

QFD 기법을 이용한 CMM 2단계 척도 선정

QFD Application to Select CMM Level 2 Metrics

김 종 윤* 정 호 원** 김 길 조***
*고려대학교 경영대학 **고려대학교 경영대학 ***시스템공학연구소

요약

SEI(Software Engineering Institute)의 CMM(Capability Maturity Model)은 소프트웨어 개발 공정 및 조직의 성숙도를 5개의 계층적인 단계로 분류·평가하며, 성숙도 측정의 기준으로 계층적인 메저먼트를 각단계별로 제시하고 있다. 본 연구에서는 CMM에 의해 제시된 메저먼트에 근거하여 정의되고 사용된 척도의 선정에 정보시스템 개발 및 품질 관리 기법인 QFD(Quality Function Deployment)를 적용하였다. 척도 사용자 그룹으로부터 CMM의 관심 영역, 사용 척도, 척도 계산에 필요한 측정치, 소프트웨어 개발 생명주기에 이르는 과정을 단계적인 흐름으로 파악하여 적절한 척도를 선택하는데 본 연구의 목적이 있다.

1. 서론

소프트웨어 개발 프로젝트의 평가에 있어서 최종 산출물인 소프트웨어 제품 자체의 품질[14]에 대한 평가 뿐만 아니라 소프트웨어 개발 공정 및 조직에 대한 평가 또한 중요하다[11]. 그것은 잘 정의되고 계획되고 예측 가능하며 명확한 제품 개발의 목표를 가진 조직으로부터 개발된 소프트웨어 제품의 품질이 그렇지 않은 조직의 제품보다 좋으리라는 것에 근거한다[13].

유형성의 특징을 갖는 일반제품과는 달리 무형적인 소프트웨어의 특징으로 인하여 계량적인 품질측정이 어렵다. 따라서 체계적이고 지속적인 소프트웨어의 품질관리 방안이 마련되어야 하며[3], 품질관리는 소프트웨어 개발 생명주기[10]의 초기 단계부터 적용되어야 한다[16].

일반적으로 소프트웨어 개발 공정 및 조직에 대한 평가 방법으로는 SEI의 CMM, SPR(Software Productivity Research)의 측정 방법(Assessment Method), 미국 상부성의 Malcolm Baldrige Assessment, ISO 9000-3 품질 시스템 등이 있는데[11], 특히, CMM은 소프트웨어 개발 조직의 성숙도를 5개의 계층적인 단계로 나누고 소프트웨어 개발 공정의 지속적인 개선을 위한 지침을 제공해 준다[13]. 본 연구에서는 CMM의 5개의 단계중 반복가능 단계인 2단계를 연구의 대상으로 하였다.

소프트웨어의 개발에 있어서 단순원가, 생산성, 품질 등의 일차원적인 개념으로부터 고객의 가치창조와 관련된 종합적인 품질의 개념이 중시되는 추세이며, 이를 구체화할 수 있는 방안으로 고객의 요구사항과 소프트웨어 제품개발의 연결도구인 QFD, 조직적 차원에서의 품질관리팀(Quality Team) 등의 활용 등이 있다. 본 연구에서는 QFD를 CMM 2단계의 척도 선정을 위한 도구로 사용하였으며, 척도 사용자 그룹을 고

객으로 간주하였다.

기존에 일반적으로 많이 사용되어지는 품질 척도의 선택 방법으로는 Motorola사에서 시행했던 목표/질문/척도 접근법(Goal/Question/Metric approach: GQM)[13]이 있다. GQM에서는 먼저 척도 작업 그룹(Metrics Working Group: MWG)을 구성하고 소프트웨어 개발을 위한 품질 정책에 의거하여 척도를 적용할 분야를 정하고 목표를 식별한 후, 그에 따르는 문제를 제시하고 그 문제에 적절한 척도를 적용한다.

본 연구에서는 GQM과는 달리, CMM 2단계의 성숙도를 달성하기 위한 척도의 선정에 QFD 기법을 이용하여 먼저 척도 사용자 그룹을 분류하고 그들의 역할을 규정한 후, 그들이 관심을 갖는 CMM 2단계 척도 및 측정치, 측정치를 수집할 수 있는 소프트웨어 개발 생명주기의 공정을 캐스캐이드한 방법을 통해 제시한다. 즉, 소프트웨어 개발의 목표 중심의 척도 선정이 아닌, 척도 사용자 중심의 척도 선정 방법 및 소프트웨어 개발 생명주기의 각 공정에 적절한 측정치등을 제시한다. QFD 기법 적용의 결과는 품질의 집(House of Quality: HOQ)[6]의 작성이며, 이것은 일련의 연속적인 선택 및 관리의 과정으로 나타난다.

본 연구의 구성은 2장은 관련연구로서 CMM, QFD 및 CMM 2단계에서의 소프트웨어 개발 생명주기에 대해서 설명하고, 3장에서 CMM 2단계의 성숙도를 달성하기 위한 척도 선택에 QFD를 적용했다. 마지막으로 4장은 결론 및 향후 연구 과제이다.

2. 관련 연구

2.1 CMM

CMM은 조직의 소프트웨어 개발 공정의 성숙도를 초기(initial), 반복가능(repeatable), 정의(defined), 관리(managed), 최적화(optimizing) 등의 5개의 계층적인 등급으로 나누어, 조직의 공정 개선활동을 지원하는 모델이다. CMM의 5단계는 혼란스럽고 미성숙된 상태의 공정으로부터 성숙하고, 잘 훈련된 소프트웨어 공정으로의 변화 과정을 단계별로 나타내고 있다. 이들 5단계의 성숙 단계는 연속적인 공정 개선에 대한 지속적인 단계를 묘사하며, 조직의 소프트웨어 공정에 대한 성숙도를 측정하는 일반적인 메저먼트를 정의해준다. 또한, CMM은 조직이 소프트웨어 개발 공정에서 성숙도를 달성하기 위해 중시해야 하는 영역을 규정하는 KPA(Key Process Area)[13]를 각 단계별로 제시하고 있다

본 연구의 대상인 효과적인 소프트웨어 개발 공정 관리가

가능해지기 시작하는 CMM 2단계에서의 KPA는 기본적인 프로젝트 관리에 대한 통제를 가능케 하는 지침을 제공한다 [13]. CMM 2단계의 KPA와 해당하는 목표는 다음의 <표 1>과 같다.

<표 1> CMM 2단계의 KPA

KPA	목표
요구사항 관리(RM)	고객의 요구사항과 프로젝트에 대한 공통적인 이해 확립
소프트웨어 프로젝트 계획(PP)	소프트웨어 엔지니어링과 프로젝트 관리에 대한 합리적인 계획 수립
소프트웨어 프로젝트 추적 및 감독(PT)	실제 공정에 대한 적절한 가시성(visibility) 확립
소프트웨어 하청계약 관리(SM)	자격 있는 하청업자의 선정 및 효과적인 관리
소프트웨어 품질	소프트웨어 품질 보증에 대한 적절한 가시성 제공
소프트웨어 조정 관리(CM)	소프트웨어의 무결성(integrity) 유지

CMM에 근거하여 2단계의 성숙도를 달성하기 위해서 정 의되고 사용된 척도는 다음의 <표 2-a> 및 <표 2-b>와 같다[11].

<표 2-a> CMM 2단계에서의 사용 척도

분류	척도	척도의 정의 및 계산
크기	추정 크기(SI1)	KSLOC
크기	초기 측정치와 현재 추정치의 비율(SI2)	(현재 추정 크기/초기 추정 크기)×100
스케줄	개발 소요 시간(SC1)	소프트웨어 개발에 소요된 시간(단위: 개월)
스케줄	개발 소요 시간의 계획치와 실제 경과 시간의 비율(SC2)	(실제 경과 시간/예산 소요 시간)×100
스케줄	계획된 인수 기일과 하청 계약에 의한 인수 기일의 비율(SC3)	제품 인수 기일/계획된 인수 기일
비용	개발 소요 시간과 예산과의 비율(CO1)	예산(단위: \$)/LM(혹은 LH)
비용	예산과 하청 계약에 소요되는 경비의 비율(CO2)	하청 계약에 소요되는 비용/개발 예산
품질	KSLOC 당 결함의 수(QU1)	결함의 수/KSLOC
품질	KSLOC 당 PTRs의 수(QU2)	PTRs/KSLOC
품질	전체 PTRs와 완료된 PTRs의 비율(QU3)	(완료된 PTRs/전체 PTRs)×100
안정성	계획 소요 시간과 스텝의 비율(SB1)	(스텝의 수/개발 소요 시간)×100
안정성	ECPs(SB2)	ECPs의 수
안정성	정의되지 않은 요구사항의 비율(SB3)	(정의된 요구사항/전체 요구사항)×100
상태	검사된 unit의 비율(ST1)	(검사된 unit의 수/전체 unit의 수)×100
상태	통합된 unit의 비율(ST2)	(통합된 unit의 수/전체 unit의 수)×100
상태	테스트된 요구사항의 비율(ST3)	(테스트된 요구사항의 수/전체 요구사항의 수)×100
상태	설계된 요구사항의 비율(ST4)	(설계된 요구사항의 수/전체 요구사항의 수)×100
상태	요구사항의 변경 회수(ST5)	요구사항 변경 회수
Computer Resources	소프트웨어 I/O의 비율(CR1)	(메세지의 길이×도착율)/처리 속도

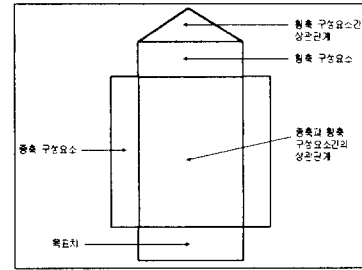
<표 2-b> CMM 2단계에서의 사용 척도

분류	척도	척도의 정의 및 개념
Computer Resources	메모리를 차지하는 비율(CR2)	사용된 CPU의 크기/이용 가능한 저장 능력의 크기
N/A	N/A	소프트웨어 개발의 각 단계에서의 독립적인 감사(audit)활동

2.2 QFD

QFD는 고객의 요구를 제품 개발 각 단계에서의 기술적 요구사항으로 전환하여 최종 제품에 충실히 반영되도록 하는 체계적인 제품 개발 및 품질 관리 방법이다[4]. 즉, QFD에 의하면 모든 개발 단계는 '고객의 소리(voice of customer)'에 의해서 좌우된다[8].

QFD는 일련의 품질의 집으로 이루어져 있으며 각각의 집 들은 고객의 요구사항으로부터 좀 더 세부적이고 기술적인 설계·구현 단계에까지 이동한다[4]. 개별적인 품질의 집의 모양은 다음의 (그림 1)과 같다[4].

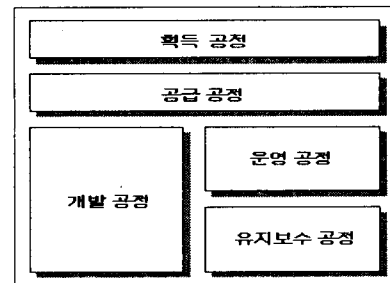


[그림 1] QFD의 모형

2.3 소프트웨어 개발 생명주기

소프트웨어 개발 생명주기에 관한 ISO 12207[10]의 주요 내용은 기본과정과 이를 지원하는 지원과정, 그리고 조사과정 등으로 구성되는데 본 연구에서는 기본과정을 대상으로 한다.

소프트웨어 개발 생명주기의 기본과정은 소프트웨어 생명 주기상의 주요 사용자들과 관련되는 직접적인 5개의 세부공정 단계인 획득공정, 공급공정, 개발공정, 운영공정, 그리고 유지보수공정으로 구성된다[10]. 그리고 각공정은 하위의 구체적인 활동들로 구성된다. 기본과정은 다음의 (그림 2)와 같다



[그림 2] 소프트웨어 개발 생명주기

3. QFD 기법의 적용

3.1 척도 사용자 그룹과 KPA

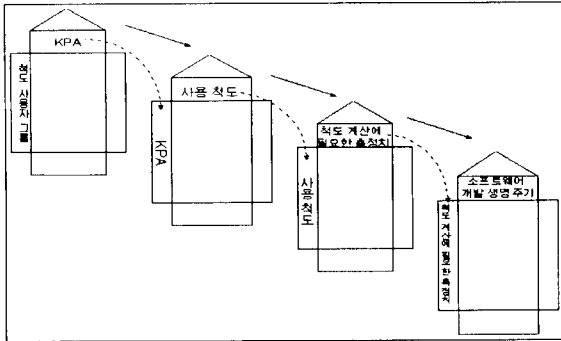
척도 사용자 그룹은 다음과 같이 분류된다[5,11]. 이것은 부여받은 직무의 성격에 따라 관심을 갖는 척도가 다르며

	소프트웨어 개발 생명주기				
	획득공정	분석공정	개발공정	운영공정	유지보수 공정
척도 계산에 필요한 측정치	KSLQC				
	개발 기간	●	●		
	인수 기일	●	●		
	소요 비용	●	●		
	하중계와 비용	●	●		
	결함의 수			●	●
	KPI의 수			●	
	스펙의 수	●	●		
	ECP의 수			●	
	요구사항의 수			●	
	변경 횟수			●	
	요구사항 변경 횟수			●	
	메세지 길이			●	
	도착률			●	
처리 속도			●		
사용된 CPU			●		
저장 능력			●		
기타	●	●	●	●	

[그림 6] 측정치와 소프트웨어 개발 생명주기
[기호 설명] ● 상관이 있음

3.5 연속적인 품질의 집

개별적인 품질의 집은 다음의 (그림 7)과 같이 서로 연결되어 단계적이고 연속적인 척도 및 소프트웨어 생명주기에 관한 의사결정에 이용된다.



[그림 7] 연속적인 품질의 집

위의 그림을 간략히 설명하면, CMM 2단계의 성숙도를 달성하기 위해서 소프트웨어 품질 담당자는 KPA QA와 CM 영역에 주로 관심을 갖는다(그림 3 참조). KPA CM 영역의 목표를 위해서는 개발 조직의 안정성을 측정하는 척도인 SB2, SB3 척도를 먼저 선택해야 한다. 즉, ECPs의 수라든지 정의되지 않은 요구사항의 비율 척도 등이 선택되어야 한다(그림 4 참조). 척도가 선택되었으면 척도 측정에 필요한 측정치가 수집되어야 하는데, 척도 SB2를 구하기 위해서는 측정치로 ECPs의 수가 측정되어야 하며 척도 SB3를 구하기 위해서는 정의되지 않은 요구사항의 수와 전체 요구사항의 수가 측정되어야 한다(그림 5 참조). 마지막으로 소프트웨어 개발 생명주기 상에서의 적절한 측정치 수집이 필요한데, ECPs의 수는 획득공정에서 요구사항의 수는 개발공정에서 수집되어야 한다(그림 6 참조). 이상과 같은 과정을 통해서 CMM 2단계의 성숙도가 소프트웨어 개발 생명주기에 걸쳐 달성된다.

4. 결론 및 향후 연구 과제

본 연구에서 소프트웨어 개발 공정 및 조직 성숙도의 계량적인 척도를 사용자의 관심사를 고려하여 선택하는 하나의

방법을 제시하였으며, 본 연구는 소프트웨어 개발 공정의 효율적인 진행을 가능케 하며, 척도 사용자 그룹간 의사소통을 원활하게 한다. 척도 프로그램의 구현에서 CMM 2단계의 모든 척도를 구현할 수는 없으므로, 그 중 어떤 척도가 중요한지에 대한 연구가 필요하며 이는 다른 연구에서 언급되었다. 소프트웨어 개발에 정보시스템의 요구사항 분석 및 설계, 품질관리 도구인 QFD 기법을 적용함에 있어서 이론상 부적당한 점이 있었고, 현장에서 얻어진 데이터에 의한 실증적인 연구가 없었다는 것이 본 연구의 한계점이며 이에 대한 연구가 현재 진행중이다.

참고 문헌

- [1] 이종무, "소프트웨어 품질 평가 투입요소 결정에 관한 연구," 고려대학교 대학원, 박사학위 논문, 1997.
- [2] Akao, Y., *Quality Function Deployment*, Productivity Press, Cambridge MA, 1990.
- [3] Basili, V. R. and G. Caldiera, "Improve Software Quality by Reusing Knowledge and Experience," *Sloan Management Review*(Fall 1995), pp.55-64.
- [4] Fatemeh Zahedi, *Quality Information System*, boyd & fraser publishing company, Boston, 1995.
- [5] Gatner Group, "Getting Value From Your AD Performance Measurement Program," 1994.
- [6] Hauser, J. R. and D. Clausing, "The House of Quality," *The Harvard Business Review*, No. 3(May-June 1988), pp.63-73.
- [7] <http://mijuno.larc.nasa.gov/dfc>
- [8] Humphrey, W. S., *Managing the Software Process*, Reading, Mass., Addison-Wesley Publishing Company, Massachusetts, 1989.
- [9] ISO 9000-3, *Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintainance of software*, ISO, 1991.
- [10] ISO/IEC 12207-1, *Information Technology - Software - Part 1: Software life-cycle processes*, ISO, 1995.
- [11] Gaffney, J., Cruickshank, R., Werling, R., and Felber, H., *Software Measurement Guidebook*, Int'l Thomson Computer Press, Boston, 1995.
- [12] Jones, C. L., "A Process-Integrated Approach to Defect Prevention," *IBM Systems Journal*, Vol. 24, No. 2, 1985.
- [13] Mark, C., Paulk, Bill Curtis, Mary Beth Chrissis and Charles, V. Weber, "*Capability Maturity Model for Software version 1.1*," Software Engineering Institute, Pittsburgh, 1993.
- [14] Daskalantonaskis, M. K., "A Practical View of Software Measurement and Implementation Experiences Within Motorola," *IEEE Transactions on Software Engineering*, Vol. 18, No. 11(1992).
- [15] Moller, K.-H. and Paulish, D.J., *Software Metrics: Practitioner's guide to improve product development*, Chapman & Hall Co., London, 1993.
- [16] Prasad, S., "Total Quality: Out of Reach or Within Reach ?" *ANTEC '96*(1996), pp.3422-3426.
- [17] Richard, Bache, *Software Metrics for Product Assessment*, McGraw-Hill Book Co., NY, 1994.
- [18] Stephen, H. Ka., *Metrics and models in Software Quality Engineering*, Addison-Wesley Publishing Company, Massachusetts, 1994.