

CMMI 기반의 프로세스 개선을 위한 6시그마 활용방안

김 한 샘[†] · 한 혁 수^{**}

요 약

CMM/CMMI를 이용하여 프로세스를 정립하고 더 나은 프로세스로 개선하려는 조직이 국내외적으로 많이 늘고 있다. 그러나 SEI에서 제시하는 CMMI와 IDEAL은 각각 프로세스 성숙도와 개선에 초점을 맞춰 우수 프랙티스(Best Practice)들을 나열하고 권고하고 있지만, 구체적인 구현방법이나 도구들에 대해서는 다루고 있지 않다. 이에 본 논문에서는 6시그마의 프로세스 개선 방법인 DMAIC 과정과 여기서 사용되는 다양한 도구 및 통계적 방법들을 프로세스 개선영역 식별과 문제도출 및 문제우선순위 파악에 적용하는 방법을 제시하였다. 제시된 방법은 프로세스의 내용에 대해서는 CMMI를 기반으로 하며 개선의 기법으로서 6시그마를 이용함으로써, CMMI를 기반으로 프로세스의 구현을 시작하려는 조직이나, 구체적인 개선 방법이 필요한 조직, 체계적인 개선사항의 식별 및 정량적 개선결과와 파악이 필요한 조직에서 활용될 수 있다.

키워드 : 소프트웨어 프로세스, CMMI, 소프트웨어 프로세스 개선, IDEAL, 6 시그마, DMAIC, 품질기능전개, 고장유형영향분석

Applying 6 sigma techniques in CMMI based software process improvement

Han-Saem Kim[†] · Hyuk-Soo Han^{**}

ABSTRACT

There are increasing numbers of foreign and domestic organizations that are using CMM/CMMI to establish their processes and keep improving them. CMMI and IDEAL models of SEI provide the best practices of processes and guide the organization using them based on processes maturity levels. However, they do not deal with their tools or methods that describe how to implement the processes in the organization.

Therefore, in this paper, we developed a method in which various tools and statistical methodology of 6 sigma are applied to identify the process areas to be improved, to extract problems in those areas and to prioritize them.

We expect this paper can contribute to the organizations that are searching for practical way of implementing CMMI based software process improvement and of identifying improvement items systematically. Also this method will be used to understand the result of improvement activities quantitatively.

Key Words : Software Process, CMMI, Software Process Improvement(SPI), IDEAL, 6 sigma, DMAIC, QFD, FMEA

1. 서 론

소프트웨어는 현재 다양한 제품과 기기에 포함되어 사용되면서 우리생활에서 없어서는 안될 중요한 부분으로 인식되고 있다. 그러나 개발환경의 다양화, 내용의 복잡화로 인해서 소프트웨어 프로젝트 관리에 많은 어려움을 겪고 있다[1]. 이러한 어려움을 극복하고자 프로젝트 및 프로세스 관리 기법 등이 사용되고 있다. 그 중 소프트웨어 프로세스 관리 및 개선을 위한 도구 중 하나로 CMMI(Capability Maturity Model Integration)[2]가 도입되어 국내외의 많은 조직들이 소프트웨어 개발 및 유지보수 프로세스를 제도화

하고 내재화(Institutionalization)시키고 있는 상태이다[3].

프로세스 모델들은 다양한 조직의 실무자들이 경험한 바를 종합하여 우수 프랙티스(Best Practice)들을 제공함으로써, 프로세스 구현 및 관리를 지원한다. 특히 프로세스 관리 활동을 다루는 모델들이 빠지지 않고 강조하고 있는 사항은 구현된 프로세스에 안주하는 것이 아닌 프로세스를 지속적으로 점검하고 개선 하는 것이다.

이러한 까닭에 최근 소프트웨어 개발 기업들은 소프트웨어 프로세스를 조직문화로 내재화시키는 동시에 조직 역량의 극대화를 위해 프로세스의 개선을 지속적으로 추진하고 있다. 이의 일환으로 소프트웨어 개발조직도 일반 기업조직과 마찬가지로 6시그마를 적용하여 기업역량의 강화 및 혁신을 꾀하고 있다[4, 5].

그러나 소프트웨어 분야의 개선활동에서는 몇 가지 문제

* 본 연구는 2006년도 상명대학교 소프트웨어미디어 연구소 지원으로 수행되었음.

† 준 회원 : 상명대학교 대학원 컴퓨터과학과 박사과정

** 정 회원 : 상명대학교 소프트웨어학부 교수

논문접수 : 2006년 2월 16일, 심사완료 : 2006년 4월 11일

가 있다. 첫째, SEI에서 배포한 프로세스 개선모델인 IDEAL은 개선의 과정을 자세히 설명하고 있지만 개선활동에 필요한 도구나 방법, 현재상태 및 개선성과의 측정 등을 통한 정량적 개선 방법에 관해서는 설명하고 있지 않다[6]. 둘째, 대부분의 실무자들은 CMMI 기반의 프로세스 개선과 6시그마를 이용한 경영혁신 및 공정개선에 혼돈을 느낀다. CMMI에서 제공하는 소프트웨어개발 조직이 갖춰야 하는 프랙티스들을 적용하고 조직에 맞게 조정(Tailoring)하는 것과 6시그마의 개선활동을 완전히 분리하여 생각하여서 과중한 업무를 발생시키는 결과를 낳거나, 이 두 가지를 통합하는 방법에 대한 어려움을 느끼고 있다. 셋째, 여러 개의 모델을 적용하면서 생기는 업무의 과중 때문에 개선활동을 거부하는 실무자들을 많이 볼 수 있다.

본 논문에서는 이러한 문제를 해결하고자 6시그마의 기법과 프로세스 개선활동과의 통합방안에 대해 연구하였다. 특히 CTQ(Critical to Quality) 즉, 핵심 개선과제를 선정하고 이에 대한 현재수준 파악 및 문제 원인을 식별하는 방법에 관해 연구하였다. 이로써 6시그마 활동을 수행중인 소프트웨어 조직은 CMMI를 통해서 개선의 문제나 목표를 선정할 때 도움을 얻을 수 있고, CMMI를 구현하는 조직은 6시그마를 통해 체계적인 개선 과정을 도움 받고, 개선 동기나 결과를 정량적으로 판단할 수 있다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 프로세스 개선방법들에 대해 설명하고, 3장에서는 본 논문에서 제안하는 6시그마를 이용한 CMMI기반의 프로세스 개선사항을 도출하는 과정을 설명한다. 4장에서는 6시그마를 이용한 개선사항 도출과정을 예로 설명함으로써 연구 내용의 효율성을 보여준다.

2. 프로세스 개선

프로세스(process)에 대해 예전에는 주어진 목적을 달성하기 위한 순서적인 절차 틀[7]이라는 작은 범위로만 내용을 파악하고 사용했으나 최근에는 제품의 생산 및 진화에 사용되는 활동, 방법 등의 집합으로서 인력, 절차, 방법, 장치 및 도구를 통합하여 원하는 목표를 이루는 수단 의미로 사용되고 있다[8]. 좋은 결과물을 얻고자 할 때 그 과정의 중요성을 누구나 인식하듯이 요즘 많은 조직에서는 프로세스의 효과적 관리를 통해 고품질의 제품을 얻고자 한다.

조직에 맞는 효과적인 프로세스를 개발하는 과정, 즉 현재의 프로세스를 점검(as-is)하고 조직의 목표에 맞는 효과적인 프로세스(to-be)를 찾아내는 활동을 프로세스 개선이라고 한다.

프로세스 개선의 기본은 데밍(Deming, W.E.)의 PDCA에서부터 시작된다. 종래의 산업사회에서부터 품질관리에 대한 인식은 이미 있었으며, 그 과정은 QC(Quality Control)에서부터, SQC(Statistical Quality Control), TQM(Total Quality Management), 6시그마로 연결된다. QC 시기에서 개선활동에 대한 인식이 시작되었는데 이것이 PDCA 모델

이며, 6시그마 시기의 프로세스 개선 모델은 DMAIC로 발전되었다. 이러한 프로세스 개선 모델에 견줄 수 있는 소프트웨어 분야의 프로세스 개선 모델로는 SEI가 개발한 IDEAL이 있다.

2.1 PDCA(Plan, Do, Check, Act)

슈하트(W. Shewhart)에 의해 개발되었으나 1950년대 데밍(W. Edwards Deming)에 의해 대중적으로 배포[9]되어 데밍사이클이라고도 불리는 PDCA모델은 제품에 대한 편차를 식별하고 원인을 규명하여 조치를 취하며 지속적인 개선이 가능한 순환적 프로세스로 구성되어 있다[10]. 현재까지도 모든 관리사이클의 기본으로써 인식되고 있는 PDCA는 다음과 같은 과정으로 수행된다.

- 계획(plan): 문제인식을 위한 자료를 수집, 분석하고 개선을 위한 계획을 개발한다. 계획을 평가하기 위한 척도(measures)를 상세히 한다.
- 실행(do): 계획을 이행한다. 이 단계에서 어떤 변화가 있었는지 문서화한다. 평가를 위해 자료를 체계적으로 수집한다.
- 검토(check): 실행단계에서 모아진 자료들을 평가한다. 계획단계에서 설정된 원래 목표와 결과가 얼마나 밀접하게 부합되었는지를 확인한다.
- 조치(act): 만일 결과가 성공적이었다면 새로운 방법을 표준화하고 공정에 관련된 모든 사람들에게 새로운 방법을 전달한다. 만일 결과가 성공적이지 않았다면 계획을 수정하고 공정을 되풀이하거나 계획을 중단한다[10].

2.2 DMAIC(Define, Measure, Analyze, Improve, Control)

6시그마는 업무의 효율성 향상과 고객만족을 통해 경영성과를 높이고자 하는 경영혁신기법으로서, 결함 원인 파악, 분석, 개선, 통제하는 활동 등을 포함한다[11].

6시그마 품질수준이란 3.4PPM(Parts Per Million)으로서, 100만개 중 평균 3.4개 정도의 불량 발생한다는 것을 의미한다. 5시그마는 1백만 번에 233회, 4시그마는 6210회 불량이 발생하는 수준이다. 시그마 앞의 계수 값이 커질수록

〈표 1〉 DMAIC 단계별 정의

단계(Phase)	수행 내용
정의 (Define)	기업전략을 고려하여 개선의 수행목표와 대상, 범위, 개선기회 등을 파악 프로젝트 팀을 결성
측정 (Measure)	CTQ (Critical to Quality)를 선정 데이터를 수집하여 현재수준을 파악
분석 (Analyze)	개선대상을 명확히 하는 단계 결함의 형태와 원인을 규명하여 잠재원인을 파악하기 위한 데이터를 확보 현실문제를 통계문제로 전환하여 기술
개선 (Improve)	분석단계에서 도출한 개선대상의 최적수준을 위해 솔루션을 창출하는 과정 솔루션은 현장에 적용되고 개선효과가 분석단계에서 파악한 현재수준과 비교된다.
관리 (Control)	개선의 효과를 지속시키기 위해 규칙기반으로 점검

불량률은 기하급수적으로 줄어들고 6시그마는 실제업무에서 실현될 수 있는 가장 낮은 불량률 발생시키는 수준이다. 이러한 6시그마에서 품질개선(혁신)의 방법으로 사용되고 있는 것은 DMAIC이다[12]. DMAIC의 각 단계 정의는 <표 1>과 같다.

2.3 IDEAL(Initial, Diagnosing, Establish, Acting, Learning)

CMM/CMMI를 기반으로 프로세스를 구현하는 조직이 개선활동 시 주로 이용하는 개선 프레임워크(Framework)가 IDEAL 이다. IDEAL 역시 SEI에서 개발된 개선모델로서 조직의 프로세스 개선의 로드맵을 제공하며, 특히 지속적인 심사 및 개선 활동, 개선 효과의 유지 등을 강조하고 있다 [6]. IDEAL의 각 단계의 내용은 <표 2>와 같다.

<표 2> IDEAL의 각 단계별 내용

단계(Phase)	수행 내용
착수 (Initiating)	초기에 개선을 위한 기반을 마련하는 단계 동기, 개선대상, 영향과익 및 개선을 위한 지원체계형성
진단 (Diagnosing)	프로세스의 진단을 통해 조직의 현재 상태에 대한 기준선을 수립하는 단계 현재상태 및 개선목표의 정의: 필요사항 권고 및 위험요소 파악
확립 (Establishing)	개선 이슈에 대해 우선순위를 결정 목표 달성을 위한 접근 방법을 개발 실행 계획을 수립
실행 (Acting)	솔루션 개발 및 이에 대한 파일럿 테스트를 수행 솔루션을 정제 조직에 배포
학습 (Learning)	실행단계를 통해 얻어지는 경험을 수집 향후 개선 활동에 대한 권고안 개발

CMMI는 시스템 또는 소프트웨어 개발 조직에서 갖춰야 할 프로세스 영역들과 이러한 영역이 제대로 수행되기 위한 활동의 지침을 제공하고 있다. 그러나 CMMI는 25개의 영역 중 어떤 것이 조직 비즈니스 요구 또는 고객의 요구와 관련된 영역인지, 어떤 활동들이 현재 수행중인 프로세스에서 부족한지 파악하는 단계적 방법이나 기술들은 제공하고 있지 않다. IDEAL 방법은 개선 활동에 대한 단계 및 구체적인 활동 내용에 대해서는 설명하고 있으나, 현재의 프로세스 수준이나 목표수준을 정량적으로 설정하고 정량적인 개선결과를 평가하는 방법이나 개선 활동 단계별 사용할 수 있는 기법은 제공하고 있지는 않다.

이에 본 논문에서는 CMMI를 이용하여 프로세스를 구현 관리하는 조직에서 프로세스 개선 시 6시그마를 이용하여 정량적이고 체계적인 개선을 할 수 있는 방법을 제안하였다.

실제 6시그마는 국내외적으로 다양한 분야에서 적용되어 그 효과가 입증되고 있다[13]. 또한 6시그마를 적용하고 있는 많은 조직에서는 CMMI를 적용하여 프로세스를 구현하고 조직의 성숙도를 높이고자 노력하고 있다[14]. 그러나 구체적인 6시그마와 CMMI의 연계방법을 다루는 연구는 미비하며, CMMI와 6시그마 두 가지를 상호보완적으로 적용하는 실무조직에서는 6시그마를 CMMI의 4단계수준(프로세스의 정량적 관리), 5단계수준(프로세스의 최적화)을 위한 도구로서 사용하고 있다[15, 16].

6시그마의 효과에 대한 보다 정확한 결과나 성과를 얻기 위해서는 구체적이고 많은 데이터가 있을수록 좋기는 하지만, 새롭게 CMMI를 도입하려는 조직에서도 현재 상태를 판단하고 프로세스를 구현 또는 개선시의 효과를 알기 위한 개선의 방법이 필요하다. 따라서 본 논문에서는 CMMI기반의 프로세스 구현 및 개선 시 6시그마의 DMAIC 방법을 이용하여 조직에 필요한 CMMI의 영역을 식별하고 문제를 분석하는 구체적인 체계를 제시하였다.

3. 6시그마를 이용한 프로세스 개선사항 도출

본 논문에서 제시하는 6시그마와 CMMI, IDEAL을 통합하는 방법은 (그림 1)과 같다.

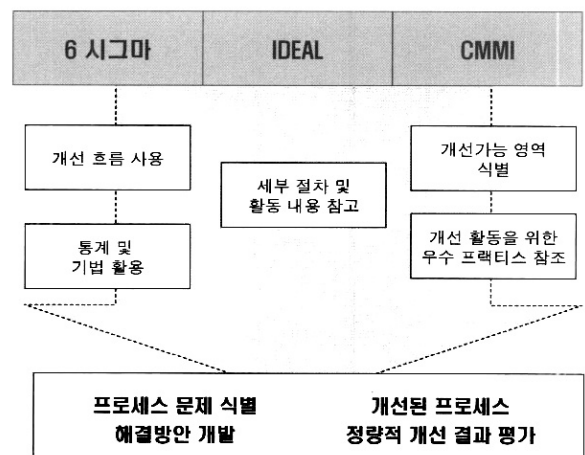
6시그마의 개선프로세스인 DMAIC 흐름으로 개선절차의 큰 흐름을 잡고, 각 활동에서 세부적인 과정이나 방법은 IDEAL을 참조한다. IDEAL은 실제로 DMAIC와 큰 틀은 비슷하지만 프로세스 개선에 필요한 세부적인 활동 지침을 제공하고 있다. 그러나 모든 활동을 적용하기에는 활동 내용이 많고 복잡하므로 활동 내용만을 참고하기로 하였다.

또한 6시그마에서 사용되는 몇 가지 기법을 프로세스 개선영역 식별 및 원인 분석에서 사용하였다.

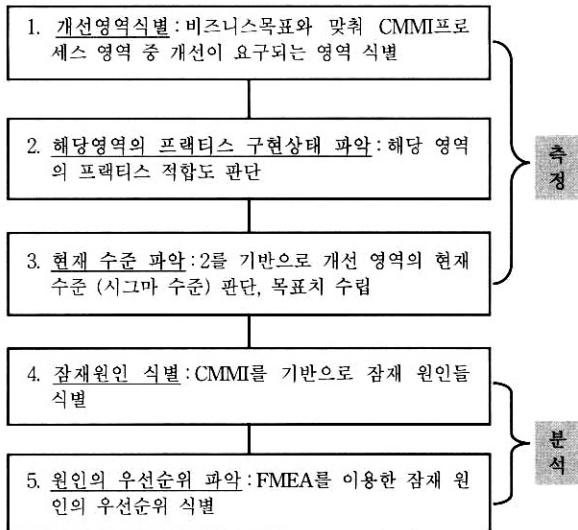
이때 개선 영역의 범위 설정은 CMMI의 프로세스 영역을 사용하였고, 실제 프로세스를 기반한 문제 식별이나 개선의 방향 등은 CMMI의 세부 프랙티스들을 참고하였다.

앞의 모델을 적용하여 개선점을 식별하는 과정을 도식화하면 (그림 2)와 같다. 6시그마의 DMAIC과정 중 M(측정)과 A(분석)부분에 해당하는 부분이다.

측정 단계에서는 조직의 목표 및 고객의 요구사항에 맞춰 개선의 영역을 식별하고, 그 영역에 대해 CMMI의 영역별 프랙티스들과 조직의 실제 업무들을 비교하여 프랙티스의 구현상태를 파악하고, 이에 대한 현재수준을 평가한다. 분석 단계에서는 측정 단계에서 파악된 부적합한 프랙티스 즉, CMMI의 프랙티스들과 맞지 않는 실무활동들에 대해서 원인을 식별하며, 위험도를 측정하고, 우선 순위를 정한다.



(그림 1) 6시그마와 CMMI의 통합 방법



(그림 2) 6시그마를 이용한 개선점 파악과정

3.1 개선영역의 식별

DMAIC를 진행하면서 다양한 도구 및 통계적 기법이 사용된다. 이중에서도 CTQ(Critical To Quality)를 식별하고 문제의 원인을 분석하는 데 많이 사용되는 QFD(Quality Function Deployment)는 품질 기능을 여러 프로세스의 속성들과 관계시키는 방법이다[17]. 효과적인 QFD의 수행을 위해서 매트릭스들로 구성된 HOQ(house of quality) 라는 도구를 사용할 수 있다. 본 논문에서는 HOQ를 응용하여 개선 영역 식별의 도구로서 사용하였다.

기업전략(Business Strategy)은 조직의 방향을 결정하고 행동을 이끌어 나가는 프레임워크이다[18]. 따라서 소프트웨어 프로세스 개선의 목표도 조직의 비즈니스 목표에 맞춰서 설정되어야 한다. 이는 <표 3>과 같이 HOQ로서 반영될 수 있다. <표 3>은 비즈니스 목표와 CMMI PA와의 관계를 파

악하는 표로써 품질특성 부분에 CMMI의 PA를 둠으로써 개선 영역을 식별하는데 사용할 수 있다.

세부 요구사항과 PA가 만나는 셀에 관련 정도에 따라 1~10점을 부과할 수 있으며 너무 많은 점수 분포가 부담스럽다면, 상·중·하, 정해진 점수(예를 들면 10, 5, 3), 또는 기호를 써서 표시할 수 있고, 이를 다시 점수화 할 수 있다.

이 때에 가중치에 관한 객관성 및 요구사항과 프로세스 영역 간의 관계점수에 대한 정확성을 높이기 위해서는 점수 부여 주체가 CMMI와 실무에 대한 경험과 지식이 있는 전문가집단이어야 하며, 가중치를 위해서는 중요도를 결정하는 기법으로 잘 알려져 있는 AHP(Analytic Hierarchy Process)[19]와 같은 방법을 사용하고, 요구사항과 PA간의 관계에 대한 점수 부과는 세부 체크리스트를 사용하여 정확성을 높일 수 있다.

3.2 해당영역의 프랙티스 구현상태 파악

식별된 개선 영역에 대해서는 실제프로젝트 활동사항과 CMMI의 프랙티스들 비교하고 분석함으로써 현재 수준을 파악할 수 있다. <표 4>와 같은 형식을 이용해서 CMMI 프랙티스 별로 실제 프랙티스들의 적합 정도를 파악하고, 이를 통합하여 PA의 적합도를 계산한다.

프랙티스들에 대한 구현상태 파악은 여러 개의 프로젝트에서 검토되어야만 프로세스의 구현 수준의 결과를 정확하게 판단할 수 있다. 이 때에 조직의 프랙티스의 구현 정도는 검토된 프로젝트들의 구현정도 중에서 낮은 수준으로 결정된다. 프랙티스에 대해 잘 수행한 프로젝트도 있고 그렇지 않은 프로젝트도 있을 경우, 그 결과를 평균인 보통으로 판단하는 것이 아니라, 통일되고 정확한 프로세스가 없어서 일관적인 결과가 나오지 않은 것으로 판단 하여야 하기 때문이다.

<표 3> 개선영역식별의 HOQ

		PP	PMC	SAM	REQM	...	(3) CMMI PAs				
		W									
(1) 비즈니스 목표	(2) 비즈니스 목표의 세부 속성 (소프트웨어 개선의 목표)	(5) 가중치									
			(4) PA와 비즈니스 목표와의 관계치 (10점 척도)								
			(6) 각 PA들의 점수								

(1) 비즈니스 목표: 조직에서 추구하여 프로세스에 반영이 되어야 하는 목표
 (2) 비즈니스 목표의 세부속성: 프로세스와 직접적인 관계설정을 쉽게 할 수 있도록 선정된, (1)을 구체화시킨 목표 특성, 소프트웨어 개선의 목표.
 (3) CMMI PAs: CMMI에서 제공하는 25개 프로세스 영역
 (4) 관계치: 해당 행, 열의 PA와 비즈니스 목표의 세부속성과의 관계 정도(10점 척도 등의 점수화 방법 이용)
 (5) 가중치: 비즈니스 목표 속성의 중요성을 반영하는 값
 (6) PA들의 점수: (4) × (5), 세부속성의 가중치와 그 속성에 해당하는 영역별 값의 곱의 총합

〈표 4〉 프로세스 적합도 매트릭스

	SP1 · 1.1	SP1 · 1.2	SP1 · 2.1	SP1 · 2.2	SP1 · 3.1	SP1 · 3.3	SP1 · 4.1	◎	○	△	-	총 프랙티스 수	(5) PA의 적합도
	(2) CMMI 프랙티스									(4) 적합수준별 프랙티스 수					
PP	-	-	-	-	-	-	-	-	-	0	0	0	25	25	0
PMC	-	-	-	(3) 프랙티스들의 적합 정도	-	-	-	-	-	0	0	0	21	21	0
SAM	-	-	-	-	-	-	-	-	-	0	0	0	19	19	0
(1) 개선영역	-	-	-	-	-	-	-	-	-	0	0	0	17	17	0
	-	-	-	-	-	-	-	-	-	0	0	0	19	19	0
.....															

- (1) 개선영역 : 식별된 개선이 요구되는 영역들
- (2) CMMI 프랙티스 : CMMI에서 제공되는 영역별 프랙티스 번호
- (3) 프랙티스들의 적합 정도 : 해당 PA의 프랙티스들과 현재 조직에서 구현되고 있는 프랙티스들과의 적합 정도. 아래의 표시와 같이 적합 정도를 나타내는 것과 같은 기호나 점수 등으로 표현될 수 있다.

기호 및 의미	적합여부
◎ : CMMI에서 제공하는 내용을 그대로 수행하고 있으며 이에 대한 증거자료가 있을 때	적합
○ : 내용은 수행하고 있지만 자료가 미흡할 때	적합
△ : 부분적으로만 수행되고 정확한 프로세스가 없으므로 모든 경우에 다 수행한다고 볼 수 없을 때	부적합
- : 수행되지 않을 때	부적합

- (4) 적합수준 별 프랙티스 수 : 해당 PA에서 받은 적합수준 심볼의 수
- (5) PA의 적합도 : 적합프랙티스 수/해당영역의 총 프랙티스 수

〈표 5〉 PA 공정능력 산출표

	(1) 프로젝트 별 적합수준 프랙티스 수						Total	(3) 공정능력산출																			
	Inst. 1		...	Inst. n		적합도		◎	○	△	-	적합도	◎	○	△	-	적합도	Defect	Unit	Opt	TOP	DPU	DPO	DPMO	shift	sigma	목표 : 0 defect
(2) 개선영역	PP	◎	○	△	-		적합도																				
	PMC																	0	0	1	0	0	0	0	1.5	6.00	6
	SAM																	0	0	1	0	0	0	0	1.5	6.00	6
																	0	0	1	0				1.5	6.00	6
																	0	0	1	0				1.5	6.00	6
																	0	0	1	0				1.5	6.00	6
																	0	0	1	0				1.5	6.00	6

- (1) 프로젝트별 적합수준 프랙티스 수 : 프로젝트 별로 프로세스 적합도 매트릭스의 (4)를 산출하여 통합시킨 것
- (2) 개선영역 : 식별된 개선이 요구되는 영역들
- (3) 공정능력산출 : 프랙티스들의 적합수준을 기반으로 산출하는 개선영역의 공정능력
 - Defect : 적합성을 검토한 프로젝트들에서 나타난 부적합 프랙티스의 총 수
 - Unit : 검토된 전체 프랙티스들의 수
 - Opt : 각 프랙티스당 Defect가 나올 수 있는 기회 수
 - TOP : 전체 프랙티스에서 Defect가 나올 수 있는 수
 - DPU : Defect/Unit
 - DPO : Defect/Opt
 - DPMO : 백만 개의 프랙티스에서 나올 수 있는 Defect의 수
 - Shift : 장기공정과정으로의 이동 값
 - Sigma : 공정능력(DPO 즉, 관찰치에 해당하는 표준정규분포의 값 + shift)

3.3 공정능력 산출

앞서 파악된 프랙티스들의 적합도는 통합되어 조직의 프로세스에 대한 역량 즉, 6시그마 용어로는 공정능력이 파악

될 수 있다. 프랙티스들 중 적합한 프랙티스와 부적합인 프랙티스의 개수를 각각 확인하고 부적합 프랙티스를 Defect라 정하면 Defect 수와 전체 프랙티스의 수 등을 통해 <표 5>

와 같이 해당 PA의 공정능력이 산출될 수 있다.

산출된 프로세스의 현재 능력수준은 프로세스 개선 프로젝트의 수행 후의 결과와 비교되어 개선의 성공여부를 확인할 수 있다.

부적합 프랙티스들을 파악하고 PA의 구현수준을 판단하는 차원이 아닌 문제의 측면에서 좀더 세부적이면서도 프랙티스들에서 나타나는 문제들을 통합해서 파악하고자 할 때는 또 한 단계의 QFD를 전개하면 된다. 이 때에는 프랙티스 번호가 세로축에, 프랙티스의 문제에 영향을 미치는 공정 또는 요인 들이 가로축에 놓여 이 요인들에 대한 점수들이 매겨지는 방식으로 전개될 수 있을 것이다.

3.4 부적합 유형 영향 분석을 통한 개선사항 식별

3.2절에서 식별된 부적합 프랙티스들에 대해서는 CMMI의 프랙티스들과 프로세스맵, 특성요인도 기법을 이용하여 문제의 원인들을 파악한다. 이는 잠재적 문제로서 그 원인에 대해서는 위험 순위를 매김으로써 개선 대상의 우선 순서로 사용할 수 있다.

우선순위를 결정하기 위해서는 6시그마에서 사용되는 기법인 FMEA(Failure Mode Effect Analysis)를 사용할 수 있다. FMEA란 문제의 원인에 따라 업무에 미치는 영향의 심각도, 발생 정도, 탐지가능 정도를 분석하여 위험도를 측정하고 위험도에 따라 우선순위를 정하는 기법이다[20]. FMEA를 응용한 부적합 유형영향 분석 도구는 <표 6>과 같다.

만약 너무 많은 문제요소들이 식별된다면 문제들에 대해 우선순위를 분석하기 전에 유사한 문제형태는 하나로 통일시키고 심각성, 발생성 등의 점수를 매기는 과정에서 두 가지의 상이한 문제가 한 문제로 결정된 경우에는 분리시키는

등의 정리작업을 수행한다.

4. 적용의 예

본 논문에서 제안하는 “6시그마를 이용한 프로세스 개선 사항 도출과정”을 사례를 통해 진행하여 그 사용 가능성을 검증해보고자 한다. 각 단계에 대한 자세한 설명은 앞의 연구모델의 설명부분에 있으므로 중복되는 부분은 설명을 생략하였다.

우선 첫 번째 단계인 개선영역식별을 위해서는 조직이 개선활동을 시작하게 된 동기, 즉 개선의 목표를 정의하고 개선영역을 구별하기 쉬운 단계까지 구체화 시켜야 한다. 예로써 대부분의 조직에서 가장 중시하는 프로젝트 기간 준수라는 목표 하나만을 고려해보았다. 프로젝트 기간 준수를 위해서는 다양한 요소가 고려될 수 있지만 우선 전제되어야 하는 부분은 정확한 계획이겠고, 계획에 맞게 제대로 프로젝트 활동을 수행하는지 검토하는 부분이겠다. 또한 재작업을 감소시키고, 갑작스럽게 일어날 수 있는 위험 등에 대해 고려하여 대처 안을 만드는 것이 필요할 것이다. 이러한 내용들을 고려하여 프로젝트 기간 준수에 대한 세부 요구사항들을 개발하고, 세부요구사항들과 PA들과의 관계 정도를 파악하였다.

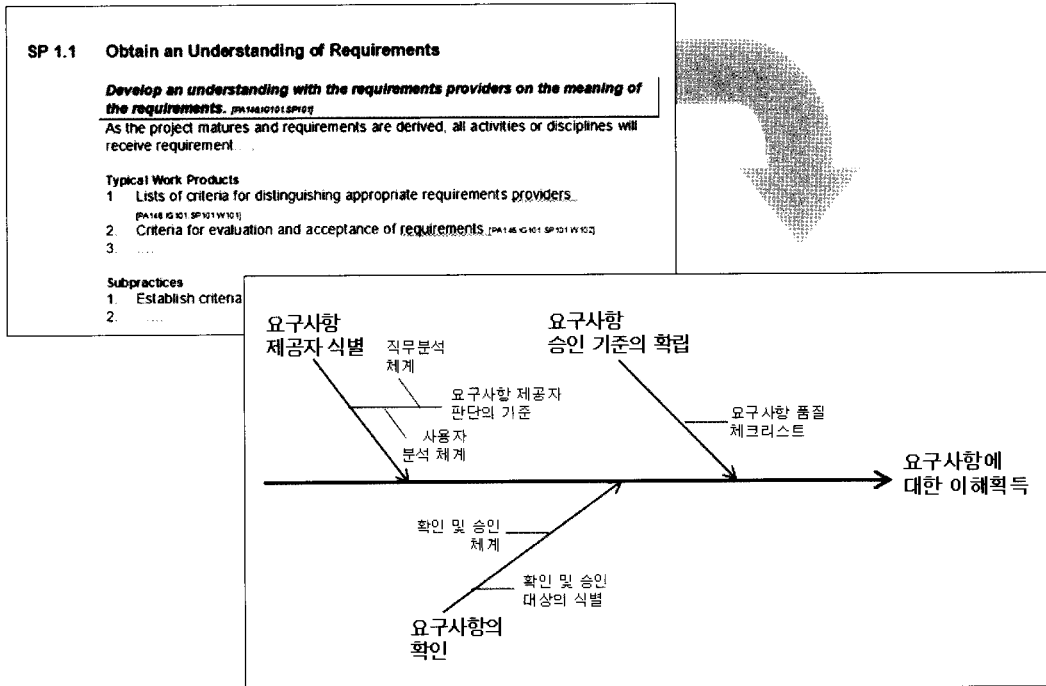
식별된 영역은 가장 점수가 높은 PP와 REQM 부분이다. 모든 영역이 프로젝트 기간 준수에 영향을 조금씩은 줄 수 있을 것이다. 그렇지만 개선의 비용을 고려하여 조직이 원하는 수준까지만 개선영역을 선택하여야 한다.

개선영역으로 정해진 프로세스들에 대해서는 조직의 현재

<표 6> 부적합 유형 영향 분석

부적합 유형 영향 분석 (NCMEA: Non-compliance Mode and Effect Analysis)											
프로세스 영역: 담당:								작성일자: 수정일자:			
부적합프랙티스	PA에 미치는 영향	심각성	부적합원인	발생성	현재 통제 방법	탐지성	위험도	위험순위	조치 계획	담당/ 목표일	현재 상태
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)			

- (1) 부적합 프랙티스: 결합의 형태, 종류, 프로세스맵이나 특성요인도를 통해 찾아낸 잠재 인자. 또는 이러한 과정이 생략되었을 때 부적합 프랙티스.
- (2) PA에 미치는 영향: PA의 목표를 기반으로 파악
- (3) 심각성: 1이 2에 미치는 영향의 정도를 점수화(1~10)
- (4) 부적합원인: 현재의 프로세스 상의 문제 및 PA에서 제안하는 산출물 또는 방법의 부재 등으로 유도.
- (5) 발생성: 4의 발생빈도를 점수화(1~10)
- (6) 현 통제방법: 부적합원인을 관리하거나 탐지해 내는 방법
- (7) 탐지성: 6의 탐지 정도를 점수화(1~10). 탐지가 어려울수록 고점수 부여
- (8) 위험도: 3 * 5 * 7
- (9) 위험순위: 8을 우선순위화 시킨 것



(그림 7) CMMI와 특성요인도(fishbone)를 이용한 원인식별

않은 것으로 판단해야 하므로 낮은 수준을 현재의 프랙티스 구현상태로 파악하고 부적합 프랙티스를 판단한다.

프랙티스의 부적합 원인을 알아보기 위해서는 일반 산업체에서는 공정을 다시 한번 점검하거나, 관련자들이 모여 원인에 대한 브레인스토밍을 수행한다. 그러나 CMMI에서는 이미 많은 조직에서 수행되어 검증된 우수사례들을 제공하고 있으므로[2] 이를 실무활동과 비교 검토하여 1차적인 원인을 밝혔다. 그리고 이에 대한 구체적인 세부요인은 실무에서 행해져야 하는 구체적인 활동 및 산출물들로 분석, 표현한다. (그림 7)은 특성요인도를 사용하여 부적합 원인을 분석한 결과이다.

특성요인도를 통해 식별된 잠재적 문제에 대해서는 그 심각성 정도 및 발생성, 탐지수준 등을 고려하여 위험도를 결정하고, 이를 기반으로 개선항목의 선정 및 개선활동의 순서를 결정하였다. <표 7>은 이러한 문제 원인에 대한 우선순위 결정을 위해 부적합 유형 영향 분석을 하는 예이다. 수행 결과를 통해 조직의 목표에 문제가 되는 핵심 원인을 선택할 수 있다. 수행된 결과를 통해 REQM의 부적합 프랙티스 중 SP1.1-1에 대한 부적합유형 영향분석을 수행한 결과 사용자 분석 및 직무분석 체계의 미흡이 가장 심각한 문제로 도출되었다.

소프트웨어 프로세스 개선 시, 절차와 기법 측면에서는 6

<표 7> 부적합 유형 영향 분석의 예

부적합 유형 영향 분석 (NCMEA: Non-compliance Mode and Effect Analysis)									
부적합 프랙티스	잠재적 문제형태	REQM			현재통제방법	탐지성	위험도	위험 순위	조치 계획
		PA에 미치는 영향	심각성	잠재적 문제 원인					
요구사항에 대한 이해 획득 [CMMI REQM SP1.1-1]	사용자 분석체계	잘못된 요구사항 획득	10	사용자분석 체계 부재	10	승인회의	7	700	1 체계의 개발
	직무 분석체계	불완전한 요구사항 획득	10	직무분석 체계 미흡	7	승인회의	5	350	2 체계의 개발
	요구사항 품질 체크리스트	요구사항 품질 저하	7	요구사항 체크리스트 요소 부족	5	PM에 의한 검토	3	105	3 체크리스트 개정
	승인/확인 대상 식별	내외부 관계자의 요구사항 이해 부족	3	승인/확인 체계 미흡	3	PM에 의한 선정	5	45	4 사용자 및 관계자 식별 방법 개발
	승인/확인 대상 체계	부정확한 요구사항 차후로 전이	3	승인/확인 체계 미흡	3	PM에 의한 선정	3	27	5 관계자에 따른 확인 체계 마련

시그마를 이용하고, 개선가능 영역과 영역별 목표활동 대상 측면에서 CMMI를 사용함으로써, 정량적이고 체계적인 과정으로 개선의 영역을 선택하고 문제를 파악할 수 있음을 확인하였다.

본 논문에서 제시하는 프로세스 개선 기법에 대한 장점을 정리하면 다음과 같다.

첫째로 조직의 목표를 위해 다뤄져야 하는 프로세스 영역에 대한 현재 수준을 수치로 확인함으로써 개선 후에는 그 결과를 또 다시 정량화하여 비교할 수 있으며, 둘째로 그 원인 분석에 대해 CMMI의 프랙티스를 근거로 하여 실무로 확장시킴으로써 문제식별의 시작을 쉽게 하면서도 핵심적으로 다뤄야 할 부분들을 고려할 수 있으며, 셋째로 식별된 원인들에 대한 우선순위를 분석함으로써 비용과 시간이 제한된 개선활동에서 가장 먼저 해결되어야 할 문제들을 찾을 수 있다.

5. 결 론

소프트웨어 프로세스가 프로젝트의 성공률을 높이는 효과적인 수단이라는 것은 이미 여러 기업을 통해서 검증이 되었으며, 이러한 효과를 지속하기 위해서 대부분의 조직들은 프로세스의 개선활동을 수행하고 있다.

국내외적으로 소프트웨어 프로세스 모델로 업체표준으로 되어 사용되고 있는 CMM, CMMI의 사용취지는 검증된 프로세스 활동들을 기반으로 프로세스를 구현·개선하여 조직에 적합한 프로세스 구현 시 시행착오를 줄이자는 것이다.

이처럼 CMM/CMMI는 성숙된 프로세스에 대한 표본을 보여주지만, 어떻게 구현하는지, 어떻게 개선하는지에 대한 방법은 제시하고 있지 않다. IDEAL 역시 프로세스 개선에 대한 과정 및 활동을 자세히 설명하고 있으나 이에 대한 도구나 방법, 개선에 대한 정량적 평가체계는 다루고 있지 않다.

따라서 본 논문에서는 6시그마(품질 혁신기법)에서 제시하는 문제해결 프로세스인 DMAIC를 사용하여 소프트웨어 프로세스 개선, 특히 CMM/CMMI 기반의 프로세스 구현 조직에서 개선 활동을 수행할 때 개선영역의 식별 및 문제식별 시 사용할 수 있는 방법을 구체적으로 제시하였고 이에 대한 사용가능성을 확인하였다.

본 논문에서 다룬 부분은 DMAIC의 다섯 단계 중 분석 단계까지의 내용이다. 식별된 문제에 대한 개선안 개발, 개선안 실행, 성과의 확인, 지속적인 관리 등 개선과 관리에 대한 체계화는 추후에 연구해야 할 과제이다.

참 고 문 헌

[1] Mary Beth Chrissis, et al., "CMMI guidelines for process

integration and product improvement", Addison Wesley, 2003.

[2] CMMI Product Team, "CMMISM for Systems Engineering/ Software Engineering/ Integrated Product and Process Development/ Supplier Sourcing, Version 1.1, Staged Representation (CMMISM)/SE/SW/IPPD/SS, V1.1, Staged", CMU/SEI, 2002.

[3] Mike Phillips, "CMMI-Version1.2 and Beyond", CMU/SEI, 2005.

[4] 이병희, "6시그마 중소기업 경영 화두 됐다", 전자신문, 2006.

[5] 황경태, "IT서비스관리(ITSM)", 전자신문, 2006.

[6] Bob McFeeley, "IDEALSM: A User's Guide for Software Process Improvement, CMU/SEI-96-HB-001", CMU/SEI, 1996.

[7] IEEE Std 610.12-1990, "IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology", IEEE, 1991.

[8] Humphrey, W.S., "Managing the Software Process", Addison Wesley, 1990.

[9] Andrea Gabor, "The Man Who Discovered Quality", Penguin, 1990.

[10] Paul Arveson, "The Deming Cycle", <http://www.balancedscorecard.org/bkgd/pdca.html>, Balanced Scorecard Institute, 1998.

[11] 최승용 외, "6시그마 기반의 프로세스 정의 기법", 한국정보처리학회 소프트웨어공학논문지, 제8권 제3·4호, 2005.

[12] 아오키 야스히코 외, "6시그마 경영", 21세기북스, 1999.

[13] 배영일, "6시그마의 현황과 미래", 삼성 경제 연구소, 2005.

[14] "List of Published SCAMPI Appraisal Results", http://seir.sei.cmu.edu/PARS/pars_list_iframe.asp, CMU/SEI

[15] Nancy Fleischer, "Raytheon's Six Sigma Process and Its Application for CMMI", NDIA/SEI CMMI Technology Conference, 2003.

[16] Sivy, Jeannine, "Software Technology Review(Six Sigma Section)", <http://www.sei.cmu.edu/str/descriptions/sigma6.html>, 2001.

[17] 김재배, "Six Sigma 추진 단계에서의 QFD의 효율적 활용 방안", 웹진-시그마경영, 2000.

[18] Urban, G. L. and Hauser, J. R., "Design and Marketing of New Products", Prentice-Hall, 1993.

[19] Thomas L. Satty, "The Analytic Hierarchy Process", New York, McGraw-Hill, 1980.

[20] Robin E. McDermott, et al., "The Basics of FMEA", Quality Resources, 1996.

김 한 샘



e-mail : puru@smu.ac.kr
1997년 상명대학교 경영학과(학사)
2000년 상명대학교 정보통신대학원
멀티미디어학과(이학석사)
2003년 상명대학교 컴퓨터과학과(박사수료)
2003년~2004년 시스템통합기술연구원
연구팀 팀장

2002년~현재 상명대학교 소프트웨어학부 강사
관심분야 : 프로세스개선, 측정, 6시그마, 소프트웨어품질 등

한 혁 수



e-mail : hshan@smu.ac.kr
1985년 서울대학교 계산통계학과(학사)
1987년 서울대학교 대학원 계산통계학과
(이학석사)
1992년 Univ of South Florida 전산학과
(공학박사)

2000년~2003년 시스템통합기술연구원 원장
2003년 소프트웨어진흥원 소프트웨어공학 연구소장
2004년~2005년 상명대학교 소프트웨어대학 학장
1993년~현재 상명대학교 소프트웨어학부 교수
관심분야 : 소프트웨어 프로세스, 소프트웨어 품질, 소프트웨어
사용성 평가 등