

단계기반 결점 프로파일을 이용한 소프트웨어 품질 평가

이 상 운*

요 약

전형적인 소프트웨어 생명주기 모델은 결점이 추가되거나 제거되는 단계들의 일련의 순서로 구성되어 있다. 우리가 원하는 수준의 품질을 달성하기 위해서는 소프트웨어 개발 전 과정에서 결점 제거를 수행하여야 한다. 잘 알려진 단계기반 결점 프로파일은 Gaffney 모델이 있다. 이 모델은 결점 제거 프로파일이 Rayleigh 분포를 따르며 단계 인덱스 번호를 모수로 하고 있다. 실제 개발되는 소프트웨어에 Gaffney 모델을 적용시 제거된 결점이 최대값이 되는 점을 위치 모수가 표현하지 못하는 문제가 있다. 그러므로 Gaffney 모델은 실제 결점 프로파일을 표현하지 못한다. 본 논문은 2개의 다른 모델을 제시한다. 하나는 수정된 Gaffney 모델로 위치 모수를 교체하기 위해 Putnam의 SLIM 모델의 모수를 도입하였다. 다른 하나는 누적 결점 프로파일이 S자 형태를 보여 성장곡선 모델을 제시하였다. 제안된 모델은 5개의 다른 소프트웨어 프로젝트로부터 얻어진 결점 프로파일 분석에 의해 검증하였다. 실험 결과 제안된 모델이 Gaffney 모델 보다 좋은 결과를 얻었다.

키워드 : 결점, 결점 프로파일, 개발 단계, 소프트웨어 품질, 소프트웨어 위기

An Evaluation of Software Quality Using Phase-based Defect Profile

Sang-Un Lee*

ABSTRACT

A typical software development life cycle consists of a series of phases, each of which has some ability to insert and detect defects. To achieve desired quality, we should progress the defect removal with the all phases of the software development. The well-known model of phase-based defect profile is Gaffney model. This model assumes that the defect removal profile follows Rayleigh curve and uses the parameters as the phase index number. However, these is a problem that the location parameter cannot present the peak point of removed defects when you apply Gaffney model to the actual situation. Therefore, Gaffney model failed to represent the actual defect profile. This paper suggests two different models: One is modified Gaffney model that introduce the parameter of Putnam's SLIM model to replace of the location parameter, the other is the growth function model because the cumulative defect profile shows S-shaped. Suggested model is analyzed and verified by the defect profile sets that are obtained from 5 different software projects. We could see from the experiment, the suggested model performed better result than Gaffney model.

Key Words : Defect, Defect Profile, Development Phase, Software Quality, Software Crisis

1. 서 론

소프트웨어 개발의 성공 여부는 고객이 요구하는 품질 수준을 만족하는 제품을 주어진 일정과 예산범위 내에서 납품할 수 있는나로 결정된다[1]. 즉, 품질(quality), 일정 (schedule)과 비용(cost)이 소프트웨어 성공여부를 결정하는 주요 요소가 된다. 현재의 소프트웨어 개발 환경은 보다 좋게(better), 보다 빠르게(faster), 보다 저렴한 비용 (cheaper)으로 개발하려는 경향을 띠고 있다[2,3]. 이와 같은 개발 환경에서 소프트웨어 품질은 개발비용, 납품일정과 더불어 사용자 만족도

에 영향을 미치기 때문에 소프트웨어 품질 관리의 중요성이 점차 증대되고 있다[4].

일반적인 관리활동과 동일하게 품질도 계획되고 계획대비 진척상황을 파악하고 필요시 적절한 시정조치활동이 수행된다. 실제 개발에 투입된 노력(또는 비용)이 계획된 노력과 비교되는 것과 같이 실제 발견된 결점(defect)들도 측정될 수 있고 계획된 결점과 비교될 수 있다. 계획과 실제 결점에 대한 비교는 프로젝트 성능을 최적화하기 위한 시정조치 활동을 유발시킨다.

소프트웨어 개발은 단계(또는 프로세스)들의 연속으로 구성되어 있으며, 각 단계에서 결점이 삽입되거나 발견될 수 있다. 이와 같이 소프트웨어 개발의 각 단계에서 삽입되고 발견되는 결점이 어떠한 분포를 따르는지를 나타내는 결점 프로파일 (defect profile)은 품질에 소요되는 비용을 계획하

* 정 회 원 : 강릉대학교 컴퓨터정보공학부 조교수
논문접수 : 2007년 7월 23일
수정일 : 1차 2007년 12월 28일, 2차 2008년 2월 25일
집사완료 : 2008년 2월 27일

고 관리하는데 중요한 영향을 미친다[5]. 그러나 소프트웨어 개발 과정에서 삽입된 결점들이 모두 발견될 때까지는 각 단계에서 얼마나 많은 결점이 삽입되었는지 정확히 파악할 수가 없으며 삽입된 결점 수에 대한 적절한 확신은 단지 시스템이 납품된 이후에만 가능하다[6]. 따라서 결점 수 프로파일에 기반하여 품질을 계획하고 관리하는 대부분의 방법들은 발견되는 결점 수에 초점을 맞추고 있다[5].

발견되는 결점이 어떤 분포를 따르는지에 대한 결점 프로파일 관련연구는 경험 모델 (empirical model)과 분석 모델 (analytical model)이 있다[5]. 분석모델은 시간 기반 (time-based)과 단계 기반 (phase-based) 모델로 구분된다. 시간 기반 모델은 발견되는 결점을 달력시간의 함수로 표현하는 방법으로 Stark[7]과 Putnam과 Myers[8] 모델이 있으며 모두 Rayleigh 분포를 따르는 모델이다. 단계 기반 모델은 소프트웨어의 개발단계 인덱스 값에 대한 함수로 표현하는 방법으로 Gaffney 모델[9]이 있다. Gaffney 모델[9]은 소프트웨어 개발 각 단계에서 투입되는 인력의 프로파일에 비례하는 Rayleigh 분포 모델을 제안하였다. 이 모델은 특히 시간에 따른 결점 프로파일을 제시하는 시간-기반을 채택하지 않고 소프트웨어의 개발 단계에 대한 인덱스 값에 기반한 단계-기반 모델이라는 특징을 갖고 있다. 그러나 이 모델을 적용하였을 경우 실제 소프트웨어 개발 각 단계에서 발견되는 결점 수 (또는 결점율)를 적절히 표현하지 못하는 단점을 보이고 있다.

본 논문은 발견되는 결점 수 (또는 결점율)의 분포를 추정하기 위한 기존의 Gaffney 모델[9]의 문제점을 보완한 수정된 Gaffney 모델 (Modified Gaffney Model)과 더불어 성장곡선 모델도 제안한다. 2장에서는 결점 프로파일을 표현하는 모델의 문제점을 살펴본다. 3장에서는 수정된 Gaffney 모델과 성장곡선 모델을 제시하고 4장에서는 제안된 모델을 실제 수집된 다양한 형태의 결점 프로파일에 적용하여 모델의 적합성을 검증한다.

2. 결점 프로파일 관련연구와 문제점

결점 프로파일은 경험 모델과 분석 모델로 구분될 수 있다. 한 개발조직이 계획한 프로젝트의 결점 프로파일을 생명주기의 각 단계에서의 발생하는 결점 삽입율과 검출율에 대한 실제 데이터를 수집하여 이에 적합시키는 방법이 경험모델이다. 경험 모델은 다음 가정에 기반하여 모델을 개발한다[5].

- [가정 1] 소프트웨어 규모는 쉽게 정량화 할 수 있으며, 규모는 결점 수와 밀접하게 관련이 있다.
- [가정 2] 소프트웨어 프로세스가 안정되게 유지되는 한 결점 삽입율과 검출율은 비교적 일정하게 유지된다.
- [경험모델]
- [1] 수집된 결점 데이터로부터 프로젝트의 규모를 파악한다.

- [2] 각 단계에서 발견된 결점 수의 초기 프로파일은 Rayleigh 분포형태를 따른다.
- [3] 기존에 개발된 프로젝트와 새로 개발될 프로젝트에 대한 규모 차이를 계산하여 비율에 따라 초기 결점 프로파일을 결정한다.
- [4] 계획된 프로젝트의 특성을 반영하기 위해 결정된 초기 결점 프로파일은 다시 조절된다.

반면에 분석모델은 결점 프로파일을 분석적으로 생성하는 방법으로 다음 가정에 기반을 두고 있다.

- [가정 1] 발견되는 결점 수는 개발에 투입되는 인력에 대략적으로 비례하는 Rayleigh 분포를 따른다. 즉, 보다 많은 노력이 투입되면 보다 많은 실수 (착오)가 유발되어 결점들이 삽입된다.
- [가정 2] 투입되는 인력 프로파일은 단일한 형태를 가진다.
- [가정 3] 소프트웨어 개발 단계들은 유사한 기간이 소요된다.

시간에 기반하여 발견되는 결점의 분포를 표현하는 모델로는 Stark[7]과 Putnam과 Myers[8]이 있다. Stark[7]은 임의의 시간(월)에 발견된 결점 수 D_m 는 식 (1)의 Rayleigh 분포를 따른다고 제안하였다.

$$D_m = \frac{K}{t_d^2} t e^{-\frac{t}{2t_d}} \quad (1)$$

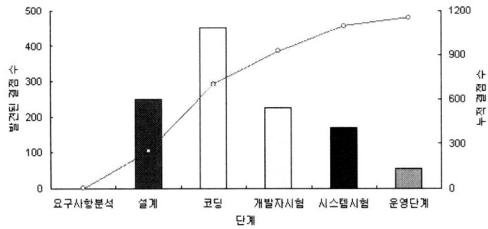
여기서 K 는 예상되는 총 결점 수이며, t_d 는 상설설계 검토 종료 시점이다. Putnam과 Myers[8]는 매월 발견되는 결점 수는 식 (2)의 Rayleigh 분포를 따른다고 제안하였으며, t_d 는 개발 종료 시점이다.

$$D_m = \frac{6K}{t_d^2} t e^{-\frac{3t}{t_d}} \quad (2)$$

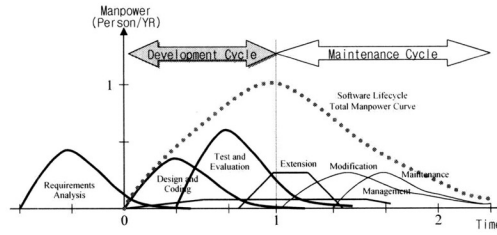
지금까지의 모델들은 모두 시간 (월)에 기반하여 발견되는 결점 수를 표현하는 모델인데 반해 소프트웨어 개발단계의 인덱스 값에 기반한 Gaffney[9] 모델이 있다. Gaffney[9] 모델은 시간 t 까지 발견된 총 결점 수는 식 (3)의 Rayleigh 분포로 표현된다.

$$V_t = D(1 - e^{-bt^2}) \quad (3)$$

여기서 D 는 소프트웨어 라이프사이클 동안 각 개발단계에서 삽입된 총 결점 수, t 는 모델을 프로젝트 단계들에 적용하기 위해 각 단계를 순서대로 숫자로 표현한 값으로 단계 인덱스 값이라고도 한다. 예로, 요구사항 분석, 설계, 구현, 단위시험, 통합시험과 수락시험의 6 단계를 수행하는 프로젝트의 경우, $1 \leq t \leq 6$ 이 된다. 또한 b 는 결점이 가장 많이 발견된 단계의 인덱스 값으로 만약 3번째 단계에서 발견된 결점이 가장 많을 경우 $b=3$ 이 된다. 이 모델은 시간에



(그림 1) 단계별 발견된 결점 수



(그림 2) 소프트웨어 인력 분포

<표 1> Gaffney 모델[9]을 이용한 결점 프로파일 추정 성능

단계	단계 인덱스 값	발견된 결점 수	누적 결점 수 (V_i)	Gaffney 모델 추정 누적 결점 수 (\hat{V}_i)	MMRE
요구사항분석	1	0	0	1093	-
설계	2	250	250	1150	359.99%
코딩	3	450	700	1150	64.29%
개발자시험	4	225	925	1150	24.32%
시스템시험	5	169	1094	1150	5.12%
운영	6	56	1150	1150	0.00%
평균					90.75%

기반을 두지 않고 단계에 기반을 두고 있어 단계기반 결점 프로파일 (phase-based defect profile)이라고 한다.

시간 t 까지 발견된 총 결점 수 V_i 로부터 라이프사이클의 특정한 하나의 단계에서 발견된 결점 수는 $\Delta V_i = D(e^{-b(t-1)^2} - e^{-bt^2})$ 로 구할 수 있다. 식 (1)의 Gaffney 모델[9]을 (그림 1)의 데이터에 적용하여 보자.

이 데이터는 Card[5]로부터 획득된 데이터로 총 삽입된 결점 수는 1150개이며, t 는 요구사항 분석부터 운영까지 $1 \leq t \leq 6$ 의 범위를 가지고 있다. 3번째로 수행된 코딩 단계에서 가장 많은 결점이 발견되었으므로 b 는 3이 된다. 이들 모수 값을 적용하면 Gaffney 모델[9]로 추정된 누적 결점 수는 $\hat{V}_i = 1150(1 - e^{-3t^2})$, $t=1, 2, \dots, 6$ 가 되며, \hat{V}_i 와 모델의 성능을 판단하기 위한 평균 상대오차 (MMRE, mean magnitude of relative error)[10,11]는 <표 1>과 같이 구해진다. 여기서, MMRE는 다음과 같이 구한다. 상대오차 (RE, relative error) = (추정치 - 실측치)/실측치, MRE (magnitude of RE) = |RE|, MMRE (mean MRE) = $100/n \sum_{i=1}^n MRE$. MMRE가 작은 값이면 평균적으로 좋은 모델임을 알 수 있다.

<표 1>에서 알 수 있듯이 Gaffney 모델[9]로 추정된 누적 결점 수 \hat{V}_i 는 실제 값인 V_i 를 거의 표현하지 못하는 단점을 갖고 있다.

3. 단계기반 결점 프로파일 대안 모델

3.1 수정된 (Modified) Gaffney 모델

각 개발 단계에서 발견되는 결점 프로파일은 개발에 투입되는 인력 프로파일에 비례한다는 가정하에 Rayleigh 분포

를 따르는 Gaffney 모델[9]이 제안되었다. Gaffney 모델의 문제점을 해결하기 위해 개발인력 분포를 표현하는 대표적인 Putnam[12]의 SLIM (software life-cycle model) 모델을 살펴보자.

Putnam[12]은 전형적인 소프트웨어 생명주기 전체(설계 단계부터 유지보수단계까지) 소요되는 총 인력과 각 단계에 투입되는 인력도 (그림 2)와 같이 Rayleigh 분포를 따름을 관찰하였다. 임의의 시간 t 시점까지 프로젝트에 투입된 누적 인력 y 를 식 (4)로, 임의의 시간 t 시점에서 프로젝트에 투입되는 인력 \dot{y} 를 식 (5)로 표현된다.

$$y = E(1 - e^{-at^2}) \text{ MY} \quad (4)$$

$$\dot{y} = 2Eate^{-at^2} \text{ MY/YR} \quad (5)$$

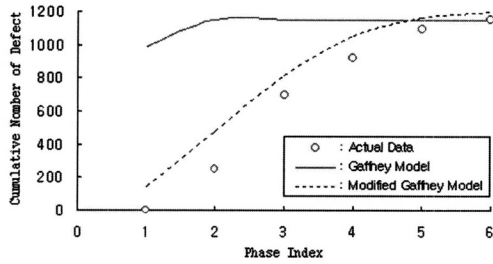
여기서 E 는 Rayleigh 곡선 아래에 있는 영역으로 년 인원으로 표시되며, 소프트웨어 생명주기 전반에 걸쳐 투입된 총 인력이다. $a = 1/2t_d^2$ 이며, t_d 는 \dot{y} 이 최대가 되는 시점으로 경험적으로 볼 때, 개발이 종료되고 납품이 되는 시점이 된다.

식 (3) 모델이 실제 데이터를 표현할 수 있도록 하는 가능한 방법은 Putnam[12]이 제안한 인력 프로파일인 식 (4)의 모수 a 의 값을 고찰해볼 필요가 있다. $a = 1/2t_d^2$ 에서 t_d 는 식 (3)의 b 값과 동일하다. t_d 의 값을 b 로 치환하여 a 를 c 라 하면, $c = 1/2b^2$ 이 된다. 이 값을 식 (3)의 Gaffney 모델[9]에 적용하면 식 (6)를 얻을 수 있다. 이 모델을 수정된 Gaffney 모델 (modified Gaffney model) 이라 부른다.

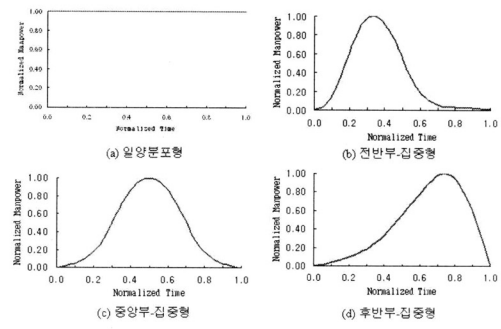
$$V_i = D(1 - e^{-at^2}), c = \frac{1}{2b^2} \quad (6)$$

(그림 1)의 데이터에 대해 식 (6)으로 추정된 누적 결점 수 $\hat{V}_i = 1150(1 - e^{-\frac{1}{2 \cdot 3^2}t^2})$ 가 된다. 이와 같은 값을 적용하여 수정된 Gaffney 모델의 누적 결점 수 추정과 MMRE를 구한 결과는 <표 2>에 제시하였다. Gaffney와 수정된 Gaffney 모델의 성능을 비교하기 위해 <표 1>과 <표 2> 데이터의 누적 결점 수 추정 결과는 (그림 3)에 제시하였다.

(그림 3)에서 알 수 있듯이 수정된 Gaffney 모델은 실제 데이터를 Gaffney 모델[9]보다 잘 표현함을 알 수 있다. 또한, <표 2>의 Gaffney 모델 성능과 비교시 MMRE를 약 64% 감소시킬 수 있다.



(그림 3) 누적 결점 수 비교



(그림 4) 가능한 개발인력 프로파일 유형

<표 2> Modified Gaffney 모델을 적용한 결점 프로파일 추정 성능

단계	단계 인덱스 값	발견된 결점 수	누적 결점 수 (V_i)	Modified Gaffney 모델 적용 추정 누적 결점 수 (\hat{V}_i)	MMRE
요구사항분석	1	0	0	143	-
설계	2	250	250	478	91.23%
코딩	3	450	700	821	17.22%
개발자시험	4	225	925	1051	13.57%
시스템시험	5	169	1094	1162	6.18%
운영	6	56	1150	1202	4.48%
평균					26.54%

<표 3> 소프트웨어 개발 단계별 소요 기간 예

구분	요구사항 분석	개략 설계	상세 설계	구현	시스템시험	수락 시험
노력	6%	8%	16%	40%	20%	10%
기간	12%	8%	15%	30%	20%	15%

3.2 성장곡선 모델

결점 프로파일에 대한 분석 모델이 근거로 하는 가정에는 다음과 같은 문제점을 갖고 있다.

[문제점 1] 개발인력 프로파일이 Rayleigh 분포를 따른다는 가정에 비하여 발견되는 결점 프로파일도 Rayleigh 분포를 따른다는 가정에 대해, 개발 각 단계에서 투입되는 인력이 증가할수록 결점이 삽입될 확률은 인간이 갖고 있는 본질적인 특성인 실수로 인해 비례하여 증가한다. 그러나 결점 프로파일은 삽입되는 결점이 아닌 발견되는 결점에 관한 사항으로 결점을 찾는데 투입되는 총 인력의 양을 해당 단계의 결점당 투입 인력의 양으로 나누어 계산할 수 있다.

[문제점 2] 개발에 투입되는 인력 프로파일은 단일한 형태를 갖는다는 가정 2에 대해, 이상훈[13]은 개발에 투입되는 인력 분포는 (그림 4)와 같이 다양한 형태를 따를 수 있음을 제안하였다.

[문제점 3] 소프트웨어 개발 라이프사이클의 단계들은 유사한 기간이 소요된다는 가정 3에 대해, 소프트웨어 개발 단계들 각각에 투입되는 기간은 <표 3>에서 알 수 있듯이 유사한 기간이 소요되지 않는다[14].

이와 같은 문제점으로 인해 본 논문에서는 다음과 같은 가정에 근거하여 모델을 제안한다.

[가정 1] 각 단계에서 발견되는 결점 수는 각 단계에 결점을 발견 하는데 투입되는 노력 (시간)을 결점

1개당 발견에 투입되는 노력의 양으로 나눈 값으로 결정된다.

[가정 2] 소프트웨어 결점을 찾아내는데 소요되는 인력 프로파일은 프로젝트마다 다양한 형태를 따를 수 있다.

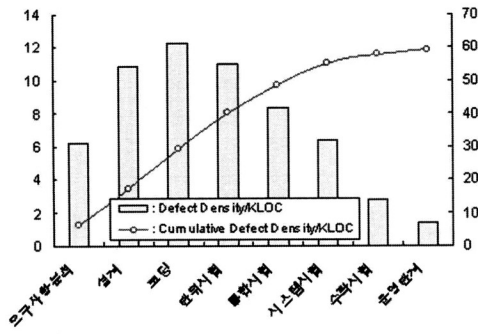
[가정 3] 소프트웨어 개발 단계들 각각은 유사한 기간이 소요되지 않는다. 그러나 선택되는 단계들은 순서적으로 인덱스 값으로 치환된다.

위 가정에 근거하여 다양한 결점 발견 프로파일에 적합한 모델을 제시하기 위해서는 발견되는 결점 프로파일은 개발에 투입되는 인력 분포에 비례한다는 가정을 하지 않은 상태에서 단지 단계별로 발견된 결점 프로파일 데이터만을 분석하여 공통점을 찾는 과정이 필요하다. 따라서 실제 데이터를 분석하여 보자. (그림 5)는 다양한 결점 프로파일에 대한 실제 데이터를 보여주고 있다.

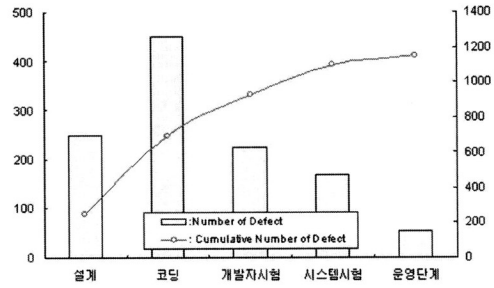
누적 결점 수나 결점율에 대해 (그림 5)의 5개 데이터 모두 개략적으로 S자 형태를 따른다고 할 수 있다. 따라서 S자 형태를 취하는 통계적 모델을 적용시키면 다양한 형태를 취하는 결점 프로파일을 보다 적절히 표현할 수 있을 것이다. 본 논문에서는 S자 형태의 곡선을 표현하는 대표적인 성장곡선 (growth curve) 중에서 식 (7)의 Gompertz 곡선 [19]을 적용한다. 그러나 다른 성장곡선을 적용시킬 수도 있을 것이다.

$$V_i = D e^{-ae^{-bt}} \tag{7}$$

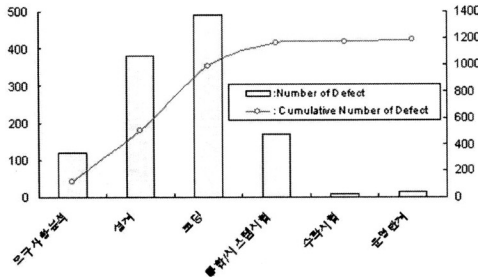
여기서 D 는 V_i 의 상한 점근선으로 소프트웨어에 삽입된 총 결점의 수 또는 결점 발견율의 합계, t 는 단계 인덱스 값, a 는 S자 곡선의 위치를 결정하는 하한 점근선, b 는 곡선



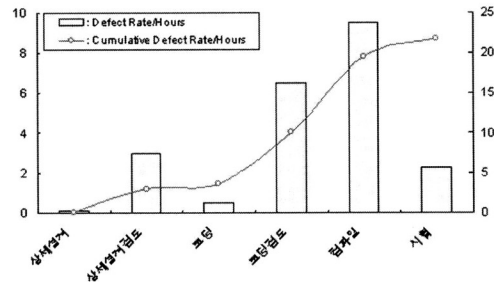
(a) Data 1: 결점을 프로파일[15]



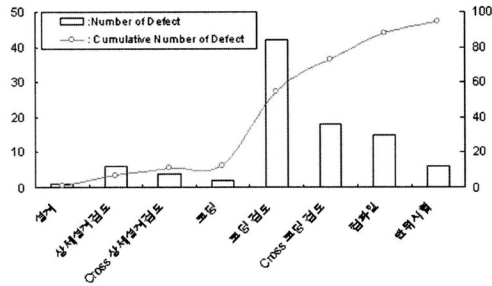
(b) Data 2: 결점 수 프로파일[5]



(c) Data 3: 결점 수 프로파일[16]



(d) Data 4: 결점 수 프로파일[17]



(e) Data 5: 결점 수 프로파일[18]

(그림 5) 개발 단계별 결점 프로파일

의 모양을 결정하는 값이 된다. 제안된 모델은 식 (8)과 같이 두 번의 자연로그 변수변환을 거치면 선형방정식이 얻어지며 이로부터 상용 소프트웨어 도구를 이용하여 선형회귀 분석을 수행하면 모수들의 값을 쉽게 구할 수 있다.

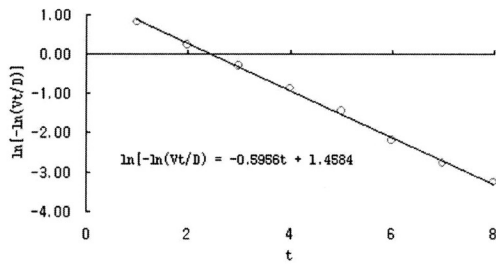
$$\ln[-\ln(\frac{V_i}{D})] = \ln a - bt \tag{8}$$

4. 모델 평가

(그림 5)의 5개 데이터에 대해 Gaffney 모델, 수정된 Gaffney 모델과 Gompertz 곡선 모델을 적용하여 성능을 평가해 본다. 수정된 Gaffney 모델의 모수 D 는 최소 자승법으

로 구하였다. Gompertz 곡선 모델의 모수 D 는 $\ln[-\ln(\frac{V_i}{D})]$ 와 t 사이의 상관계수가 최대가 되는 방법을 적용하였다. 이 방법이 식 (6)을 이용하는 가장 합리적인 추정법이라 할 수 있다. 이 값을 식 (6)에 대입하여 Gompertz 곡선 모델의 모수 a 와 b 값을 얻을 수 있다. 예로 Data 1에 대한 식 (8)의 선형회귀분석을 수행한 결과는 (그림 6)과 같다.

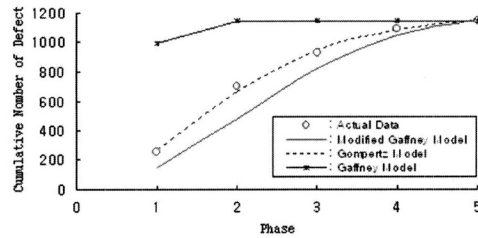
따라서 모수 $b = -0.5956$, 모수 a 는 $\ln(a) = 1.4584$ 로부터 $\hat{a} = e^{\ln a} = e^{1.4584} = 4.2991$ 을 얻을 수 있다. 5개 데이터에 대한 모수 추정 결과는 <표 4>와 같이 얻었다. 이와 같이 추정된 모수들을 각 모델에 적용한 결과 얻은 모델들은 <표 5>와 같다. <표 5> 모델들을 실측 데이터와 비교한 결과는 (그림 7)에, 모델의 성능은 <표 6>에 제시하고 있다.



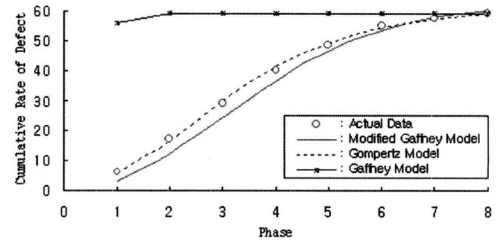
(그림 6) 선형회귀분석 (Data 1)

<표 4> Gompertz 곡선 모델의 모수 추정

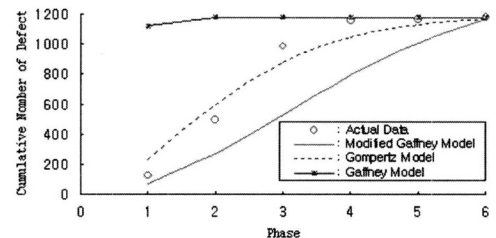
구분	\hat{D}	선형회귀분석 식	\hat{a}	\hat{b}
Data 1	61.5	$\ln[-\ln(\frac{V_t}{D})] = -0.5956t + 1.4584$	4.2991	-0.5956
Data 2	1,192	$\ln[-\ln(\frac{V_t}{D})] = -0.9373t + 1.3438$	3.8336	-0.9373
Data 3	1,200	$\ln[-\ln(\frac{V_t}{D})] = -1.0740t + 1.7268$	3.8236	-0.8438
Data 4	32	$\ln[-\ln(\frac{V_t}{D})] = -0.5391t + 2.1985$	9.0115	-0.5391
Data 5	110	$\ln[-\ln(\frac{V_t}{D})] = -0.5078t + 2.2282$	9.2831	-0.5078



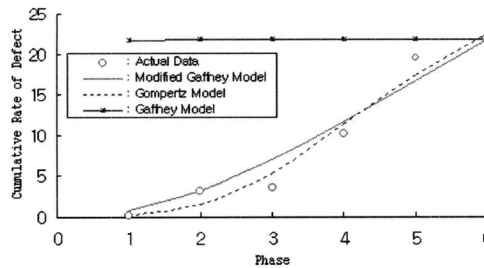
(a) Data 1



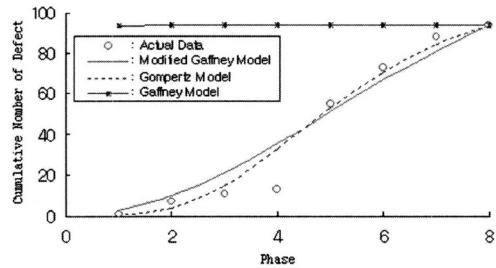
(b) Data 2



(c) Data 3



(d) Data 4



(e) Data 5

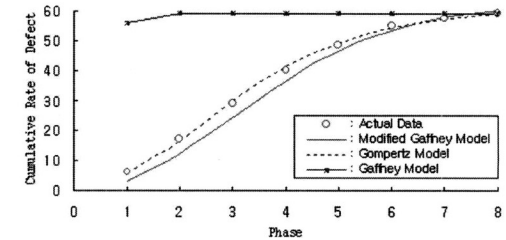
(그림 7) 실측 값과 추정 값 비교

결론적으로 Gaffney 모델에 비해 수정된 Gaffney 모델이, 수정된 Gaffney 모델에 비해 Gompertz 곡선 모델이 보다 좋은 성능을 보여주고 있다.

실제 적용하는 측면에 있어서 기존에 획득한 결점 프로파일 데이터들이 충분히 많을 경우, 이 데이터들로부터 모수 추정을 통해 Gompertz 곡선 모델을 적용할 수 있을 수 있

<표 5> 모수 추정 결과 얻은 모델

구분	모델		
	Gaffney 모델	수정된 Gaffney 모델	성장곡선 모델
Data 1	$59.2(1 - e^{-3t^2})$	$62(1 - e^{-0.0556t^2})$	$61.5e^{-4.2991e^{-0.5956t}}$
Data 2	$1150(1 - e^{-2t^2})$	$1215(1 - e^{-0.125t^2})$	$1192e^{-3.8336e^{-0.9373t}}$
Data 3	$1183(1 - e^{-3t^2})$	$1350(1 - e^{-0.0556t^2})$	$1200e^{-3.8236e^{-0.8438t}}$
Data 4	$21.8(1 - e^{-5t^2})$	$42.5(1 - e^{-0.02t^2})$	$32e^{-9.0115e^{-0.5391t}}$
Data 5	$94(1 - e^{-5t^2})$	$130(1 - e^{-0.02t^2})$	$110e^{-9.2831e^{-0.5078t}}$



다. 그러나 획득된 데이터가 적을 경우에는 수정된 Gaffney 모델을 적용하는 것이 보다 타당할 것이다. 왜냐하면, (그림 5)의 5개 데이터를 살펴보면 결점이 가장 많이 발생하는 단계는 대부분이 코딩단계이므로 모수 c의 값을 추정하기가 쉽지 때문이다.

〈표 6〉 모델 성능

구분	MMRE		
	Gaffney 모델	수정된 Gaffney 모델	성장곡선 모델
Data 1	152.49%	13.65%	2.41%
Data 2	78.28%	18.17%	2.90%
Data 3	166.02%	29.59%	22.52%
Data 4	250.12%	26.60%	25.40%
Data 5	1,475.45%	61.53%	37.28%

5. 결론 및 향후 연구과제

소프트웨어 개발의 각 단계에서 발견되는 결점 프로파일은 품질관리에 소요되는 비용을 계획하고 관리하는데 중요한 영향을 미친다. 이에 대한 대표적인 분석 모델로 단계기반의 Gaffney 모델[9]이 있다. 그러나 이 모델을 실제 데이터에 적용하였을 경우 개발의 각 단계에서 발견되는 결점을 적절히 표현하지 못하는 단점을 보이고 있다.

본 논문은 먼저, Gaffney 모델의 단점을 보완하기 위해 결점이 가장 많이 발생하는 단계에 대한 인덱스 값에 대한 모수를 Putnam[12]이 소프트웨어 개발에 투입되는 인력 프로파일 유도 과정에서 제시한 모수의 값으로 치환하여 수정된 Gaffney 모델을 제시하였다. 또한, 결점 프로파일의 누적 곡선이 S자 형태를 따른다는 점에 착안하여 성장곡선 모델을 제시하였다. 실제 발견된 다양한 결점 프로파일 데이터를 이용하여 제안된 모델의 적합성을 검증하였다. 실험 결과 Gaffney 모델[9]에 비해 수정된 Gaffney 모델이, 수정된 Gaffney 모델에 비해 성장곡선 모델이 보다 좋은 성능을 나타내었다.

본 논문에서 제안된 수정된 Gaffney 모델과 성장곡선 모델을 성장곡선 모델을 선택하는 기준은 기존에 결점 프로파일 이력 데이터를 얼마나 많이 갖고 있는냐에 따라 결정될 수 있다. 즉, 결점 프로파일 이력 데이터가 적을 경우에는 수정된 Gaffney 모델을, 데이터가 충분히 많이 있을 경우에는 성장곡선 모델을 적용하는 것이 타당해 보인다.

본 논문은 소프트웨어 개발 방법론을 순차적인 개발 방법인 전형적인 폭포수 프로세스를 따를 경우에 대해서만 연구되었다. 최근 들어 반복 프로세스가 보다 많이 적용되고 있으며, 이 프로세스는 소프트웨어를 반복적이고 점진적인 형태로 반복으로 구분된다. 하나의 반복에서는 폭포수 프로세스의 모든 단계가 동시에 수행되는 형태를 나타내고 있다. 이러한 개발 방법론을 적용하였을 경우 각 반복에서 발견되는 결점 프로파일은 Rayleigh 분포와 다른 형태를 따를 수 있다. 따라서 추후 이 분야에 대한 연구를 수행할 것이다.

참 고 문 헌

[1] A. J. Shenher, "Improving PM: Linking Success Criteria to

Project Type," Project Management Institute, Creating Canadian Advantage through Project Management Symposium, Calgary, 1996.

[2] H. E. McCurdy, "Faster, Better, Cheaper: Low-Cost Innovation in the U.S. Space Program," Johns Hopkins University, 2003.

[3] B. M. Block, "Faster, Better, and Cheaper: The Software Development Life Cycle," Intercom, 2001.

[4] R. M. Green, "Improving Software Quality: Steps to Software Quality," Robelle Solutions Technology Inc., 2004.

[5] D. N. Card, "Managing Software Quality with Defects," The Journal of Defense Software Engineering, 2003.

[6] M. Criscione, J. Ferree, and D. Porter, "Predicting Software Errors and Defects," SMASM Conference, 2001.

[7] G. E. Stark, "Defect Prediction and Impact Schedule Project XYZ," 1998.

[8] L. H. Putnam and W. Myers, "Familiar Metric Management - Reliability," QSM The Intelligence Behind Successful Software Projects, http://www.qsm.com/frmm_03.pdf

[9] J. Gaffney, "Some Models for Software Defect Analysis," Lockheed Martin Software Engineering Workshop, Gaithersburg, MD, 1996.

[10] T. Foss, E. Stensrud, and B. Kitchenham, "A Simulation Study of the Model Evaluation Criterion MMRE," IEEE Trans. on Software Eng., Vol.29, No.11, pp.985-995, 2003.

[11] A. Hodgkinson, "Measuring Effort Estimation Model Goodness," Department of Electronics and Computer Science, University of Southampton, UK, 1995.

[12] L. H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Trans. on Software Eng., Vol.SE-4, No.4, 1978.

[13] 이상운, "소프트웨어 개발 라이프사이클 인력 프로파일," 한국정보처리학회논문지D, 제11-D권 제5호, pp.1123-1132, 2004.

[14] NASA, "Manager's Handbook for Software Development," SEL-84-101, 1990.

[15] A. P. Nikora, "Extending the Use of Measurement," Quality Assurance Office, Jet Propulsion Lab., California Institute of Technology, 2003.

[16] S. Harbaugh, "Crusader Software Quality Assurance Process Improvement," Technical Report, Integrated Software, Inc., 2002.

[17] W. S. Humphrey, "The Software Quality Profile," Software Engineering Institute, Carnegie Mellon University, <http://www.sei.cmu.edu/publications/articles/quality-profile>, 2004.

[18] D. M. Roy, "Synergy of Review Techniques from PSP(SM) to Formal Inspections," Software Technology, Process and People, Inc.,

[19] B. Gompertz, "On The Nature of The Function Expressive of The Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies," Phil. Trans. Roy. Soc. London. Vol.123, pp.513-585, 1832.



이 상 운

e-mail : sulee@kangnung.ac.kr

1983년~1987년 한국항공대학교
항공전자공학과(학사)

1995년~1997년 경상대학교 컴퓨터학과
(석사)

1998년~2001년 경상대학교 컴퓨터학과
(박사)

1992년~2002년 국방품질관리소 항공전자장비 소프트웨어
품질보증 담당

2003년 강원도립대학 컴퓨터응용과 전임강사

2004년~2007년 2월 국립 원주대학 여성교양과 조교수

2007년 3월~현 재 강릉대학교 컴퓨터정보공학부

멀티미디어정보공학 전공 조교수

관심분야: 소프트웨어 프로젝트 관리, 소프트웨어 개발 방법론,
소프트웨어 척도 (소프트웨어 규모, 개발노력, 개발기
간, 팀 규모), 분석과 설계 방법론, 소프트웨어 시험
및 품질보증, 소프트웨어 신뢰성, 신경망, 뉴로-퍼지,
그래프 알고리즘