project. The Agile methodology can be understood as reducing process. However, if Agile/agility is missing, reducing process cannot define the Agile methodology [7].

The application of the agile methodology to projects is limited so far. Projects run in a chaotic way in reality by overlapping with other development methodologies and management processes. On the other hand, there are few cases that advantages of Agile processes were not accepted by incomprehension of the agile process and maladaptation to the change. Problems and improvements in these instances should be detected so that the quality of the Agile applicable projects will be increased, and these improvements play a key role in information system audit. Most of the system development domain can be applied to the audit but the audit model, which checks Agile project management and development process comprehensively, is positively necessary. As a result, Agile methodology audit model should be studied so that Agile project audit will be more effective. This thesis proposes Agile methodology audit model by comparative analysis between Agile methodology and the current audit model.

II. The Problems in the Current Information System Audit Model

Business, which applies Agile to the Information system development, has currently been increasing but audit guidelines for Agile do not exist, and audit is performed by applying an object-oriented/component based model. The audit is stepwise including requirement definition, design, and termination. The business' size and risk of danger should

Manuscript received November 20, 2012; revised March 18, 2013.

Dong Hyup Kim is with the Samsung SDS, Seoul, 135878 Korea (e-mail: hyupdong.kim@samsung.com).

Dong Soo Kim is with the Graduate School of Information System, Konkuk University, Seoul, 143701 Korea (e-mail: dskim@kisac.co.kr).

Chan KOH is with the Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, 139743 (e-mail: chankoh@seoultech.ac.kr).

Hee Wan Kim is with the Division of Computer Science and Engineering , Shamyook University, Seoul, 143742 Korea (e-mail: hwkim@syu.ac.kr).

DOI: 10.7763/IJIET.2013.V3.284

be considered to determine if it is necessary to perform an additional audit or more auditors can be stationed at the field to audit. If working expenses is under two million dollars or duration of work is under six months, the requirement definition step can be omitted. In this case, the ordering body should check directly whether tasks in the requirement definition document are reflected or not [8].

TABLE I: AGILE PROCESS AND CURRENT AUDIT CRITERIA [8]-[12]

Steps		Activity/action	Product	Current audit criteria
Requirement Definition		Defining User story	User story, Product backlog	X
		Establishing test plan	Test protocol	O
		Defining Architecture	Architecture definition document	O
Release planning		Establishing release schedule	Release protocol	X
Architecture/ construction	Sprint launch	Establishing sprint plan	Sprint backlog	X
	Analysis/design	Constructing Architecture	Architecture definition document, UI standard, development standard	O
		Application analysis/design	Monitor definition document , ERD	O
	Development	Application development	Source code	O
	Test	Measure test implement	Test scenario, result	O
	Sprint Review	Customer review	Sprint review (customer) result	X
		Retrospect	Retrospect result	X
Examination		Integrated test implementation	Integrated test scenario, results	O
		System test implementation	System test scenario, results	O
		Acceptance test	Acceptance test results	O
Execution		Manual making	User manual	O
Project launch		Project planning	Project plan, WBS	O
Project performance regulation		Requirements alterations and administration (management)	Product backlog, User story	X
		Issues and danger management	List of issues and risks	O
		Progress management	User backlog, Sprint backlog, A burn-down chart	X
Project termination		Project termination	System handover, final exam confirmation	O

However, Agile process proceeds in these steps: requirement definition → release plan → construction → test → execution. Since Agile methodology has its sprint cycle repeated many times depending on the release plan, there is much difference between the current audit and the audit with Agile methodology applied.

Since the audit domain is not a process by which the Agile process does not aim for a specific field or system, it is possible to apply equally the current audit domain. However, the company's change management such as active customer participation, culture, and organizational change during a frequently repeated Agile process is an important factor for the company that is accustomed to the existing development step. The current audit domain does not consider these change management, though.

The audit perspective/inspection standard is that a working process and its product are not equal according to an Agile development process. Therefore, the development environment and process should be understood and the audit viewpoints and standard should be applied accordingly.

The product backlog and sprint backlog, Agile's representative outputs, are similar to the requirement definition, but audit can be performed only when the product contents are comprehended thoroughly. However, relevant check points or detailed review factors are not considered in the current audit inspection framework.

Moreover, review factors that can guarantee the adequacy such as a burn-down chart, sprint review, and retrospect results, which represents project management and quality assurance document, are not considered. New methods should be sought to secure the visibility and traceability of the requirements, because Agile products are variant.

The Agile project process and main outputs, as well as the comparison with the current audit criteria are organized as below Table I.

III. PROPOSAL OF AGILE METHODOLOGY AUDIT MODEL

A. Agile Methodology Audit Model

Agile methodology audit model reuses the current information system development audit model and combines important considerations adequately during the Agile methodology audit. The fact that audit of the Agile-based development project is processed effectively and the audit quality of the project can be enhanced is a great significance.

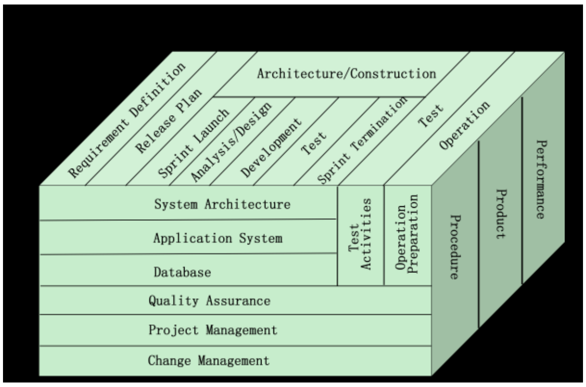


Fig. 1. Agile methodology audit model.

1) Audit point

The Agile process was applied to the audit point, which is composed of requirement definition, analysis/design, implementation, test, and operation steps of the current information system audit model. On the other hand, Agile methodology audit model consists of requirement definition, release plan, architecture/construction, test, and operation steps.

The release plan was added after requirement definition, especially because of iteration, which is a key component of Agile. Architecture/construction phase was repeated accordingly with the release plan. This phase was composed of sprint launch, analysis/design, development, test, and sprint termination. It was applied with frequent release and repetitive development process [11], [12].

The audit point of the current information system audit model and the Agile methodology audit model is illustrated in Table II.

Requirement definition phase has the user story as a foundation. Defining the requirements check whether the system scope is omitted or not. It also ranks priorities of the user story and computes story points. Moreover, it checks whether development plan, which is placed in each sprint, was established reasonably or not.

Release planning phase checks for the range of performance during the release and determine if schedule will proceed as planned. It also inspects whether the iteration period and complete schedule are defined or not.

Architecture/construction phase sets the sprint plan based on user story and architecture and verifies the possibility of developing and testing the code. Its inspection focuses on the completeness of the user story, which is assigned to the sprint by customer reviews

The test phase checks for whether integrated test, system test, and acceptance test are performed comprehensively. The implementation stage finally inspects for an appropriate termination of release-functional manual, education, transition, and examination.

TABLE II: AUDIT TIME OF THE CURRENT AND AGILE METHODOLOGY AUDIT MODEL [8], [11], [12]

Classification	Audit Time								
Current audit model	Requirement analysis		Analysis/design		Implementation			Test	Operation
Agile methodology audit model	Requirement definition	Release plan	Architecture/construction					Test	Operation
			Sprint launch	Analysis /design	Develop -ment	Test	Sprint Review		

TABLE III: CHECK LIST DEDUCED FROM THE DOMAIN OF THE CHANGE MANAGEMENT [5], [11], [12]

Audit Point	Audit Domain	Activity	Remarks	Remarks
Launch/ Plan	Change Management	Establishing plans for change management	Analyzing the interested parties, planning scrum/agile education training, assigning change management, and other planning	Additional
Execution/ Regulation		Performing change management	Distinguishing/eliminating organizational obstacles according to the environment of Agile development and performing programmatic change management	Additional
Termination		Evaluating effectiveness and performance of business	Measuring the team performance according to the Agile project measuring index	Additional

2) Audit domain

System architecture, applied system, and database apply the audit domain of the current information system audit model equally. However, the domain of change management, which is deduced from Agile process, was applied additionally. Audit points are classified into launch/plan, execution/regulation, and termination, and this classification is similar to the project management domain. On the other hand, change management activities are divided into establishing plans for change management, performing change management, and evaluating effectiveness and performance of the business.

3) Audit viewpoint

Procedures and products can be applied with the same standard as the information system audit standard. Business accomplishment has added productivity to reflect Agile characteristics. Agile cooperate by a self organizational project team and communicate continuously with customers. This team creates successful products in each sprint. They self-reflect on each sprint by repetitive plan, tracing, and evaluating the team speed, and also prepare for the next sprint. The productivity is one of the important factors in business because it enables to predict the developer, scrum team, and release date of the final product beforehand [5]. Inspection standards according to audit viewpoints are listed below.

B. Agile Methodology Audit Checkitems

Basic check items were applied with Agile characteristics based on the current audit check items. The existing requirement analysis was named as requirement definition for the audit time, and the basic check items were applied. However, basic check items were moved to carry out the plan of an early unit/integrated test because it is important to do an early test for the Agile project. The requirements were analyzed, and release planning was added to establish release planning and repetitive plan. Moreover, basic check items, which were previously divided into analysis, design, and implementation, were integrated into architecture / construction. This process simplified check items and allowed the quality inspection of Agile project, by which analysis, design, and development run parallel to each other. Table V proposes the basic check items of applied system domain as below.

TABLE IV: AUDIT VIEWPOINT AND INSPECTION STANDARD OF THE AGILE MODEL [5], [8]

Audit Viewpoint	Inspection Standard	Related characteristics	Remarks
Process	Plan Reasonability	Business management planning construction/management, evaluation of suitability of establishing and following procedures	Existing
	Process Reasonability	Establishing development/management/maintenance procedure and reasonable establishment of risk/schedule/quality/form/human resources/change management procedures	existing
	Compliance	Following plans and risk / schedule / quality / form / human resources / change management procedures fairly	Existing
Product	Functionality	Sufficiency, completeness, accuracy, interoperability, and connectivity of the Functionality function	Existing
	Integrity	Data integrity and accuracy	Existing
	Usability	User convenience, management convenience, and learning	Existing
	Stability	System stability, service continuity, quick restoration	Existing
	Security	System confidentialness and safety	Existing
	Efficiency	Efficiency of using information resources (human resources, server, etc), work efficiency, quick answers, system extendability, technology development compatibility	Existing
	Compliance	Following criteria / procedure / standard / methodology of the product	Existing
	Consistency	Analysis, alteration, existing, traceability, maintainability	existing
Performance	Sufficiency	Satisfaction of work/technical requirements, achieving performance goals, sufficient task scope	Existing
	Realizability	Concreteness, feasibility, efficiency of investment, achieving the performance goals, system availability	Existing
	Productivity	Speed and receptiveness of Agile, frequency of release and repetition, test productivity, business accomplishment and productivity	Additional

TABLE V: BASIC CHECK ITEM OF APPLICATION SYSTEM DOMAIN OF AGILE METHODOLOGY AUDIT [8], [11], [12]

Audit Point	Basic Check Items	Remarks
Requirement Definition	01. Analyzing current work and checking whether possible solutions to the problem are proposed or not.	Existing
	02. Sufficiency and adequacy of deducing and analyzing users' requirements	Existing
	03. Adequacy of use-case model specification level about system function	Existing
	04. Checking whether conceptual analysis class was deduced sufficiently	Existing
	05. Checking whether user interface prototyping plan was set fairly	Existing
	06. Checking whether test plan was set appropriately	Change
Release Plan	01. Checking whether release planning was set appropriately by analyzing requirements	Additional
	02. Setting repetitive plan and appropriate selection of story for the release	Additional
Architecture/ Construction	01. Sufficiency and completeness of work and detailed analysis of user's requirement/design/implementation of functions	Change
	02. Sufficient refinement of use-case model	Existing
	03. Appropriate performance of user interface prototyping	Existing
	04. Designing for convenience of user interface and its report	Change
	05. Sufficiency and appropriateness of refining analysis class and designing class in details	Change
	06. Sufficiency and appropriateness of interior /exterior interface analysis/design/implement	Change
	07. Appropriate analysis/design/implement of access privileges and control	Change
	08. Detailed design for introducing/implementing the component	Change
	09. Unit testing	Existing
	10. Sprint review for each sprint	New

IV. VERIFICATION OF THE AGILE METHODOLOGY AUDIT MODEL

The survey about the necessity, audit time, and audit domain of the Agile methodology audit model was conducted (see Table VI). Ninety-six to hundred percent of the results indicated that Agile applied projects are needed.

The survey about the appropriateness of check lists indicates that basic check lists for the applied system is hundred percent suitable. Change management was 89.3 percent suitable, and Agile product detailed review items are

99.5 percent suitable (see Table VII).

TABLE VI: THE NECESSITY FOR AUDIT OF THE AIGLE METHODOLOGY PROJECT

Classification	Yes	No
Necessity of Agile applied projects	100%	0%
Suitability of the audit time	100%	0%
Necessity of adding the change management domain	96.4%	3.6%
Contribution of quality improvement during the audit, which is suitable to Agile methodology applied projects	100%	0%

TABLE VII: A COMPLETE LIST OF THE SUITABILITY OF AGILE METHODOLOGY AUDIT CHECKLIST

Classification	Items	Suitable	Average	Not Suitable
Basic check lists of the Applied System	10	100%	0%	0%
Basic check lists of change management	7	89.3%	8.2%	0%
Review lists of Agile products	20	99.5%	0.5%	0%

V. CONCLUSIONS

This paper proposes the audit model that perfectly fits to Agile characteristics by comparatively analyzing Agile methodology, which is currently applied in South Korea, and the current audit model. Audit time is when release and repetitive planning are established by analyzing a project-specific repetitive period and its form. When repetition is terminated and reviewed, audit is performed and a reinforcement of constant audit is required additionally. Audit check lists were formed throughout analysis, design, and development stages and added detailed review lists of major Agile products. It also reflected on check lists that reviews whether change management is practiced in Agile environment or not.

A proposed Agile methodology audit model in this paper will contribute greatly to performing audit in future. This research offers Agile methodology audit model based on a comparative analysis between a large IT organization's Agile methodology and the current audit model.

It is expected that combined strengths of Agile and various forms of methodologies will be applied and evolve together. Furthermore, information system audit should be studied continuously in order to take a prompt action in a complex environment.

REFERENCES

[1] H. Eriksson *et al.*, *UML2 toolkit.*, Indianapolis, USA: Wiley Publishing Inc, 2004.

[2] S. Bennett, S. McRobb, and R. Farmer, *Object oriented systems analysis and design using UML*, 2nd ed., Berkshire, UK: MCGraw-Hill Inc., 2002.

[3] I. Sommerville, *Software engineering.* 7th ed., Essex, England: Person Education Limited, 2004.

[4] H. Holmstrom *et al.*, "Agile practices reduces distance in global software development," *Information System Management*, vol. 23, pp. 7-18, 2006.

[5] H. Jegal, J. Lee, and T. Kim, *Scaling software agility: best practices for large enterprises*, Euiwang, Korea: Euiwang Publishing Inc., 2008.

[6] K. Henrik, W. Sim, W. Eum, and J. Han, *Scrum and XP from the trenches*., Insight, Korea: Insight, 2009.

[7] C. Kang, "A Study of Comment-oriented Documentation for Agile Software Development Method," M.S. thesis, School of Information System, Seogang Univ., Seoul, 2006.

[8] Korea National Information Society Agency, *Information System Audit Guideline Manual V3.0*- 2008.

[9] Korea National Information Society Agency, *Information System Audit Guideline V1.0*- 2009.

[10] Korea Ministry of Public Administration and Security, *Information System Audit Guideline Revision Manual*- 2010.

[11] A company, *Agile Process Definition Documentation V1.1*- 2011.

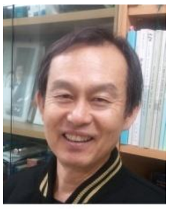
[12] B company, *Audit Report for Information System Construction*-2011.

Dong Hyup Kim received his M.S. degree in Information System Audit from Konkuk University, Korea in 2012. He received his BS degree in Computer Engineering from Kangneung University, Kangwon, Korea in 1995.

He worked as a team manager at the Daegyo Company. Now he is a top manager in the department of Delivery Innovation Center at Samsung SDS, Seoul, Korea. His research interest includes Information System Audit, IT Outsourcing, Enterprise Architecture, and DB Security.

Dong Soo Kim received his Ph.D. degree in Management Information System from Kookmin University, Korea in 2005. He received his MS degree in Computer Science from Seoul National University of Science and Technology, Seoul, Korea in 2001. He also received his BS degree in Computer Science from Kwangwoon University, Seoul, Korea in 1981. He has three Certificate as a Professional Engineer(P.E.) in Information Systems Management, Computer Application System, and Computer Communications from the Korean Ministry of Science and Technology. He also has a Certificate as a Chief Information System Auditor from the Korean Ministry of Science and Technology.

He is a chief consultant in the department of Information System Audit at Korea Information System Audit & Consulting (KISAC) Company and an adjunct professor at Graduate School of Information and Communications, Konkuk University, Seoul, Korea. He has published research papers in Information System Audit in the journal of Korea Digital Policy and Management. His research interest includes U-City Audit, Information System Audit, DB Quality, IT Outsourcing, Enterprise Architecture, and DB Security.



Chan Koh was born in Euisung, Korea. His first Ph.D. is in Electronic Engineering fron Kyunghee University, Seoul, Korea in 1991 and his second Ph.D. is in Economics from Seoul National University, Seoul, Korea in 2008. He also has a MS in Computer Science from Yonsei University, Seoul, Korea in 1983, and a BS in Mechanical Engineering from Kyunghee University, Seoul, Korea in 1974. He has a Certificate as a Professional Engineer(P.E.) in Information Systems Management from the Korean Ministry of Science and Technology in 1987.

He served in the R.O.K Navy officer for three years as a systems analyst at R.O.K. Navy Command & Staff College. Now, He is a professor in the department of Computer Science & Engineering at Seoul National University of Science and Technology (SeoulTech), Seoul, Korea. Also, he is a Professor of Culture Policy Program, Graduate School of IT Policy at SeoulTech. He has been teaching courses at Helsinki Metropolia University of Applied Science in Finand since 2005. He has published research papers in IT convergence in the journal of Korea Digital Policy and Management. His research interest includes Computer Graphics, Game Design, Culture Technology, Cultural Policy and Management.

Dr. Koh is a vice president in the Society of Korea Digital Policy and Management, a director in the Society of Korea Digital Content, a director in the Society of Convergence Knowledge, sn editorial staff in the Korean Association for Regional Information Society.

Hee Wan Kim received his MS and Ph.D. degrees in Computer Engineering from Sungkyunkwan University, Seoul, Korea in 1995 and 2002. He received his B.S. degree in Computer Science from Kwangwoon University, Seoul, Korea in 1987. He has two Certificate as a Professional Engineer(P.E.) in Information Systems Management and a Chief Information System from the Korean Ministry of Science and Technology in 1995 and 2001.

He worked as a computer programmer for 4 years at Korea Electric Power Cooperation (KEPCO). Now, he is a professor in the division of Computer Science & Engineering at Shamyook University, Seoul, Korea. He has published research papers in Information System Audit in the journal of Korea Digital Policy and Management. His research interest includes Database, Information System Audit, Project Management, and Software Engineering. He is an editorial staff in the Korean Association for IT Service Society.