

PSP/TSP와 Six Sigma의 활용을 통한 정량적 소프트웨어 프로세스 개선 방안의 수립 및 연관 지원도구의 개발에 대한 연구

송일선* · 박현철** · 정경학*** · 최호진****

1. 서 론

현재 소프트웨어의 복잡도는 날이 갈수록 빠르게 증가하고 있다. 이로 인하여 조직적인 소프트웨어 품질의 중요성 또한 높아지고 있다. 소프트웨어 품질을 극대화하기 위해서는 무엇보다도 프로세스를 개선해야 한다. 만일 철저한 개발 프로세스를 적용하지 않는다면 잦은 요구사항의 변경과 파악된 요구사항의 불확실성으로, 결국에는 소프트웨어의 품질 저하를 초래하기 때문이다.

프로세스 개선을 위해 개인과 팀 차원에서 소프트웨어 개발에 사용할 수 있는 기법에는 SEI에서 개발한 PSP/TSP(Personal Software Process / Team Software Process)가 있다. PSP/TSP는 각각 개인과 조직 차원에서 프로세스 개선을 구체적으로 어떻게 적용하여 성과를 창출 할 수 있는지에 대한 절차와 수단을 제공해준다.

소프트웨어 개발의 시작은 개개인의 개발 프로세스에 기초하기 때문에 PSP는 무엇보다도 중요하다 할 수 있다. PSP에서 제시되는 방법에는 측정 가능한 프로세스, 과거 프로젝트 데이터에 기반한 소프트웨어 크기 및 개발시간 예측, 설계 및 코드 검토, 프로세스 품질 측정 등이 포함된다. 이러한 방법을 통해 정량적인 프로세스의 개선을 수행할 수 있다. 그러나 PSP/TSP에서 제공하는 메트릭 분석기법은 한정적이기 때문에, 단순히 PSP/TSP만으로는 정량적인 프로세스 관리를 수행하기 어렵다.

최근 PSP/TSP에서 제공하는 메트릭 분석기법의 부족함을 보완해주고, 정량적인 소프트웨어 품질 관리를 위한 기법으로 식스 시그마가 주목을 받고 있다. 품질개선 운동과 통계적 기법을 결합함으로써 제품과 서비스의 품질을 전사적으로 혁신하는 활동인 식스 시그마는 소프트웨어 개발 프로세스의 개선에도 적용이 될 수 있기 때문이다.

이와 같은 정량적인 프로세스의 개선을 이루기 위해서는 신뢰도 높은 데이터의 수집이 중요하다. 하지만 데이터의 수집 작업을 수동으로 수행할 경우, 단순 반복적인 작업의 특성상 실수를 야기할 수 있고, 이로 인하여 분석결과에 오류가 발생할 수 있다. 뿐만 아니라 식스시그마를 이용하여 효율적으로 프로세스를 변경, 유지 관리할 수 있도록 지원해주는 소프트웨어 도구 또한 흔히

※ 교신저자(Corresponding Author) : 최호진, 주소 : 대전광역시 유성구 문지동 한국정보통신대학교 F604 (305-732), 전화 : 042)866-6162,6211 E-mail : hjchoi@icu.ac.kr

* 한국정보통신대학교 공학부 석사과정
(E-mail : isceyou@icu.ac.kr)

** 한국정보통신대학교 공학부 석사과정
(E-mail : aimania@icu.ac.kr)

*** 한국정보통신대학교 공학부 석사과정
(E-mail : hcparker@icu.ac.kr)

**** 한국정보통신대학교 공학부 교수

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었다(IITA-2006-(C1090-0603-0032)).

않고, PSP/TSP를 수행하는 동안 적시적소에 식스시그마 도구를 사용함 있어 개발자들이 많은 어려움을 겪고 있다.

본 논문에서는 자동으로 데이터를 수집 및 통계적 분석 결과를 제공해주는 도구와 PSP/TSP의 적시적소에 식스시그마 기법을 사용함으로써 PSP/TSP의 부족한 분석기법을 보완해 줄 수 있는 프레임워크, 그리고 전체 프로세스에 걸친 통합적인 식스시그마 프로젝트 지원도구 도구를 소개한다.

2. PSP 지원도구

2.1 연구목적 및 기대효과

소프트웨어 프로세스의 개선은 조직, 프로젝트 레벨 뿐만 아니라 개인, 팀 레벨에서 소프트웨어 품질을 향상시키기 위한 효율적인 방안이다. PSP는 개인 개발자들이 개인 프로세스를 개선시킴으로써 소프트웨어 품질을 향상하도록 도와주는 방법이다. PSP에서 제시하는 방법에는 측정가능한 프로세스, 과거 프로젝트 데이터에 기반한 소프트웨어 크기 및 개발 시간 예측, 프로세스 품질 측정 등이 포함된다.

PSP가 제시하는 다양한 방법들 중 측정 및 분석의 지속적인 수행을 통해 개발자들은 개인 프로세스의 약점을 파악할 수 있고, 파악된 약점을 해결할 수 있는 계획을 세울 수 있다. 이러한 이득을 얻기 위해서는 신뢰성 있는 데이터의 수집이 필수적이다. 하지만 데이터의 수집을 수동으로 수행할 경우, 단순 반복적인 작업의 특성상 실수를 일으킬 수 있다.

PPMT(Personal Process Management Tool)는 데이터 자동 수집을 통해 이러한 문제점을 해결해 준다. 또한 수집된 데이터를 이용하여 소프트웨어의 크기 측정, 소프트웨어 크기 및 개발시간 예측,

설계 및 코드 검토, 프로세스 품질 측정과 같은 정량적 통계 분석을 제공함으로써 정량적인 프로세스의 개선을 이루는 효과를 가져올 수 있다[1].

2.2 PSP 지원도구 아키텍처

PPMT는 데이터 수집의 오버헤드와 문맥전환을 감소시키기 위한 프로그램으로 자동 데이터 수집과 차트, 그래프 형태로 다양한 데이터 분석 결과를 제공하고, 프로젝트 계획, 획득가치를 이용한 프로젝트 추적을 지원한다. PPMT는 그림 1과 같이 서버-클라이언트 구조로 되어 있다. 클라이언트는 데이터 자동 수집을 담당하는 센서들로 이루어져 있고, 데이터 자동 수집을 제외한 모든 기능을 수행하는 서버는 웹 브라우저를 통해 사용자와 상호작용하는 웹 애플리케이션으로 개발되었다. PPMT의 주요 컴포넌트는 다음과 같다.

- Sensor: 개발 관련 도구(예 이클립스, JUnit)에 플러그인 형태로 부착되어서 필요한 데이터를 자동으로 수집하고 수집된 데이터를 PPMT Client로 전송한다.

- PPMT Client: PPMT Client의 주요 기능은 센서에서 받은 데이터를 PPMT Server로 전송하는 것이다. 또한 PPMT Client는 서버와의 연결이 끊겼을 경우 센서 데이터를 임시 보관하고, 필요한 경우 센서 데이터를 전처리(pre-processing)하는 역할을 담당한다. 모든 센서들이 공통적으로 수행해야 하는 이러한 기능을 PPMT Client가 처리함으로써 센서 개발을 용이하게 한다.

- PPMT Server: PPMT Server는 데이터 자동 수집을 제외한 기능들-데이터 저장 및 분석, 데이터 수동 입력, 계획 및 획득가치 추적 지원, 사용자/프로젝트 관리-을 수행한다. 서버는 자바 기반(자바 서블릿, JSP, 자바빈즈)으로 구현되었으며 서블릿과 JSP 실행을 위해 Apache Tomcat를 이용한다.

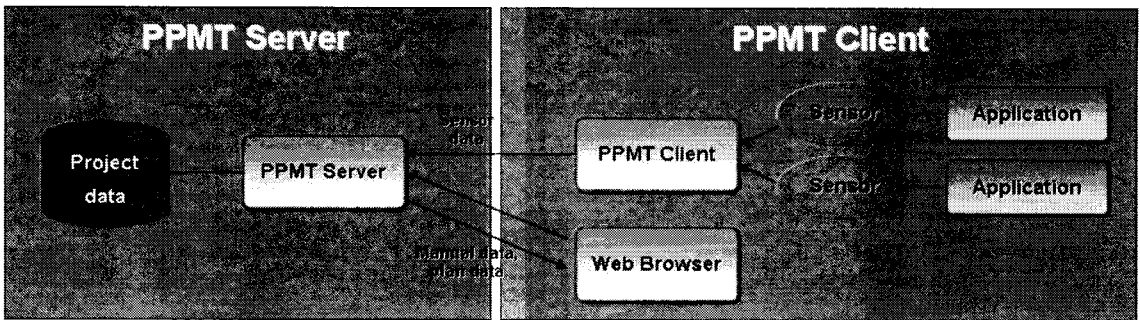


그림 1. PPMT 아키텍처

- Project Data: Project Data 데이터베이스는 센서 데이터, 시간 및 결함, 작업 및 스케줄 계획 등의 수동으로 수집된 데이터, 사용자/프로젝트 관련 정보를 저장한다.

PPMT는 센서, PPMT Client, PPMT Server 간에 센서 데이터 전송을 위해 XML 포맷을 이용하고 있다. 센서는 하나의 프로그래밍 언어가 아닌 다양한 언어로도 개발될 수 있으므로, 센서 데이터는 특정 언어에 종속적이지 않는 XML의 포맷으로 작성될 필요가 있다. PPMT에서 제공되는 주요 기능들은 다음 절에서 자세히 설명한다.

2.3 센서 기반 데이터 자동 수집

시간, 결함, 소프트웨어 크기 데이터 기록을 돕기 위해 PPMT는 센서 기반 데이터 자동 수집을 제공한다. 자동 수집된 시간, 결함 데이터는 각각 시간, 결함 일지의 항목으로 기록되는데 이는 앞서 언급한 것처럼 자동 수집된 시간, 결함 데이터는 PSP가 요구하는 모든 정보를 포함하지 않기 때문에 개발자가 필요한 정보, 이를테면 작업 시간이나 결함일지를 추가하거나 기존 데이터를 수정할 수 있게 하기 위함이다.

개발자가 작성 및 수정하는 소프트웨어 산출물 또는 개발자가 사용하는 소프트웨어 도구를 계속적으로 모니터링 함으로써 소스코드 작성, 수동

테스팅, 설계문서 작성, 코드 검토 등에 사용된 시간을 자동 수집할 수 있다. 현재 구현에서는 소스코드 작성, 윈도우즈 애플리케이션과 웹 애플리케이션의 수동 테스트에 사용된 시간의 자동 수집이 지원된다. 소스코드 작성에 사용된 시간은 소스코드 파일의 크기를 계속적으로 모니터링 함으로써, 수동 테스트에 사용된 시간은 해당 애플리케이션에서 발생하는 마우스 또는 키보드 이벤트를 모니터링 함으로써 자동 수집된다. 현재 구현에 포함된 센서는 다음과 같다.

- Eclipse 센서: 자바 소스 파일의 크기를 주기적으로(예 30초) 모니터링 함으로써 소스코드 작성에 사용된 시간을 자동으로 수집한다. 또한 Eclipse에서 개발자가 실행한 윈도우즈 애플리케이션에 마우스 또는 키보드 이벤트를 감지하는 윈도우즈 후킹(Windows Hooking) 모듈을 삽입하여 실행된 애플리케이션의 수동 테스트에 사용된 시간을 자동으로 수집한다.

- IE 센서: Internet Explorer에서 발생하는 마우스, 키보드 이벤트를 감지하여 웹 애플리케이션 수동 테스트에 사용된 시간을 자동으로 수집한다.

이렇게 수집된 시간 센서 데이터는 다음과 같은 방법으로 시간 일지에 기록된다. 사용자가 설정 가능한 간격(예 5분) 동안에 최소한 하나의 시간 데이터(파일 크기 변경, 마우스 또는 키보드

이벤트)가 있으면 이 간격 동안 개발자가 작업을 했던 것으로 간주한다. 이러한 간격들이 연속적이면 이 연속된 시간들을 하나의 항목으로 합쳐서 시간 일지에 기록한다. 시간 일지에 기록된 시간 데이터가 어떠한 작업과 관련된 것인지를 개발자가 파악하는데 도움을 주기 위해 수정된 파일 또는 사용된 도구에 대한 정보를 함께 기록한다.

결함은 단위 테스트, 버그, 컴파일 에러 등을 수집함으로써 자동 수집된다. 현재 버전에서는 실패한 단위 테스트, 컴파일 에러, 런타임 에러를 자동으로 수집하여 결함 일지의 항목으로 기록한다. 표 1에 나타나 있듯이 수집된 결함 데이터에 대해서 제거 단계, 상세 설명, 결함 종류, 발견된 날짜 정보가 자동으로 결함 일지에 기록된다. 또한 보다 다양한 결함 분석을 할 수 있도록 결함이 발생된 소스 파일 정보를 추가로 기록한다.

자동으로 수집된 결함 데이터를 보여주는 결함 일지 화면의 일부가 그림 2에 나타나 있다. 현재 버전에서는 JUnit을 이용한 단위 테스트의 결과, 자바 컴파일 에러, 자바 프로그램 실행 중에 발생하는 자바 예외(exception)의 수집을 Eclipse 센서가 담당하고 있다.

표 1. 실패한 단위 테스트, 컴파일 에러, 런타임 에러의 결함 정보[1]

	Failed unit tests	Compile errors	Runtime errors
Remove phase	Test	Compile	Test
Description	The stack trace of the exception	The description of the syntax error	The stack track of the exception
Defect type	The exception type	Syntax	The exception type
Found date	(automatic)	(automatic)	(automatic)
Inject phase	(manual)	(manual)	(manual)
Fix time	(manual)	(manual)	(manual)
Resource	The test source file, the failed unit test method	The source file	The source file

소프트웨어 크기는 라인 카운팅 도구에 의해 LOC (Lines Of Codes)로 측정되어 자동 수집될 수 있다. 현재 버전은 LOCC로 측정된 LOC결과

<< Data View - View DefectLog

Defect Log: Lists your defects for the given day... View

Project:

Start: End:

Number	Project	FoundDate	Type	Inject	Remove	Fix Time	Desc
1	Jasmine	2007-03-02	Syntax Error		compile	0	fdfdsf cannot be resolved
2	Jasmine	2007-03-02	Syntax Error		compile	0	Syntax error, insert "AssignmentOperator Expression" to complete Assignment
3	Jasmine	2007-03-02	Syntax Error		compile	0	Syntax error, insert ";" to complete BlockStatements
4	Jasmine	2007-03-02	java.lang.Error		test	0	Exception in thread "main" java.lang.Error: Unresolved compilation problems: Syntax error, insert
1	Jasmine	2007-03-10	Syntax Error		compile	0	Syntax error, insert "AssignmentOperator Expression" to complete Assignment
2	Jasmine	2007-03-10	Syntax Error		compile	0	Syntax error, insert ";" to complete BlockStatements
3	Jasmine	2007-03-10	Syntax Error		compile	0	fdfdsf cannot be resolved
4	Jasmine	2007-03-10	java.lang.Error		test	0	Exception in thread "main" java.lang.Error: Unresolved compilation problems: Syntax error, insert

그림 2. 결함 일지[1]

를 자동으로 수집한다. 전적으로 센서에 의해서 수집되는 시간, 결함과 달리 소프트웨어 크기 수집을 위해서는 개발자가 라인 카운팅 도구를 실행해야 하는데 이를 위해 Eclipse 센서에서 그림 3과 같이 Eclipse 프로젝트에 대해서 라인 카운팅 도구를 실행할 수 있게 하는 메뉴를 제공하고 있다.

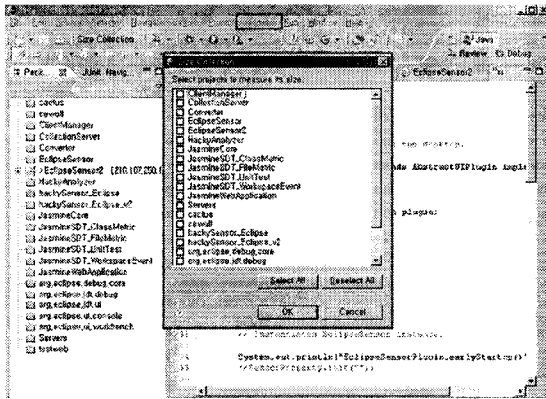


그림 3. 크기 수집 메뉴[1]

2.4 데이터 수동 입력 지원

PPMT는 자동으로 수집되지 못하는 설계/코드 검토 시간, 설계 검토에서 발견된 결함 등의 데이터를 기록하거나 기존 데이터를 수정할 수 있게 하는 기능을 제공한다.

2.5 계획 및 획득가치 추적 지원

PSP에서 개발자들은 계획단계에서 작업 및 스케줄 계획을 세우고 획득가치를 이용하여 프로젝트 진행상황을 추적하는 것이 요구된다. 이러한 활동을 도와주기 위해 PPMT는 작업 및 스케줄 계획 템플릿 작성을 위한 폼을 제공하고, 수집된 시간 일지의 데이터와 계획 데이터를 바탕으로 획득가치를 자동으로 계산한다.

2.6 데이터 분석 및 리포트 생성

PPMT는 개발자의 요청에 따라 수집된 데이터에 대해 차트 또는 표 형태로 다양한 데이터 분석과 분석 결과를 요약해서 보여주는 리포트를 제공한다. 제공되는 분석에는 시간 경과에 따른 데이터의 변화를 보여주는 경향차트(Trend Chart), 계획가치(Planned Value)와 획득가치를 보여주는 획득가치차트, 결함 분석을 위한 파레토 차트(Pareto Chart)가 있다. 또한 단계 결함률(Phase Yield), A/FR(Appraisal to Failure Ratio) 등의 품질 측정변수(Quality Measure) 분석이 제공된다.

3. 정량적인 식스 시그마 프로젝트 관리를 위한 프레임워크 및 도구의 개발

3.1 연구의 필요성 및 기대효과

식스 시그마는 등장 초기에 주로 제조업 분야에서 활용되어왔지만, 최근에는 소프트웨어 개발 및 소프트웨어 품질 보증에서도 이를 활용할 수 있을 것이라 보고 이의 적용을 위한 활발한 연구가 이루어지고 있다[2,3]. 그러나 이러한 식스 시그마의 활발한 도입 및 활용에도 불구하고 식스 시그마 프로젝트의 수행을 지원하는 소프트웨어 도구는 흔치 않았다. 이러한 이유로 정량적인 식스 시그마 프로젝트 관리에 대한 필요성이 제기 되었고, 이를 위해 디지털 식스 시그마가 등장하게 되었다[4]. 그러나 디지털 식스 시그마에서도 IT 기술의 도입과 이를 통한 핵심 프로세스의 추적에 대해서만 언급하고 있다. 이러한 이유로 식스 시그마 프로젝트의 전체 프로세스에 걸친 통합적인 식스 시그마 프로젝트에 대한 측정 및 분석, 측정 및 분석결과에 대한 저장 및 리포팅 기능을 지원하는 프레임워크나 도구가 필요하다. 이러한 이유로 이러한 기능들을 지원하는 정량

적인 식스 시그마 프로젝트 관리 프레임워크에 대한 연구 및 이 프레임워크에 기반한 식스 시그마 프로젝트 지원도구의 개발을 진행하고 있으며, 이를 본 장의 나머지에서 소개한다. 식스 시그마 프로젝트 지원도구의 개발을 통해 식스 시그마 프로젝트를 수행하는 조직이나 사용자는 정량적인 프레임워크와 도구의 사용에 바탕을 둔 식스 시그마 프로젝트의 관리를 통해 식스 시그마 프로젝트의 정량화, 체계화, 통합화라는 이득을 얻을 수 있을 것이다.

3.2 정량적 식스 시그마 프로젝트 관리를 위한 프레임워크

식스 시그마 프로젝트에서는 해결해야 할 문제의 식별과 함께 문제를 같이 해결할 프로젝트 팀을 구성해야 하며, 프로젝트 팀은 벨트 시스템이란 식스 시그마만의 독특한 시스템에 의해 구성된다. 즉 식스 시그마 프로젝트를 통해 해결할 문제를 파악하고 식스 시그마 프로젝트팀의 블랙벨트들을 가이드 하는 마스터 오브 블랙 벨트(Master of Black Belt), 식스 시그마 프로젝트를 전담하며 프로젝트를 이끄는 블랙벨트(Black Belt), 일반 업무와 식스 시그마 프로젝트를 동시에 수행하는 실무자인 그린벨트(Green Belt), 그리고 경영진으로서 식스 시그마 프로젝트의 수행을 지원하고 적절한 조언을 하는 챔피언(Champion)이 그것이다.

또한 식스 시그마 프로젝트는 DMAIC나 DFSS와 같은 식스 시그마 방법론에 기반하여 진행되며, DMAIC 방법론의 경우 정의(Define), 측정(Measure), 분석(Analyze), 개선(Improve), 통제(Control)의 다섯 단계를 통해 다양한 데이터에 대한 측정 및 통계적 분석에 기반하여 기존의 프로세스에서의 문제 식별 및 해결을 통한 기존 프

로세스를 향상시킨다[5]. DFSS(Design For Six Sigma)는 새로운 제품이나 서비스의 개발을 위해 제시된 식스 시그마 방법론이며 정의(Define), 측정(Measure), 분석(Analyze), 설계(Design), 검증(Verify)의 다섯 단계로 이루어지며, 이러한 일련의 다섯 단계를 거친 식스 시그마 프로젝트를 통해 새로운 제품이나 서비스의 개발을 정량적, 체계적으로 수행한다[6]. 정량적인 식스 시그마 프로젝트 관리 프레임워크에서의 각각의 요소간의 관계와 다른 어플리케이션이나 데이터와의 관계를 나타내면 그림 4와 같다.

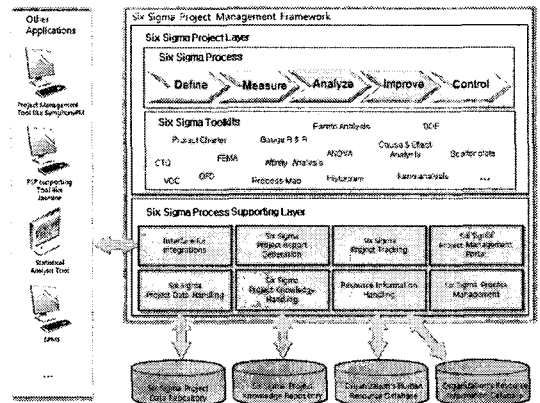


그림 4. 정량적인 식스 시그마 프로젝트 관리를 위한 프레임워크

정량적인 식스 시그마 프로젝트 관리 프레임워크의 중심에는 DAMIC 방법의 식스 시그마 프로젝트가 있으며, 이 방법의 각 단계에서 필요한 워크시트, 템플릿의 구현을 통하여 이들 단계에서의 데이터의 측정을 돕는다. 또한 각 단계에서 수행해야 할 각종 식스 시그마 도구를 프레임워크 내에서 구현하여 사용자가 식스 시그마 프로젝트를 수행하며 수집하는 데이터에 대한 측정 및 분석을 프레임워크 내에서 체계적이고 통합적으로 수행하도록 한다.

한편 다양한 통계적 분석 도구 및 기법 혹은 프로세스와 관련된 도구 및 기법들이 식스 시그마에서 사용되며 식스 시그마 도구란 이름으로 불리고 있으며, 식스 시그마 프로젝트 관리 프레임워크에서는 이러한 각종 식스 시그마 도구 중 기본적으로 필수적인 것들을 구현하여 식스 시그마 프로젝트에서의 데이터 수집 및 측정을 지원하고 수집된 데이터에 대한 분석 및 원인의 파악, 문제의 해결을 위한 방안의 도출을 용이하도록 한다. 이러한 문제의 해결 과정을 돕는, 즉 소프트웨어 식스 시그마에서의 경우 소프트웨어 품질 향상을 꾀하는 식스 시그마 프로젝트에서 유용한 식스 시그마 도구에 대한 연구가 몇몇 연구자에 의해 수행되었으며, 이러한 연구의 결과 소프트웨어 식스 시그마에서 유용하다고 도출된 식스 시그마 도구로는 다음의 표 2과 같은 것이 있다 [2,3].

표 2. 소프트웨어 품질 향상에 유용한 식스 시그마 도구

단계	도구
정의	프로젝트 차터
	FEMA
	QFD
측정	Gauge R&R
분석	파레토 분석
	회귀 분석
	상관 분석
	산점도
개선	원인 및 영향 다이어그램
통제	컨트롤 차트

그러나 정량적인 식스 시그마 프로젝트 관리 프레임워크에서 아무리 많은 식스 시그마 도구를 구현하여 다양한 측정 및 분석 기능을 제공한다 하더라도, 각각의 기능에 특화된 전용 어플리케이션만큼 다양하고 세부적인 기능을 제공하기는

힘들다. 이러한 이유로 다른 여타의 어플리케이션과의 연동이 중요하게 되며, 식스 시그마 프레임워크와 연동될 다른 어플리케이션의 예로는 MS Project 등의 프로젝트 관리 도구, PSP 지원 도구와 같은 데이터 측정 도구, Minitab과 같은 통계적 분석 도구 및 MS Visio와 같은 다양한 다이어그램 도구가 있다. 예를 들면, 정량적인 식스 시그마 관리 프레임워크는 프로젝트 관리 도구와의 연동을 통해 프로젝트 관리 도구에 저장된 조직의 인사정보 및 기존의 프로젝트 수행에 대한 각종 데이터를 획득할 수 있으며 이를 통해 식스 시그마 프로젝트 중의 문제 정의 및 원인 파악을 위한 근거로서 활용 가능하다. 또한 프로젝트 관리 도구에서 제공하는 일반적인 프로젝트에서의 일정 관리와 관련된 Gantt 차트나 PERT 차트와 같은 도구는 정량적인 식스 시그마 프레임워크에서의 식스 시그마 프로젝트 일정 관리를 위한 목적으로 사용될 수 있다. 또한 PSP 지원 도구는 PSP의 수행을 통해 개인 프로세스와 관련된 데이터의 수집을 수행하며 이러한 개인 프로세스와 관련된 데이터 역시 정량적인 식스 시그마 관리 프레임워크에서 문제의 식별 및 원인의 파악, 해결방안의 도출에 활용될 수 있다. 이렇듯이 정량적인 식스 시그마 프레임워크와 다양한 어플리케이션과의 연동은 소프트웨어 개발 프로세스 향상을 통한 소프트웨어 품질 향상 및 고객 만족 실현이라는 식스 시그마 프로젝트의 근본 목적을 충실히 달성할 수 있게 된다.

3.3 식스 시그마 프로젝트 지원도구

현재 식스 시그마 프로젝트 지원도구란 이름으로 본 논문에서 기술한 정량적인 식스 시그마 프로젝트 관리 프레임워크를 실질적인 도구로 구현하는 작업을 진행 중이다. 식스 시그마 프로젝트

지원도구는 DMAIC 방법론을 지원하며 DMAIC의 각 단계에서의 다양한 워크시트나 템플릿, 다양한 식스 시그마 도구의 지원을 통해 정량적이고 통합적인 식스 시그마 프로젝트의 수행을 가능케 하는 요소로서 작용할 수 있을 것이다. 또한 MS Project와 같은 프로젝트 관리도구, Minitab과 같은 통계분석도구, PSP 지원도구, BPMS 등의 다양한 외부 어플리케이션과의 연동을 지원하여 다양한 기능을 사용자에게 제공하도록 한다.

그림 5는 식스 시그마 프로젝트의 정의 단계에서 사용되는 프로젝트 평가에 대한 워크시트를 식스 시그마 지원도구에서 구현한 것이며, 사용자로 하여금 프로젝트 평가 요소의 점수와 가중치를 입력하도록 하여 프로젝트에 대한 평가를 점수화하는 도구이다.

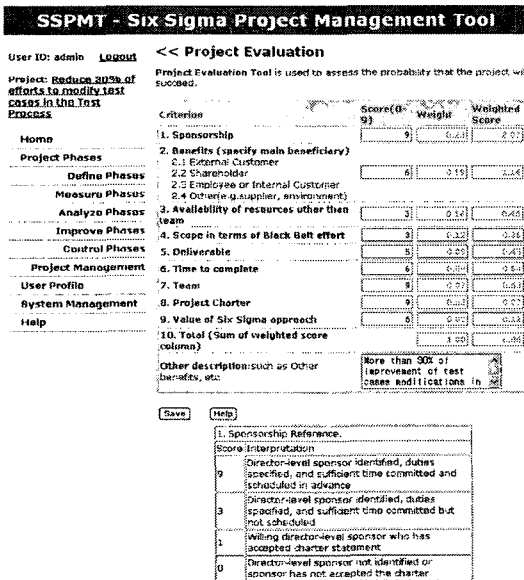


그림 5. 식스 시그마 지원도구의 프로젝트 평가 화면

3.4 향후 연구

본 연구에서는 정량적인 식스 시그마 프로젝트 관리를 위한 프레임워크와 이에 기반한 도구의

개발을 제안하였다. 이를 통해 식스 시그마 프로젝트를 수행중인 조직이나 사용자는 소프트웨어 도구를 통한 식스 시그마 프로젝트의 관리를 통해 식스 시그마 프로젝트의 정량화, 체계화, 통합화라는 이득을 얻을 수 있을 것이다. 이러한 정량화, 체계화, 통합화는 보다 효과적이고 효율적인 식스 시그마 프로젝트의 수행을 위해서 반드시 필요한 부분이며, 식스 시그마 프로젝트를 통해 얻게 될 비즈니스 가치의 향상에 도움을 줄 것이다[7].

또한 본 연구에서 제안하는 정량적인 식스 시그마 프로젝트 관리를 위한 프레임워크와 식스 시그마 프로젝트 관리 도구를 확장하여 더욱 다양한 식스 시그마 방법 및 식스 시그마 도구를 구현하는 것에 대한 연구를 통해 성공적인 소프트웨어 식스 시그마의 도입 및 이를 통한 소프트웨어 품질 향상, 더 나아가 기업의 경쟁력 향상에 도움이 될 수 있을 것이다.

4. 식스 시그마 통합 프레임워크

4.1 개발 배경

PSP/TSP가 개인과 팀 차원에서 소프트웨어 개발에 사용할 수 있는 구체적인 기법들을 기술하고 있지만 PSP/TSP에서 수집되는 메트릭에 대한 분석기법은 여전히 부족하다. 따라서 PSP/TSP 수행 시 발생할 수 있는 문제를 방지하고 프로세스가 지속적이고 효율적으로 변경 및 유지 관리될 수 있도록 하기 위해서는 식스시그마의 다양한 통계분석도구와 의사결정도구의 사용이 필요하다. 기존에도 식스시그마 도구를 PSP/TSP에 적용하려는 노력은 있었으나 이를 적시적소에 사용하기에는 무리가 있었다. 이는 식스시그마 도구가 PSP/TSP의 다양한 단계에

서 각기 다른 데이터를 요구하는 것에서 비롯되었다.

본 장에서는 이러한 문제점을 해결하기 위해 식스시그마 도구를 PSP/TSP 각 활동에 연계시키는 프레임워크에 대해서 기술한다. 프레임워크의 정의를 위하여 PSP/TSP 요소 파악 및 PSP/TSP에 적합한 식스시그마 도구를 선정하였다. 다음으로는 각 선정된 식스시그마 도구를 적절한 PSP/TSP 활동에 연관시킴으로써 프레임워크를 개발하였다[8].

4.2 TSP/PSP에 적합한 식스시그마 도구

식스시그마 프로젝트 진행 시 사용하는 식스시그마 도구의 표준집합은 존재하지 않으며 도구의 선택은 접근법이나 조직에 따라서 달라진다. 또한 소프트웨어에 적용할 수 있는 모든 식스시그마 도구들이 PSP/TSP에 적용될 수 있는 것이 아니기 때문에 프레임워크 개발을 위해서는 PSP/TSP에 적합한 식스시그마 도구를 우선적으로 선별하는 것이 필요하다.

PSP/TSP에 적합한 식스시그마 도구 선택을 위해 다양한 PSP/TSP 요소를 고려하였다. PSP/TSP 매트릭, 추정 모델, 목표 등 분석이 필요한 요소들을 우선적으로 선별하였으며 표 3에 지속적인 프로세스 개선을 위해 분석이 필요한 PSP/TSP 요소, 해당 요소에 필요한 분석의 종류, 분석을 수행하는 식스시그마 도구들을 차례로 나열하였다.

예를 들어 PSP/TSP에서 결함 방지는 발생하는 결함들을 우선순위화 하여 가장 많이 발생하는 결함 형태를 파악한 후 결함의 원인을 추적하고 해당 원인을 방지할 수 있는 대책 마련이 필요하다. 이러한 대책에는 해당 결함원인이 다시 발생하는 것을 방지하기 위해서 설계표준이나 코딩

표준을 갱신하는 것을 예로 들 수 있다. 이때 PSP/TSP에서 결함 데이터의 우선순위화에는 Pareto analysis를 사용할 수 있으며 결함의 원인 파악에는 cause and effect diagram을 사용할 수 있다. 이와 같이 PSP/TSP에서 분석이 필요한 요소를 파악하고 해당 요소에 필요한 분석을 정의한 후 분석을 수행하는 식스시그마 도구를 파악하여 표 3에서 보이는 것과 같이 총 13개의 식스시그마 도구를 식별하였다.

표 3. PSP/TSP 요소와 식스 시그마 도구[8]

PSP/TSP Elements		Corresponding Six Sigma Tools
PSP/TSP measures	Defect measure	Pareto analysis
		Cause and effect diagram
	Other measures	Control charts
PROBE		Scatter plots, correlation analysis, regression analysis
Process		ANOVA
Customer needs		Process mapping
Estimation data		Kano analysis, SQFD, affinity diagram
Project risks		Two-sample t-test
		SWFMEA

PSP/TSP에 적합한 식스시그마 도구 선별과 더불어 하나의 식스시그마 도구가 여러 개의 다른 이름으로 사용되기 때문에 대표 이름 선택 또한 필요하다. 다양한 이름으로 사용되는 도구의 경우에는 가장 널리 사용되는 이름을 선별하였으며 소프트웨어 사용을 목적으로 이름이 따로 존재하는 식스시그마 도구의 경우에는 소프트웨어 적용을 위해 명명된 이름을 선별하였다.

PSP/TSP에 적합한 13개의 식스시그마 도구 중 일부는 팀 기반 도구이다. 이러한 팀 기반 도구는 affinity diagram, SQFD, Kano analysis,

SWFMEA이며 이러한 도구는 팀을 기반으로 하는 TSP에는 적용가능 하나 개인 개발자들을 대상으로 하는 PSP에는 적용 가능하지 않다. 또한 PSP에는 process mapping 을 사용하여 가시화하고 분석할만한 복잡한 프로세스가 존재하지 않기 때문에 PSP 에 적합한 도구에서 제외하였다. ANOVA는 선형회귀의 적합성 판단을 위해 사용할 수 있으나 TSP에는 선형회귀가 포함되지 않기 때문에 TSP에 적합하지 않다. 따라서 PSP에 적합한 식스시그마 도구는 팀 기반도구와 process mapping을 제외한 8개 도구이며 TSP에는 ANOVA를 제외한 12개의 도구가 적용 가능하다. 표 4는 PSP/TSP에 적합한 식스시그마 도구, 해당 도구에 대해 설명하고 있는 대표 참고문헌, PSP/TSP 에 해당 도구의 적용 가능성을 보여준다.

표 4. PSP/TSP 에 적합한 식스 시그마 도구(8)

Six Sigma Tools for PSP/TSP	Tool Applicability		Basic Reference
	PSP	TSP	
Process mapping	No	Yes	[9]
Pareto analysis	Yes	Yes	[10]
Cause and effect diagram	Yes	Yes	[11]
Control charts	Yes	Yes	[12]
Analysis of variance (ANOVA)	Yes	No	[13]
Two-sample t-test	Yes	Yes	[14]
Scatter plots	Yes	Yes	[13]
Correlation analysis	Yes	Yes	[13]
Regression analysis	Yes	Yes	[13]
Affinity Diagram	No	Yes	[15]
Software Quality Function Deployment (SQFD)	No	Yes	[16]
Kano analysis	No	Yes	[17]
Software Failure Modes and Effects Analysis (SWFMEA)	No	Yes	[18]

4.3 식스 시그마 도구와 PSP 활동과의 관계

프레임워크 정의를 위해서는 PSP에 적합한 식스시그마 도구와 PSP 활동과의 연관관계를 파악하는 것이 필요하다. 식별된 식스시그마 도구의 특성에 기반하여 같은 식스시그마 도구가 적용되는 PSP 활동을 그룹화 함으로써 식스시그마 도구가 사용되는 PSP 활동들을 파악할 수 있다. 이러한 연관관계 파악은 식스시그마 도구의 입력 데이터에 기반하여 각각의 도구에 입력 데이터를 제공하는 PSP활동들을 그룹화함으로써 수행하였다. 다음은 식스시그마 도구와 PSP에서 도구 사용을 위해 필요한 입력, 입력을 제공하는 PSP 활동을 보여준다.

그림 6은 PSP에 적합한 식스시그마 도구와 각각의 식스시그마 도구가 사용되는 PSP 활동을 보여준다. 왼쪽에 나열된 식스시그마 도구를 해당 PSP활동에 사용함으로써 해당 활동에서 산출된 매트릭이나 데이터를 분석하게 된다. 그림에서 PSP의 Postmortem 단계에서 가장 많은 식스시그마 도구들이 사용되는 것을 볼 수 있는데 이는 많은 식스시그마 도구들이 데이터 분석에 사용되고 PSP에서 데이터 분석이 Postmortem에서 수행되기 때문이다.

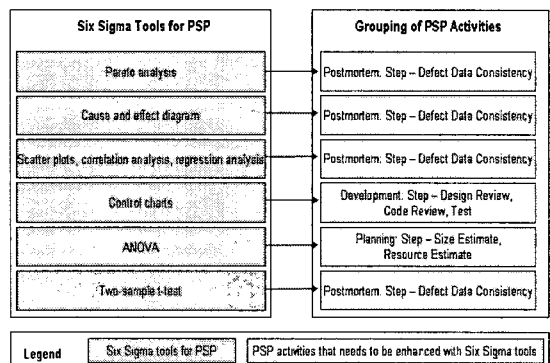


그림 6. PSP 활동과 식스 시그마 도구의 대응(8)

4.4 식스 시그마 도구와 TSP 활동과의 관계

PSP활동이 식스시그마 도구의 입력에 기반하여 그룹화된 것과 같이 TSP 활동 또한 식스시그마 도구의 입력에 기반하여 그룹화 시킴으로써 TSP를 위해 식별된 식스시그마 도구들이 어떠한 TSP활동에서 사용될 수 있는지 파악할 수 있다. TSP는 총 8개의 단계로 구성되며 각각의 단계는 다른 많은 활동으로 구성되어 있다.

그림 7는 TSP에 적합한 식스시그마 도구가 어떤 TSP 활동에서 사용되는지를 보여준다. TSP에 적합한 식스시그마 도구들은 모든 TSP 활동에서 적용되어 지속적인 프로세스 개선을 위해 사용될 수 있다. PSP와 마찬가지로 TSP의 Postmortem 에서 식스시그마 도구들이 가장 많이 사용되는데 이는 많은 식스시그마 도구들이 데이터 분석에 사용되며, TSP의 데이터 분석은 Postmortem에서 수행되기 때문이다.

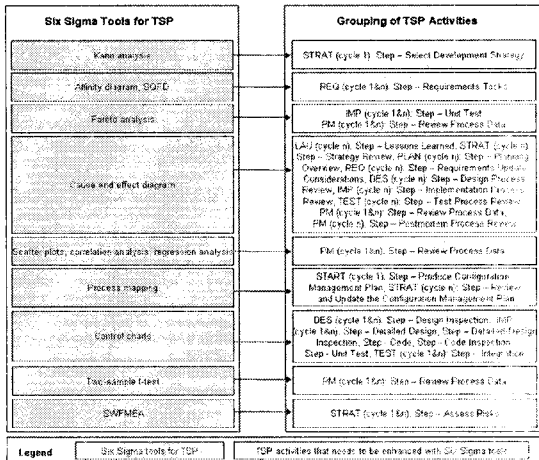


그림 7. TSP 활동과 식스 시그마 도구의 대응(8)

4.5 PSP/TSP - 식스 시그마 프레임워크의 목적

PSP/TSP-식스시그마 프레임워크는 PSP/TSP를 사용하여 개인과 팀 프로세스 개선 수행 시

지속적이고 효과적인 프로세스 개선을 위해 필요한 식스시그마 도구와 PSP/TSP 활동 사이의 연관관계와 더불어 해당 활동에서 각 도구의 목적을 기술한다. 소프트웨어 개발자와 팀의 프레임워크 활용을 돕기 위해 두 가지 관점에서 프레임워크를 제공한다.

- 식스시그마 도구 관점: 식스시그마 도구 관점은 PSP/TSP에 적합한 식스시그마 도구를 먼저 나열하고 해당 도구가 사용되는 PSP/TSP 활동을 기술한다. 이는 개발자가 특정 식스시그마 도구에 대한 지식이 있으며 해당 도구가 적용되는 PSP/TSP 활동을 알고 싶을 때 유용하다. 이 관점을 이용하여 소프트웨어 개발자나 팀은 특정 식스시그마 도구의 특성을 알 수 있다.

- PSP/TSP 활동 관점: PSP/TSP 활동 관점은 소프트웨어 개발자나 팀이 PSP/TSP 활동을 기준으로 각 활동에서 사용할 수 있는 식스시그마 도구를 알고 싶을 때 사용할 수 있다.

표 5와 표 6는 PSP/TSP 활동 관점에서의 PSP/TSP - 식스시그마 프레임워크를 보여준다. 두 관점 모두 같은 내용을 다루고 있기 때문에 본 논문에서는 PSP/TSP 활동 관점만을 제공한다. PSP - 식스시그마 프레임워크는 각각의 PSP 단계에서 사용하는 식스시그마 도구와 목적, 도구의 사용법에 대해 기술하고 있는 참고문헌을 나열한다. TSP - 식스시그마 프레임워크에서는 각각의 TSP 단계와 활동, 해당 활동이 속해있는 주기, 각 단계에서 사용하는 식스시그마 도구와 목적을 차례대로 기술한다. 표 6는 각 단계 중 일부를 예시로 나타낸다.

5. 결론

본 논문에서는 개발자들의 PSP 수행 지원을 목적으로 개발된 PPMT도구와 식스 시그마 지원도

표 5. PSP - 식스시그마프레임워크: PSP 활동관점[8]

PSP Phase	PSP Step	Six Sigma Tools	Purpose	Basic Reference
Planning	Size Estimate	ANOVA	Determination of the goodness of "fit" of PROBE	[13]
	Resource Estimate			
Development	Design Review	Control charts	Statistical process control, evaluation of the significance of a process change	[12]
	Code Review			
	Test			
Postmortem	Defect Data Consistency	Cause and effect diagram	Identification of root causes	[11]
		Scatter plots	Identification of a correlation between two measures	[13]
		Two-sample t-test	Determination of estimation accuracy	[14]
		Correlation analysis	Test of statistical significance of the relationship between two measures	[13]
		Regression analysis	Description of relationship between two measures precisely by means of an equation that has predictive value	[13]
		Pareto analysis	Prioritization of defects types	[10]

표 6. TSP - 식스시그마프레임워크: TSP 활동관점(Partial)[8]

TSP Phase	TSP Cycle	TSP Step	Six Sigma Tools	Purpose	Basic Reference
Team Launch	1	Lessons Learned	Cause and effect diagram	Identification of root causes	[11]
Development Strategy	1	Select Development Strategy	Kano analysis	Prioritization of product functions	[17]
	1&n	Assess Risks	SWFMEA	Identification of possible risks	[18]
	n	Strategy Review	Cause and effect diagram	Identification of root causes	[11]
	1	Produce Configuration Management Plan	Process mapping	Documentation of process flow	[9]
	n	Review and Update the Configuration Management Plan			

구, 식스 시그마 통합 프레임워크를 제안하였다. PPMT는 수동 데이터 입력의 문제점을 완화하기 위해 데이터의 수집을 자동으로 수행할 뿐만 아니

라, 수집된 데이터의 다양한 통계적 분석을 제공한다. 이러한 기능을 통해 개발자들은 개인 프로세스의 약점을 파악하여 프로세스 개선 계획을

세울 수 있고, 계획이 올바르게 수행되었는지 또는 어떠한 효과를 가져왔는지를 파악할 수 있다.

PSP/TSP에서 제공하는 매트릭 분석기법의 부족함은 식스 시그마 지원도구를 통해 보완할 수 있다. 식스 시그마 지원도구는 프로젝트에 대한 측정 및 분석, 분석결과에 대한 저장 및 리포팅 기능을 지원함으로써 정량적인 소프트웨어 품질 관리를 도와준다. 그리고 PSP/TSP의 적시적소에 식스시그마 기법을 적용할 수 있도록 프레임워크를 제공한다.

이와 같은 소프트웨어 품질을 개선하기 위한 도구의 구현과 프레임워크의 연구는 앞으로 해결해야 할 많은 숙제를 가지고 있다. 하지만 정량적인 프로세스 관리 수행에 대한 가능성을 제시하고 있고, 이러한 도구를 이용하여 축적한 데이터의 분석은 향후 프로젝트에서 발생하는 문제점을 해결하는 데에 도움을 줄 것이다.

참 고 문 헌

- [1] 신현일, "Jasmine: PSP 지원도구," 2007 한국 소프트웨어공학 학술대회 논문집, 제9권 제1호. pp. 217-225.
- [2] Cvetan Redzic and Jongmoon Baik, "Six Sigma Approach in Software Quality Improvement," International Conference on Software Engineering Research, Management and Applications (SERA'06), August 2006.
- [3] Amanda Harte, "Six Sigma Software Development," AUERBACH, 2002.
- [4] Howard Smith and Peter Fingar, "Digital Six Sigma: Integrating continuous improvement with continuous change and continuous learning," <http://www.bpmi.org/downloads/LJB-2003-12-1.pdf>.
- [5] Donald P. Lynch, Suzanne Bertolino and Elaine Cloutier, "How To Scope DMAIC Projects," Quality Progress, Vol. 36, No. 1, pp. 37-41, January 2003.
- [6] Thomas Pyzdek, "The Six Sigma Handbook : A Complete Guide for Green Belts, Black Belts, and Managers at All Levels," McGraw-Hill, 2003.
- [7] Frank T. Anbari and Young Hoon Kwak, "Success Factors in Managing Six Sigma Projects," Project Management Institute Research Conference, 2004.
- [8] 박영규, "PSP/TSP - 식스시그마 통합 프레임워크," 2007 한국 소프트웨어공학 학술대회 논문집, 제9권 제1호. pp. 190-199.
- [9] Damelio, Robert, "The Basics of Process Mapping," Portland, Oregon: Productivity Inc, 1996.
- [10] Scholtes, P.R. et al., "The Team Handbook," pp. 2-25. Madison, WI: Joiner Associates, 1988.
- [11] Brassard, M., "The Memory Jogger: A Pocket Guide of Tools for Continuous Improvement," pp. 24-29. Methuen, MA: GOAL/QPC, 1988.
- [12] Wheeler, D.J., & Chambers, D.S., "Understanding Statistical Process Control (2nd Ed.)," Knoxville, TN: SPC Press, 1992.
- [13] Thomas Pyzdek, "The Six Sigma Hand book (II Ed.)," McGraw Hill, 2003.
- [14] Miller, Rupert G. Jr., "Beyond ANOVA: Basics of Applied Statistics," New York: John Wiley & Sons, 1986.
- [15] Beyer, H., Holtzblatt, K., "Contextual Design: Defining Customer-Centered Systems", Morgan Kaufmann, 1998.
- [16] Haag, S., M.K. Raja and L.L. Schkade, "Quality Function Deployment Usage in Software Development," Communications of the ACM, 39(1), pp. 41-49, 1996.
- [17] Kano, Noriaki, Seraku, N., Takahashi, F., and Tsuji, S., "Attractive and Normal Quality," Quality, (14:2), pp. 39-48, 1984.
- [18] Reifer, Donald J., "Software Failure Mode and Effects Analysis," IEEE Transactions on Reliability, Vol. 28, No. 3, August 1979.



송 일 선

- 2007년 한국기술교육대학교 디지털시스템(공학사)
- 2007년~현재 한국정보통신대학교 공학부 석사과정 재학중
- 관심분야 : 소프트웨어 공학, 인공지능



정 경 학

- 2007년 단국대학교 전기전자컴퓨터공학부(공학사)
- 2007년~현재 한국정보통신대학교 공학부 석사과정 재학중
- 관심분야 : 소프트웨어 공학, 인공지능



박 현 철

- 2001년 숭실대학교 컴퓨터학부 (공학사)
- 2001년~2003년 헨디소프트 주임연구원
- 2006년~현재 한국정보통신대학교 석사과정 재학중
- 관심분야 : Software Process, Software Six Sigma, Software Testing, Software Measurement



최 호 진

- 1982년 서울대학교 컴퓨터공학과 학사
- 1985년 영국 뉴캐슬대학교 컴퓨팅학과 석사
- 1995년 영국 임페리얼공대 컴퓨팅학과 박사
- 1982년~1989년 ㈜데이콤 중앙연구소 선임연구원
- 1995년~1996년 임페리얼공대 박사후연구원
- 1997년~2002년 한국항공대학교 전자공학부 교수
- 2003년~2003년 미국 카네기멜론대학교 방문교수
- 2002년~현재 한국정보통신대학교 공학부 교수
- 2003년~현재 카네기멜론대학교 겸임교수
- 관심분야 : 유비쿼터스컴퓨팅, 인공지능, 소프트웨어공학, 그리드컴퓨팅, 데이터마이닝