

중소기업 정보화방법론 개선 연구

(A Study on Improvement of Information Methodology for SMEs)

순남순

(㈜이지스웨어 연구소장, 한밭대학교 겸임교수, sns174@daum.net)

Abstract

Information competitiveness accounts for substantial parts of business competitiveness necessary for business management in the knowledge-information society in the 21st century. To improve quality, productivity and competitiveness through information in the fields of SMEs particularly having difficulties under rapidly changing business environment, the government has operated "SME Information Support Project" for the past 8 years. The methodology for developing the standard for this project known as EISDM (Enterprise Information System Development Methodology) provides communication between IT businesses and SMEs participating in this project, and standardized output document formats and how to make out such documents.

Unfortunately, the number of personnel partaking in the development project for SMEs is no more than 2~4 per site on average. Further, they are required to complete demand analyses, development, testing and operation in about 6 months, which is a very short period. Moreover, there is too much demand for documentation, which is likely to end up being formal work process just for supervision and inspection. That is, the documentation could be for noting but documents, which will prove useless outputs after the project finishes.

Therefore, this study proposes an improvement approach as an information system development methodology taking into account SMEs' characteristics and environment so as to relieve developers from such excessive burden of documentation, to save time and resources through efficient management of software development as the original purpose of the methodology, and to produce required quality software.

Key Words : EISDM; agile; scrum; Methodology; SMEs'

1. 서론

21세기 지식정보화 사회의 기업 경영은 정보화의 경쟁력이 기업의 경쟁력에 상당 부분을 차지한다. 급변하는 환경에서 특히 어려움을 겪고 있는 중소기업의 현장에 정보화를 통한 품질 및 생산성 향상 그리고 경쟁력을 높이기 위하여 정부에서는 "중소기업정보화지원사업"을 지난 8년 동안 시행하고 있다. 이 사업의 표준 개발 방법론 (EISDM :

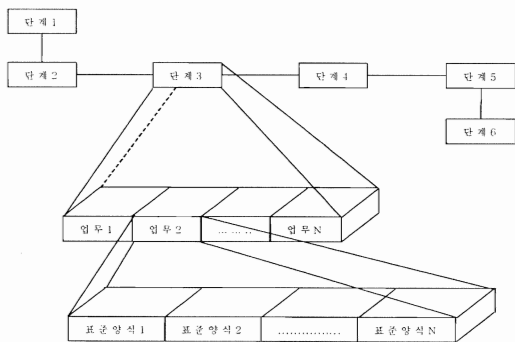
Enterprise Information System Development Methodology)은 사업 참여 중소기업과 IT업체 간의 의사소통과 표준화된 산출물 문서 양식 및 작성 방법을 제공하고 있다[2]. 그러나 중소기업의 개발 참여 인력이 한 사이트 당 평균 2~4명이 투입되어 약 6개월 정도의 짧은 기간에 요구분석에서 개발 및 시험과 운용까지 완료하여야 하는 상황임에도 불구하고 많은 문서 작성업무는 개발자에게 과중한 업무량이 되어 자칫 프로젝트의 품질을 저하시키는 원인으로, 감리 수검에서 실제로 구

현된 프로그램과 산출물 문서의 정합성의 문제가 종종 발견되며 이는 프로젝트 종료 후 유지보수 시에도 사용성과 유용성이 떨어질 수 있는 우려가 있다. 따라서 본 연구에서는 이러한 과도한 문서작성의 부담을 개발자에게서 줄여 주고 방법론의 본래 목적인 소프트웨어 개발을 효율적으로 관리하여 시간과 자원의 낭비를 줄이고, 중소기업 현장의 다양한 요구사항을 충실히 반영하여 요구되는 품질의 소프트웨어를 만들어내기 위해서, 중소기업의 특성과 환경을 감안한 정보시스템개발방법론으로 구조적 및 절차적 모델에 기반한 현재의 EISDM 방법론을 보다 경량적이면서도 짧은 개발 주기를 반복(iteration)하고, 의도된 방향으로 프로젝트가 진행되는지 수시로 확인하여 위험요소를 최소화시킬 수 있는 Agile 방법론으로 개선을 제안한다.

II. 선행연구

1. 중소기업정보화사업 표준 방법론

중소기업정보화사업의 표준 방법론인 EISDM의 구성은 구조적 및 절차적 구성 요소로 살펴 볼 수 있다. 먼저 EISDM의 구조적 구성 요소는 <그림 1>과 같이 단계(Phase), 업무(Activity), 표준 양식으로 구성되어 있다. 각 단계는 그 단계에서 수행하여야 할 여러 가지 업무로 구성되어 있으며, 각 업무는 세분화된 기능으로 구성되어 있다. 이 기능들의 산출물을 표준 양식으로 작성함으로써 상호 업무의 영역을 구분 하고 있다.



<그림 2> EISDM의 구조적 구성 요소

EISDM의 절차적 구성 요소는 <그림 2>와 같이 6단계로 구성되어 있으며, 크게 상반부(3단계)와 하반부(3단계)로 구성된다. 상반부는 정보시스템을 구축하기 위한 제반 활동으로 타당성 조사, 업무

분석 및 설계, 기술 설계로 구성되며, 하반부는 상반부에서 작성된 자료를 기초로 하여 코딩, 테스트 및 이행, 운영 지원으로 구성된다.

○ 타당성 조사 (FS : Feasibility Study)

프로젝트 계획을 수립하기 위해 현행 업무의 개괄 파악, 여러 가지 대체 안 작성 및 선택, 선택된 새로운 시스템의 개념적 구성, 기대 효과 및 경제성 분석을 실시하는 단계이다.

○ 업무 분석 및 설계 (AD : Business Analysis & Design)

프로젝트 관리 계획수립, 시스템 요구사항 설계를 위해 현행 업무 내용을 세부적으로 조사 및 분석하는 단계이다.

○ 기술 설계 (TD : Technical Design)

물리적 데이터 설계, 정보시스템을 내부적으로 구현하기 위한 상세 설계, 전환설계, 시험수립, 교육설계 등을 수행하는 단계이다.

○ 코딩 (CD : Coding)

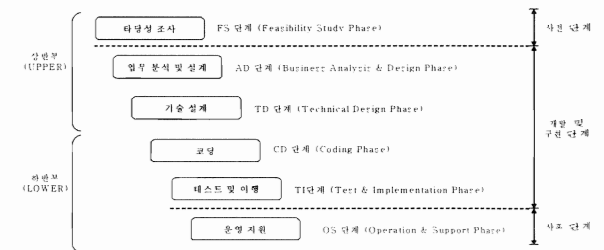
업무 분석 및 설계와 기술 설계 산출물을 응용 프로그램으로 전환시키기 위한 필요한 업무들을 수행하는 단계이며, 프로그램 작성 및 단위 테스트를 수행하는 단계이다.

○ 테스트 및 이행(TI:Test & Implementation)

시스템 요구사항 검증을 위하여 정보시스템의 전 기능을 종합적으로 테스트 및 보완 그리고 필요한 S/W 및 H/W 설치, 데이터변환, 지침서 작성, 병행처리 및 조정, 사용자 교육 등을 실시하는 단계이다.

○ 운영 지원 (OS : Operation & Support)

신규 시스템 적용 이후 정상적인 운영 및 통제를 위한 요구사항 변경 및 추가로 인해 시스템 보안을 실시하는 단계이다.



<그림 3> EISDM의 절차적 구성 요소

EISDM은 기존의 시스템 개발 방법론(PSDM : Production System Development Methodology) 및 패키지적용 방법론(정보화구축사업방법론)으로 패키지적용 사업과 시스템 개발 사업으로 양분되

었던 “중소기업정보화지원사업”이 2010년 시스템 개발 사업으로 통합됨에 따라 방법론도 “EISDM : 기업정보화시스템 개발 방법론”으로 수정·보완하여 통합되었다. EISDM의 단계별 표준 양식 및 작성 목록은 <표 1>과 같이 구성되며 개발 산출물은 모두 111종류로 필수적으로 작성하여야 하는 산출물이 사업 유형에 따라 최소 58개에서 최대 63개이며 나머지는 개발 현장의 상황에 따라 선택적으로 작성하도록 되어있다.

<표 1> EISDM 단계별 산출물 목록

단계구분	업무구분	산출물 수
FS 타당성 조사	FS1 타당성조사 계획수립	9
	FS2 현행업무과목	5
	FS3 대체안 작성	4
	FS4 개념적 시스템구성	2
	FS5 투자대비 기대효과분석	1
	FS6 결과보고서 작성	1
AD 업무분 석 및 설계	AD1 현상분석(AS-IS)	15
	AD2 개선방안 수립(TO BE)	8
	AD3 논리적 데이터분석설계	6
	AD4 사용자 입출력 명세	4
	AD5 결과보고서 작성	1
TD 기술설 계	TD1 시스템 설계	1
	TD2 물리데이터 설계	6
	TD3 프로그램 설계	5
	TD4 전환계획 수립	1
	TD5 시험계획 수립	1
	TD6 교육계획 수립	1
CD 코딩	CD1 프로그램 작성	1
	CD2 단위 테스트	3
TI 테스트 및 이 행	TI1 시스템 요구사항 검증	3
	TI2 지침서 작성	2
	TI3 시스템 이행	3
	TI4 사용자/운영자 교육 실시	1
	TI5 사용자 승인 테스트 실시	1
	TI6 프로젝트 완료 검토	1
OS 운영지 원	OS1 운영지원계획수립	1
	OS2 시스템운영 및 통제	3
단계구분	업무구분	산출물 수
PM 프로젝트관리	PMP 착수/계획	14
PM 프로젝트관리	PMD 실행/통제	9
	PMF 종료	4

최근의 정보시스템은 IT기술의 발전과 더불어 복잡도가 날이 증가하고, 많은 업무 프로세스를 자동화하기 위한 사용자 요구가 다양하여짐에 따라 시스템 전략 계획에서부터 철저한 사용자의 요구 분석과 설계를 통해 자동화 도구에 의한 개발 공정에 이르기까지 그 생산 방법의 자동화 및 체계화가 필요하게 되었으며, 이러한 측면에서 EISDM

은 중소기업 정보시스템을 개발하기 위한 프로젝트 관리 기능 제공, 표준 양식 및 작업 방법이나 절차, 기법 등을 체계적으로 제시하여 중소기업정보화지원사업의 프레임워크로 크게 공헌하였다.

그러나 EISDM은 구조적 및 절차적 개발 모델로 프로젝트 중간에 추가로 도출되는 사용자 요구사항에 대한 민첩한 대응이 어렵고 자주 발생하는 변화에 따른 산출물의 현행화도 어려운 것이 현실이다. 무엇보다도 개발방법론의 본연의 목적은 소프트웨어 개발을 효율적으로 관리하여 시간과 자원의 낭비를 줄이고, 요구되는 품질의 소프트웨어를 만들어내기 위한 것이므로 사용자요구사항을 보다 충실히 반영하여 요구되는 품질의 요건을 만족하는 소프트웨어 개발 프로세스를 제공하고 정부지원사업임을 감안한 검증의 요소를 갖춘 개발 방법론으로의 개선이 필요한 것으로 분석되었다.

2. EISDM 보완의 Agile 방법론 제안

2.1 Agile 방법론 제안

현재 사용하는 중소기업정보화방법론인 EISDM은 분석, 설계, 개발, 검증, 이관 등의 단계를 순차적으로 거치는 방식으로 1960년대 복잡한 군사용 소프트웨어 개발을 위해 미국 해군에서 고안된 폭포수 모델(waterfall model)에 기원을 두고 있다. 이 모델의 공통적인 특징은 소프트웨어 개발에 있어 정형화된 프로세스와 단계별 검증방안을 제공하고 있는데, 각 단계의 끝에서 프로젝트 팀은 최종 점검까지 모두 끝낸 후 고객의 승인을 받게 되고, 고객이 만족하지 않는 한 다음 단계로 넘어가지 않는다[3]. 이 때문에 소프트웨어의 구현 및 테스트 단계에 이를 때까지 잠재적인 문제들과의 대면을 미루게 되며 요구 사항, 디자인, 코딩에 숨어 있는 모든 문제들이 프로젝트가 끝나기 직전에 갑자기 부상되어 고통스러운 현실을 만들어 버리는 경우가 발생하곤 한다. 이러한 구조적 모델 구현 프로세스의 주요 문제점은 사용자의 요구를 정확하게 반영하기 힘들며, 지속적으로 변화하는 요구사항을 적절히 처리할 수 없고, 각자 개발한 소프트웨어 모듈들이 잘 조합되지 않을 수 있으며, 방대한 산출물들의 관계와 정보들을 체계적이고 독립적으로 관리하기 힘들어 유지보수에 큰 부담을 주며 반영시킨 요구사항에 대한 프로그램의 품질 역시 떨어지게 될 우려가 있다[4]. 이러한 기존의 개발 방식은 소프트웨어 개발주기가 길며 사용자

의 요구사항을 효과적으로 반영하기가 어렵다 [10]. 더욱이 중소기업에서는 급변하는 기업경영 환경의 변화에 신속히 대응하여 기업경쟁력을 높일 필요가 있는데, 최소의 소프트웨어 개발 투자를 통해 극대화된 효과를 원하기 때문에 새로운 개발 방식을 찾게 하는 동기를 제공하게 되었다. 이러한 배경으로 기존의 개발방법론을 개선할 대안으로 Agile 방법론을 제안한다.

2.2 Agile 방법론 특징 및 구조

Agile 방법론은 과거 방법론의 위험 및 실패요소를 바탕으로 더 효과적인 프로젝트를 수행하기 위해 만들어졌는데 여기서 말하는 Agile이란 기본적으로 소프트웨어를 빨리 개발하여 비즈니스에 적용한다는 것 이상의 내용을 포함하고 있다. 그 내용을 살펴보면 아래와 같다.

- 프로젝트에 해를 끼칠 수 있는 것들에 대해 미리 대처할 수 있는 것

- 지속적으로 발생하는 변화에 대하여 적절하게 대처할 수 있는 것

Agile 방법론은 소프트웨어 개발을 수행하는 개념적인 프레임워크라 할 수 있다. 대부분의 Agile 방법론에서는 짧은 개발 주기를 반복(iteration)하고, 의도된 방향으로 프로젝트가 진행되는지 수시로 확인하여 위험요소를 최소화 시키고 있다. 반복은 계획수립, 요구사항 분석, 디자인, 개발, 테스트, 문서작업 등의 모든 태스크를 포함한 작은 단위의 프로젝트 개념이다. 이러한 반복은 요구사항을 다 반영하지 못하더라도 릴리즈(release)하는 경향이 있는데 이것은 각 반복 주기마다 구현된 기능들을 확인하고 새로운 요구사항을 받아들이기 위함이다. 또한 Agile 방법론은 실시간 커뮤니케이션과 고객과의 밀착된 개발 분위기를 강조하며 다른 방법론과 비교하여 문서화 작업에 큰 비중을 두지 않는다. <표 2>는 대표적인 Agile 방법론들의 특징을 나타내고 있다.

<표 2> Agile 방법론의 종류 및 특징

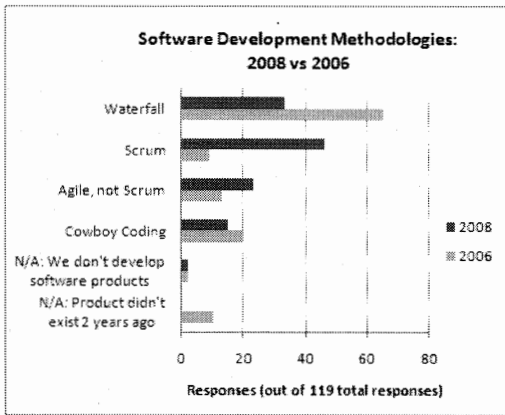
종류	특징	비고
XP	-테스팅 강조, 4가지 가치와 12개의 실천항목, 1~3주 iteration	신속한 피드백 단순함을 전체 점진적 수정
SCRUM	-프로젝트를 스프린터(30일 단위)	Iteration 계획과

종류	특징	비고
	iteration)로 분리, 팀은 매일 스크럼 (15분 정도) 미팅으로 계획 수립 -팀 구성원이 어떻게 활동해야 하는가에 초점 -통합 및 이수 테스트가 상세하지 않음	Tracking에 중점
DSDM	-기능모델, 설계화 구현, 수행 3단계 사이클(2~6주)로 구성	영국에서 주로 사용
FDD	-짧은 iteration(2주), 5단계 프로세스	설계, 구축 프로세스 반복
Crystal	-프로젝트 상황에 따라 알맞은 방법론을 적용할 수 있도록 다양한 방법론 제시 -Tailoring하는 원칙 제공	프로젝트 중요도와 크기에 따른 메소드 선택 방법 제시

프로젝트에서 중요한 포인트 중의 하나는 팀의 운영과 관리이다. 프로젝트에 Unified Process나 Waterfall model과 같은 기존의 방법론은 프로젝트 단계별 공정관리 프로세스가 잘 정립되어 있지만 자세한 Task 관리 프로세스에 대해서는 거의 정의가 되어 있지 않다. 반대로 Agile 방법론의 경우, Task 관리에 대한 전략과 수행 방법을 기술하고 있지만, 실제 프로젝트를 관리하는 관점에서는 전체 공정에 대한 예측과 관리가 어렵기 때문에 명확히 정의되어있지 않다. 이로 인하여 일반 SI 프로젝트 등에서는 쉽게 적용하는데 어려움이 있다.

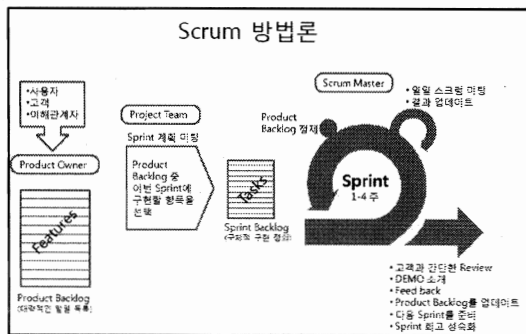
본 연구에서는 위 두 가지 접근 방법의 문제점을 상호 보완하여, 전체 프로세스 공정 관리는 기존의 전통적 방법론을 따르고, 전체 프로세스의 각 세부적인 개발 태스크 관리는 Agile 방법론으로 보완하여, 유연하고 현장 적응성이 뛰어난 관리 방법론으로 개선할 것을 제안한다. Agile 방법론에는 대표적으로 Scrum, XP(Extreme Programming), FDD(Feature-Driven Development), DSDM(Dynamic Systems Development Method), Crystal Clear 방법론들이 있는데, <그림 3>에서 나타난 바와 같이 근래에 들어서는 Scrum이 많이 사용되고 있다[1,6,7]. 이는 실제로 개발에 적용되어 적용의 효과가 입증되고 프로젝트 관리와 개발 업무 최적화 지원 측면에서 지속적으로 개선되고 있음을 나타내고 있다고 볼 수 있다. 또한 Agile 방법론의 단점인 산출물의 가시성 확보와 통제가 어려운 점을 해결하기 위한 다양한 도구들이 타 방법론에 비교하여 상용화가 진전되어 프로젝트에

적용하기 용이한 점도 있다.



<그림 4> 소프트웨어 개발 방법론 현황

따라서 본 연구에서는 EISDM의 개선으로 Agile 방법론 중 Scrum 방법론의 적용을 제안한다. Scrum은 기본적으로 점진적 반복 개발 방법에 기반하고 있으므로 공학적 접근이 용이하고 기존의 계획중심방법론과 결합한 프로세스로 Agile 방법론의 단점인 프로젝트 단계별 공정관리의 약점을 보완할 수 있다고 보기 때문이다.



<그림 5> Scrum 방법론 개발 주기

Scrum은 <그림 4>와 같이 몇 개의 반복으로 구성되며 이를 Sprint라고 한다. 각 Sprint는 1~4 주 정도의 기간을 갖는다. 이는 일종의 Timebox의 개념을 가지며 한번 정해진 Sprint의 기간은 변경될 수 없다는 것을 전제로 한다. 다음으로 Scrum에는 고객의 요건으로부터 작성된 Product Backlog(대략적인 할일 목록)라는 것을 가지고 있다[5]. 이 Product Backlog 항목은 팀 미팅을 통해서 이번 Sprint에 구현할 우선순위를 통해 선택이 되고, 구체적으로 구현하기 위해서 task로 쪼개진다. 이 task 목록은 이번 Sprint에서 구체적으로 해야 할 일을 정의하고 되고 이 목록 리스

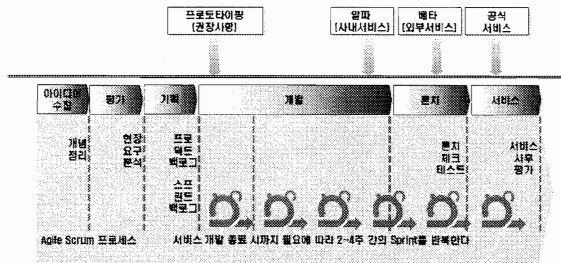
트를 Spring Backlog 라고 부른다. Sprint Backlog의 Task들은 공정이 지정되어 팀의 담당자에게 할당된다. 팀은 이 Sprint Backlog를 가지고 정해진 기간 동안의 Sprint를 구현한다. Sprint중에 매일 아침 팀원들은 10~20분 정도의 Daily Scrum Meeting 이라는 회의 시간을 갖는데, 이 시간 동안, 팀원들은 어제 한일과 오늘 한일에 대해서 간략하게 보고하고, 기술적인 질의 사항에 대한 문의나, 리스크 사항 등 현재 Sprint의 진행 상황을 공유할 수 있다. Sprint가 끝난 후에 팀은 프로젝트의 고객(stakeholder)과 간단한 Review 미팅을 갖는데, 주로 이번 Sprint 에서 구현한 내용을 간단한 형태로 시연 하고 피드백을 받아서 Product Backlog를 업데이트 하고 다음 Sprint를 준비한다. Review가 끝난 후에는 팀원끼리 Sprint에 대한 회고(Retrospective)의 시간을 갖는데, 이 시간에는 Sprint에서 Scrum 방법론을 적용했을 때의 잘되었던 점과 잘못되었던 점을 토론했을 때의 Scrum 운영 프로세스에 반영하여 팀의 운영 방식을 성숙화 시킨다[8].

III. 연구 설계

1. 중소기업정보화방법론의 개선 모델 설계

중소기업정보화방법론의 개선 모델 설계는 기존의 계획 중심 방법론의 장점과 스크럼을 결합한 형태로 <그림 5>와 같이 관리 프로세스는 아이디어 수집, 평가, 기획, 개발, 론치, 서비스 단계로 분류하였다. 이에 대응하는 Scrum은 task 수행 프로세스로 개념정리, 현장요구분석, 프로덕트 백로그, 스프린트 백로그의 점진적인 반복, 론치테스트, 서비스 사후평가로 연계되어 프로젝트관리와 자세한 개발 task 관리가 상호 연동되어 진행되도록 결합한 모델이다. 프로젝트의 진행은 아이디어 수집과 평가단계로 일정이 관리되고 구체적인 개발 task는 개념정리와 현장요구사항 도출을 수행한다. 이어서 기획단계로 공정이 진행되는데, 이때 스프린트 시작 전 전체 프로젝트 팀이 참석하는 스프린트 계획 미팅을 통해 사용자의 요구사항을 기술한 프로덕트 백로그(product backlog)를 바탕으로 해당 스프린트의 목표를 설정한다. 그리고 이에 따른

작업 범위, 개발 담당자, 작업 예상 시간 등 구체적인 개발 계획이 포함된 스프린트 백로그가 작성된다. 이는 개발 단계로 일정이 이어져 새로운 스프린트가 시작되면 스크럼 마스터가 주축이 되어 일일 스크럼을 운영한다. 이를 통해 프로젝트 진행 사항, 협력이 필요한 부분 등을 하루 단위로 점검하며, 매 스프린트 종료 시점에 전체 팀이 참여하는 스프린트 리뷰 미팅에서는 짧은 데모를 통해 해당 스프린트의 산출물을 공유하고 새로이 발견된 문제점이나 개선점 등을 논의한다. 이러한 과정을 거쳐 팀과 고객의 피드백을 다음 스프린트에 신속하게 반영하고 프로젝트 수행 과정을 점진적으로 개선해 나가면서 알파테스트와 베타테스트를 거쳐 공식 서비스로 프로젝트가 종료된다.



〈그림 6〉 계획 중심 방법론의 장점과 스크럼을 결합한 프로세스

IV. 결론

기업환경은 지속적으로 변화하고 있으며 그 변화의 주기도 짧아지고 있다. 또한 그 변화의 방향 역시 예측하기 어려운 상황이다. 이러한 환경에서 기업이 생존하기 위해서는 변화에 대한 신속한 대응을 필요로 하며 이를 위해 기업 시스템은 어떠한 비즈니스 요구사항에도 유연하고 민첩하게 지원할 수 있어야 한다. 그 대응방법의 하나로 등장한 Agile 방법론은 비즈니스 요구사항에 대하여 업무에 빠르게 적용될 수 있도록 신속하게 소프트웨어를 개발하며, 이렇게 전달되는 소프트웨어의 품질을 향상시키고, 비즈니스 관련자와의 협업 및 관계를 강화시켜주는 역할을 한다[9]. 그러나 Agile 방법론 적용은 소규모 프로젝트에서 신속한 개발과 요구되는 품질의 소프트웨어를 생산해 내는데 보다 효과적일 수 있지만 표준화된 산출물의

가시성 확보가 미흡한 문제점이 해결하여야 할 과제이다. 프로젝트가 진행되면서 발생하는 공정별 산출물은 개발 과정의 적정성을 점검할 수 있는 가시적인 문서이고 향후 유지보수 시에 참조되며, 소프트웨어의 품질보증과 성공적인 완성을 증명할 수 있는 감리 수검의 필수 자료가기 때문에 이러한 문제점은 반드시 해결하여야 한다. 그러므로 Agile 방법론을 중소기업정보화지원사업의 표준 방법론으로 적용하기 위해서는 표준화된 리포트 환경제공 도구와 전략 수립을 위한 관리도구, 요구관리 수행도구, 품질관리 수행도구, 그리고 전략적인 관점에서 어플리케이션 라이프사이클 관리 도구가 자동화 도구로 사용되어야 해결할 수 있다. 또한 개발자의 방법론 숙지 교육과, 컨설팅 서비스가 이루어져 정보화 지원 기관인 개발업체의 프로젝트 수행 능력으로 자리 잡아야 될 것이다. 이러한 자동화 관리 도구 측면에서 최근 상용화가 상당히 진전되어 있으나, 상당히 고가이므로 지원 기관의 영세성을 감안할 때 자체적으로 도입하기가 용이하지 않으며 또한 개별적 도구 적용은 표준화의 문제가 대두된다. 따라서 정부차원의 주도 및 지원이 필요하다고 보며 지원의 형태로는 기존의 상용관리 도구를 구매하여 개발 및 참여 기업에 제공하는 방안도 있겠으나, 우리나라 중소기업의 특성을 충분히 감안한 중소기업기술정보진흥원 고유의 관리 도구와 관련 표준을 전략적으로 개발하여 보급시킴과 동시에 변화하는 환경에 적응하도록 지속적인 유지관리와 감독 및 교육을 실시하는 것이 바람직할 것이다.

참고 문헌

- [1] <http://www.javastudy.co.kr>
- [2] 중소기업기술정보진흥원, 기업정보화시스템개발방법론(EISDM), 2009
- [3] 이영진, 소프트웨어 Agile 개발방법론, 가치창조, 2007.
- [4] 공채원, 심우곤, 백인섭, "국내 소프트웨어 개발 환경에서의 Extreme Programming 개발방식의 적용성 연구" 정보과학회 봄 학술발표논문집, Vol.28 No1, 577-597.
- [5] 이세영, 용환승, "소규모 프로젝트를 위한 애자일 프레임워크 설계 및 평가", 정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제11호, 2009.11, 851-855.
- [6] O. Salo, P. Abrahamsson, "Agile Methods in European Embedded Development

Organizations: a survey study of Extreme Programming and Scrum," *IET Softw.*, vol.2, 2008, 58-64.

[7] T. Dyba, T. Dingsøy, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol.50, no.9-10, 2008, 833-859.

[8] B. Boehm, R. Turner, "Using Risk to Balance Agile and Plan-Driven Methods," *IEEE Comput.*, vol.36, no.6, 2003, 57-66.

[9] T. Chow, D. Cao, "A survey study of critical success factors in agile software projects," *The Journal of Systems and Software*, vol.81, 2008, 961-971.

[10] L. Wang, "Agility counts in developing small-size software," *IEEE Potentials*, vol. 26, no.6, 2007, 16-23.