

분석과 설계 단계에서의 초기 소프트웨어 신뢰도 모델

류연호^o 백두권

고려대학교 컴퓨터과학기술 대학원

yhyru@korea.ac.kr, baik@software.korea.ac.kr

An Early Software Reliability Prediction Model based on Analysis and Design Phase

Yeon-Ho Ryu, Doo-Kwon Baik

Graduate School of Computer Science and Technology, Korea University

요 약

소프트웨어 개발 중 분석과 설계 단계에서 만들어진 산출물을 이용하여 개발될 소프트웨어의 신뢰도를 예측하는 초기 소프트웨어 신뢰도 모델은 소프트웨어의 품질을 보증하는 유용한 도구로써 사용될 수 있다. 기존의 소프트웨어 신뢰도 모델은 개발될 소프트웨어의 품질에 한정하여 신뢰도를 예측함으로써 기존 시스템과 연계된 신뢰도를 예측하는데 한계가 있다는 단점과 개발자의 개발 능력이 개발될 소프트웨어 신뢰도에 미칠 영향을 고려하는데 한계가 있다는 단점이 있었다. 그러므로, 본 논문은 기존 시스템과의 연관성을 고려한 초기 소프트웨어 신뢰도 모델을 기준으로 개발자의 개발 능력을 고려한 종합적인 초기 신뢰도 모델을 제시하였다.

1. 서 론

소프트웨어 신뢰도는 주어진 일정 기간동안 같은 환경 아래에서 소프트웨어가 고장없이 명세서대로 정상적으로 사용될 수 있는 확률로 정의된다[1]. 기존의 소프트웨어 신뢰도에 대한 연구와 제안된 모델들은 시스템을 구현한 후에 시스템을 시험하거나 운영하는 동안 수집된 오류 자료 등을 이용하여 시스템의 신뢰도를 예측하는 것이었다. 그러나 시험 단계나 운영 단계가 아닌 개발 초기 단계에서 시스템의 신뢰도 예측이 가능하다면 시스템 개발 자원을 효율적으로 할당함으로써 최소 비용으로 시스템의 신뢰도를 최적화할 수 있을 것이다. 그리고 설계 단계의 신뢰도 예측을 통하여 시스템의 재설계 필요성을 판단할 수 있다. 즉, 가장 중요한 품질 인자의 하나인 신뢰도를 개발 초기 단계에서 정량화 함으로써 설계 결과의 품질 평가 기준을 마련하는 것이다[2]. 본 논문에서는 기존 시스템과의 연관성을 고려한 초기 소프트웨어 신뢰도 모델을 기준으로 개발자의 개발 능력을 고려한 종합적인 초기 신뢰도 모델을 제시하고 평가하였다.

2. 관련 연구

2.1 소프트웨어 신뢰도 모델 분류.

소프트웨어 신뢰도 모델은 신뢰도 향상 방법론, 신뢰도

측정 및 예측 기법, 고장 인성 기법으로 구분된다.

소프트웨어 신뢰도 향상 방법론은 결함 예방 또는 결함을 위해 에러가 삽입되는 것을 방지하는 방법들과 실행 도중 에러 발생을 예측하고 예외처리 기능들을 소프트웨어 내부에 삽입시키는 회피 방법들이 있다. 신뢰도 예측 기법은 소프트웨어 구조 분석이나 통계적 추정 기법을 이용하여, 측정 기법으로는 Input Domain Model, Error Seeding Model을 이용한 이산적 기법과 고장간 시간형 모델 또는 에러수형 모델을 이용한 연속적 기법으로 분류할 수 있다. 소프트웨어 고장 인성 기법으로는 Voting, N-version Approach 등을 이용한 중복 기법과 Recovery Block으로 분류할 수 있다[3].

2.2 소프트웨어 신뢰도 성장 모델(SRGM).

SRGM을 소프트웨어 개발 단계에 따라 에러는 줄고 신뢰성은 높아 간다는 점을 기본 가정으로 하고 있다. 신뢰도 성장 모델은 시간 예측 모델, 개수 예측 모델, 가용시간 모델로 구분될 수 있다.

시간 예측 모델은 소프트웨어 고장 발생 시간 혹은 소프트웨어 결함 발견을 시간을 기초한 확률, 통계 모델로써 Littlewood-Verrall Model, Moranda Model, Moranda Model 등이 있다.

개수 예측 모델은 발생한 소프트웨어 고장 수 혹은 발견

된 소프트웨어 결함수에 기초한 확률, 통계 모델로서, Shooman Model, Musa Model, Musa-Okumoto Model 등이 있다.
가용 시간 모델은 소프트웨어의 시간적 흐름을 소프트웨어 고장이 발생하지 않은 상태와 소프트웨어 고장이 발생하여 소프트웨어 실행이 중단된 상태에 기초한 확률, 통계 모델로서, Shooman Model, Shooman-Trivedi Model, Okumoto-Goel Model 등이 있다[4].

2.3 초기 소프트웨어 신뢰도 모델.

초기 소프트웨어 신뢰도 모델은 복잡도가 높은 모듈일수록 오류가 발생할 가능성이 높다는 점을 가정으로 하고 있다.

초기 소프트웨어 신뢰도 모델은 개발 초기 단계에서 개발된 소프트웨어의 신뢰도를 예측하는 모델이다.

초기 소프트웨어 신뢰도 모델은 오류 자료와 복잡도 측정치를 기준으로 하는 예측 모델을 만드는 분야와 신뢰도 측정의 매트릭스를 정의하고 타당성을 입증하여 시스템 신뢰도를 예측하는 것으로 구분할 수 있다. 이중 판별 분석법, 분류 트리법, 신경망 기법 등이 저자에 속하며 데이터 바인딩 기법, Zaga의 연구, Agresti의 연구등이 후자에 속한다[5].

2.4 SRGM 과 초기 소프트웨어 신뢰도 모델의 문제점.

SRGM의 단점은 예측되어지는 신뢰도가 테스트 및 구현 단계에서 제공된다는 점과 기존 Legacy 시스템과의 연관성을 고려하지 못하고 개발될 소프트웨어 자체만을 신뢰도 예측 대상으로 하고 있다는 점을 들 수 있다.

초기 소프트웨어 신뢰도 모델은 소프트웨어 개발 초기에서 소프트웨어 신뢰도를 예측할 수 있는 기능을 제공하지만 기존 Legacy 시스템과의 연관성을 고려하지 못하고 있다.

SRGM 과 초기 소프트웨어 신뢰도 모델은 모두 개발자의 개발 능력을 신뢰도 평가 항목으로 제공하지 못하고 있다.

3. 제안모델 (ESRPM : Early Software Reliability Prediction Model)

본 논문의 제안 모델 ESRPM 은 다음과 같은 가정을 전제로 한다.

- 1) 복잡도가 큰 모듈은 오류 발생율이 높다.
- 2) 소프트웨어 개발 비용은 고려하지 않는다.
- 3) 개발 방법론의 적용은 폭포수모델을 이용한다.

3.1 복잡도 매트릭스.

ESRPM의 신뢰도 측정 매트릭스는 시스템 개발과정 중 분석과 설계단계의 산출물로부터 만들어진 다음 신뢰도 매트릭스를 이용한다.

1) 구조 검증도(a)

구조 검증도는 분석, 설계 단계에서의 산출물을 이용하

여 시스템이 갖추어야 하는 요구사항과 요구사항을 만족할 수 있는 설계가 이루어졌는지를 정형화한다.

정형기법[6]을 이용하면, 요구사항 및 설계의 기술과 이의 검증이 시스템의 안정성을 향상시킬 수 있다. 그러므로, <표 2>와 같이 분석, 설계 단계에서 신뢰도 예측 대상이 되는 각각의 모듈들은 정의된 요구사항에 대하여 설계 검증 여부에 따라 정형화된 구조 검증도를 정의하였다.

요구사항설계	현업요구사항정의	
	확인	미확인
확인	5	3
미확인	2	1

<표 2> 구조 검증도

2) 내부 복잡도(β)

내부 복잡도는 개발 또는 수정되는 모듈이 개발될 소프트웨어의 신뢰성에 영향을 미치는 정도이다.

McCall 의 순환수 V(G)는 식(1)과 같다.

$$V(G) = e \square n + 2p \text{ ----- (1)}$$

(e:간선들의 수, n:node들의 수, p:연결성분들의 수)

이를 기본으로 한 단순성은 식(2)과 같다.

$$1 - \frac{Bn}{TMn} \text{ ----- (2) [7]}$$

(TMn: 총 모듈의 수, Bn: 규칙위반 모듈의 수)

본 논문에서 내부 복잡도는 프로그램 유형을 고려하여 식(3)과 같이 정의하였다.

$$\left(1 - \frac{Bn}{TMn}\right) \left(\frac{1}{SP}\right) \left(\frac{1}{MP}\right) + \left(1 - \frac{BPn}{SPn}\right) \left(\frac{1}{MP}\right) + \left(1 - \frac{BNn}{MPn}\right) \text{ ----- (3)}$$

(Bn : 공통모듈의 개발, 수정되는 모듈수,
BPn : Sub-Program 의 개발, 수정되는 모듈수,
BMn : Main-Program 의 개발, 수정되는 모듈수,
TMn : 공통모듈의 총 모듈수,
SPn : Sub-Program 의 총 모듈수,
MPn : Main-Program 의 총 모듈수
SP : 공통모듈을 사용하는 Sub-Program 의 수,
MP : 공통모듈을 사용하는 Main-Program 의 수)

3) 외부 복잡도(γ)

외부 복잡도는 개발, 수정되는 모듈이 연관되어 있는 타 시스템에 영향을 미치는 정도로써 식(4)와 같이 정의하였다.

$$\left(1 - \frac{BPn}{SPn}\right) \sum_{i=1}^m \frac{BNi}{LMi} \text{ ----- (4)}$$

(BPn : Sub-Program 의 개발, 수정되는 모듈수,
SPn : Sub-Program 의 총 모듈수,
LMi : 연관된 Legacy System 의 총 모듈수

BNi :연관된 Legacy System 에 영향을 받는 모듈수)

28.04%, 12.37%임을 나타내고 있다.

4) 개발자의 개발 능력

개발자의 개발 능력은 분석 및 설계 단계의 작업 결과에 대한 신뢰성을 표현하는 기능 개선율과 코딩 및 테스트 단계의 작업 결과에 대한 신뢰성을 표현하는 에러 수정율로 분류할 수 있다.

기능 개선율(δ) = 기능개선건수/유지보수건수.
 에러 수정율(ϵ) = 에러수정건수/유지보수건수.

5) 신뢰도

개발될 각 기능별 신뢰도(A) = $(\beta + \gamma) * a / 5$

개발될 SW 의 신뢰도(B) =

$$\left(\sum_{i=1}^n (\beta_i + \chi_i) * \alpha_i \right) / \left(\sum_{i=1}^n (\beta_i + \chi_i) * 5 \right)$$

개발될 SW 의 기능 개선율(C) = $B * \delta$

개발될 SW 의 에러 수정율(D) = $B * \epsilon$

(α = 구조검증도, β = 내부복잡도, γ = 외부복잡도,
 δ = 기능개선율, ϵ = 에러수정율, i = 개발될 각 기능)

3.2 구현 및 평가

1,270개의 Legacy 시스템의 프로그램에 대한 내부 복잡도와 외부 복잡도를 산출하였다. 그리고 2003년 1월부터 6월까지 완료된 424건의 유지 보수건을 대상으로 개발자의 기능 개선율과 에러 수정율에 대한 평균값을 산출하였다. 두 자료를 기준으로 2003년 완료된 프로젝트 A에 대한 신뢰도를 산출하였다.

프로젝트 A는 현업요구사항이 5가지 항목으로 되어 있고 개발 기간이 2003.6월부터 2003.7.20일까지 소요되었다. 이 프로젝트 A의 분석 및 설계가 완료 되었을 때 개발될 신뢰도는 <표 3>과 같다.

기능	내부복잡도(A)	외부복잡도(B)	구조복잡도(C)	신뢰도((A+B)*C/5)	복잡도합(A+B)*C	이상적신뢰도
1	94.643	0	5	94.643	473.215	473.215
2	93.836	0.2	5	94.036	470.18	470.18
3	50	0	3	30	150	250
4	88.235	0	3	52.941	264.705	441.175
5	79.31	2	4	65.048	325.24	406.55
					1683.34	2041.12

<표 3> 각 기능별 신뢰도.

개발될 SW 신뢰도(B) = $1683.34 / 2041.12 = 0.8247$

개발될 SW 기능개선율(C) = $B * 0.34 = 0.2804$

개발될 SW 에러수정율(D) = $B * 0.15 = 0.1237$

(프로젝트 A 개발자의 기능개선율과 에러수정율은 각각 0.34, 0.15임)

소프트웨어 신뢰도 예측치는 분석 및 설계 단계에서의 산출된 복잡도를 기준으로 구조 검증도를 적용하였을 때, 82.47%의 신뢰도를 나타내었고, 소프트웨어 개발이 완료된 이후 예상되는 기능 개선율과 에러 수정율은 각각

4. 결 론

소프트웨어 분석 및 설계 단계에서 만들어진 산출물을 이용하여 개발될 소프트웨어의 구조 검증도, 내부 복잡도를 산출하였고 기존 Legacy 시스템과의 연관성을 고려한 외부 복잡도를 계산하여 신뢰도를 예측하였다. 그리고 일정 기간 개발자의 개발 능력에 대한 데이터를 이용하여 기능 개선율과 에러 수정율의 평균을 적용하여 소프트웨어 개발 이후에 예상되는 기능 개선율과 에러 수정율을 예측하였다.

분석과 설계 단계에서 만들어진 신뢰도 예측 모델을 이용하여 개발될 소프트웨어의 신뢰도를 사전에 예측하고 기능 개선율과 에러 수정율의 예측을 통하여 분석 설계 및 코딩 테스트 단계의 작업 결과에 대한 품질 관리를 효율적으로 할 수 있다. 향후 연구과제는 신경망을 이용한 내부 복잡도 및 외부 복잡도의 산출을 개발하고 RAD, Prototyping, OOD같은 다양한 개발 방법론에 적용할 수 있도록 초기 소프트웨어 신뢰도 예측 모델을 발전시켜야 한다.

참고 문헌

- [1] 문외식, " 소프트웨어 품질평가를 위한 최적 신뢰도 평가도에 관한 연구", 경남대학교 대학원 박사학위 논문, pp 6, 1997.
- [2] 정혁철, 홍의석, 김갑수, 홍성백, 우치수, "설계복잡도 매트릭스를 이용한 초기 신뢰도 예측 모델", 한국 정보과학회 가을 학술대회, pp 1~3 정보과학회, 1996.
- [3] 문외식, " 소프트웨어 품질평가를 위한 최적 신뢰도 평가도에 관한 연구", 경남대학교 대학원 박사학위 논문, pp 10~12, 1997.
- [4] 권대곤, " 결합 중요도에 따른 소프트웨어 신뢰도 성장 모델 제안 및 평가", 경남대학교 대학원 석사학위 논문, pp 3~10, 1995
- [5] 정혁철, " 설계 복잡도 매트릭스를 이용한 초기 신뢰도 예측 모델 및 구현", 서울대학교 대학원 석사학위 논문, pp 3, 1998.
- [6] 방기석, 유희준, 최진영, " 프로그래밍 언어의 관점에서 본 소프트웨어/하드웨어의 정형 검증 방법 소개", 고려대학교 컴퓨터학과, pp 3, 2002.
- [7] 변정우, " 효율적인 소프트웨어 평가와 모듈 집단화에 관한 연구", 경희대학교 대학원 공학박사학위논문, pp 67~78, 1999.