

애자일 방법론을 적용한 정보시스템의 감리 품질 향상에 관한 연구

박동아[†], 박만곤^{**}

A Study on the Quality Improvement of Information System Auditing for Agile Methodology

Dong-Ah Park[†], Man-Gon Park^{**}

ABSTRACT

In case of auditing of the information system development project applying agile methodology, it is not appropriate to carry out a comprehensive check on the establishment of information system with only the existing check on software. This study considers the characteristics of the agile methodology in terms of Information System Auditing. To improve inspection quality of development project with agile methodology by deriving detailed check items of test activities at each stage, this study proposes a strategy to improve the check on software for the test activities of the supervisory model that is suitable for agile methodology, which emphasizes repetitive work.

Key words: Agile; Information System Auditing; Auditor; Information System Developing; System Integration

1. 서 론

정보화 사회로 급변에 따른 고객 요구사항에 최적화된 서비스를 제공하고자 수많은 정보화관련 프로젝트가 수행되고 있으며 각 프로젝트는 개별적 상황과 특징에 따라 적절한 개발방법론이 적용되고 있다. 폭포수모델과 같은 전통적 개발방법론은 논리적이고 구조적으로 정형화되어 고객의 요구사항을 적시에 제공하고, 고품질 소프트웨어 제품을 제공해왔다. 그러나 다양한 요구사항에 효과적인 대응에는 한계가 존재하여 애자일 방법론(Agile Methodology)과 같은 새로운 개발방법론이 제시되었다. 고객의 다양한 요구에 대응하기 위해 공정과 도구보다 개발자간

의 상호작용, 포괄적인 문서보다는 작동하는 소프트웨어를, 계약 협상보다는 고객과의 협력을, 계획을 따르기 보다는 변화 대응에 더 많은 가치를 두며 정보시스템 개발 프로젝트에서 두각을 나타내고 있다 [1]. 그렇지만 애자일 방법론을 적용한 시스템 개발 프로젝트에 대한 정보시스템 감리 수행 시 기존 감리 체계에서 제공되는 점검 항목만으로 정보시스템 구축에 대한 종합적인 점검을 실시하기 적합하지 않다. 본 연구는 정보시스템 감리측면에서 애자일 방법론에 대한 특징을 고찰하고, 각 단계별 테스트 활동에 대한 감리 점검항목을 기존 항목에서 미비한 점을 추가적으로 정의하여 애자일 방법론이 적용된 정보시스템 개발 프로젝트의 감리 에 대해 품질향상 기법

※ Corresponding Author: Man-Gon Park, Address: (48513) Yongso-Ro 45, Nam-Gu, Busan, Rep. of Korea, TEL: +82- 51-629-6240, FAX: +82-51-629-6230, E-mail: mpark@pknu.ac.kr

Receipt date: Dec. 12, 2016, Revision date: Mar. 21, 2017
Approval date: Apr. 3, 2017

[†] Dept. of Information Systems, Pukyong Nat. Univ., Rep. of Korea, E-mail: zenoparkk@naver.com

^{**} Dept. of IT Convergence and Application Engineering, PuKyong Nat. Univ., Rep. of Korea, mpark@pknu.ac.kr

을 제안하고자 한다. 본 연구의 구성은 개발방법론에 대한 문헌적인 비교분석과 애자일의 특징 및 장점에 대해서 기술하고 정보시스템 개발 프로젝트의 테스트활동 중요성과 애자일이 적용된 감리 체계에 대한 개선모형과 점검항목에 대한 개선 방안을 제시하였다.

2. 개발방법론 비교분석

2.1 개발방법론의 변천과정

현대 사회 발전 속도에 비례하여 소프트웨어 수요 증가, 정보시스템 복잡화 및 다양화, 높은 신뢰도가 필요한 소프트웨어 사용자를 만족시키기 위한 다양한 소프트웨어 개발방법론 (Software Development Methodology)이 적용되고 있으며, 이러한 소프트웨어 개발방법론은 정보화 프로젝트 관리의 필수 사항으로 인식되고 있다[2]. 1950년대부터 적용되기 시작한 논리 중심의 기능과 데이터 흐름을 정의하는 단순한 방법론에서 현재의 애자일 방법론까지 발전해 왔다. 1950년대는 비구조적, 무원칙적 상향식개발 방식으로, System flowchart, Flow diagrams 등의 기법이 적용된 수작업 분석 기법이었고[3], 1960년대부터 소프트웨어 개발방법론의 형식이 완성되었으며, 1970년대는 분석/설계를 정형화한 폭포수와 같은 구조적 방법론, 1980년대는 데이터의 중요성을 강조한 정보공학적 방법론, 1990년대는 객체를 부각시킨 객체지향 방법론이 적용되었다[4]. Table 1에서처럼, 이후

2000년대부터 소프트웨어 컴포넌트 조합으로 재사용을 중요 이슈로 부각시킨 CBD(Component Based Development)기반 방법론이 제시되었으며, 사용자와의 지속적인 의견교환으로 반복적 개발을 중요시한 애자일(Agile)기반 개발방법론까지 나타났다[5].

2.2 개발방법론의 특·장점

폭포수 개발 방식은 계획단계부터 요구사항 분석과 프로그램 설계, 구현 및 테스트단계를 상세하게 문서화한다. 각 개발 단계별 작업 시간을 추정, 합산하여 전체 개발시간을 산정하고, 이해당사자들의 검토와 승인 후 응용프로그램을 개발 및 구현하고, 고객 인수에 필요한 테스트를 진행한다. 구조적 방법론이란 시스템 개발과정을 보다 체계적으로 지원하는 방법론으로, 개발 프로젝트를 효과적으로 진행하고, 유지보수가 쉬운 시스템 개발 방법이다. 구조적 방법론은 초기 단계에서 수집된 자료를 활용, 구조적 명세서를 작성하고 전체 시스템을 부분 시스템으로 분할한다. 이는 사용자의 요구사항을 정확히 파악하기 위한 객관적 측정기준에 따른 하향식 개발(Top-down)이며, 대상 시스템의 분석, 설계 및 코딩의 가시성을 제공한다[7]. 정보공학적 방법론은 비즈니스 관점에서 정보시스템 개발을 위해 공학적 방법을 접목한 것이다. 구조적 개발방법론이 응용프로그램 중심의 개발 관리가 목적이었다면, 정보공학적 개발방법론은 기능 위주의 다양한 정보를 동시에 처리하는

Table 1. Evolution of Software Development Methodology

Classified Methodology	Structured	Information Engineering	Object-Oriented	CBD
Period	1970s	1980s	1990s	Since 2000
Implementation Techniques	Function	Data Structure	Object-Oriented	Component
Content	Abstraction Structured	Operation area analysis; System design & implementation	Requirement definition Object oriented analysis & design	Needs Analysis Modeling of case Design development
Advantage	Function deployment program	For Data	For Reuse	Improving productivity, quality, costs & risks
Disadvantage	Unable to data hide; Low performance reuse & maintenance	Functional design of the application; Low of reuse features.	Basic Software technology needs.	Component distribution environment should be improved

데이터 중심 방법론이다. 절차적 관점보다, 비즈니스 측면에서 어떤 정보를 어떻게 처리할 것인지에 보다 높은 가치를 차지하고 있다. DFD, ERD로 표현하고 분할, 계획, 분석, 개발 및 운영의 명확한 구조적기반으로 정의되며, 정보전략 계획수립, 업무영역분석, 시스템설계 및 시스템구축 단계로 구분된다. 장점은 일관성 있고 통일된 시스템 구축이 가능하고, 데이터 중심업무절차 및 환경변화에 유연한 대처가 가능하다. 객체지향 방법론은 복잡한 구조의 현실 세계를 인간이 이해하는 방식으로 시스템에 적용하고자, 객체(object)와 객체들의 범주를 나타내는 클래스(class), 객체간의 상호작용을 위한 메시지(message)를 기본 모형으로 제시하였다[8,9]. 객체지향 방법론은 구조적 방법론과 정보공학 방법론과는 완전히 다른 프레임워크를 가지고 있다. 해당 분야의 주요 객체를 파악 후, 그 객체에 기능적 절차를 맞춰 나가며, 요구사항이 변경되어도 비교적 안정적인 수정 및 보완이 가능하다. 차이점은 정보공학적 방법론이 반복적이고 순차적인 분석, 설계, 구현 단계에서 데이터와 프로세스를 분리하여 구현되는 것과는 다르게 반복과 점진적으로 데이터와 데이터를 조작하는 함수를 객체로 묶어 처리한다. 컴포넌트 기반 개발 방법론은 소프트웨어 컴포넌트 조합으로, 빠르게 변화하는 비즈니스 프로세스에 적극적으로 대처하고, 소프트웨어 개발 생산성을 향상 시켰다[10]. 여러 개의 컴포넌트로 분할하여 각 컴포넌트에 비교적 간단한 고유 기능을 부여하고, 이들을 하나의 응용프로그램에 통합하여 재 사용성과 유지보수 편의성을 확보하고 각각의 컴포넌트는 단 하나의 응용프로그램에서만 사용되는 것이 아니라, 컴포넌트가 제공하는 고유한 기능이 필요한 다른 여러 응용프로그램에 재사용되므로 코딩 작업량을 감소시켰다. 각각의 컴포넌트는 고유한 기능을 제공하는 간단한 코드형식으로 그 기능을 변경시키는 작업은 쉬워졌고, 조그만 기능 개선을 위해 전체 응용프로그램을 컴파일하고 배포하는 절차를 생략하게 되었다. 이는 신속성, 용이성, 플랫폼 독립성에 의한 개발 원가절감, 생산성과 품질을 향상 시켰다[11]. 마지막으로 애자일 방법론은 계획 중심의 개발방법론을 경량화 시킨 모델로, 개발 과정의 요구사항 변경에 유연하고, 민첩하게 대응하며, 불완전한 사용자와 개발자 사이의 의사소통의 애로사항을 해결하고자 개발된 방법론으로 기존 모델을 거부

하는 것이 아니라 이를 적극적으로 수용하되, 각각의 개발 단계를 짧게 분해하여 의사소통을 향상시키는 것이 핵심사항이다.

2.3 애자일 방법론의 특징 및 종류

애자일 방법론은 기민하고 빠른 변화를 수용하는 프레임워크, 태도 및 접근법으로 사용자의 업무요구에 효율적 및 능동적으로 대응한다[12]. 애자일 방법론의 핵심 가치는 2001년 Snowbird에서 개최된 회의 결과로 만들어진 애자일 선언문(The Manifesto for Agile Software Development)에 나타나 있다. 요구사항의 변화를 수용하여 실제로 동작하는 소프트웨어를 끊임없이 제공하고 소프트웨어를 개발할 때 중요한 것을 최우선 실행하는 것이 가장 중요한 항목이라고 언급하고 있다[1]. 애자일 방법론의 정의를 살펴보면 급변하는 고객 요구에 전통적인 소프트웨어 개발 방법론이 합리적으로 대응하지 못한 것으로부터 시작하여, 변화되는 업무요구사항에 능동적으로 대응하는 능력을 말하며, 궁극적으로 변화를 수용하고 절차보다는 사람중심으로 변화에 유연하고 신속하게 적용하면서 효율적으로 시스템을 개발하는 것이다. 애자일 방법론은 과거 방법론의 위험 및 실패 요소를 바탕으로 더 효과적인 프로젝트를 수행하기 위해 만들어졌으며 기본적으로 소프트웨어를 일정에 맞게 개발하여 비즈니스에 적용하는 것이며[13], 개발자로 하여금 프로젝트에 능동적으로 대응하도록 하며, 민첩하게 사용자의 요구사항에 대응하고 개발자 상호간의 의사소통을 원활하게 하여 고품질의 제품이 출시되도록 개발에만 몰두해야 한다. 이러한 애자일 방법론은 역동적인 요구사항들, 책임감 있고 의욕이 많은 개발자들과, 이해력이 높은 고객들을 통해 조직은 시스템의 특성이나 개발 팀의 특성에 따라 정보시스템 개발을 위한 다양한 접근방법이 필요하거나, 규모가 큰 프로젝트 보다는 소규모 프로젝트에 보다 적합하고, 특히 안전이 중요한 시스템의 경우 최상의 효과를 나타낸다. 또한 목표시스템의 변동성이 높은 경우[14], 목표는 신뢰성이 높은 소프트웨어를 빨리 제작하는 것에 최적의 결과를 보여준다. 현재까지 정보시스템 개발에 적용된 주요 애자일 방법론을 정리하면 다음의 Table 2 및 Fig. 1과 같다.

스크럼(Scrum)은 1986년 Hirotaka Takeuchi와 Ikujiro Nonaka가 최초 적용하였으며, 제품 개발 프

Table 2. Type and Characteristic of Agile Methodology

Type	Characteristic	Advocate
Scrum	an iterative and incremental agile software development framework for managing product development	Ken Schwaber Jeff Sutherland
eXtreme Programming (XP)	improve software quality and responsiveness to changing customer requirements, 12 practice	Kent Beck Erich Gamma
Scrumban	hybrids of Scrum and Kanban	Ladas, Corey
Kanban	focuses on the customer and work which meets their needs, rather than individuals' activities with a kanban board	David Anderson Don Reinertsen
Lean Development	the traditional lean principles in a modified form, as well as a set of 22 tools and compares the tools to agile practices	Marry & Tom Poppendieck
Dynamic System Developing Method (DSDM)	provide some discipline to the rapid application development (RAD) method	Dane Falkner
Feature-Driven Development (FDD)	a model-driven short-iteration process that consists of five basic activities	Peter Coad Jeff De Luca

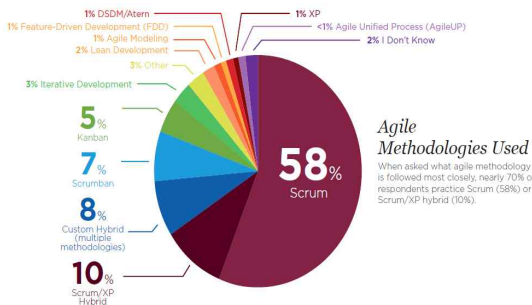


Fig. 1. Agile Methodologies Used.

로세스 유형을 가리키는 용어로 사용되며, 특징은 요구사항을 수렴한 후에 가장 중요한 요구사항기능의 개발 및 테스트를 위해 스프린트를 2~4주기로 반복적 가동시킨다. 스프린트(Sprint)는 고객과 함께 검토하고 피드백 사항을 재개발하며 요구사항이 충족될 때 까지 반복 수행된다[17]. 스프린트 계획회의(Sprint Planning Meeting)를 통하여 전체 요구사항을 제품 백로그(Product Backlog)에 설정 후 우선순위가 높은 요구사항을 백로그종목(Backlog Item)으로 선정, 2~4주 단위의 소규모 스프린트를 반복적으로 진행하고 일일 스크럼 미팅(Daily Scrum Meeting)을 통하여 점검한다. 스프린트의 결과물은 작동되는 프로그램으로 스프린트 반복을 통하여 최종 제품이 완성된다[4].

익스트림 프로그래밍 (eXtreme Programming, 이하 XP)은 1996년 Chrysler Comprehensive Compensation System Payroll Project의 PM이었던

Kent Beck이 처음으로 적용한 방법론으로, XP는 주로 엔지니어링 개발 프로세스에 초점을 두고 효과적이고 단순한 개발방법을 적용하여 개발일정을 단축시키는데 목적을 두고 있다. XP은 소규모 개발인원이 현장에서 고객과 함께 개발을 진행하며, 순차적으로 사용자 요구사항이 반영된 프로그램을 릴리스와 인도의 반복(Iteration)절차를 수행한다. 요구사항은 사용자가 요구하는 새로운 기능들을 보여주는 유저 스토리(User Story)로 구체화 과정을 거쳐, 2명의 프로그래머가 공동 작업(Pair Programming)을 하고 코딩표준에 따른 단위테스트를 수행하며, 요구사항, 아키텍처, 설계는 프로젝트 진척에 따라 진행된다. XP는 10명 이하의 개발자 구성이 가장 적절하며 고객은 이 팀과 함께 작업하거나 근접장소에서 업무를 처리한다.

2.4 정보시스템 테스트활동

정보시스템 테스트활동은 개발된 프로그램을 작동시켜 요구정의사항의 기능과 비교, 기능의 정상적 실행 여부를 확인하는 과정이다[19]. 테스트 활동은 IEEE 정의에 의하면 소프트웨어와 관련된 프로젝트 계획, 설계문서, 코드, 테스트 사례 사용자 매뉴얼 등을 개발하고 유지 보수하기 위하여 사용하는 활동, 방법, 기법 등이 일련의 순서를 가지는 집합으로 정의하고 있다[10]. 이러한 전체 테스트활동 과정 중 가장 비중 있는 부분은 테스트 실행 단계이지만, 효율적인 실행을 위해서는 테스트를 계획하고, 다양한

테스트케이스 설계와 준비 테스트 진행 상태를 확인하고 평가하는 활동을 선행해야 한다. 정보시스템 테스트활동의 특징은 첫째, 개발과정에서 소프트웨어의 결함을 발견하고 결함을 찾아내기 위해서 가능한 많은 장애(Failure)상황을 만들어 내면서, 둘째, 요구 정의된 결과로 시스템이 동작하는지 확인하고, 요건에 맞는지 확인하는 과정이며, 셋째, 특정시간에 시스템 개발을 종료하기 위한 여러 위험을 개발 프로젝트 책임자에게 전달하고, 넷째, 개발과정에서 변경작업이 일어나는 경우 새로운 결함이 발생하였는지 확인하는 회귀테스트활동(Regression testing)과정을 포함하며, 다섯째, 신뢰성 및 가용성과 같은 시스템의 특성을 평가하는 것이다. 단계별 테스트활동은 목적, 대상, 수행 주체 등을 고려하여 효율적인 테스트활동이 되어야하고, 종류는 단위테스트, 통합테스트, 시스템테스트, 인수테스트 등이 있으며, 각 단계별 각기 다른 테스트계획으로 실시된다. 테스트활동은 별도의 테스트설계, 실행, 완료 및 보고, 마감활동과 테스트팀에 의해 관리되며, 각기 다른 테스트 활동 단계들의 일반적인 목표와 테스트 케이스 도출에 필요한 개발 입력 산출물, 테스트 대상 및 발견되는 결함과 장애, 테스트 도구 및 테스트 접근법이 존재한다.

단위테스트는 최소 단위의 모듈을 작동시키는 테스트를 말하며, 단위 프로그램 마다 분할적으로 테스트함으로써 결함을 발견한다. 단위 테스트는 프로그램 개발코드 분석부터 시작하는 화이트 박스 테스트와 시스템 및 컴포넌트, 개발 프로그램 내부 구조에 대한 사전 지식 없이, 요구사항의 명세서에 따른 기능의 정상작동 여부를 파악하는 블랙박스 테스트 기법이 있다[20,21]. 통합테스트는 컴포넌트 또는 단위 모듈 사이의 인터페이스를 테스트하는 것으로, 시스템을 구성하는 각 단위 모듈을 종합적으로 테스트하는 것으로, 시스템 요구사항 명세서 기능의 실행여부와 모듈간 인터페이스를 테스트하는 것이 주 목적이며, 또한 프로그램 구성요소 사이의 인터페이스 테스트를 수행 후 메모리 누출문제 등 시스템관련 분석을 실시하며, 운영체제시스템, 파일시스템, 하드웨어 시스템의 다른 부분과 원활한 작동여부를 테스트한다[9]. 시스템테스트는 통합테스트를 실시 후 소프트웨어의 성능 및 부하에 대한 테스트를 실행하고, 개발 프로젝트 차원의 전체 시스템이나 제품의 정상적인

작동여부를 확인한다[22]. 시스템테스트는 실제 최종 사용자환경과 동일하거나 유사한 환경에서 실시한다. 인수테스트는 개발된 프로그램을 사용할 실제 사용자 또는 관계자에 의하여 실시되며, 목적은 시스템 전체 또는 일부의 기능적 및 비기능적 요소를 총체적으로 점검하여, 고객이 시스템을 사용가능한지 여부를 평가하는 것이다. 인수테스트가 성공적으로 수행되어도 다른 종류의 시스템테스트 및 통합테스트가 실행될 수 있으므로 인수테스트가 테스트의 최종 단계는 아니다.

2.5 애자일 프로젝트의 테스트활동 감리의 필요성

테스트활동은 소프트웨어 개발프로젝트에 대한 감리 수행 시 기능점검표를 통하여 사용자가 요구한 응용프로그램 및 시스템의 동작과 성능이 수행되는지 확인하기 위한 감리의 중요한 부분이다. 정보시스템 감리관점에서 테스트활동은 시스템 사용자의 요구사항에 맞게 구현되고 기대수준에 만족여부를 확인하고 이를 통해 최종적으로 결함 발견하고 이를 정량적 값으로 소프트웨어 개발 프로젝트의 완료여부와 관련된 정보를 감리의뢰인에게 보고하는 것이다. 개발 프로젝트 전 과정에서 각종 개발 산출물을 테스트활동 관점에서 검토하고, 테스트를 통한 각종 결함 발견을 매우 중요하게 생각하고 있다.

현행 정보시스템 감리점검해설서 V3.0의 'SD-14-1' 부분에서 테스트활동에 대한 각종 검토항목, 주요검토 대상 산출물 및 검토내용에 대해서 언급하고 있지만[25], 이는 애자일 방법론과 다른 특성을 가지고 있는 구조적개발방법론 등에 최적화된 감리 점검해설로 통합테스트와 시스템테스트에 대한 활동만 규정되어 있을 뿐, 애자일 방법론에 대한 테스트활동은 일체 누락되어져있다.

애자일 방법론은 각 스프린트마다 유저스토리에 대한 기능구현시점에 테스트활동이 수행된다. 현행 다른 개발방법론이 적용된 프로젝트에 대한 감리 시기는 요구정의단계, 설계단계, 종료단계 등 시점별 단계별로 수행되므로, 이를 애자일 방법론에 적용할 경우 감리 시점에서 종료된 스프린트만 테스트 가능하며, 해당 스프린트의 경우에는 구현 단계에 대한 점검사항을 포함하여 감리가 수행될 필요가 있다. 또한 소명차트, 백로그, 유저스토리 등 애자일 방법론만이 가지는 특유의 산출물을 점검항목에 추가해야

한다. 제한적으로 시스템 관련 개발영역은 현재의 감리 점검항목을 적용할 수 있지만, 테스트영역의 경우 개발 프로젝트의 총체적 점검을 위해 애자일 방법론에 특화된 테스트 점검항목 개선은 반드시 필요하다.

3. 애자일 방법론에 최적화한 감리기법

반복적으로 설계와 구현을 실행하는 애자일 방법론에 대한 특징 및 주요 절차를 분석하여 테스트활동과 관련된 개선된 감리점검항목을 제안하고자 한다. 현행 테스트활동 점검영역인 단위테스트, 통합테스트, 시스템테스트, 인수테스트 등 일련의 테스트과정에 애자일 방법론의 감리 영역인 요구정의, 릴리즈계획, 아키텍처/구축, 테스트, 운영에 맞게 단계별 테스트활동을 단위테스트, 인수테스트, 통합테스트, 시스템테스트로 제한하며, 시스템테스트단계에서는 성능테스트 및 부하테스트를 추가로 제시하였다.

3.1 애자일 방법론 테스트활동에 대한 개선된 감리 모형

현행 감리 지침은 요구분석, 분석·설계, 구현, 테스트, 전개에 따른 단계별 테스트활동을 계획하고 수행한다. 하지만 애자일 프로젝트는 짧은 주기의 반복적 개발로 기존의 감리기법을 적용하기에 다소 부적절하다. 선행 연구에서 현행 정보시스템 감리모형을 이용하여 보다 애자일 방법론이 적용된 프로젝트에 적합하도록 제안하였다[23]. Table 3의 제시된 애자일 방법론을 적용한 감리방안에서 애자일 방법론 감리의 테스트활동의 경우 현행 감리모형의 테스트활동을 동일하게 적용하여 반복적 개발로 진행되는 애자일 방법론의 테스트활동에 대한 감리활동으로는 다소 부적절하다.

본 연구의 테스트활동이 강화된 감리모형은 Dean

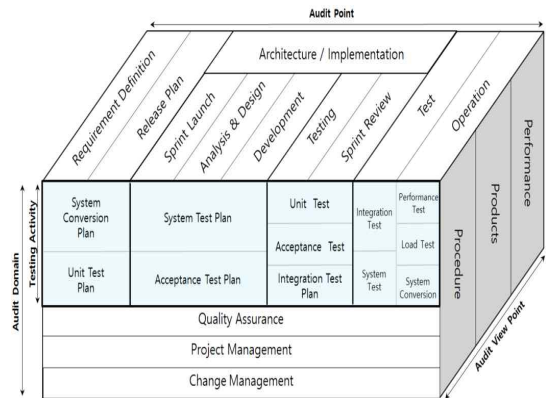


Fig. 2. System Audit Model for Agile.

Leffingwell의 단계별 테스트활동인 단위테스트, 인수테스트, 통합테스트, 시스템테스트(성능테스트·부하테스트)를 기반으로 적용하였다[24]. 반복되는 스프린트마다 유저스토리에 대한 기능이 구현되므로 단위테스트 이후 인수테스트를 제시하였다. 감리시점 영역에 따른 단계별 테스트활동으로 요구사항정의·릴리즈계획에서는 시스템전환계획, 단위테스트계획을 아키텍처/구축 단계의 스프린트착수 및 분석·설계·개발에서는 시스템테스트계획, 인수테스트계획을 테스트·스프린트종료 단계에서는 단위테스트, 인수테스트, 통합테스트계획을 테스트단계에서는 통합테스트, 시스템테스트를 운영단계에서는 성능테스트, 부하테스트 및 시스템전환을 수행하도록 제시하였다.

선행 연구된 애자일 기반 감리모형[23]과 정보시스템 감리점검해설서[25]를 참조하여 반복적인 릴리즈와 개발 프로세스에 대한 테스트활동을 강화한 개선된 감리모형을 Fig. 2에서 제시하였다. 일반적으로 성능테스트는 인수테스트단계에서 수행되던 절차이지만 반복적 개발인 애자일 방법론에서는 인수테스

Table 3. Current and Agile Method Audit Model

Class		Step of Audit						
Current	Requirement Analysis		Analysis & Design		Implementation			Deployment
	Agile	Req. Define	Release Plan	Architecture / Implementation				
SL		A&D	Dev		SR			

SL: Sprint Launch, A&D: Analysis and Design
Dev: Development, SR: Sprint Review

트에서 실행된 성능테스트를 통합테스트 이전에 실행하는 것은 무의미하므로 통합테스트 이후 성능테스트 및 부하테스트를 실행하면 프로그램에 대한 신뢰성 및 결함 발견율을 향상시킬 수 있다. 그리고 부하테스트는 전체 시스템의 부하를 점진적으로 증가시켜 어느 수준의 부하에서 시스템이 오작동 여부를 점검하여 최종적으로 시스템 전환계획을 수립할 수 있도록 하였다.

3.2 애자일 방법론 테스트활동에 대한 개선된 감리 점검항목

테스트활동에 대한 감리는 정보시스템 감리 점검 사항 중 소프트웨어의 품질을 점검하는 단계이므로 정보시스템의 신뢰성을 보증하는 중요한 활동이다. 애자일 방법론의 테스트활동에 대한 개선된 감리 점검항목은 현행 정보시스템 감리점검해설서를 기준으로 애자일 방법론만의 특징을 반영하였다[25].

릴리즈계획 단계에서는 애자일 방법론의 핵심인 반복(Iteration)에 적합한 요구사항이 유저스토리(User Story)로 충분히 표현되고 스프린트(Sprint)별 반복 계획 수립여부를 확인하기 위해 단위테스트 계획과 시스템전환계획을 제시하였다. 기존 감리 점검 해설서의 점검 항목인 ①개선모델로 전환계획의 수립 ②단위테스트 계획의 수립의 내용을 포함하고 ①테스트케이스 설계 및 테스트 우선순위 결정여부 ②테스트케이스에 필요한 테스트데이터 포함여부 ③요구사항의 유저스토리로 상세한 표현여부 ④스프린트별 반복계획 수립의 적절성 여부를 추가적으로 제시하였다.

스프린트착수/분석·설계/구현 단계에서는 아키텍처/구축의 스프린트착수/분석·설계단계에 해당하는 시스템테스트계획과 각각의 계획된 스프린트별 유저스토리의 기능구현종료시점에 프로그램 사용자가 직접 인수테스트를 할 수 있도록 인수테스트계획을 제시하였다. 점검 항목으로는 기존 해설서의 사용자 인수테스트에 대한 계획의 수립여부를 포함하고 ①시스템구축을 위한 테스트 계획 수립여부 ②기능구현 종료시점별 사용자가 인수테스트를 수행할 개별적인 스프린트 유저스토리 수립여부 ③스프린트별 반복계획 수립의 적절성 여부를 추가하였다.

테스팅/스프린트리뷰 단계에서는 스프린트가 완료되면 모든 유저스토리가 단위테스트를 통과하고

결함에 대한 개선사항 반영 여부를 점검하는 단위테스트단계를 제시하였으며 개별 스프린트 단위로 인수테스트를 실시하도록 하였으며, 이미 완료된 각 스프린트별 단위테스트 및 인수테스트를 통하여 종합적인 테스트가 가능하도록 통합테스트계획을 제시하였다. 점검 항목으로는 기존 항목인 ①단위테스트의 수행 ②인수테스트를 통한 최종 사용자의 승인 획득 ③통합테스트 계획 수립을 포함하여 ①완료 스프린트 내 모든 유저스토리가 단위테스트 기준을 통과하고 결함/개선사항의 반영 여부 ②해당 스프린트 단위로 인수테스트 수행여부 ③완료된 스프린트별 단위테스트 및 인수테스트를 반영한 통합테스트계획의 수립여부를 추가하였다.

테스트단계에서는 컴포넌트와 하위시스템이 인터페이스 되어 단일 시스템으로 통합되고 개발된 제품이 실제 운영 환경에서 기본적인 기능들이 정상적으로 실행되는지 점검할 수 있도록 통합테스트와 시스템테스트를 제시하였다. 점검항목으로 기존 해설서의 ①테스트환경이 실제 운영환경을 반영한 테스트 수행여부 ②통합테스트 계획에 따른 실시 여부 ③내·외부 시스템간의 연계 및 구현된 시스템간의 기능 완전성과 데이터 무결성 확보 여부 ④수립된 시스템테스트 계획에 따른 실시 여부 ⑤시스템테스트를 통한 성능 및 가용성, 보안성 검증 여부 ⑥테스트결과 결과 관리 여부 ⑦통합 및 시스템테스트 결과를 반영한 시스템 최적화 여부 ⑧시스템 사용자 및 운영지침서의 작성 여부를 포함하고 ①통합테스트 결과의 개선 및 회귀테스트 진행 여부 ②시스템테스트 결과의 개선 및 회귀테스트 진행 여부 ③컴포넌트 하위 시스템이 통합되고 실제 운영 환경에서 정상적 기능 동작 점검여부라는 점검항목을 추가하였다. 테스트단계 점검항목 중 회귀테스트는 기존 정보시스템감리해설서에는 존재하지 않지만 현업에서 일상적으로 사용되는 점검항목으로서, 회귀테스트는 이전의 실행테스트를 다시 수행하며 이전의 오류가 재발생되는지 검사하므로, 반복적 개념인 애자일 방법론의 테스트활동에 적합하므로 반드시 포함되어야 하는 점검항목이다.

마지막으로 운영단계는 시스템테스트의 확장으로 추가적인 병목현상을 찾아 측정하여 제거하고 향후 회귀테스트를 위한 성능 측정 기준을 수립할 수 있는 성능테스트단계 및 시스템의 부하를 점진적으

로 증가시켜 시스템의 오작동 여부를 점검하는 부하 테스트단계와 사용자 승인여부와 테스트활동의 적절성을 평가하는 시스템전환 단계를 제시하였다. 기존 해설서의 점검항목인 사용자 인수테스트를 통한 사용자 승인 여부를 포함하고 ①추가적인 병목현상을 측정 후 제거 및 회귀테스트의 기준 수립을 위한 성능테스트 수행여부 ②시스템의 부하를 점진적으로 증가시켜 어느 수준의 부하에서 시스템이 오작동하는지를 발견하는 부하테스트단계의 수행여부와 같은 점검항목을 추가적으로 제시하였다.

점검 산출물은 기존 감리 시 점검하는 산출물을 포함하여 유저스토리, 제품 백로그, 테스트계획서, 릴리즈계획서, 스프린트 백로그, 스프린트 리뷰, 소멸 차트 등과 같이 애자일방법론에 특화된 산출물에 대한 점검을 수행해야 한다. 유저스토리의 점검항목은 ①유저스토리 구현 시 사용자 요구사항의 인식 및 적용 여부 ②주요 요구사항 및 공통 기능의 우선순위 설정 후 일정관리 여부이며, 제품 백로그는 ①유저스토리에서 발생된 결함목록 존재 여부 ②유저스토리가 위험우선순위 또는 구현을 고려하여 스프린트 할당 후 백로그에 우선순위를 부여 여부와 릴리즈계획서는 ①릴리즈 일정 수립여부 ②릴리즈 대상 스토리 선정 여부를 점검하고 스프린트 백로그는 ①유저스토리가 스프린트 백로그에 포함되고 유저스토리를 완료하기 위한 기준 수립 ②스프린트를 진행하는 동안 작업시간 등의 수정사항이 발생한 경우, 변경사항이 스프린트 백로그에 반영되었는지 여부 등이다. 소멸차트에는 제품 백로그 혹은 스프린트 백로그의 정보가 충실하게 업데이트 및 소멸차트에 반영되는지 확인하고 유저스토리 기반으로 스프린트의 일별 계획 및 실적을 관리하는지 여부를 점검해야 한다.

3.3 애자일 방법론 테스트활동에 대한 개선된 감리 점검항목 검증

본 연구에서 제시된 개선된 감리점검 항목에 대하여 약식 설문조사와 델파이기법을 적용하여 검증하였다. 공공사업 감리 현업에서 활동하는 10여명의 정보시스템 기술사와 감리사에게 질문하여 개선된 감리항목에 대한 설문조사와 개선된 점검항목에 대한 의견을 수렴하였다.

15명에게 설문조사를 요청하여 10명으로부터 설

문지를 회수하였으며 설문조사 결과는 다음과 같다. ①애자일방법론이 적용된 프로젝트 감리 시 기존의 감리기법으로 가능여부의 첫 번째 질문에서는 필요하다 응답자가 6명(60%), 불필요하다 응답자가 4명(40%)로 집계되었으며, ②애자일방법론이 적용된 프로젝트 감리 시 별도의 감리기법 필요여부의 두 번째 질문에는 6명(60%)은 필요하다, 4명(40%)은 불필요하다는 의견이 있었으며, ③애자일방법론이 적용된 프로젝트 감리 시 감리시점에 대한 세 번째 질문에는 기존방법 유지(70%), 상시감리형식(30%)의 응답이 있었으며, ④애자일방법론이 적용된 프로젝트 감리 시 테스트항목이 특별히 중요하다의 네 번째 질문에는 5명(50%)는 그렇다, 5명(50%)는 그렇지않다로, ⑤애자일방법론이 적용된 프로젝트 감리 시 점검항목에 대한 개선이 필요성의 다섯 번째 질문에는 7명(70%)는 그렇다, 3명(30%)는 그렇지않다라고 답변하였다.

애자일방법론에서 개선 제안된 테스트 점검항목에 대한 적정성 검토로서 상기 문항 5번항목 중 개선이 필요하다고 답변한 전문가와 개선이 필요하지 않다고 응답한 전문가 모두가 개선된 내용이 적합하다고 동의하였으며, 일부 전문가는 다음과 같은 추가적인 의견을 제시하였다.

① 전문가 A : 애자일 기반 정보시스템 감리 시 제안된 테스트활동을 강화한 감리모형은 적절하다고 판단되며, 애자일의 특성인 스프린트착수와 관련된 각각의 기능구현 종료시점별 고객이 인수테스트를 수행할 각각의 스프린트별 유저스토리 수립여부와 반복계획 수립의 적절성 여부에 관한 항목을 추가한 것은 적절하다고 판단됨.

② 전문가 B : 반복적인 릴리즈와 개발 프로세스에 대한 테스트활동을 강화한 감리모형은 변화되고 있는 개발환경에 대한 발전적인 감리 방법의 제시라 생각되며, 애자일의 특성인 반복적 개발에 대한 테스트 방법으로 회귀테스트를 명문화한 것은 실제 현장 감리의 내용을 잘 반영한 항목이라 생각함.

③ 전문가 C : 애자일 방법론이 적용된 정보시스템에 대한 감리모형으로서 제시된 모형은 적절하다고 판단되며, 테스트부분에서 개선된 점검항목들은 일부 항목들에 있어서 현실에 적용하기엔 일부 과한 측면도 존재하지만 실제 감리적용에는 무리가 없다

고 판단된다. 점검 산출물에 대한 점검항목을 명문화 부분은 반드시 포함되어야 하는 것으로 유저스토리, 제품 백로그, 테스트계획서, 릴리즈계획서, 스프린트 백로그, 스프린트 리뷰, 소멸 차트 등을 점검 항목으로 포함한 것은 당연한 것으로 판단됨.

상기와 같이 감리점검항목의 개선 및 추가에 대한 검증으로 10여명의 소수를 대상으로 설문조사하였지만, 설문조사 대상인원이 감리 현업에서 수석감리원으로 활동 중인 전문가들이므로 개선된 점검항목에 대해서는 적절성에 대하여 유의미한 결과라고 해석할 수 있다.

4. 결 론

본 연구는 현업에서 애자일 방법론을 적용한 정보시스템 개발 프로젝트에 대한 감리 수행 시 기존 감리점검항목으로 감리를 수행하기가 부적절한 상황이 빈번하게 발생하였다. 따라서 애자일 방법론이 적용된 정보시스템의 구축, 운영에 대하여 감리 수행 시 종합적인 점검 및 평가가 가능하도록 하였으며, 특히 애자일 방법론의 특징인 짧은 주기의 반복적인 개발이 강조된 애자일 방법론에 적합한 감리 모형과 테스트활동에 대한 구체적인 점검항목을 제시하였다.

본 연구의 결과를 요약하면 애자일 방법론을 적용한 정보시스템 개발의 테스트활동 수행 시 테스트의 실행 여부와 테스트 효율성에 대한 점검을 할 수 있도록 기존 감리 점검항목을 보완하였고, 기존 감리 점검항목이 제시하고 있지 않은 유저스토리, 제품 백로그, 릴리즈 계획서, 스프린트 리뷰 등에 대한 점검과 정보시스템감리 점검모형에서의 테스트활동 절차를 단위테스트, 인수테스트, 통합테스트, 시스템테스트로 제시하여 애자일 기반의 정보시스템 개발 프로젝트에 대하여 효과적인 감리가 수행되도록 제시하였다.

본 연구의 시사점은 정보시스템감리분야에서 애자일 방법론의 특징이 반영된 테스트활동에 특화된 감리점검항목에 대한 연구이며, 한계점은 수많은 애자일 개발방법론 중 현업에서 많이 사용되는 스크럼과 XP의 관점에서 연구를 수행하였으며, 수행되는 프로젝트 규모에 따른 추가된 점검항목 적절성에 대한 연구가 필요하다고 생각한다.

REFERENCES

- [1] Manifesto for Agile Software Development, <http://agilemanifesto.org/iso/ko/manifesto.html>, (accessed Oct., 1, 2016).
- [2] S.C. Yang, H.M. Kim, S.Y. Shin, and S.D. Kim, "A Corporate-wide Integrated Software Development Methodology," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 4, No. 1, pp. 63-74, 1998.
- [3] J.J. Ju, S.D. Kwon, and S.H. Ko, "A Study on the History of IS Development Methodology," *Information Systems Review*, Vol. 10, No. 2, pp. 211-234, 2008.
- [4] S. Ki, "A Scrum Development Process of an Agile Methodology using Concurrent Testing," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 39, No. 2, pp. 84-90, 2012.
- [5] S.B. Seo and J.H. Kang, "An Analysis of Agile Methodologies' Trends and Introduction Cases of the Methodologies at the ESA Ground Segment Software Development," *Aerospace Engineering and Technology*, Vol. 11, No.1, pp. 111-117, 2012.
- [6] K.U. Kim and H.Y. You, "An Evaluation of Software Development Methodology Applicability at Medium and Small Business through AHP," *Journal of Information Processing Systems*, Vol. 2, No. 10, pp. 691-696, 2013.
- [7] Y.R. Kwon, *Software Testing*, Life&Power Press, Paju, 2010.
- [8] I.S. Kim, Object Oriented Data Modeling, *Kyungbuk College*, Vol. 19, pp. 199-211, 2000. <http://www.riss.kr/link?id=A75002415> (accessed Oct. 1, 2016).
- [9] J.S. Kim, S.G. Choi, K.H. Kim, and T.W. Kyung, "A Case Study on the Software Quality Improvement : Information Systems Advancement Project of A Company," *The Korea Contents Association Journal*, Vol. 11, No. 7, pp. 80-88, 2011.
- [10] J.E. Cha and H.G. Kim, "Construction of

- Component Repository for Supporting the CBD Process,” *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 29, No. 7, pp. 476-486, 2002.
- [11] Y. Heo, W.Y. Kim, C.S. Kim, and J.S. Kim, “A Development of the Remote Teacher’s Training Cyber System Applied by CBD Methodology,” *Proceeding of Conference of the Korea Information Processing Society*, pp. 485-488, 2000.
- [12] Y.D. Kim, *Requirement Tracing Method Using Matrix in Agile Development Process*, Doctor’s Thesis of Junnam University Graduate School, 2013.
- [13] S.W. Jung, H.J. Jun, W.H. Park, and J.W. Lee, “How ICT Company Should Manage the Potential Risk of Open Source Software on its Application,” *Proceeding of Korean Society for Internet Information Conferences*, pp. 65-66, 2012.
- [14] S.S. Hwang, *An Empirical Analysis on the Effects of Agile Methodology on Job Characteristics of Software Development and Work Performances*, Doctor’s Thesis of Jungang University Graduate School, 2009.
- [15] Agile Methods, https://en.wikipedia.org/wiki/Agile_software_development#Agile_methods, (accessed Oct., 1, 2016).
- [16] Visionone 10th Annual State of Agile Report, <http://www.versionone.com/about/press-releases>, (accessed Oct., 29, 2016).
- [17] S.G. Kim and S.S. Hwang, “An Analysis of Measurement in Scrum Method and Its Case Study,” *Information Systems Review*, Vol. 10, No. 3, pp. 185-205, 2008.
- [18] Scrum Framework, <http://http://blogs.moham-madsami.com>, (accessed Oct., 29, 2016).
- [19] C.J. Seo, “A Study on the Factors and Measurement of Quality of System Integration Service”, *Journal of the Korean Society for Quality Management*, Vol. 27, No. pp. 20-41, 1999.
- [20] K.I. Seo and E.M. Choi, “Comparison of Major Black-Box Testing Methods in Object-Oriented Software,” *Korean Institute of Information Scientists and Engineers*, Vol. 33, No. 1, pp. 1-16, 2006.
- [21] S.H. Jun, J.D. Kim, and D.K. Beak “A Design Technique of a Flexible Unit Testing Tool for Test Driven Development,” *Journal of Computing Science and Engineering*, Vol. 35, No. 1, pp. 53-54, 2008.
- [22] H.S. Bae, “Survey on Test Automation Tools,” *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 23, No. 3, pp. 33-44, 2005.
- [23] D.H. Kim, C. Ko, D.S. Kim, and H.W. Kim, “A Study on the Agile-based Information System Audit Model,” *The Journal of Digital Policy & Management*, Vol. 11, No. 8, pp. 95-108, 2013.
- [24] Dean Leffingwell, *Scaling SoftwareAgility: BestPracticesforLarge Enterprises*, Pearson Education, Boston, 2008.
- [25] H.M. Oh, J.W. Jun, and E.J. Kim, *Information System Auditing Guide V3.0*, NIA, Seoul, 2008.
- [26] S.K. Park and M.G. Park, “A Model to Estimate Software Development Effort Based on COSMIC-FFP Using System Complexity,” *Journal of Korea Multimedia Society*, Vol. 13, No. 11, pp. 1575-1585, 2010.



박 동 아

1996년 2월 부경대 경영학과 (경영학사)
2001년 2월 부산대 경영대학원 (경영학석사-경영정보 전공)
2017년 2월 부경대 대학원 정보시스템학과 (공학박사 취득)

2008년~현재 한국정보시스템감사통제협회 이사
1997년~2012년 더존비즈온 ERP컨설팅팀 근무
2015년~현재 비앤디컨설팅 감사사업부 이사
관심분야 : BPR, ISP, Clouding Computing, 정보시스템 감리



박 만 곤

경북대학교 수학교육(이학사)
경북대학교 전산통계학(이학박사)
Philippine Women's University (국제행정학석사)
University of Rizal System, Philippines(명예 기술학박사)

Dept. of Electrical and Computer Engineering, University of Kansas (Post Doc.)
1981년~현재 부경대학교 IT융합응용공학과 교수
1997년~현재 한국멀티미디어학회(KMMS), 초대 총무 이사, 수석부회장, 회장 및 명예회장
2002년~2007년 정부간 국제기구 CPSC (콜롬보플랜기 술교육대학교) 사무총장 (Director General and CEO)
2004년~2007년 Asia-Pacific Accreditation and Certification Commission (아태지역 인증검증위원회) 위원장
2005년~2007년 유네스코 (UNESCO-UNEVOC) 자문 위원, 아시아개발은행 (ADB) 자문관
관심분야 : 소프트웨어 공학 및 재공학, 소프트웨어 신뢰성공학, 소프트웨어 안전성 공학, 비즈니스 프로세스 재공학 (BPR), ICT-기반 HRD, 전자 정부 및 전자교수학습 시스템 구축