

CMMI 기반 결함 예측 성과 모델을 이용한 소프트웨어 개발 노력 분배 연구

곽미경*, 안영정**, 최진영**

*고려대학교 소프트웨어공학과

**고려대학교 컴퓨터학과

e-mail : *meiqing@korea.ac.kr, **{yjahn, choi}@formal.korea.ac.kr

A Study on Software Development Effort Allocation using Defect Prediction Performance Model based on CMMI

Mi-Kyung Kwak*, Young-Jung Ahn**, Jin-Young Choi**

*Dept. of Software Engineering, Korea University

**Dept. of Computer Science and Engineering, Korea University

요 약

소프트웨어 프로젝트를 진행할 때, 소프트웨어 개발에 투입할 노력의 정확한 추정과 더불어 소프트웨어 생명주기 단계별 적절한 개발노력을 투입하는 것은 프로젝트 성공을 위해 필요한 요소 중 하나이다. 조직의 과거 데이터를 활용한 기존의 개발노력 분배 방식은 단계별로 발생하는 결함의 양에 따라 개발노력의 투입량 변동이 발생할 수 있다. 본 연구에서는 CMMI 조직 프로세스 성과(Organization Process Performance) 프로세스 기반의 결함 예측을 이용한 개발노력 분배 성과모델을 제시하고, 제시한 성과모델의 예측값과 프로젝트 수행 결과 값의 비교를 통해서 제시한 성과모델의 유효성 및 결함과 개발노력 분배의 연관성에 대해서 검증 하고자 한다.

1. 서론

2003 년에 발행된 미국의 IT 프로젝트에 관한 조사 보고서(The Standish Group 의 CHAOS Report)에 따르면 2002 년 미국에서 실시된 소프트웨어 프로젝트 중 성공한 프로젝트는 전체의 34%에 불과했다. 실패한 프로젝트의 51%는 품질, 납기, 비용 중 하나 이상의 항목을 만족시키지 못했으며, 15%는 중도에 멈춘 프로젝트라고 보고되었다.[1] 위에 보고된 프로젝트의 실패 원인으로는 프로젝트 관리의 부재, 범위 관리 능력 부족, 우수 프로그래머 부족 등 여러 원인이 있겠지만, 소프트웨어 규모 및 개발노력 추정의 부정확성을 중요한 원인 중 하나로 들 수 있다.

2004 년도 정보통신부가 국가 및 공공기관의 정보화 사업의 S/W 개발비 산정방식을 프로그램 라인수만을 계산해 산정하는 LOC(Lines Of Code)방식에서 기능점수(Function Point)방식으로 개정하면서, 과거 소프트웨어 규모에 대한 기준 조차 명확하지 않은 상태로 소프트웨어 개발 비용 및 규모를 예측하던 때에 비해서는 소프트웨어 규모 산정을 위한 많은 연구와 개선 작업이 이루어지고 있다. 그럼에도 기업들이 소프트웨어 개발 프로젝트를 진행할 때, 소프트웨어 개발에 투입할 노력의 추정이 어려운 게 현실이다. 계획 단계에 개발노력에 대해서 정확히 산정했을지라도 단계별 투입할 개발노력 산정의 부정확성으로 단계별 적절한 인력을 투입하지 못 했을 경우, 납기준수가 어려워 짐에 따라 개발 단계별 적절한

개발노력을 투입하는 것은 프로젝트 성공을 위해 필요한 요소 중 하나이다.

본 연구에서는 소프트웨어 생명주기 단계별 개발노력을 산정하는데 변수로 작용할 수 있는 결함을 CMMI 기반의 결함 예측 성과모델을 활용하여 단계별 개발노력 분배 방법을 향상시키기 위한 방안을 제시하고자 한다.

본 연구의 배경 및 관련 연구에 대해 논하고, 결함 예측 성과모델을 활용한 소프트웨어 개발노력 분배 방법을 제시한 후, 사례를 통해 이를 검증하고 결론에 대해 논하고자 한다.

2. 연구 배경 및 관련 연구

2.1 연구 배경

일반적으로 기업은 개발노력 산정을 위해 과거의 프로젝트들의 평균 생산성 데이터를 수집하고, 이를 바탕으로 총 개발노력을 산정하며, 개발노력은 프로젝트 특성을 반영한 여러 보정계수를 적용하여 다음과 같이 산정한다.

$$\text{개발노력} = \text{S/W 개발규모} * \text{평균생산성}^1 * \text{보정계수}^2$$

1) 평균생산성 = 개발노력(MM) ÷ S/W규모(FP)

2) 개발유형, 투입인력 경험년수, 유사업무 경험정도 등 프로젝트 특성을 반영한 보정계수

이렇게 산정된 총 개발노력은 소프트웨어 생명주기 단계별로 분배되어 추진하게 되는데, 각 단계별 개발노력은 분석, 설계, 구현, 시험 단계별 생산성을 예측하여 분배하거나, 조직의 과거 프로젝트의 단계별 개발노력 분포 데이터를 이용하여 분배하고 있다. 이러한 형태로 개발노력을 분배해서 프로젝트를 진행할 경우, 단계별로 발생 되는 결함의 양에 따라서 개발노력 투입량이 변동될 수 있으며, 결함 발견 시기가 늦어 질수록 결함 제거를 위한 개발노력의 추가 투입 문제가 발생하게 되는 문제점이 발생된다.

2.2 관련 연구

2.2.1 개발노력 단계별 분배

소프트웨어 생명주기 전반에 걸쳐 투입될 총 개발노력이 결정된 후, 실제로 소프트웨어를 개발하는 각 세부 단계인 요구사항 분석, 설계, 구현, 시험과 유지보수의 각 단계별로 투입될 노력과 기간을 확정하는 연구로는 Putnam[2]과 Yamada et al.[3]등이 있다.

Putnam[2]은 1960~1970년대 개발된 150개 프로젝트에 대한 경험치를 적용하여 요구사항 명세화 단계에 투입될 노력과 기간은 전체의 약 20%가 소요되며, 개발 시점 이후에 프로젝트 개발 및 유지보수 단계에서 소요된 총 개발노력만을 보았을 때는 설계 및 코딩단계에 15%, 시험 및 검증단계에 20%, 변경, 유지보수와 확장 단계는 55%와 관리부분에 10%가 소요가 된다고 보고했다. 즉, 프로젝트 진행 전 과정을 통해 전체 투입 노력에 대해 설계 및 구현단계는 12%, 시험 및 검증단계는 16%, 변경 및 유지보수와 확장단계는 44%, 관리는 8%의 노력이 소요된다고 했다. 또한 Yamada et al.[3]은 소프트웨어 개발 자원의 약 40%~50%가 시험단계에 소요된다고 보고했다. 이들 연구 결과를 종합했을 때 결과, 시험단계 또는 시험단계 이후 소프트웨어 완성도를 위해 투입되는 개발노력이 매우 많다는 것을 알 수 있으며, 이는 현재 프로젝트에 투입하고 있는 단계별 개발노력 비율과는 상이한 점을 보이고 있다.

2.2.2 CMMI

CMMI(Capability Maturity Model Integration)는 소프트웨어, 시스템, 하드웨어, 서비스와 관련된 모든 프로세스 개선 활동을 지원하기 위해 카네기멜론대학 소프트웨어공학연구소(SEI)가 만든 모델로 조직의 전체적인 프로세스 능력을 판단 하는 단계적 표현방법(Staged-Representation)과 개별 프로세스 영역별 평가를 하는 연속적 표현방법(Continuous-Representation)으로 표현된다. 단계적 표현방법(Stage-Representation)은 5단계의 성숙도(Maturity) 레벨로 관리되고, CMMI 성숙도 중 4레벨은 프로젝트 결과를 정량적으로 관리(Quantitatively Managed) 하는 조직 수준으로 프로세스의 성과를 정량적으로 예측하고, 이상 원인을 찾아 적절하게 시정 조치를 할 수 있는 단계를 의미한다. CMMI 성숙도 4레벨 프로세스로는 정량적 프로

젝트 관리(Quantity Project management) 프로세스와 조직 프로세스 성과(Organization Process Performance)프로세스로 구성된다. 성숙도 4레벨 프로세스인 조직 프로세스 성과(Organization Process Performance) 프로세스는 조직의 표준 프로세스들의 수행결과를 바탕으로 품질 및 프로세스 성과 목표를 수립하고, 성과 목표를 관리하기 위한 프로세스 성과 베이스라인과 프로세스 성과모델 등을 적용하여 조직의 정량적 관리를 하기 위한 프로세스이다.[4]

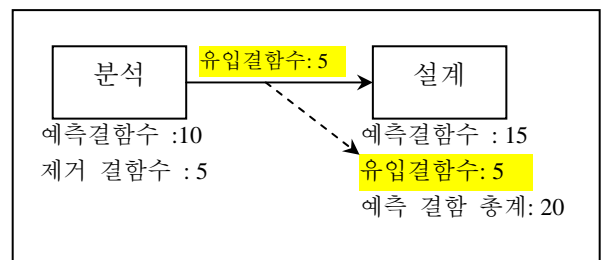
2.2.3 프로세스 성과모델

CMMI의 성숙도 4레벨의 프로세스인 조직 프로세스 성과(Organization Process Performance) 프로세스에서 관리되는 모델로 일명 예측 모델로도 일컬어진다. 프로세스 성과모델은 프로세스 측정치와 작업 산출물 간의 관계를 통계적으로 표현하고, 프로젝트의 후반부가 되어야 측정할 수 있는 값(결함, 투입공수, 규모 등)들을 예측하는데 사용된다. 프로세스 성과모델은 프로세스 성과 베이스라인³⁾을 기초로 만들어지며, 거의 모든 프로젝트 목표들에 대해 만들어질 수 있다.[5]

3. CMMI 기반 결함 예측 성과모델

CMMI 성숙도 4레벨부터는 조직의 프로세스 성과모델 개발을 통해 조직의 비즈니스 목표 달성을 위한 프로세스 성과를 관리한다. 프로세스 성과모델은 프로젝트의 통계적 데이터 관리를 통해 프로젝트에서 발생하는 데이터가 통제할 수 있는 범위 수준으로 관리되고 있는지를 판단하고, 또한 프로젝트 결과값에 대한 예측이 가능하게 한다.

프로젝트에서 결함 발견 활동으로는 단계별 동료 검토(Peer Review), 구현 단계의 단위테스트(Unit Test), 시험 단계의 통합테스트(Integration test), 시스템테스트(System Test), 인수테스트(Acceptance Test)등이 있다.

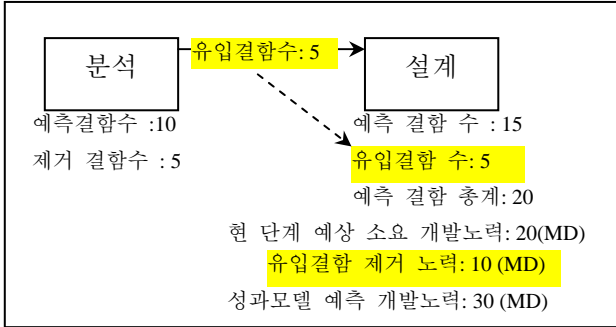


(그림 1) 결함 유입

(그림 1)은 결함 예측 성과모델에서의 결함 유입과정을 설명한 것이다. 결함 예측 성과모델은 조직의 결함밀도 성과 베이스라인을 기준으로 단계별 결함수를 예측할 수 있게 한다. 이전 단계에서 제거되지 못한 결함들은 잠재적 결함으로 판단하고, 다음 단계

3) 과거 조직의 실적 데이터를 가지고 컨트롤 차트를 이용하여 중앙값(Control Limit), 상한값(Upper Limit), 하한값(Low Limit)으로 관리하는 방식으로 프로세스의 안정성(Stability)을 판단하는데 사용된다.

결함으로 유입시킨다. 이전 단계에서 유입된 결함 수와 현 단계의 예측된 결함 수의 합이 현 단계에서 제거되어야 할 결함 수가 되고, (그림 2)에서와 같이 현 단계의 예상 소요 개발노력(과거 노력 분포 기준)과 전 단계에서 유입된 결함을 제거하는데 필요한 유입 결함 제거 노력이 더해져서 성과모델 예측 개발노력으로 산정된다.



(그림 2) 개발노력 산정

결함 예측 성과모델은 <표 1>과 같이 구성되고, 세부 항목에 대한 설명은 아래와 같다.

<표 1> 결함 예측 성과모델

<결함 예측 성과모델>		분석	설계	구현	시험
(A)과거 노력 분포(%)		15%	22%	32%	31%
(B)결함밀도 성과 베이스라인		0.029	0.043	0.075	0.057
결함	(C)규모(FP)	3,400	3,400	3,800	4,010
	(D) 예측 결함 수 (성과베이스라인 기반)	99	146	285	229
	(E) 유입 결함 수	0	60	140	121
	(F) 예측 결함 총계	99	206	425	349
	(G) 제거 결함 수	39	66	304	299
	(H) 예상 개발노력(MD)	7,215			
	(I) 결함 제거 평균 노력	0.063	0.150	0.375	0.750
개발 노력	(J) 노력 예측값(MD) (과거 노력 분포 기반)	1,082	1,587	2,309	2,237
	(K) 유입 결함 제거 예상 노력(MD)	0	197	1,153	91
	(L) 성과모델 예측 개발노력(MD)	1,082	1,784	3,462	2,327
	(M) 실제 투입 노력(MD)	156	518	3,995	2,701

- (A)과거 노력 분포(%): 조직에서 과거에 수행된 프로젝트들의 단계별 투입노력 분포
- (B)결함밀도 성과 베이스라인⁴⁾: 결함수 ÷ 규모(FP)
- (C)규모 : 단계별 산정된 규모

4) 프로젝트들의 단계별 결함밀도를 조직의 프로세스 성과베이스라인으로 관리하는 지수

- (D)예측 결함 수 : 단계별 예상되는 결함 수 (B) x (C)
- (E)유입 결함수: 이전 단계에서 예측된 결함 중 제거되지 못하고 다음 단계로 유입될 결함 수
- (F) 예측 결함 총계: (D) + (E)
- (G) 제거 결함 수: 단계별 제거된 결함 수
- (H) 예상 개발노력: 전체단계에 투입될 개발노력
- (I) 결함 제거 평균 노력: 결함당 평균 제거 노력
- (J) 노력 예측 값(과거노력분포): (A) x (H)
- (K) 유입 결함 제거 예상 노력 : (E) x (I)
- (L) 성과모델 예측 개발노력 : (I) +(K)
- (M) 실제 투입 노력 : 프로젝트 실제 투입 개발노력

4. 사례 연구를 통한 검증

본 연구를 검증하기 위해 CMMI 성숙도 레벨 4 수준을 적용하고 있는 A社의 정량적(통계적) 관리 지표를 활용하여 결함예측 성과모델을 수립하고, 결함예측 성과모델의 예측 값과 2007~2008년에 수행된 5개의 프로젝트의 수행 결과 값의 비교를 통해서, 제시한 성과모델의 유효성 및 결함과 노력 분배의 연관성에 대해서 검증 하고자 한다.

<표 2> 결함 예측 성과모델 적용 결과

<결함 예측 성과 모델>		분석	설계	구현	시험		
과거 노력 분포(%)		19%	24%	32%	25%		
결함 밀도 성과 베이스라인		0.029	0.043	0.075	0.057		
P1	규모	3,400	3,400	3,800	4,010		
	결함	예측 결함수(성과베이스라인 기반)	99	146	285	229	
		유입 결함 수	0	60	140	121	
		예측결함 총계	99	206	425	349	
		제거 결함 수	50	100	304	299	
	개발 노력	예상 개발 노력	4,134				
		결함 제거 생산성	0.063	0.150	0.375	0.750	
		노력 예측값(과거 노력 분포 기반)	785	992	1,323	1,034	
		유입 결함 제거 예상 노력	0	160	782	57	
		성과모델 예측 개발 노력	785	1,153	2,105	1,090	
	실제 투입 노력	785	800	2,368	1,112		
	P2	규모	3,572	3,572	4,245	4,578	
		결함	예측 결함수(성과베이스라인 기반)	104	154	318	261
			유입 결함 수	0	17	88	77
			예측결함 총계	104	170	407	338
제거 결함 수			87	82	330	285	
개발 노력		예상 개발 노력	4,366				
		결함 제거 생산성	0.063	0.150	0.375	0.750	
		노력 예측값(과거 노력 분포 기반)	828	1,045	1,394	1,089	
		유입 결함 제거 예상 노력	0	55	728	57	
		성과모델 예측 개발 노력	828	1,100	2,121	1,146	
실제 투입 노력		729	950	2,340	1,280		
P3		규모	6,310	6,310	7,512	9,829	
		결함	예측 결함수(성과베이스라인 기반)	183	271	563	560
			유입 결함 수	0	85	67	106
			예측결함 총계	183	356	631	666
	제거 결함 수		98	289	525	610	
	개발 노력	예상 개발 노력	8,052				
		결함 제거 생산성	0.063	0.150	0.375	0.750	
		노력 예측값(과거 노력 분포 기반)	1,530	1,932	2,577	2,013	
		유입 결함 제거 예상 노력	0	280	555	79	
		성과모델 예측 개발 노력	1,530	2,213	3,132	2,092	
	실제 투입 노력	1,125	1,890	3,195	2,280		
	P4	규모	7,001	7,001	8,010	8,567	
		결함	예측 결함수(성과베이스라인 기반)	203	301	601	488
			유입 결함 수	0	131	237	188
			예측결함 총계	203	432	838	676
제거 결함 수			72	195	650	470	
개발 노력		예상 개발 노력	9,254				
		결함 제거 생산성	0.063	0.150	0.375	0.750	
		노력 예측값(과거 노력 분포 기반)	1,758	2,221	2,961	2,313	
		유입 결함 제거 예상 노력	0	432	1,956	141	
		성과모델 예측 개발 노력	1,758	2,653	4,917	2,454	
실제 투입 노력		1,023	2,340	5,320	2,543		
P5		규모	8,636	8,636	9,012	9,780	
		결함	예측 결함수(성과베이스라인 기반)	250	371	676	557
			유입 결함 수	0	130	262	199
			예측결함 총계	250	502	938	756
	제거 결함 수		120	240	739	680	
	개발 노력	예상 개발 노력	9,834				
		결함 제거 생산성	0.063	0.150	0.375	0.750	
		노력 예측값(과거 노력 분포 기반)	1,868	2,360	3,147	2,459	
		유입 결함 제거 예상 노력	0	430	2,160	149	
		성과모델 예측 개발 노력	1,868	2,791	5,307	2,608	

<표 2>는 결합 예측 성과모델을 적용한 결과를 나타낸다. 이전 단계의 잠재 결합 유입이 많은 단계에서는 과거 노력 분포 예측값에 비해 성과모델 예측값이 실제 개발노력 투입결과와 유사한 결과가 예측되었음을 알 수 있었다.

<표 3> 개발노력 분배 성과모델 적중율(%)

구분(MD)		분석	설계	구현	시험
P1	과거노력분포 예측값	785	992	1,323	1,034
	성과모델 예측값	785	1,153	2,105	1,090
	실제 투입 개발노력	<u>785</u>	<u>800</u>	<u>2,368</u>	<u>1,112</u>
	성과모델 적중율(%)	-	69%	112%	102%
P2	과거노력분포 예측값	828	1,045	1,394	1,089
	성과모델 예측값	828	1,100	2,121	1,146
	실제 투입 개발노력	<u>729</u>	<u>950</u>	<u>2,340</u>	<u>1,260</u>
	성과모델 적중율(%)	-	86%	110%	110%
P3	과거노력분포 예측값	1,530	1,932	2,577	2,013
	성과모델 예측값	1,530	2,213	3,132	2,092
	실제 투입 개발노력	<u>1,125</u>	<u>1,890</u>	<u>3,195</u>	<u>2,280</u>
	성과모델 적중율(%)	-	85%	102%	109%
P4	과거노력분포 예측값	1,758	2,221	2,961	2,313
	성과모델 예측값	1,758	2,653	4,917	2,454
	실제 투입 개발노력	<u>1,023</u>	<u>2,340</u>	<u>5,320</u>	<u>2,543</u>
	성과모델 적중율(%)	-	88%	108%	104%
P5	과거노력분포 예측값	1,868	2,360	3,147	2,459
	성과모델 예측값	1,868	2,791	5,307	2,608
	실제 투입 개발노력	<u>1,020</u>	<u>2,256</u>	<u>5,412</u>	<u>2,825</u>
	성과모델 적중율(%)	-	81%	102%	108%

<표 4> 단계별 성과모델 평균 적중율

분석	설계	구현	시험
-	81.8%	107%	106.6%

<표 3>의 결합 예측을 적용한 개발노력 분배 성과모델의 적중률과 <표 4>의 단계별 성과모델 평균 적중률을 분석한 결과, 설계 단계에 비해 구현단계 이후부터 실제 투입된 개발노력과 근접한 예측 결과를 보여 주고 있다. 이는 구현단계 이후부터 단위시험, 통합시험과 같은 결합발견 활동의 비중이 높아지고, 분석, 설계단계에 조기에 제거되지 못한 결합 증가에 따라 예상보다 초과된 결합제거에 소요되는 노력이 증가됨을 알 수 있었다. <표 2>, <표 3>, <표 4>의 사례 적용 결과에서 볼 수 있듯이 단계별 예상 결합 수와 이전단계에서 제거되지 못한 잠재결합이 단계별 개발노력 분배에 중요한 영향을 미치는 변수로 작용되었음을 알 수 있었다.

5. 결론

이번 연구를 통해서 결합이 개발노력 분배의 중요한 요소로 작용될 수 있고, 단계별로 제거되지 못한 결합은 프로젝트 후반부로 갈수록 더 높은 결합 제거 비용이 소요됨에 따라 조직의 과거 노력 분포 기반의 개발노력을 예측한 값보다 개발노력이 추가 투입되었다는 것을 알 수 있었다. 또한 프로젝트의 성공을 위해서는 정확한 개발노력의 추정 및 단계별 개발노력 분배와 더불어 결합의 조기 제거 및 결합의 유입을 차단하는 활동을 통해 비용의 증가를 예방하는 노력이 무엇보다 필요하다는 것을 알 수 있었다.

이번 연구에서 제시한 결합 예측 성과모델은 CMMI 성숙도 3 레벨 이상의 프로세스들이 안정적(Stability)으로 유지되고, CMMI 성숙도 4 레벨 프로세스인 조직 프로세스 성과 프로세스가 적용되는 조직에 적용할 수 있다는 제한적 요소가 있다. 또한 소프트웨어 생명주기 전체에 대한 개발노력 분배 기준보다는 결합제거 활동이 증가되는 구현 이후 단계의 개발노력 분배 기준으로 활용할 수 있다는 한계점이 있다.

향후 관련 연구는 단계별 개발노력 분배에 영향을 줄 수 있는 추가적 요소를 식별하여 보다 정확한 개발노력 배분 방법을 제시하고자 한다.

참고문헌

- [1] The Standish Group International Inc, "Lastest Standish Group CHAOS Report Show Project Success Rates Have Improved by 50%", <http://www.standishgroup.com/press/>, 2003.
- [2] L,H Putnam, *A General Empirical Solution to the Macro Software Sizing and Estimating Problem*, IEEE Trans, on Software Eng, Vol:SE-4, Issue: 4, pp 345-361, July 1978.
- [3] S.Yamada, J.Hishitani, and S.Osaki, *Software-Reliability Growth with a Weibull Test-Effort: A Model & Application*, IEEE Trans, on Reliability, Vol42, No1, pp100-106, 1993.
- [4] CMMI Product Team, "CMMI® for Development Version 1.2", Aug 2006
- [5] 이민재, 박남직, "CMMI 의 이해", (주)피어슨 에듀케이션 코리아, 2006.