

소프트웨어 프로세스의 계량적 관리

한국전자통신연구원 안유환·김길조
고려대학교 오세원·김종윤

1. 서 론

최근 산업이 고도화되면서 현대 사회의 소프트웨어에 대한 의존도가 급격히 증가하고 있으며, 소프트웨어는 대부분 첨단제품의 핵심부품으로 다양한 산업에서 중추적 역할을 하고 있다. 그러나, 개발 측면에서 소프트웨어는 일반 제품의 개발과정과는 상이한 무형성과 복잡성으로 인해 사실상 효과적이고 체계적인 관리가 이루어지지 않고 있다[1].

소프트웨어 개발 프로세스를 효과적으로 관리하기 위해서는 계량적으로 프로세스를 측정할 수 있어야 한다. 측정에 근거한 소프트웨어 프로세스 관리는 개발 프로세스 산출물들이 정량적 목표를 달성할 수 있도록 유도하고, 점진적으로 목표가 달성된 정도를 평가하는 것을 의미한다[2]. 즉, 개발 프로젝트와 프로세스에 대한 가시성(visibility)을 높이기 위해서는 프로세스의 계량적 관리를 통해 제품의 크기, 비용, 일정, 품질 등에 대한 이해 가능한 측정이 필요하다.

소프트웨어 개발에서 품질의 중요성이 부각되면서 프로젝트 관리, 프로세스 품질 및 최종 제품 품질에 대해 계량적으로 측정하고 평가하는 것이 소프트웨어 개발자들에게 필수적인 업무가 되었다. 따라서 본고는 소프트웨어 개발 프로세스의 계량적 관리를 위해 개발 단계별로 필요한 척도(metrics)를 수집하고 분석하는 방법을 설명하기 위한 것이다.

척도란 소프트웨어 프로세스/제품/프로젝트 특성을 계량적으로 결정하는 방법으로 척도 값을 결정하는 공식, 그 값을 나타내는 도표, 그

리고 이를 사용하고 해석하기 위한 지침을 의미한다[3]. 소프트웨어 공학관점에서 척도는 제품과 프로세스를 통제할 수 있게 하고, 작업 생산성, 효과성, 품질의 상태를 보여주며, 소비자의 신뢰성 획득 및 품질 개선이 필요한 부분을 찾아내는 역할을 한다. 한편 관리적 측면에서 척도는 시간, 자원, 품질에 대한 추정을 하고 잠재적 문제점들을 인지하여 수집할 수 있게 해주며, 자원, 선정, 일정에 대한 지침을 마련하여 프로세스 개선을 위한 데이터를 제공하는 역할을 한다[4].

효과적인 척도 프로그램의 개발을 위해서는 SEI(Software Engineering Institute)에서 개발한 CMM(Capability Maturity Model)이나 ISO/IEC 15504 TR2(Technical Report Type 2)인 SPICE(Software Process Improvement Capability dEtermination)등의 국제적 품질 기준에 부합하면서 조직환경에 적응성이 높은 척도를 선정하는 것이 필수적이다. 본 고에서는 수행능력 평가모형 중의 하나인 CMM에 근거한 관리척도(management metrics)를 중심으로 계량적으로 프로세스를 관리하는 방법에 대해 설명한다.

본고의 구성은 2장에서는 CMM과 관리척도에 대해 설명하며, 3장에서는 관리척도의 계산을 위한 측정값을 소프트웨어 생명주기에 따라 분류하고 측정값 데이터 수집 방법과 측정값 우선순위 부여방법에 대해 분석한다. 4장에서는 이러한 관리척도를 조직에서 활용하게 하는 척도 프로그램을 도입하는 방법과 지원도구를 설명하고 5장에서 결론을 맺는다.

2. CMM과 관리척도

CMM은 조직의 프로세스 개선활동을 지원하는 성숙도 평가모형으로서 그림 1과 같이 소프트웨어 개발 프로세스의 성숙도를 5개의 계층적이고 연속적인 수준으로 구분한다[5]. 이 모형은 제품 자체의 관리보다는 제품을 생산하는 프로세스의 효과적인 관리에 초점을 맞추어, 혼란스럽고 미성숙된 프로세스 상태(initial)로부터 보다 체계적인 프로세스로 성숙되어가는 과정을 단계별로 나타내고 있다. 이 접근방법은 CMM을 적용하면 지속적인 프로세스 개선이 가능하며, 개선된 프로세스는 제품 품질에 반영된다는 기본가정에 근거한다. 따라서 성숙된 소프트웨어 개발 프로세스 관리체계를 가진 조직은 1) 소프트웨어 크기, 비용, 품질, 그리고 개발 일정의 예측가능성 향상, 2) 프로세스의 관리를 통한 제품 품질의 향상, 3) 소프트웨어 개발 프로세스의 가시성(visibility) 증대와 같은 조직적 이득을 가져온다는 것이다 [2].

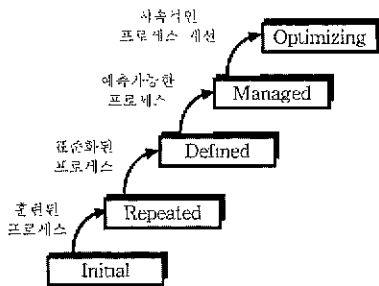


그림 1 CMM의 성숙단계 구조

CMM에서는 소프트웨어 개발 프로세스에서 해당 성숙도를 달성하기 위해 중시해야 하는 영역을 KPA(Key Process Areas)로 규정하여 표 1과 같이 각 수준별로 제시하였다[5]. 표 1에서 볼드체로 된 부분이 측정과 깊은 관련이 있는 KPA이다.

Gaffney 등은 소프트웨어 프로젝트 상태를 추적하고 감시하는데 사용될 수 있는 38개의 관리 척도를 제시하여 프로젝트 목표에 맞는 척도를 선택하여 설명할 수 있도록 하였다[2]. 38개의 관리척도는 소프트웨어 제품 크기, 비용, 개발 일정, 프로젝트 기술 안정성, 프로젝

표 1 CMM의 KPA

성숙 단계	Key Process Area	약어
5. Optimizing	Prevent defects	(DP)
	Managed process change	(PC)
	Managed technology innovation	(TI)
4. Managed	Process measurement and analysis	(PA)
	Management of quality	(QM)
3. Defined	Focus on organization process	(PF)
	Define organization process	(PD)
	Training program	(TP)
	Integrated software management	(IM)
	Software product engineering	(PE)
	Intergroup coordination	(IC)
Peer review	(PR)	
2. Repeatable	Manage requirements	(RM)
	Plan software projects	(PP)
	Track and oversee software projects	(PT)
	Manage software subcontracts	(SM)
	Software quality assurance(SQA)	(QA)
	Software configuration management (SCM)	(CM)

트 상태, 품질, 획득가치(Earned Value), 컴퓨터 자원으로 구성된 8개의 범주로 구분된다.

Gaffney는 이 38개의 관리척도를 CMM의 KPA별로 연결하였다. 이 내용을 CMM의 각 수준별로 정리하면 표 2, 표 3, 표 4와 같다 [2]. CMM 2수준의 KPA별로 연결된 관리척도는 표 2에서 보는 바와 같이 총 25개 항목에 대해 연결되어 있다. 이 표에서 PP2(KPA PP의 두 번째 척도)와 PT2(KPA PT의 두 번째 척도)와 같이 적용 KPA는 다르나 동일한 내용을 가진 척도가 포함되어 있는데 이러한 것들은 두 KPA에서 공통적으로 필요로 하는 척도라고 할 수 있다. CMM 3수준의 KPA별로 연결된 관리척도는 표 3에서 보는 바와 같이 총 8개 항목에 대해 연결되어 있다. 이 수준에서는 프로세스 단계별 결함과 에러에 대한 수집에 비중을 두고 있다. 마지막으로 표 4에서 보는 바와 같이 CMM 4수준의 KPA별로 연결된 관리척도는 총 8개의 항목이고, 프로세스 단계별 결함 또는 에러에 대한 정보뿐만 아니라 CMM 성숙도가 높아지면서 측정되는 측정값을 데이터베이스로 구축하여 활용하는 전사적 의미의 척도 프로그램으로의 확장에 초점을 맞추고 있다.

표 2 CMM 2단계 관리척도

KPA	번호	관리척도	범 주
PP	1	새로운, 제 사용된, 총 KSLOC(또는 FP)	크기
	2	LM, LH, 달러(\$), 비율로 표현된 LM 또는 LH	비용
	3	경과 달 수	일정
	4	제품 완결성 [전체 소프트웨어 완결 비율(KSLOC, FP 등등)]	획득가치
QA	1	단계별 소프트웨어 개발프로세스를 위한 독립 감사	N/A
CM	1	ECP의 수	안정성
	2	(정의된 요구사항 수)/(전체 요구사항 수) 100	
PT	1	새로운, 제 사용된, 총 KSLOC(또는 FP), KESLOC(현재/최초) 100	크기
	2	LM, LH, 달러(\$), 비율로 표현된 LM 또는 LH	비용
	3	(경과된 달 수/예상 달 수)100, 경과된 달 수	일정
	4	사용된 CPU/사용 가능한 CPU, 사용된 저장공간/사용 가능한 저장공간[메모리 활용 비율]	컴퓨터자원
	5	(목표 mips/host mips)*(mips의 기능크기/host 처리시간)=표준 기능을 위한 예상 목표 mips[표준 기능 처리를 위한 목표 CPU 처리 속도]	
	6	(메시지 길이)(도착 비율)/(처리 속도) [I/O capacity 사용비율]	
	7	(설계된 요구사항/전체 요구사항)100, (코딩된 요구사항/전체 요구사항)100, (설계된 유닛/전체 유닛)100, (코딩된 유닛/전체 유닛)100	상태
	8	(시험된 요구사항/전체 요구사항)100, (통과된 시험/전체 시험)100, (시험된 유닛/전체 유닛)100	
	9	(통합된 유닛/전체 유닛)100	
	10	(배치된 인력/계획된 인력)100	안정성
	11	(완결된 SAI/전체 SAI)100	
	12	결함 또는 에러/KSLOC(실제 또는 예측된 KSLOC) [코딩 단계]	품질
	13	예상 결함/KSLOC [배포 단계]	
	14	(처리된 PTR/전체 PTR)100	
	15	PTR/KSLOC [통합시험 단계]	
	16	PTR/KSLOC [시스템시험 단계]	
	17	제품 완결성 [전체 소프트웨어 완결 비율(KSLOC, FP 등등)]	
	18	정기적 경영점토를 보장하는 공식화된 절차	N/A

<용어설명>

- DDR : Detailed Design Review, •ECP : Engineering Change Proposal, •FP : Function Point,
- KSLOC : Thousand Source Lines of Code, •LH : Labor Hour, •LM : Labor Month,
- MIPS : Million Instructions Per Second, •PTR : Program Trouble Report,
- PDR : Preliminary Design Review, •SAI : Software Action Item

표 3 CMM 3단계 관리척도

KPA	번호	관리척도	범 주
PD	1	결함 또는 에러/KSLOC(실제 또는 예측된 KSLOC)[PDR, DDR단계]	품질
	2	유닛(모듈) 개발 진행 상황에 대한 공식기록	N/A
PE	1	(통과된 테스트 수/전체 테스트 수)100	상태
PR	1	결함 또는 에러/KSLOC(실제 또는 예측된 KSLOC) [PDR, DDR단계]	품질
	2	(완결된 SAI/전체 SAI)100 [코딩단계에서 발생된 SAI 추적]	
	3	결함이나 에러/KSLOC(실제 또는 예측된 KSLOC) [코딩 단계]	
	4	내부 소프트웨어 디자인 검토	N/A
	5	소프트웨어 코드 검사 수행	

표 4 CMM 4단계 관리척도

KPA	번호	관리척도	범 주
PA	1	계획 대비 실제 결함 또는 에러수(결함 또는 에러/KSLOC) [PDR, DDR단계]	품질
	2	계획 대비 실제 결함 또는 에러수(결함 또는 에러/KSLOC) [코딩 단계]	
	3	프로세스 데이터베이스 설립(전체 프로젝트 프로세스척도 대상)	경험 데이터베이스
	4	제품에 잔존하고 있는 에러의 분포와 특성을 파악(코딩과 시험 단계에서의 에러 데이터를 토대)	
	5	주요 프로세스단계를 위한 소프트웨어 생산성 분석	
QM	1	결함 또는 에러/KSLOC (실제 또는 예측된 KSLOC) [PDR, DDR]	품질
	2	단계별 기능성 시험의 시험범위는 측정되고 기록된다	
	3	검토 데이터(review data) 분석 [PDR, DDR, 코딩 단계]	

3. 관리척도의 측정값 수집

3.1 척도와 측정값의 관계

관리척도는 그 계산에 필요한 측정값들의 함수이다. 척도와 측정값간의 관계는 3가지로 요약될 수 있다. 첫번째는 측정값이 그대로 척도가 되는 경우로 KSLOC(Thousand Source Lines of Code) 등이 그 예이다. 두번째는 두 개이상의 측정값이 함수연산을 통해서 하나의 척도를 만드는 경우이며, 크기 또는 비용의 비율척도가 여기에 속한다. 마지막으로 하나의 척도 계산에 소프트웨어 개발 프로젝트의 성격 및 유형에 따라 적용되어지는 계산공식이 달라지는 경우로 LM(Labor Month) 척도가 그 예이다. 이상은 그림 2와 같이 나타낼 수 있다.

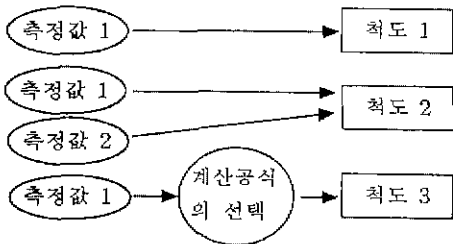


그림 2 척도와 측정값의 관계

그림 2와 같은 척도와 측정값간의 관계를 기반으로 하여 만들어질 수 있는 자동화된 척도 프로그램 도구는 그림 3과 같은 데이터 흐름을 갖는다. 그림 3에서는 각 측정값 데이터로 계산할 수 있는 척도를 보여주고 있다. 예로서

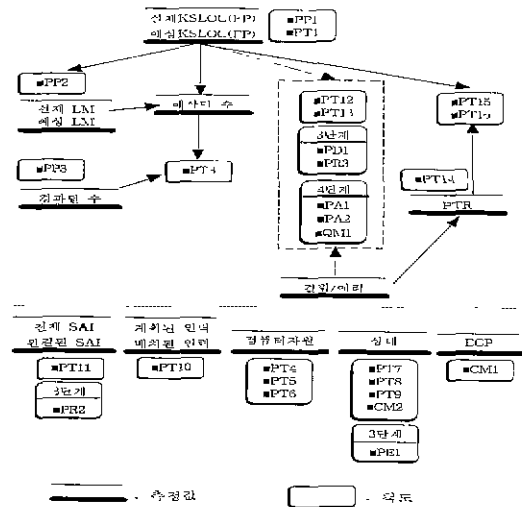


그림 3 전체 데이터 흐름도

KSLOC와 FP(Function Point)는 그 자체의 값만으로도 PP1과 PT1의 척도를 산출할 수 있으며, LM과 예상기간 등의 척도값에 영향을 주기도 한다. 그리고 PT15와 PT16과 같이 다른 척도값인 PTR(Program Trouble Report) 과 동시에 KSLOC와 FP를 사용하여 함수연산을 통해서 하나의 척도를 구하기도 한다.

Gaffney 등은 8개의 관리지침으로 분류된 척도를 실제 척도 프로그램으로 운영하기 위해 측정값들을 수집할 때 이를 측정범주로도 이용할 수 있음을 제시하였다[2]. 다음 표 5는 측정범주별로 분류된 CMM 각 수준과 연결된 관리척도를 구하기 위해 필요한 측정값들을 소프트웨어 개발 생명주기 과정에서 분석한 것이다.

표 5 관리척도의 측정

프로세스		준비	요구사항	디자인		코딩	시험		배포
단계	척도			PDR	DDR		통합 시험	시스템 시험	
2단계	PP1	단계별 예상 KSLOC의 FP				KSLOC, FP	KSLOC, FP	KSLOC, FP	KSLOC, FP
	PP4	단계별 예상 KSLOC와 FP				KSLOC, FP	KSLOC, FP	KSLOC, FP	KSLOC, FP
	PT1	단계별 예상 KSLOC와 FP				KSLOC, FP	KSLOC, FP	KSLOC, FP	KSLOC, FP
	PT12					결함/에러, KSLOC			
	PT13								예상결함, KSLOC
	PT14						PTR수, 처리된 PTR수	PTR수, 처리된 PTR수	PTR수, 처리된 PTR수
	PT15						PTR, KSLOC		
	PT16							PTR, KSLOC	
	PP2	예상 LM, LH	LM, LH	LM, LH	LM, LH	LM, LH	LM, LH	LM, LH	LM, LH
	PP3								경과달 수
	PT3	예상될 수							경과달 수
	PT4	사용 가능한 CPU, 사용가능한 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간	사용된 CPU, 사용된 저장공간
	PT5	host 처리시간	목표 mips				host mips, mips의 가능크기		
	PT6	처리속도					메시지 길이, 도착 비율		
3단계	PT7	전체 요구사항	실제된 요구사항, 전체 유닛, 설계된 유닛			코딩된 요구사항, 코딩된 유닛			
	PT8	전체 요구사항	전체 유닛				시험된 유닛, 시험, 통과된 시험, 시험된 요구사항	시험, 통과된 시험, 시험된 요구사항	
	PT9		전체 유닛				통합된 유닛	통합된 유닛	
	CM2	전체 요구사항, 정의된 요구사항							
	PT10	계획된 인력	배치된 인력	배치된 인력	배치된 인력	배치된 인력	배치된 인력	배치된 인력	배치된 인력
	PT11	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI
	CM1	ECP	ECP	ECP	ECP	ECP	ECP	ECP	ECP
	PD1			결함/에러, KSLOC	결함/에러, KSLOC				
	PR3					결함/에러, KSLOC			
	PD2			전체 유닛, 설계된 유닛	전체 유닛, 설계된 유닛	코딩된 유닛	시험된 유닛, 통합된 유닛	시험된 유닛, 통합된 유닛	
4단계	PE1						시험, 통과된 시험	시험, 통과된 시험	
	PR2	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI	SAI, 완결된 SAI
	PR4			설계검토 횟수	설계검토 횟수				
	PR5					코드 검사			
	PA1	단계별 예상 결함/에러, KSLOC		결함/에러, KSLOC	결함/에러, KSLOC				
	PA2	단계별 예상 결함/에러, KSLOC				결함/에러, KSLOC			
4단계	전체 척도	전체 척도	전체 척도	전체 척도	전체 척도	전체 척도	전체 척도	전체 척도	
PA4					결함/에러	결함/에러	결함/에러		
PA5	생산성 분석	생산성 분석	생산성 분석	생산성 분석	생산성 분석	생산성 분석	생산성 분석	생산성 분석	
QM2						시험범위	시험범위		
QM3			검토데이터	검토데이터	검토데이터				

3.2 측정값 데이터 수집

척도를 관리에 활용하기 위한 첫 단계는 각 척도 사용자와 척도 계산을 위해 수집하여야 할 측정값을 얻을 수 있는 대상을 식별하는 것이다[6]. 프로젝트 계획 관리자, 형상관리자, 품질 보증담당자, 그리고 각 개발 단계별 담당자들은 측정값의 데이터 수집을 위하여 그 역

표 6 프로젝트의 각 담당자를 대상으로 수집하여야 할 측정값

측정값	범주	PPM	CM	QA	분석	설계	코딩	시험
예상 KSLOC	크기	×						
KSLOC		×	×				×	
예상 FP		×						
FP		×	×				×	
예상 LM	비용	×						
LM		×	×	×	×	×	×	×
결합의 수	품질			×			×	×
예상 결합의 수		×		×		×		×
PTR				×				×
경과 달 수	일정	×						
예상 달 수		×						
컴퓨터 자원	컴퓨터 자원	×			×			×
전체 요구사항	인정성, 상태	×			×	×		
설계된 요구사항	상태					×		
코딩된 요구사항							×	
시험된 요구사항								×
전체 유닛						×		
설계된 유닛						×		
코딩된 유닛							×	
통합된 유닛								×
전체 시험					×			×
통과된 시험					×			×
정의된 요구사항		안정성	×					
매치된 인력	×					×		
계획된 인력	×							
전체 SAI	×					×		
완결된 SAI	×			×	×	×	×	×
ECP	×		×			×		

<용어 설명>

- PPM: Project Planning Manager
- CM: Configuration Manager
- QA: Quality Assurance

할과 책임이 다르며, 이들이 각각 개발 단계에서 수행하고 있는 작업에 따라 수집하여야 할 데이터도 다르다. 표 6은 각 담당자들이 수집해야 할 측정값 데이터를 예로서 분류하여 정리한 것이다.

측정값 데이터 수집에는 우선적으로 수집되어야 할 데이터가 고려되어야 한다. 소프트웨어 개발 프로젝트를 수행할 때 소프트웨어의 크기, 개발비용, 개발일정을 정확하게 추정할 수 있다면 개발 효율성 및 효과성을 높일 수 있고, 높은 품질의 소프트웨어를 생산할 수 있다. 특히 계량적 프로세스 관리를 위해서는 소프트웨어 크기가 우선적으로 추정될 필요가 있다. 왜냐하면 이 값은 대부분 개발비용, 개발일정 추정의 계산에 초기 입력값으로 사용되며, 잘못 추정된 소프트웨어 크기는 부정확한 개발비용 및 개발일정 예측을 초래하기 때문이다. 이외의 나머지 척도 사용을 위한 측정값 수집의 우선순위는 조직환경에 따라 다르게 설정이 가능하다(그림 3 참조).

4. 소프트웨어 척도 프로그램 및 지원 도구

소프트웨어 척도 프로그램(metric program)은 조직내에서 소프트웨어 척도를 수집하고 해석하기 위한 정형화된 절차이다. 따라서 척도 프로그램은 개발절차와 자연스럽게 연관되어 있으며, 척도 결과는 개발 프로세스로 피드백 되는 경우가 많다. 척도 프로그램을 통해 해당 프로세스의 효과성을 식별하고 감시할 수 있으며, 프로세스가 어떻게 성장하는가를 관찰할 수 있다. 최근에는 소프트웨어 척도 프로그램을 조직의 소프트웨어 개발 프로세스와 결합시킴으로써, 프로세스를 개선하고 원하는 프로세스 능력 수준을 달성하려는 시도가 늘고 있다[8]. 다음은 이 척도 프로그램을 조직내에서 어떻게 도입하는가에 대한 간단한 접근방법을 설명한다[9]. 척도 프로그램은 다음 단계에 따라 구축될 수 있다.

- 단계 1) 목적 설정 단계(Goal Phase): 조직의 전략적 측면에서 척도 프로그램의 목적을 설정한다. 그리고 조직의 목적을 각 소프트

웨어 프로세스와 연관시킨다. 조직의 목적이 추상적이거나 범위가 클 경우 여러 개의 구체적인 부목적(subgoals)들로 나눌 수 있다. 예를 들면 도달하고자 하는 CMM의 목표 수준이 목적이 될 수 있고, 각 수준의 KPA는 부목적이라고 할 수 있다.

• 단계 2) 척도 연결 단계(Metric Phase): 프로세스 측면에서 목적설정 단계에서의 목적 또는 부목적들과 연관된 척도를 설정한다. 척도를 할당하는 방법은 GQM(Goal-Question-Metrics) 접근방법이 대표적이다. 예를 들면 관리척도들은 CMM의 각 수준별 KPA별로 관련되어 있다(표 2, 3, 4 참조).

• 단계 3) 측정값 연결 단계(Measurement Phase): 프로젝트를 수행하면서 척도값을 얻기 위해 수집할 필요가 있는 측정값을 설정한다. 예를 들면 표 5는 관리 척도 계산에 필요한 측정값을 보여준다.

• 단계 4) 역할 할당 단계(Role Phase): 측정값을 수집하거나 척도를 계산하고, 관리하고 활용하는 담당자를 할당한다. 표 6과 같이 각 측정값에 대해 담당자를 정할 수 있고, 각 관리자를 포함하여 담당자별로 다양한 뷰(view)를 갖게 하여 접근을 제한시킬 수 있다.

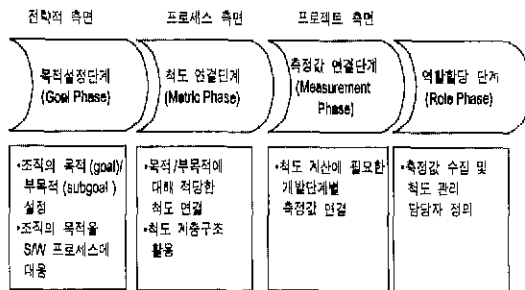


그림 4 척도 프로그램의 적용 단계

조직내에서 척도 프로그램을 활용하기 위해 필요한 데이터를 수집하고 분석하는 것은 매우 힘든 작업이다. 따라서 자동화된 도구의 지원을 필요로 한다. 최근 인터넷의 발달로 여러 분야에서 Web기술을 응용한 많은 프로그램이 개발되고 있다. 소프트웨어 공학 분야에서는 Boehm의 COCOMO가 Java언어로 구현되어 Web 브라우저를 통한 데이터 입력으로 소프

트웨어 개발 비용에 대한 관리가 가능함을 보여주고 있다. STARS(Software Technology for Adaptable, Reliable Systems)에서는 1995년에 Web에 기반한 소프트웨어 공학 환경(software engineering environment)하에서의 정보 수집을 위한 인터페이스 구축에 관한 프로젝트를 수행하였다. 이 프로젝트는 SE-Web(Software Engineering Environment Web)라는 이름으로 수행되었으며, Web을 기반으로 한 프로세스, 프로젝트, 제품에 대한 정보를 가시화 할 수 있는 체계를 제시하였다. SEWeb은 Web 기술을 활용하여 프로그램을 구축함으로써 물리적 컴퓨팅 환경의 지역적 분산이나 기종에 상관없이 사용자가 원하는 정보를 얻을 수 있게 하였다[10][11].

한국전자통신연구원에서는 고려대와 공동으로 Web 기술을 활용하여 그림 5와 같은 척도 프로그램 지원 도구를 구현하고 있다[12]. 본 지원도구는 척도 프로그램 관리자, 프로젝트 실무담당자, 관리자(프로젝트 관리자, 프로세스 관리자, 경영자 등)가 사용할 수 있다. 척도 프로그램 관리자는 조직에 필요한 척도를 등록하고, 관련 측정값을 설정하고 데이터 수집 담당자를 정한다. 각 담당자가 측정값을 수집하면 이를 분석하여 척도를 계산하는데 사용되는데 이때 외부 통계 처리 프로그램 등이 사용되어 계산의 정확성 및 신속성을 높일 수 있다. 이러한 척도 계산 결과의 추이 등은 관리자에게 제시되어 의사결정에 사용되거나 진도를 점검하고 관리하는데 사용될 수 있다. 모든 사용자는 Web 기반의 인터페이스 및 필터를 통해 데이터베이스에 저장된 척도 관련 데이터에 접근할 수 있다.

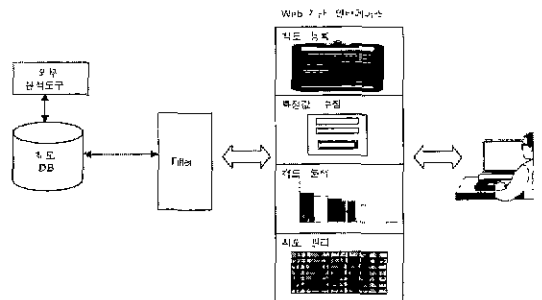


그림 5 척도 프로그램 지원도구

5. 결 론

본고에서는 계량적으로 프로세스를 관리하기 위한 방법으로 CMM 기반의 38개 관리척도를 설명하였다. 관리척도는 8개 범주 및 CMM 수준별로 분류하였고, 각 척도값의 계산을 위해 필요한 측정값을 정리하였다. 또한 측정값 수집 단계는 소프트웨어 개발 주기와 개발 담당자에 따라 분류하였다. 또한 이러한 계량적 프로세스 관리 기법을 조직에 적용하여 사용하기 위한 척도 프로그램 및 지원도구를 설명하였다.

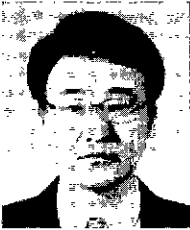
본고에서 설명한 척도를 기반으로 한 계량적 프로세스 관리 방법은 소프트웨어 개발 조직으로 하여금 국제적 품질 기준에 부합하는 수준에 빠른 시간에 도달할 수 있는 접근방법을 보여준다. 그러나 38개 관리척도는 CMM의 모든 성숙 수준을 포괄하기 못하기 때문에 아직 계량화하지 못한 척도가 존재한다. 따라서 CMM이나 SPICE 등의 국제적 프로세스 품질 기준의 전체 능력수준(capability level)을 표현할 수 있는 척도의 개발 및 적용이 필요하다. 또한 소프트웨어 개발 과정은 조직의 규모, 특성, 문화적 차이로 말미암아 다양한 형태를 가질 수 있음을 고려할 때 척도 프로그램의 도입은 조직의 현 상황을 고려하여 유연하게 접근할 필요가 있다.

계량적 프로세스 관리 기술은 그 자체가 프로세스 개선을 위한 하나의 도구로 사용되어야 하며 그 자체가 목적이 될 수는 없다. 또한 해당 기술이 성공하기 위해서는 경영층의 전폭적인 지원과 함께 조직내에 효과적인 프로세스 환경 또는 기반구조(infrastructure)가 갖추어져야 하며, 그 기술을 지원하기 위한 역할 및 체계가 정립될 필요가 있다.

참고문헌

- [1] 이종무, 소프트웨어 품질평가 투입요소 결정에 관한 연구, 고려대학교 대학원, 박사학위 논문, 1997.
- [2] Gaffney, J., Cruickshank, R., Werling, R. and Felber, H. F., *Software Measurement Guidebook*, International Thomson Computer Press, Boston, 1995.
- [3] Daskalantonakis, M. K., "A Practical View of Software Measurement and Implementation Experiences Within Motorola," *IEEE Transactions on Software Engineering*, Vol.18, No.11, pp. 998-1010, November 1992.
- [4] Giles, A. E. and Daich, G. T., "Metrics Tools," *Crosstalk*, February 1995. <http://stsc/hill.af.mil/crosstalk/1995/feb/metrics.html>.
- [5] Paulk, M. C., Curtis, B. and Chrissis, M. B., *Capability Maturity Model for Software*, Software Engineering Institute, CMU/SEI-91-TR-24, February 1991.
- [6] Möller, K. H. and Paulish, D. J., *Software Metrics*, IEEE Press, New York, 1993.
- [7] Boehm, B. W., *Software Engineering Economics*, New Jersey, Prentice-Hall, 1981.
- [8] Hall, T. and Fenton, N., "Implementing Effective Software Metrics Programs," *IEEE Software*, March/April 1997.
- [9] 김길조, 전인걸, 구자경, 안유환, "S/W 프로세스 개선을 위한 매트릭 접근 방법", 98 소프트웨어 품질관리 심포지움 논문집, 1998.
- [10] Uzzle, L., "SEWeb: A Software Engineering Environment Information Interface based on World Wide Web Technology," *STARS-PA32-AF03/001/00*, December 1995.
- [11] Schwarting, D. C. and Guy, E., "Guidelines for WWW-based Metrics Automation," *STARS-PV03-A033/001/00*, May 1996.
- [12] 정호원, "소프트웨어 품질 데이터 수집 및 분석", 한국전자통신연구원(ETRI) 위탁

연구과제, 제2차 년도 최종보고서. 1998.
11.



안 유 환

서울대학교 공과대학 산업공학과에서 학사, 한국과학기술원 경영과학과에서 석사학위를 취득하였다. 현재 한국전자통신연구원 컴퓨터·소프트웨어기술연구소 S/W 공학연구부 S/W 품질보증연구팀 팀장으로 재직중이고 SPICE 심사원이다. 관심분야는 S/W 품질 보증 및 평가, S/W 프로세스 관리, Integrated Methodology 등이다.

Email: ywahn@etri.re.kr



김 길 조

서울대학교 공과대학 산업공학과에서 학사, 한국과학기술원 경영과학과에서 석사학위를 취득하였다. 현재 한국전자통신연구원 컴퓨터·소프트웨어기술연구소 S/W 공학연구부에 선임연구원으로 재직중이고 SPICE 심사원이다. 관심분야는 S/W 개발방법론, S/W 품질보증, S/W 프로세스 개발 환경 등이다.

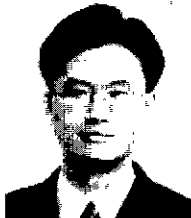
Email: kgj@etri.re.kr



오 세 원

고려대학교 경영학과에서 학사학위를 취득하고, 현재 동 대학원에서 MIS 석사학위를 준비중이다. SPICE 심사원이며, 추후 LG-EDS 시스템의 품질경영팀에서 근무할 예정이다. 관심분야는 소프트웨어 프로세스 개선 및 개별 비용 산정, 소프트웨어 품질 평가 등이다.

Email: ohsewon@kuccnx.korea.ac.kr



김 종 윤

고려대학교 통계학과에서 학사학위를 취득하고, 현재 경영학과 대학원에서 MIS 석사학위를 준비중이다. 관심분야는 소프트웨어 프로세스 개선, 소프트웨어 품질 평가, BPR 등이다.

Email: kjylyj@kuccnx.korea.ac.kr

● 제1회 한국데이터베이스 학술대회 ●

- 일 자 : 1999년 2월 26~27일
- 장 소 : 동국대학교
- 주 최 : 데이터베이스연구회
- 문 의 처 : 송실대학교 컴퓨터학부 이상호 교수
Tel. 02-820-0922