

애자일에 기반한 정보시스템 감리 모델에 관한 연구

김동협*, 고찬**, 김동수***, 김희완****
삼성 SDS*, 서울과학기술대학교 컴퓨터공학부**, 건국대학교 정보통신대학원***, 삼육대학교 컴퓨터학부****

A Study on the Agile-based Information System Audit Model

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

Delivery Innovation Center, Samsung SDS*

Dept. of Computer Science & Engineering, Seoul National University of Science and Technology**

Graduate School of Information and Telecommunications, Konkuk University***

Division of Computer Engineering, Shamyook University****

요약 애자일 개발 방법론은 소프트웨어 개발자들 간의 상호작용, 작동하는 소프트웨어, 고객과의 협업을 주요 가치로 삼고 있다. 대부분의 개발 프로세스는 테일러링이라는 과정을 거쳐 해당 프로젝트에 맞도록 조정한다. 실제로 프로젝트는 다른 개발 방법론 및 관리 프로세스와 중첩이 되어 혼란스럽게 진행된다. 그래서 애자일 프로세스의 장점을 그대로 수용하지 못하는 실정이다. 이러한 문제점들을 발견하여 애자일 적용 프로젝트의 품질을 향상시키는 것은 정보시스템 감리의 중요한 역할이다. 따라서 애자일 프로젝트의 효과를 높이기 위해서는 애자일 방법론 감리 모형에 대한 연구가 필요하다. 본 논문에서는 애자일 방법론을 적용하여 정보시스템 감리를 위한 모델을 제안하였다. 애자일 기반의 정보시스템 감리 영역 및 점검항목들을 제안하였다. 제안한 모델의 적합성을 검증하고자 점검항목의 적합성에 관한 설문 조사 결과, 모든 영역에서 89.3% 이상이 적합하였다.

주제어 : 애자일 개발 방법론, 프로젝트 품질, 애자일 감리 모형, 감리 영역, 점검항목

Abstract Agile development methods regards an interaction among software developers, operating softwares, and cooperation with customers as its main value. Most of the developing processes undergo a procedure called tailoring to meet the relevant project. Yet, in reality, projects run in disorder due to their overlap with other methods and management processes. As a result, the advantages of agile process cannot be fully used. So, the information system audit has an important role to detect these problems and improve the qualities of agile application projects. Also, researches should be conducted on Agile-based Information System Audit Model in order to enhance the effectiveness of agile projects. This paper suggests a model for an information system audit that applied agile developing methods. This model proposes audit domains and check lists for agile-based information system. In order to verify the suitability of the suggested model, more than 89.3% of the checklists were verified as suitable in all domains.

Key Words : agile development methods, project qualities, agile audit model, audit domain, check lists

* 본 논문은 2012년 삼육대학교의 학술연구비에 의하여 지원되었음.

Received 9 June 2013, Revised 5 July 2013

Accepted 20 August 2013

Corresponding Author: Hee Wan Kim(Shamyook University)

Email: hwkim@syu.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

소프트웨어 개발 품질과 생산성을 향상시키기 위해 수많은 개발 방법론이 진화, 발전해 왔다. ‘코드 작성, 수정, 코드 재작업’ 과 같은 초기 방법론에서 시작해 폭포수 모델처럼 더욱 구조화되고 정형화된 방법론으로 발전해 가면서, 소프트웨어 생산을 제어하고 관리하는 수많은 방법론을 만들고 적용해 왔다[14]. 조직에서는 조직 특성에 맞는 방법론을 제정하고 방법론을 효율적으로 유지할 수 있도록 수많은 도구와 산출물들을 정의하고 표준을 수립하여 왔으며, 이를 바탕으로 고객이 요구하는 제품을 적시에 제공하고 수준 높은 품질을 제공하고자 노력해 왔다. 2000년대에 와서는 애자일 개발 방법론이라는 새로운 패러다임이 등장하여 소프트웨어 개발자들 간의 상호작용, 작동하는 소프트웨어, 고객과의 협업, 변화에의 대응을 주요 가치로 삼고 큰 성과를 거두고 있다[16]. 대부분의 개발 프로세스는 테일러링이라는 과정을 거쳐 해당 프로젝트에 맞도록 조정한다. 표준 프로세스를 조정 작업 없이 모두 수용하는 프로젝트는 매우 비효율적이다. 애자일 방법론이 경량화를 의미하는 것으로 이해하기도 하는데 애자일의 가치가 빠져 있다면, 프로세스 경량화 자체가 애자일을 의미한다고 할 수 없다[1].

이러한 애자일을 프로젝트에 적용함에 있어 현재까지 많은 한계를 지니고 있다. 실제로 프로젝트는 다른 개발 방법론 및 관리 프로세스와 중첩이 되어 혼란스럽게 진행되는가 하면 애자일 프로세스에 대한 이해 및 조직이 변화에 적응하지 못해 애자일의 장점을 그대로 수용하지 못하는 사례들이 종종 있다. 이러한 사례들의 문제점과 개선사항들을 발견하여 애자일 적용 프로젝트의 품질을 향상 시키는 것은 정보시스템 감리의 중요한 역할이라 생각한다. 하지만 애자일 프로젝트에 적합한 감리 모형은 제시되어 있지 않다. 물론 시스템 개발 영역의 상당 부분은 감리에 적용 가능하지만 애자일 프로젝트의 관리 및 개발 프로세스를 종합적으로 점검하기 위한 감리 모형은 반드시 필요한 부분이다. 따라서 애자일 프로젝트의 감리 효과를 높이기 위해서는 애자일 방법론 감리 모형에 대한 연구가 필요하다고 판단된다. 본 논문에서는 애자일 방법론 및 현행 감리 모형을 비교 분석을 토대로 애자일 방법론 감리 모형을 제시하였다.

2. 관련 연구

2.1 애자일 방법론

애자일 소프트웨어 개발은 전통적인 폭포수 모델의 합리적이지 못한 부분들에 대한 대응으로 출발한다. 즉 전통적인 폭포수 모델은 원래의 계획에 집착함으로써 변화하는 고객의 요구사항 충족에 실패 했고 요구 사항 변경을 통제함으로써 프로젝트의 유연성이 사라지고 고객의 가치 전달 제공이 쉽지 않았다. 따라서 문서와 산출물 중심의 무거운 방법론을 대체할 대안의 필요성을 공감하고 가벼운 프로세스로 불리며 2001년 애자일 Software Development 선언을 통해 본격적인 시작을 하게 되었다 [3].

애자일 선언문에는 “우리는 소프트웨어를 개발하고, 또 다른 사람의 개발을 도와주면서 소프트웨어 개발의 더 나은 방법들을 찾아가고 있다. 이 작업을 통해 우리는 다음을 가치 있게 여기게 되었다: 공정과 도구보다 개인과 상호작용을, 포괄적인 문서보다 작동하는 소프트웨어를, 계약 협상보다 고객과의 협력을, 계획을 따르기보다 변화에 대응하기를 가치 있게 여긴다[8]. 즉 요구사항의 변화를 수용하고 이를 지원하면서도 실제로 동작하는 소프트웨어를 끊임없이 출하해야 한다는 데 초점을 맞추고 있으며 소프트웨어를 개발할 때 진정 중요한 것을 최우선으로 해야 한다는 단순한 이치야말로 오늘날 업계에 있어 가장 중요한 항목이라고 이야기한다. 애자일 선언문과 이에 담긴 원리는 애자일의 기본 철학을 보여주며 애자일의 베스트 프랙티스에 적용된 모든 것과 연계되어 있다[15].

2.2 전통적 개발 방식과 애자일 방식의 차이

전통적 개발 방법론은 소프트웨어 개발방법론 중에 가장 오래된 개발방법론이며 가장 널리 배포되어 있는 방법론이다. 크고 작은 회사들이 전통적으로 사용한 소프트웨어 개발 방식은 소위 ‘폭포수 모델’ 이라고 알려진 순차적인 생명주기 모델이다. V모델과 같이 여러 변종이 있지만, 이것은 전형적으로 계획 단계부터 상세하게 최종 제품을 설계하고 이를 세부 사항까지 문서화하는 특징이 있다. 또한 전통적인 관점의 개발 방법론은 프로젝트를 관리하고 시스템을 만들어 나가는 데에 있어서 탑다운 방식을 취하고 있다[6].

각 개별 단계에 대한 상세한 추정치들을 모두 합하여 총 개발 시간에 대한 추정치를 구한다. 프로젝트 이해 당사자들이 이 계획을 철저히 검토하여 승인을 하고 나면 비로소 팀은 일을 시작한다. 모든 작업이 끝나면 제품이 테스트 조직으로 이관되어 고객인도 전에 필요한 테스트를 수행한다. 전체 프로세스에 걸쳐 애초에 설계했던 그대로 제품이 개발되도록 하기 위해 계획이 어긋나지 않도록 철저한 통제가 이루어진다[9].

이러한 전통적이 개발 방법론의 장점은 매우 논리적 이다는 것이다. 만들기 전에 조직화된 상태로 유지한다. 단점은 사람이 개입 될 경우 문제가 발생한다. 개발 과정에서 변경되기 쉬운 고객의 요구사항에 신속하게 대응하기 어려우며 또한 고객 입장에서는 개발이 완료된 이후에야 비로소 요구사항과 구현된 기능과의 차이점과 문제점을 확인할 수 있다는 것도 중요한 한계점이다. 또한 앞 단계를 완료해야만 다음 단계로 진행할 수 있기 때문에 변화에 민첩하게 대응하기 어렵다. 그리고 다양한 인터페이스에 대한 관리영역이 불분명하다는 점과, 기능구현만을 중심으로 보고 있기 때문에 프로젝트의 전반적 관리영역에 대한 가이드 자체가 부족한 것도 대표적으로 지적되는 단점 중에 한가지이다. 이러한 이유에 의해 플랫폼 기반의 통합 개발 시 개발방법론을 구조적 개발방법론만으로 프로젝트를 관리하는 것은 한계가 있다[9].

애자일 개발 방식은 기존의 전통적인 개발 방식에서 얻을 수 없었던 요구사항의 변화에 대한 유연한 대처와 개발 과정의 효율성을 추구하기 위한 소프트웨어 개발 방식이다. 애자일 개발 방식에서는 개발자 중심의 실용적이고도 효율적인 여러 실천 방식을 도입하였다[4].

다음 <Table 1>은 애자일 방법론과 전통적 개발방법론의 차이점을 비교한 것이다.

<Table 1> Comparison of an Agile Method and a Conventional Method [3, 5, 7], Re-edition

Classification	Agile Method	Conventional Method
Plan	-Frequent plan establishment. Set up a repeatative period using a renovation experience-based process and establish a detailed plan for the next period.	- Establish a detailed plan for each step using a plan based process of a detailed plan establishment.

Classification	Agile Method	Conventional Method
Management of request items	- The Development of continuous request items and acceptance of changes are possible, making the alteration and supplement relatively easy. - Detailed plan establishment for the next iteration.	- Early collection of request items and strict management of alterations - in the stage of defining request items, finalize all the requests and establish detailed plans
Design and development	-Just-in-time design -Early testing in which coding and unit tests are performed repetitively at the same time. It includes Demonstration and embodiment of the capability and effectiveness of each system using the embodied functions.	- Detailed design in advance - In the analysis and design stage, design a detailed and specific system which will be developed and embodied based on the design. After all functions are coded in each team at the same time, testing will follow.
Review	-In each repetition, check for the activation of release software.	- Examine milestone documents.
Test	-Verification of functions through developments and testing in small units of functions (continuous/simultaneous/early testing)	-Guarantee the completeness of a system through embodiment of particular modules, unit testing and integrated testing, and system testing.

2.3 애자일 방법론 적용 프로젝트의 감리 필요성

애자일이 변화하는 현실과 모든 제약 상황을 극복하고 항상 프로젝트를 성공적으로 이끌어 준다고 할 수 없지만 최근 많은 사람들이 애자일 방식에 관심을 갖고 더 많은 사람들이 애자일 방법을 적용하고 있다.

앞서 살펴보았듯이 국내 대형 개발 업체를 중심으로 애자일 방법론을 제정하고 프로젝트에 적용하는 사례가 증가하고 있다. 애자일을 적용하는 프로젝트에서 애자일의 특성을 이해하고 이게 걸 맞는 품질을 유지하기 위해 정보시스템 감리 관점도 변화가 필요하다. 반복/점진적 개발에 따른 프로세스의 품질을 점검하기 위해 감리 수행 시기와 단계별 점검 항목의 보완, 다양한 애자일 실천 방법들의 적정성, 효과성을 입증하기 위한 감리 기법 및

세부점검항목의 발굴, 스크럼 프로세스에 대한 이해 및 분석을 통한 효율적 감리 방안 모색, 스크럼 마스터, 스크럼 팀 등의 조직 활동에 대한 감리 관점의 변화 등의 고민이 필요하다 할 수 있다. 애자일 개발 프로젝트에 현재 감리 기준을 적용한다면 실제 프로젝트와 많은 괴리가 생겨 감리에 대한 효과가 반감되고 다양한 이해당사자들에게 신뢰를 주기가 어렵다.

2.4 현행 정보시스템 감리 모형의 문제점

현재 애자일을 적용한 정보시스템 개발 사업이 늘어나고 있지만 애자일을 위한 감리지침은 별도로 존재하지 않고 객체지향/컴포넌트 기반 모델을 적용하여 감리를 수행하고 있다. 객체지향 모델을 적용한 감리모형은 기존 연구[2]에서 수행되었지만, 애자일 방법론을 적용한 시스템에 적용하기가 어렵다. 또한, 애자일은 최종 제품을 위해서 개발/테스트 중심으로 고객과 함께 상당히 유연하게 동적으로 제품을 개발하고 반복 개발을 중요시하며 개발 산출물 또한 이러한 프로세스에 적합하게 상이하고 관리 방식이 다르다.

감리 시점은 요구정의, 설계, 종료 등의 단계별 감리와 감리대상사업의 규모·위험도 등 특성을 감안하여 단계별 감리 외에 추가로 감리를 하게 하거나 감리 인력을 현장에 상주시켜 감리를 하게 할 수 있다[10].

그러나, 애자일 프로세스의 경우 요구정의 -> 릴리즈 계획 -> 아키텍처/구축 -> 시험 -> 이행단계로 진행되며 특히 릴리즈 계획에 따라 스프린트 주기가 여러 번 반복되는 형태로 분석, 설계, 구현단계가 짧은 주기로 동시에 아키텍처/구축단계에서 반복되고 있어 현행 감리 시점인 설계 단계와 많은 차이를 지니고 있다.

응용 시스템을 기준으로 현행 감리와 애자일 방법론의 감리 시점의 차이는 아래 <Table 2>와 같다.

<Table 2> Inspection Items for an Information System Auditing [11,12,13,14], Re-edition

Current Check Items		Steps
Analysis of Dem	01. Check whether or not there is a proposal of a solution plan after analyzing current problems in business	Requirement
	02. Elicit users' request items as well as	Requirement

Current Check Items		Steps
and	sufficiency and suitability of the analysis	
	03. Detail level and appropriateness of Usecase model of the system functions	Requirement
	04. Check if an analysis class of concept-level is sufficiently elicited	Requirement
Anal ysis/ Desi gn	05. Check if a prototyping plan of users' interface was appropriately established.	Requirement
	01. Check if an object work and detailed analysis of users' request items are sufficient.	Architecture/ Construction
	02. Check if the Usecase model was sufficiently refined.	Architecture/ Construction
	03. Check if a prototyping plan of users' interface was appropriately established.	Architecture/ Construction
	04. Check if the users' interface and report paper were designed to ensure convenience.	Architecture/ Construction
	05. Sufficiency and appropriateness in a refinement of analysis class and detailed design of class	Architecture/ Construction
	06. Sufficiency and appropriateness in the analysis/design of the inner/outer interface.	Architecture/ Construction
	07. Check if the analysis/design for the right of approach and regulation was appropriately performed.	Architecture/ Construction
	08. Check if the detailed design for the components' introduction / implementation were appropriately performed.	Architecture/ Construction
Emb odim ent	09. Check if an unit testing plan was appropriately established/	Requirement
	0. Sufficiency and completeness in the implementation of work functions	Architecture/ Construction
	02. Check if the users' interface was implemented to ensure convenience.	Architecture/ Construction
	03. Sufficiency and completeness in the implementation of inner/outer system interface	Architecture/ Construction
	04. Check if a right of approach and regulation functions were accurately implemented.	Architecture/ Construction
	05. Check if the unit testing was performed.	Architecture/ Construction
06. Check if the integrated testing plan is appropriately established.	Execution	

감리영역은 애자일 프로세스가 특정 분야나 시스템을 지향하는 프로세스가 아니기 때문에 현행 감리 영역을 동일하게 적용할 수 있다. 그러나, 기존 개발 단계에 익숙해 있는 기업 내에서는 애자일에서 더 찾아진 빈도수와

반복되는 과정에서의 적극적인 고객 참여, 문화, 조직의 변화 등의 기업변화관리는 중요한 요소이다. 객체지향을 적용한 감리모델의 감리 영역에는 이러한 변화관리 영역은 고려되어 있지 않다.

감리관점/점검기준은 애자일 개발 프로세스에 따라 작업 공정 및 관련 산출물이 상이하다. 따라서 개발 환경과 프로세스를 이해하고 이에 맞는 감리관점과 기준을 적용하여야 한다. 애자일의 대표적인 산출물인 제품 백로그, 스프린트 백로그 등은 요구사항 정의서와 유사하다고 할 수 있지만 해당 산출물의 내용들을 충분히 이해를 하여야 감리를 수행할 수 있다. 하지만 현행 감리점검 프레임워크에서는 관련 점검항목이나 세부 검토 항목이 고려되어 있지 않다.

지금 까지 언급한 현행 감리 점검 프레임워크의 분석 내용을 감리시점, 감리영역, 감리관점/점검기준 측면에서 애자일 적용 시 감리 개선 방향을 비교 정리하면 아래 <Table 3>과 같다.

<Table 3> Comparison between a Current Audit and Agile Method[10,11,12,13,14], Re-edition

	Current Audit	Reformed Model
Audit Point	<ul style="list-style-type: none"> - Steps : Analyze requests, Analysis/design, implementation, testing, development in 5 steps - Definition of requests, design, Auditing in the final steps, step-by-step auditing if necessary or additional and resident audit are possible to be performed - Steps are clear and distinct such as analysis, design etc. 	<ul style="list-style-type: none"> - Steps: define request, release plan, architecture/establishment, testing, performance in 5 steps, analysis/design in the architecture/establishment step, implementation, etc. Repetitive performance. - point of time: Release plan, architecture/establishment, closing auditing and importance of resident audit - Repetitive and progressive development - Analysis, design, development steps are not distinct.
Audit Domain	<ul style="list-style-type: none"> - Classified into system architecture, application system, database, quality assurance activity, and business management 	<ul style="list-style-type: none"> - Fields are the same but activity in the change management sector is important - Architecture, development centered auditing is necessary - Activity for measuring sprint periods and activity for qualities are important

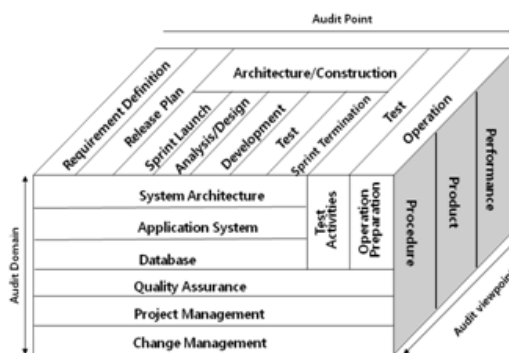
Inspection Criteria based on Auditing perspectives	Perspective : procedure, production, outcome	- perspective classifications are identical
	inspection standard : applied in a diversified matter based on auditing field	- Diverse auditing perspective or standards fit for different characteristics are required because agile products are diverse. - Understanding of agile using progress should be preceded because agile management process are diverse. - A secured plan enabling the traceability of each product is required.

3. 정보시스템 감리 모형

3.1. 애자일 방법론 감리 모형

전체적인 구성은 감리점검 프레임워크[8]를 기반으로 하고 있으며 이중 애자일의 가장 큰 특징이라 할 수 있는 감리시점, 지속적인 조직적 변화관리영역을 추가한 감리영역, 기존 방법론과 차이를 나타내는 산출물/성과 등의 감리관점/점검기준을 제안 하고자 한다.

애자일 방법론 감리 모형은 현행 정보시스템 개발 감리모형의 재사용 가능영역과 애자일 방법론 감리 수행 시 주요 고려사항을 적절히 혼용하여 최근 애자일 기반의 개발 프로젝트의 감리를 효율적으로 진행하고 프로젝트 감리 품질을 향상할 수 있다는 점에서 의의를 가진다.



[Fig.1] Information System Audit Model[11]

3.1.1 감리시점 및 형태

(1) 감리시점

감리시점에는 애자일 개발 프로세스를 적용하여 현행 정보시스템 감리 모형의 요구분석, 분석/설계, 구현, 시험, 전개 단계를 애자일 방법론 감리 모형에서는 요구정의, 릴리즈계획, 아키텍처/구축, 시험, 이행 단계로 구성하였다. 특히 애자일의 핵심 구성요소인 반복(Iteration)을 위해 요구정의 이후 릴리즈 계획을 추가하고 이러한 릴리즈계획에 의한 반복을 수행하게 되는 아키텍처/구축 단계를 스프린트 착수, 분석/설계, 개발, 테스트, 스프린트 종료단계로 구성하고 잦은 릴리즈, 반복 개발 프로세스를 반영하였다[13,14].

현행 정보시스템 감리 모형의 감리시기와 애자일 방법론 감리 모형의 절차는 다음 <Table 4>와 같다.

<Table 4> Audit Time of the Current and Agile Method Audit Model[11,13,14]

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	Development	Test		

요구정의 단계는 사용자스토리에 기반을 둔 요구사항 정의를 통해 개발할 시스템의 범위의 누락 여부 및 사용자스토리의 우선순위와 스토리 포인트를 산정하고 스프린트 별로 배치한 개발계획이 적절하게 수립되었는지를 중점적으로 점검한다. 또한 초기 시스템 구축의 전체적인 방향성을 제시하고 개발계획에 따라 조기 테스트 계획의 수립여부를 점검한다.

릴리즈 계획단계는 릴리즈에 수행할 범위와 일정 수립여부를 점검하고 릴리즈에 포함된 이터레이션의 기간, 완료 일정 등이 정의 되었는지 점검한다.

아키텍처/구축 단계는 사용자스토리와 아키텍처를 기반으로 스프린트 계획을 수립하고 실제 코드의 개발 및 테스트 수행 여부를 검증한다. 또한 고객리뷰를 통해 스프린트에 할당된 사용자스토리의 완전성을 중심으로 점검한다. 추가적으로 매 스프린트별 일일 스크럼 회의를 통해 진행상황을 점검하고 스프린트 기간 동안 반복적으

로 개발->구현->테스트 활동이 이루어져 점증적으로 시스템과 산출물의 구현 및 갱신이 되고 있는지, 스프린트 종료에는 완성된 기능을 시연하고 해당 이터레이션을 적절하게 평가하고 있는지 점검한다.

시험단계는 통합테스트, 시스템테스트, 인수테스트가 종합적으로 수행되고 있는지 점검하고 이행 단계에는 릴리즈 대상 기능의 매뉴얼, 교육, 인수인계, 검수 등의 종료 작업이 적절하게 진행되고 있는지 최종 점검한다.

(2) 감리형태

감리형태는 프로젝트 환경, 사업규모에 따라 차이가 있지만 반복횟수 및 주기에 따라 달라질 수 있다. 단계별 감리 형태는 요구정의 및 릴리즈 계획이 완료되는 시점, 아키텍처/구축 단계, 종료 단계 3단계로 구분 실시가 적절 하지만 반복 주기에 따라 아키텍처/구축 단계는 차이가 있을 수 있다. 따라서 애자일 방법론 감리의 경우 단계별 감리와 상주 및 상시 감리를 병행 하여 감리를 수행하는 형태가 가장 적절하다.

3.1.2 감리영역

감리영역에는 시스템아키텍처, 응용시스템, 데이터베이스는 현행 정보시스템 감리 모형의 감리 영역을 동일하게 적용한다. 다만 애자일 프로세스에서 도출된 변화관리영역을 추가로 반영하였다. 감리시점은 프로젝트관리영역과 동일하게 착수/계획, 실행/통제, 종료로 구분하고 변화관리 활동으로는 변화관리 계획 수립, 변화관리 수행, 비즈니스 효과 및 성과 평가가 있다.

<Table 5> Check Lists deduced from the Change Management[15, 17]

Audit Point	Audit Domain	Activity	Remarks	Remarks
Launch / Plan	Change	Establishing plans for change management	Analyzing the interested parties, planning education, training, assigning change management, and other planning	Additional
	Management	Performing change management	Distinguishing/eliminating organizational obstacles according to the environment of Agile	Additional

			development and performing programmatic change management	
Termination		Evaluating effectiveness and performance of business	Measuring the team performance according to the Agile project measuring index	Additional

3.1.3 감리관점 및 점검기준

절차와 산출물은 정보시스템 감리기준과 동일한 기준을 적용할 수 있고 사업성과는 애자일의 특성을 반영하여 생산성을 추가하였다. 애자일은 자체적으로 조직하는 프로젝트팀에 의해 협업하고 지속적으로 고객과 끊임없이 의사소통한다. 이러한 팀은 각 스프린트마다 성공적인 산출물을 만들고 반복 계획과 추적, 팀 속도 평가 등을 통해 스프린트를 반성하고 다음 스프린트를 준비한다. 애자일에서의 생산성은 개발자, 스크럼 팀, 최종 제품의 릴리즈 시기를 초기부터 예측 가능하게 하는 중요한 사업성과 요소 중 하나이다[15].

따라서 감리 관점 별로 점검기준을 정리하였고 세부 속성은 아래 <Table 6>과 같다.

<Table 6> Audit Viewpoint and Check Criteria of Agile Model[11,15]

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/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 confidentiality 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

3.2 단계별 애자일 방법론 감리 점검항목

3.2.1 감리검토항목

기본점검항목은 현행 감리 점검기준을 기준으로 애자일 특징을 반영하였다. 어플리케이션, 데이터베이스, 시스템 아키텍처 영역을 준용하여 감리시점을 요구정의, 릴리즈 계획, 아키텍처/구축으로 제안하였다. 감리시점에 있어 기존 요구분석을 요구정의 명명하여 기본 점검 항목을 준용하였다. 다만 애자일 프로젝트의 초기 테스트의 중요성을 감안하여 사업 초기 단위/통합테스트 계획을 수립할 수 있도록 기본점검항목을 이동 하였고 요구사항을 분석하여 릴리즈 계획, 반복계획 수립을 위한 릴리즈 계획 단계를 추가 하였다. 또한 기존에 분석/설계/구현으로 구분되어 있던 기본점검항목들을 아키텍처/구축으로 통합하여 점검항목을 단순화하고 분석/설계/개발이 병행 진행되는 애자일 프로젝트의 품질을 점검할 수 있게 하였다. 다음 <Table 7>과 같이 응용시스템 영역의 기본점검항목을 제시하였다.

<Table 7> Check Lists of Application System in Agile Audit Method[11,13,14]

Audit Time	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

변화관리와 애자일 개발팀을 조합해 애자일 프로젝트의 성공적 수행을 위해 변화관리 영역을 추가 감리 영역으로 제안하고 각 단계에 따라 점검항목을 제안한다. 감리 시점은 프로젝트 관리 영역과 같이 착수/계획, 실행/통제, 종료 단계로 구분하였다. 착수/계획 단계는 변화관리계획 수립, 이해관계자 식별, 변화 환경에서의 지속적 교육 계획, 전담 인력 등이 적절하게 수립 되었는지가 중점 점검항목이고 실행/통제 단계에는 변화관리계획에 따라 적절하게 수행, 통제되고 있는지를 점검한다. 또한 종료단계에는 변화관리 수행에 따라 비즈니스 효과 및 성

과가 달성되었는가를 집중 점검한다. 아래 <Table 8>은 시점에 따라 점검항목을 제시하였다.

<Table 8> Check Lists of Change Management in Agile Method[15,17]

Audit Time	Activity	Check Items	Note
Launching Plan	Establishment of Alteration Management Plan	01. Is the alteration management plan appropriately established?	Additional
		02. Are the diverse related people distinguished and filled out a list?	Additional
		03. Are continuous scrum/agile education training plans based on repetitive developing environment appropriately established?	Additional
		04. Is there a specific group/manpower that will be responsible for the alteration management sector?	Additional
Implementation Regulation	Performance of Alteration Management	01. Is the alteration management appropriately performed based on alteration management plan?	Additional
		02. Did an activity to distinguishing main cause/elimination of group obstacle due to changes in agile developing environment performed?	Additional
Close	Effectiveness and evaluation in business	01. Was a measurement of team outcomes according to an agile project measurement indicator accurately performed?	Additional

3.2.2 애자일 주요 산출물의 세부 검토 항목

애자일 산출물은 전통적인 개발 방법론과 차이가 있다. 애자일 산출물의 특성과 스크럼을 통한 프로젝트의 관리 방식을 이해해야 감리 품질이 향상된다. 요구사항의 가시성과 추적성 확보 측면에서도 다양한 방법론과 테일러링 되어 적용 될 때 어떻게 요구사항 추적이 가능할지의 문제는 애자일 산출물을 검토하기 위한 체크 항목이 정의 되어야 한다. 예를 들어 애자일 프로젝트에서 요구정의 단계에서 사용자 스토리의 기반을 둔 요구사항을 정의하고 아키텍처/구축단계에서는 사용자스토리의 ID와 제품 백로그, 스프린트 백로그의 사용자 스토리를 매핑하고 어플리케이션, 데이터베이스의 설계 산출물인 화면정의서, ERD등과 연관성을 가지고 이후 테스트 시나리오와 연계하여 추적성을 확보케 하는 것이다.

이러한 세부 검토 방법을 위해 애자일 주요 산출물에

대하여 아래 <Table 9>에 세부 검토항목을 제시하였다.

<Table 9> Check Lists for Main Products of Agile AUDIT Method[13,14,15,16]

Product	Detailed Inspection Items	Note
Users' Story	Check if the request lists related to system (function, non-function, system interface) are elicited completely.	Additional
	By giving descriptions of testing items, check if the people in charge of designing/developing can understand detailed contents of request items and apply in practical use when implementing users' stories.	Additional
	Check if a priority in important work functions or common functions are understood to enable smooth management process.	Additional
Product backlog	Check if a list of total functions that the team will implement are included in the products' backlog.	Additional
	Check if the alteration items based on alteration lists and alteration request forms are included in the products' backlog.	Additional
	Check if the defect lists from completed users' stories are included in products' backlog.	Additional
	Check if the story point of users' stories are assessed in an appropriate level.	Additional
	Check if danger assesment was performed based on a danger priority ranking elicited from users' stories.	Additional
	Check if the users' stories are allocated in the sprint according to danger ranking or relations to implementations.	Additional
	Check if modified records for testing plans are described.	Existing
Test Plan	Check if an objective for total examination is described.	Existing
	Check if a testing performance group is clearly stated.	Existing
	Check if testing steps(unit, function, system, factors) are defined.	Existing
	Check if a statement about a testing environment is present.	Existing
	Check if a statement about the tools that would be used during testing is defined.	Existing
	Check if the testing schedule is stated.	Existing
	Check if detailed contents for each testing step are described.	Existing
	Check if checklists for unit testing are defined.	Existing
	Check for the presence of contents about scenarios for integrated testing.	Existing
	Check if there is a statement about procedures in build management for an	Existing

Product	Detailed Inspection Items	Note
	integrated testing	
	Check if there is a description about defect management (defect management steps, tools, etc.)	Existing
	Check if there is a description about progress management (Progress management period, targets, content, template, etc.)	Existing
Release Plan	Check if there is descriptions about test ceasing and resuming standard.	Existing
	Check if there is a release schedule in the backlog of products	Additional
	Check if stories for release are selected.	Additional
Sprint backlog	Check if users' stories in the products' backlog were estimated.	Additional
	Check if users' stories that a team will implement are included in the sprint's backlog.	Additional
	Check if a standard to complete users' stories are clear and accurate	Additional
	Check if an operation time exceeded a sprint period	Additional
	Check if a work is split in detail to complete work within 16 hours.	Additional
Sprint Review	Check if alteration items are included in the sprint's backlog in case modification items occur during sprint operation time.	Additional
	Check if customers' opinions through customers' participation are continuously applied	Additional
Extinct Chart	Check if informations in the product backlog or sprint backlog are accurately updated to apply in extinction charts.	Additional
	Check if sprint glance plans and records are managed based on users' story.	Additional

IV. 제안의 검증

설문조사의 대상은 <Table 10>과 같이 선정하였으며, 감리원(46.4%), 프로젝트 관리/사업관리/품질관리(7.14%), 개발자(17.9%), 분석 및 설계(14.3%), 데이터베이스/DBA/모델러(3.57%), 인프라/SA/TA/솔루션/기술지원(10.7%)로 구성되어 있다.

<Table 10> Results of a Survey on Understanding Characteristics of the Surveyers

	Auditor	PM	Developer	Analyser	DBA	TA	Total
number	13	2	5	4	1	3	28
percent	46.4	7.14	17.9	14.3	3.57	10.7	100

애자일 방법론을 적용하여 프로젝트를 수행한 경험을 묻는 질문에서는 응답자의 75%가 1 ~ 2회 애자일 방법론을 경험했으며, 감리원 중 25%는 애자일 방법론 프로젝트의 감리 경험이 없는 것으로 조사되었다.

<Table 11> Results of a Survey on Experiences of Agile Development Methods

Count number	None	1	2	3 ~ 4	above 5	Total
number	7	6	15	0	0	28
percent	25	21.4	53.6	0	0	100

애자일 방법론을 적용한 프로젝트의 감리 필요 여부에 대한 질의에 대하여 “필요하다”라고 응답한 경우가 100%로 애자일 방법론에 적합한 감리 의 중요성이 인식되고 있다.

<Table 12> Results of a Survey on Necessity of Auditing in Agile Audit Model

	Auditor	PM	Developer	Analysers	DBA	TA	Total
Necessary	13	2	5	4	1	3	28 100%
Unnecessary	0	0	0	0	0	0	0 0%

4.1 애자일 방법론을 적용한 프로젝트의 감리 점검 항목에 관한 설문결과

감리 점검 항목에 대한 설문은 3장에서 제안한 애자일 방법론 감리모형의 점검 항목 중 추가, 변경된 점검항목에 대하여 각 항목마다 매우필요(5점), 필요(4점), 보통(3점), 필요없다(2점), 전혀 필요없다(1점)의 값으로 표기하도록 하였다. 설문 결과에 따른 항목 적합성 검증은 보통을 기준으로 “매우필요·필요”는 적합으로, “필요없다·전혀 필요없다”는 부적합 의견으로 정리하였다.

4.1.1 응용시스템 기본점검항목에 관한 설문결과

애자일 감리 모델에서의 응용시스템 영역에서 감리점검항목에 대한 적합성을 묻는 설문 조사에서 적합 의견은 100%로 나타났으며, 설문결과는 아래 <Table 12>와 같다.

<Table 12> Results of Check Items in Application System

Check items	Appropriate	middle-ground	Inappropriate	Average	Standard Deviation
Checking whether test plan was set appropriately	28	0	0	4.54	0.50
Checking whether release planning was set appropriately by analyzing requirements	28	0	0	4.54	0.50
Setting repetitive plan and appropriate selection of story for the release	28	0	0	4.39	0.49
Sufficiency and completeness of work and detailed analysis of user's requirement/design/implementation of functions	28	0	0	4.50	0.50
Designing for convenience of user interface and its report	28	0	0	4.43	0.49
Sufficiency and appropriateness of refining analysis class and designing class in details	28	0	0	4.50	0.50
Sufficiency and appropriateness of interior/exterior interface analysis/design/implement	28	0	0	4.50	0.50
Appropriate analysis/ design/ implement of access privileges and control	28	0	0	4.39	0.49
08. Detailed design for introducing/implementing the component	28	0	0	4.39	0.56
Sprint review for each sprint	28	0	0	4.46	0.50
Average Response Rates	28 (100%)	0 (0%)	0 (0%)	4.46	0.50

4.1.2 변화관리 기본점검항목에 관한 설문결과

애자일 감리 모델에서의 변화관리 영역의 점검항목에 대한 적합성은 88.3 %로 높은 수준이며, 보통 의견이 8.2%로 설문결과는 아래 <Table 13>과 같다.

<Table 13> Results of Check Items in Change Management

Check items	Appropriate	middle-ground	Inappropriate	Average	Standard Deviation
Is the alteration management plan appropriately established?	25	2	1	4.29	0.75
Are the diverse related people distinguished and filled out a list?	23	4	1	4.21	0.77
Are continuous scrum/agile education training plans based on repetitive developing environment appropriately established?	24	3	1	4.21	0.77
Is there a specific group/manpower that will be responsible for the alteration management sector?	24	3	1	4.18	0.76
Is the alteration management appropriately performed based on alteration management plan?	27	1	0	4.43	0.56
Did an activity to distinguishing main cause/elimination of group obstacle due to changes in agile developing environment performed?	26	1	1	4.32	0.71
Was a measurement of team outcomes according to an agile project measurement indicator accurately performed?	26	2	0	4.32	0.60
Average Response Rates	25.0 (89.3%)	2.3 (8.2%)	2.5	4.28	0.70

4.1.3 애자일 산출물에 대한 세부 검토 항목에 관한 설문결과

애자일 감리 모델에서의 산출물에 대한 세부 검토 항목에 대한 적합성은 99.5 %로 높은 수준이며, 보통 의견이 0.5 %로 나타났으며, 설문결과는 아래 <Table 14>와 같다.

<표 14> Results of Check Items in Agile Documentation

Check items	Appropriate	middle-ground	Inappropriate	Average	Standard Deviation
Check if the request lists related to system (function, non-function, system interface) are elicited completely.	28	0	0	4.75	0.43
By giving descriptions of testing items, check if the people in charge of designing/developing can understand detailed contents of request items and apply in practical use when implementing users' stories.	28	0	0	4.71	0.45
Check if a priority in important work functions or common functions are understood to enable smooth management process.	28	0	0	4.68	0.47
Check if a list of total functions that the team will implement are included in the products' backlog.	28	0	0	4.75	0.43
Check if the alteration items based on alteration lists and alteration request forms are included in the products' backlog.	28	0	0	4.68	0.47
Check if the defect lists from completed users' stories are included in products' backlog.	28	0	0	4.61	0.49
Check if the story point of users' stories are assessed in an appropriate level.	28	0	0	4.61	0.49
Check if danger assesment was performed based on a danger priority ranking elicited from users' stories	28	0	0	4.71	0.45
Check if the users' stories are allocated in the sprint according to danger ranking or relations to implementations	28	0	0	4.68	0.47
Check if modified records for testing plans are described	28	0	0	4.68	0.47
Check if an objective for total examination is described.	28	0	0	4.64	0.48
Check if a testing performance group is clearly stated.	28	0	0	4.64	0.48
Check if testing steps(unit, function, system, factors) are defined.	28	0	0	4.93	0.26
Check if a statement about a testing environment is present.	28	0	0	4.93	0.26

Check items	Appropriate	middle-ground	Inappropriate	Average	Standard Deviation
Check if a statement about the tools that would be used during testing is defined.	27	1	0	4.86	0.44
Check if the testing schedule is stated.	27	1	0	4.82	0.47
Check if detailed contents for each testing step are described.	28	0	0	4.96	0.19
Check if checklists for unit testing are defined.	27	1	0	4.86	0.44
Check for the presence of contents about scenarios for integrated testing.	28	0	0	4.89	0.31
Check if there is a statement about procedures in build management for an integrated testing.	28	0	0	4.93	0.26
Average Response Rates	27.8 (99.5%)	0.15 (0.5%)	0	4.77	0.41

4.2 애자일 방법론 감리모형의 적합성 검증 종합

본 연구에서 제안한 애자일 방법론 감리 모형을 기반으로 필요성, 감리 시점, 감리 영역에 대한 설문 결과, 96%~100%에서 “필요함” 및 “있음”으로 조사되었다.

〈Table 15〉 Results of Agile methodology Project Audit

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%

점검 항목의 적합성에 관한 설문 조사 결과, 응용 시스템 기본점검항목은 100% 적합, 변화관리 기본점검항목은 89.3% 적합, 애자일 산출물 세부검토항목은 99.5%에서 “적합” 하다고 응답을 하였다.

〈Table 16〉 Total Results of Appropriateness in Agile methodology Audit Model

Classification	Items	Suitable	Average	Not Suitable
Check lists of Application System	10	100%	0%	0%
Check lists of Change Management	7	89.3%	8.2%	0%
Review lists of Agile products	20	99.5%	0.5%	0%

V. 결론 및 향후 과제

소프트웨어 개발은 더욱 복잡해지고 개발 초기에는 일반적으로 요구사항이 명확하지 않고 설계, 개발 단계로 이어지면서 많은 시행착오를 거치면서 정형화 된다. 이러한 문제점을 해결하기 위한 많은 방법 중의 하나로 최근 애자일이 조직에 맞게 제정되고 프로젝트에 적용되고 있다. 따라서 정보시스템 감리에서도 애자일 개발 프로젝트의 품질을 보증하기 위한 적합한 감리 모형이 필요하다.

본 연구에서는 국내 적용되고 있는 애자일 방법론과 현행 감리 모형의 비교분석을 통해 애자일 특성에 맞는 감리 모형을 제안하였다. 감리 시점은 프로젝트의 특성에 따른 반복 주기 및 형태를 분석하여 릴리즈 및 반복계획이 수립되는 시점과 반복이 종료되어 평가 되는 시점, 이행시점으로 감리를 수행하고 상시 감리의 강화가 요구된다. 감리 점검 항목은 분석, 설계, 개발단계에 걸쳐 통합 구성하였고 주요 애자일 산출물에 대한 세부 검토 항목도 추가 하였다. 또한 애자일 환경에서의 변화관리의 실행 여부를 검토하기 위한 점검항목을 반영하였다.

본 연구는 대규모 IT 조직의 애자일 방법론과 현행 감리 모형의 비교 분석을 중심으로 애자일 방법론 감리 모형을 제시하였다. 이를 통하여 정보시스템 감리에 애자일의 특징 적인 영역과 점검항목을 적용 하였으나 다음과 같은 한계점을 지니고 있다.

첫째, 애자일 방법론 감리 모형의 검증을 위해 설문조사와 애자일 프로젝트의 감리 사례를 통해 검증을 하였으나 연구의 기초 자료가 되는 설문조사의 표본대상이 부족하였다. 애자일 프로젝트 수행 및 감리 경험을 지닌

대상을 추출하기가 어려웠다.

둘째, 실무에 적용되고 있는 다양한 애자일 실천 항목들을 지속적으로 수집, 분석하여 애자일 특성에 맞는 점검항목을 애자일 방법론 감리 모형에 추가적으로 반영할 필요가 있다.

소프트웨어 품질 보증을 위해 애자일은 다양한 형태의 방법론과 병행되어 각각의 장점을 결합한 형태로 적용, 진화 할 것으로 예상되고 정보시스템 감리도 이러한 환경에 신속한 대응이 가능토록 끊임없는 연구가 필요할 것으로 판단된다.

ACKNOWLEDGMENTS

This paper was supported by the Shamyook University Research Fund in 2012.

REFERENCES

- [1] Chun Hee Kang, A study of commentary documentation for Agile software methodology, Graduate School of Information Communication of Sogang University, 2006.
- [2] Dongsoo Kim, Seungryul Kim, Developing A Quantitative Evaluation Model on System Analysis Phase for Object oriented Information System, Journal of the Korean Society for Industrial and Applied Mathematics, 7(1), pp.87-105, 2003
- [3] Eun Suk Park, A study on application of the CMMI for web service product development process using agile method : focused on scrum-based process improvement practices, Korea University, Graduate School of Computer & Information Technology, Master of Engineering, 2010.
- [4] Bum Ho Baek, Progress Analysis and Quality Prediction Method for Agile Software Development, Graduate School of Sogang University. Master of Engineering, 2009.
- [5] Hee Kyung You, Analysis of the agile method effect on ERP project in university, Hanyang University Graduate School of Engineering, Master of Engineering, 2011.
- [6] Chung Yoon, Configuration management Method and Function Improvement for Agile Software Development, Graduate School of Chungnam University, Master of Engineering, 2005.
- [7] Suk Jin Yoon, Application of XP practical techniques for small and medium-size web project, Graduate School of Information Communication of Sungkyunkwan University, Master of Engineering, 2011.
- [8] Sung Ho Lee, Jin Tak Choi, Dong Soo Kim, Hee Wan Kim, The Model of Information System Operating Audit for the Service Level Agreement, The Journal of Digital Policy & Management, 10(6), pp.71-82, 2012.
- [9] Mi Jung Choi, A study on the application effect and improvement of agile methodology for project management, Hanyang University Graduate School of Engineering, Master of Engineering, 2011.
- [10] Ministry of Security and Public Administration, Information System Audit Standard Revision, 2010.
- [11] National Information Society Agency, The Commentary on the Information System Audit Standards v3.0, National Information Society Agency, 2009.
- [12] National Information Society Agency, The Commentary on the Information System Audit Standards v1.0, National Information Society Agency, 2009.
- [13] Company A, Agile Process Definition V1.1, 2011.
- [14] Company B, Agile Methodology, 2011.
- [15] Dean Leffingwell, Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise, Addison-Wesley Professional, 2011
- [16] Henrik Kniberg, Scrum and XP from the Trenches, Lulu.com, 2007
- [17] Mel Beckman, Connection of agile methods in change managements, CIO Korea, 2011
- [18] <http://agilemanifesto.org/iso/ko/>

김 동 협(Kim, Dong Hyup)



- 2012년 2월 : 건국대학교 정보통신대학원(공학석사)
- 1996년 2월 ~ 2008년 7월 : (주)대교 CNS 과장
- 2008년 8월 ~ 현재 : 삼성SDS Delivery 혁신본부 수석
- 관심분야 : 정보시스템 감리, 프로젝트 관리, 소프트웨어 공학

· E-Mail : hyupdong.kim@samsung.com

김 희 완(Kim, Hee Wan)



- 1995년 8월 : 성균관대학교 정보공학과(공학석사)
- 2002년 2월 : 성균관대학교 컴퓨터공학과(공학박사)
- 1996년 5월 : 정보관리기술사 취득
- 1997년 1월 : 정보시스템 수석감리원 자격 취득
- 2001년 3월 ~ 현재 : 삼육대학교 컴퓨터학부 교수
- 관심분야 : 정보시스템 감리, 프로젝트 관리, 데이터베이스, 소프트웨어 공학

· E-Mail : hwkim@syu.ac.kr

고 찬(Koh Chan)



- 1991년 2월 : 경희대학교 전자공학과(공학박사)
- 2008년 2월 : 서울대학교 기술정책학 전공(경제학박사)
- 1992년 3월 ~ 1993년 2월 : 미국 North Carolina State University 컴퓨터학과 Post Doc.

- 2005년 1월 ~ 현재 : 핀란드 Hmuas 대학교 겸임교수
- 1983년 3월 ~ 현재 : 서울과학기술대학교 교수
- 관심분야 : 기술정책, 문화정책, 경영정보학, 컴퓨터 그래픽스&게임디자인

· E-Mail : chankoh@seoultech.ac.kr

김 동 수(Kim, Dong Soo)



- 1981년 2월 : 광운대학교 전자계산학과(이학사)
- 2001년 2월 : 서울산업대학교 전자계산학과(공학석사)
- 2005년 2월 : 국민대학교 경영정보학과(경영학박사)
- 1991년 12월 : 전자계산조직응용기술사 취득

- 1995년 8월 : 정보통신기술사 취득
- 1998년 2월~현재 : (주)키사 대포컨설턴트
- 2008년 3월~현재 : 건국대학교 정보통신대학원 겸임교수
- 관심분야 : u_city 감리, 프로젝트 관리, 정보시스템 감리, 소프트웨어 공학

· E-Mail : dskim@kisac.co.kr