

A Method for Selecting Software Reliability Growth Models Using Partial Data

Yong Jun Park[†] · Bup-Ki Min^{**} · Hyeon Soo Kim^{***}

ABSTRACT

Software Reliability Growth Models (SRGMs) are useful for determining the software release date or additional testing efforts by using software failure data. It is not appropriate for a SRGM to apply to all software. And besides a large number of SRGMs have already been proposed to estimate software reliability measures. Therefore selection of an optimal SRGM for use in a particular case has been an important issue. The existing methods for selecting a SRGM use the entire collected failure data. However, initial failure data may not affect the future failure occurrence and, in some cases, it results in the distorted result when evaluating the future failure. In this paper, we suggest a method for selecting a SRGM based on the evaluation goodness-of-fit using partial data. Our approach uses partial data except for inordinately unstable failure data in the entire failure data. We will find a portion of data used to select a SRGM through the comparison between the entire failure data and the partial failure data excluded the initial failure data with respect to the predictive ability of future failures. To justify our approach this paper shows that the predictive ability of future failures using partial data is more accurate than using the entire failure data with the real collected failure data.

Keywords : Software Reliability, Software Reliability Growth Model, Model Selection Criteria, T-point

부분 데이터를 이용한 신뢰도 성장 모델 선택 방법

박 용 준[†] · 민 법 기^{**} · 김 현 수^{***}

요 약

소프트웨어 신뢰도 성장 모델은 고장 데이터를 사용해서 소프트웨어 출시일 또는 추가 테스트 노력을 결정하는 데 사용된다. 소프트웨어 신뢰도 성장 모델을 사용할 때 특정 소프트웨어 신뢰도 성장 모델을 모든 소프트웨어에 사용할 수 없는 문제가 있다. 또한 신뢰도를 평가하기 위해 이미 많은 수의 소프트웨어 신뢰도 성장 모델이 제안되었다. 따라서 특정 조건에 맞는 최적의 소프트웨어 신뢰도 성장 모델을 선택하는 것은 중요한 이슈가 되었다. 기존 소프트웨어 신뢰도 성장 모델 선택 방법에서는 수집된 고장 데이터 전체를 사용하고 있다. 그런데 초기에 수집된 고장 데이터는 미래 고장 예측에 영향을 주지 않을 수도 있고 경우에 따라서는 미래 고장 예측 과정에서 왜곡된 결과를 초래할 수도 있다. 이를 해결하기 위해서 이 논문에서는 부분 고장 데이터를 이용하여 적합도 평가를 수행하는 방법에 기반을 둔 소프트웨어 신뢰도 성장 모델 선택 방법을 제안한다. 이 논문에서는 고장 데이터에서 과도하게 불안정한 데이터를 제외한 부분 고장 데이터를 사용한다. 소프트웨어 신뢰도 성장 모델 선택에 사용될 부분 데이터는 전체 고장 데이터와 고장 데이터의 일부를 제외한 부분 고장 데이터의 미래 고장 예측 능력의 비교를 통해서 찾는다. 연구의 타당성을 보이기 위하여 실제 수집된 고장 데이터를 사용해서 전체 데이터를 적용한 경우보다 부분 데이터를 사용한 경우의 미래 고장 예측 능력이 더 정확함을 보인다.

키워드 : 소프트웨어 신뢰도, 소프트웨어 신뢰도 성장 모델, 모델 선택 기준, T-포인트

1. 서 론

소프트웨어는 국방 및 우주 분야와 자동차, 의료기구, 스마트폰 등 다양한 분야에서 그 사용이 증가하고 있으며 소프트웨어의 규모가 커짐에 따라 소프트웨어의 복잡도도 증가하고 있다. 소프트웨어의 사용 범위가 사람의 생명과 밀접한 분야까지 확장됨에 따라서 소프트웨어의 품질에 대한 중요성이 높아지고 있다. 소프트웨어의 품질은 기능성

※ 본 연구는 국방과학연구소의 지원으로 수행되었습니다.
(계약번호 UDI30043CD)

† 준 회 원 : 충남대학교 컴퓨터공학과 석사과정

** 준 회 원 : 충남대학교 컴퓨터공학과 박사과정

*** 종신회원 : 충남대학교 컴퓨터공학과 교수

Manuscript Received : October 21, 2014

First Revision : December 9, 2014

Accepted : December 9, 2014

* Corresponding Author : Hyeon Soo Kim(hskim401@cnu.ac.kr)

(Functionality), 이식성(Portability), 사용성(Usability), 신뢰도(Reliability) 등의 품질 속성을 사용해서 표현한다. 이중 신뢰도는 소프트웨어에서 발생하는 고장과 관련된 품질 속성이다. 소프트웨어 신뢰도는 특정 환경이나 조건에서 특정 시간 동안 소프트웨어가 고장 없이 동작할 확률로 정의되며 정량적인 값을 갖는다. 또한 목표 신뢰도와 평가 시점의 신뢰도를 비교해서 목표 신뢰도를 달성하기 위해 발견해야 하는 소프트웨어의 고장의 수와 고장 발견에 필요한 테스트 시간 등 요구되는 자원을 정량적으로 표현할 수 있기 때문에 테스트에 필요한 노력 및 소프트웨어 출시일을 결정하는 데도 사용되고 있다[1].

소프트웨어 신뢰도 모델은 그것이 적용되는 소프트웨어 개발 주기에 따라 분류될 수 있다. 테스트 단계 이전에는 과거 데이터 또는 사용하는 언어, 개발 도메인, 복잡도, 아키텍처 등의 정보를 사용하여 신뢰도를 예측하는 신뢰도 예측 모델이 사용되며, 테스트 단계 이후에는 테스트 단계에서 수집된 고장데이터를 사용해서 남아 있는 고장 또는 전체 고장을 추정하는 소프트웨어 신뢰도 성장 모델(Software Reliability Growth Model : SRGM)이 사용된다[2].

소프트웨어 신뢰도 성장 모델을 사용하기 위해서는 대상 소프트웨어에 적합한 모델을 개발하거나 이미 만들어진 모델을 사용하는 방법이 있다. 이중 새로운 소프트웨어 신뢰도 성장 모델을 개발해서 사용하는 데에는 몇 가지 어려움이 있다. 첫째, 소프트웨어 신뢰도 성장 모델 개발이 무척 어렵다는 것, 둘째, 개발된 소프트웨어 신뢰도 성장 모델에 대한 검증이 어렵다는 것, 셋째, 소프트웨어 신뢰도 성장 모델을 개발하고 구현하기 위한 노력이 소프트웨어 개발에서 많은 부분을 차지하는 문제가 발생할 수 있다. 이런 문제점 때문에 새로운 소프트웨어 신뢰도 성장 모델을 개발하기보다는 이미 만들어진 소프트웨어 신뢰도 성장 모델의 사용을 선호한다. 기존에 존재하는 소프트웨어 신뢰도 성장 모델을 사용하기 위해서는 소프트웨어에 적합한 소프트웨어 신뢰도 성장 모델을 선택해야 한다. 왜냐하면 소프트웨어 신뢰도 성장 모델은 개발 프로세스 또는 테스트 프로세스, 개발자, 소프트웨어의 규모 및 복잡도, 난이도 등에 따라 다양한 분포를 갖는 고장 데이터에 대응해서 개발되었기 때문이다[3]. 도메인별로 사용 가능한 소프트웨어 신뢰도 성장 모델을 한정지을 수 없다. 소프트웨어 신뢰도 성장 모델을 선택할 때 가장 중요한 사항은 테스트나 운영 단계에서 고장 및 결함 데이터를 수집하는 것이다. 고장 데이터를 수집하기에 앞서 고장 데이터를 정의하는 과정에서 도메인 특성을 반영할 수 있지만 소프트웨어 신뢰도 성장 모델을 적용할 때에는 수집된 데이터가 어떤 도메인에서 수집된 데이터인지보다는 그 데이터의 분포, 그 데이터의 특성 등을 고려한다. 따라서 같은 도메인 내의 소프트웨어라 할지라도 수집된 데이터의 특성이나 분포가 다르다면 같은 소프트웨어 신뢰도 성장 모델을 적용하기 어렵다.

대상 소프트웨어에 적합한 소프트웨어 신뢰도 성장 모델을 선택하기 위한 방법에는 여러 가지가 있다. 한 가지 방

법은 특정 기준을 사용해서 모델을 분류한 후 기준에 맞는 모델을 식별하는 방법이다[4]. 다른 방법은 실제 고장 데이터를 적용한 결과를 바탕으로 실제 값(수집된 고장 데이터)과 추정 값(신뢰도 모델로 계산된 값)의 적합도(Goodness-of-fit)를 평가해서 가장 잘 맞는 모델을 선택하는 방법이다[2]. 소프트웨어 신뢰도 성장 모델을 선택하기 위한 기존의 방법들은 테스트 단계에서 얻은 모든 고장 데이터를 사용하고 있다. 하지만 시스템 테스트 또는 운영 단계 초기에 발생한 고장은 시스템이 안정화된 이후에 발생하는 고장에 영향을 주지 않을 수 있다. 또한 미래에 발생할 고장을 표현하는데 먼 과거의 고장 데이터가 영향을 주지 않을 수 있다. 수집된 고장 데이터의 일부분을 사용해서 더 정확한 미래의 고장을 예측할 수 있다면 소프트웨어 신뢰도 성장 모델 선택 과정에서도 전체 고장 데이터가 아닌 고장 데이터의 일부분을 사용했을 때 더 정확한 소프트웨어 신뢰도 성장 모델을 선택할 수 있을 것이다. 이 과정에서 적용 가능한 소프트웨어 신뢰도 성장 모델들의 성능을 비교하여 가장 우수한 성능을 갖는 모델을 선택한다면 소프트웨어 신뢰도 성장 모델 선택에서의 정확성을 향상시킬 수 있을 것이다. 이 논문에서는 소프트웨어 신뢰도 성장 모델 선택의 정확성을 향상시키기 위하여 고장 데이터 중 미래에 발생할 고장에 영향을 미치는 부분 데이터를 사용하는 방법을 제안한다. 이를 위해서 고장 데이터의 일부분을 구분할 수 있는 T-포인트(T-Point)를 찾기 위한 방법과 이를 적용한 소프트웨어 신뢰도 성장 모델 선택 방법을 제시한다.

2. 관련 연구

소프트웨어 신뢰도 성장 모델을 선택하기 위한 방법은 신뢰도 모델을 식별하기 위한 기준을 사용하는 방법과 고장 데이터를 사용하는 방법으로 나눌 수 있다. 논문 [4]의 연구에서는 모델 선택을 위한 9가지 기준으로 소프트웨어 개발 생명 주기(SDLC), 사용자가 요구하는 출력(Output Desired by the User), 모델이 필요로 하는 입력(Input Required by Model), 데이터의 추세(Trend Exhibited by the data), 데이터에 일치하는 가정의 유효성(Validity of Assumptions According to Data), 프로젝트의 특질(Nature of Project), 프로젝트의 구조(Structure of Project), 테스트 프로세스(Test Process), 개발 프로세스(Development Process))를 제시하고 있으며 이 기준들에 가중치를 적용해서 조합하는 SoReMS 알고리즘을 제안하였다. 논문 [5]의 연구에서는 예측 유효성(Predictive Validity), 역량(Capability), 가정의 품질(Quality of Assumptions), 적용성(Applicability), 단순성(Simplicity)을 사용하는 방법을 제안하였다. 이 중에서 예측 유효성은 고장 데이터를 적용 후 판단하는 방법이다. 논문 [6,7]의 연구에서는 Lowest Akaike Information Criteria (AIC)를 사용하는 방법을 제안하였다. 논문 [8]의 연구에서는 현재 프로젝트와 유사한 과거 프로젝트를 찾고 과거 프로젝트에 적용된 신뢰도 모델을 선택하는 방법을 제안하였다.

소프트웨어 신뢰도 성장 모델 식별 기준을 사용하는 방법은 테스트 단계 이전에 소프트웨어 신뢰도 성장 모델을 결정할 수 있다. 또한 고장 데이터를 적용하고 적합도를 계산 및 비교하는 과정이 없기 때문에 소프트웨어 신뢰도 성장 모델 선택을 위한 시간을 줄일 수 있다는 장점이 있다.

소프트웨어 신뢰도 성장 모델에 고장 데이터를 적용한 후 그 결과를 비교하는 방법의 연구로 논문 [2]의 연구에서는 12개의 모델 선택 기준을 기반으로 순서를 매긴 소프트웨어 신뢰도 성장 모델에 대해서 거리기반 접근 방법(Distance based Approach)을 사용해서 모델을 선택하는 방법을 제안하고 있다. 연구 [9]에서는 가장 잘 맞는 하나의 모델을 선택하는 방법이 아닌 특정 고장 데이터에 적합한 모델들을 조합하는 방법을 제안하였다. 연구 [10]에서는 경험 데이터를 기반으로 소프트웨어 신뢰도 성장 모델을 선택하는 방법을 제안하였다. 이 방법은 기준치 이상의 R-Value 값을 갖는 소프트웨어 신뢰도 성장 모델을 식별한 후 상대 에러(Relative Error)를 계산하여 소프트웨어 신뢰도 성장 모델을 선택하는 방법이다. 연구 [11]에서는 소프트웨어 신뢰도 성장 모델을 식별하기 위한 기준과 통합 계획(Unification Scheme)을 결합하여 사용하는 방법을 제안하였다. IEEE 표준 문서[12]에서는 미래 예측 정확성(Future Predictive Accuracy), 과거 예측 유효성(Historical Predictive Validity), 일반성(Generality), 왜곡된 자료에 대한 둔감성(Insensitivity to Noise)을 모델 평가의 기준으로 제시하고 있는데 이 중에서 예측 유효성을 비교하기 위한 방법으로 정확성(Accuracy), 편향성(Bias), 추세(Trend), 상대 에러(Relative Error)를 제시한다. 소프트웨어 신뢰도 성장 모델 선택을 위해서 고장 데이터를 사용하는 경우에는 정량적인 값에 근거해서 소프트웨어 신뢰도 성장 모델을 선택하기 때문에 소프트웨어 신뢰도 성장 모델 선택의 정확도를 높일 수 있다는 장점이 있다.

이 논문에서는 소프트웨어 신뢰도 성장 모델을 선택하기 위하여 소프트웨어 신뢰도 성장 모델 식별 기준과 고장 발생에 영향을 주는 고장 데이터의 일부분을 사용하는 방법을 함께 사용한다. 식별 기준을 사용해서 고장 데이터를 적용할 소프트웨어 신뢰도 성장 모델의 후보군을 줄일 수 있으며 고장 데이터를 사용함으로써 소프트웨어 신뢰도 성장 모델 선택의 정확도를 높일 수 있다.

3. 소프트웨어 신뢰도 성장 모델 선택 방법의 개요

특정 소프트웨어 신뢰도 성장 모델이 모든 소프트웨어에 맞지 않기 때문에 소프트웨어 신뢰도 성장 모델을 사용하기 위해서는 소프트웨어에 잘 맞는 소프트웨어 신뢰도 성장 모델을 찾아야 한다. 이 논문에서 제안하는 소프트웨어 신뢰도 성장 모델 선택 방법은 다음과 같다.

- 소프트웨어 신뢰도 성장 모델 선택 기준을 사용해서 소프트웨어 신뢰도 성장 모델 후보를 식별한다.

- 수집된 고장 데이터와 신뢰도 성장 모델을 사용해서 추정한 값의 적합도(Goodness-of-fit : GOF)를 이용하여 가장 적합한 소프트웨어 신뢰도 성장 모델을 선택한다.

소프트웨어 신뢰도 성장 모델 선택을 위한 기준과 수집된 고장 데이터의 특성을 사용해서 소프트웨어 신뢰도 성장 모델을 찾을 수 있다. 하지만 소프트웨어 신뢰도 성장 모델 식별 기준에 의해 식별된 소프트웨어 신뢰도 모델이 여러 개가 존재할 수 있기 때문에 실제 값(수집된 고장 데이터)과 추정 값(신뢰도 모델을 사용해서 추정한 값)의 적합도(GOF)를 계산하여 가장 적합한 소프트웨어 신뢰도 성장 모델을 선택한다. 적합도를 계산하기 위해서는 실제 값(수집된 실제 고장 데이터)과 추정 값(신뢰도 모델을 통해서 얻은 추정 값)이 필요하다. 수집된 고장 데이터는 소프트웨어 신뢰도 성장 모델의 파라미터를 추정하는 데 사용되며 소프트웨어 신뢰도 성장 모델의 파라미터 추정 결과를 사용하여 추정 값을 얻는다. 정확한 추정 값을 얻기 위하여 이 논문에서는 전체 고장 데이터가 아닌 부분 데이터를 사용한다. 부분 데이터를 이용하는 방법은 다음과 같다.

- 미래 고장 발생에 영향을 주지 않거나 미래 고장 예측을 왜곡시키는 고장 데이터 부분을 찾기 위해서 T-Point를 사용한다.
- 전체 고장 데이터에서 T-Point까지의 고장 데이터를 제외한 나머지 고장 데이터를 사용해서 소프트웨어 신뢰도 성장 모델을 선택한다.

4. T-Point의 결정

이 절에서는 소프트웨어 신뢰도 성장 모델 선택에 사용할 고장 데이터를 얻기 위한 방법을 제시한다. Fig. 1은 고장 데이터 분포에 대한 하나의 사례이다. Fig. 1에서 보듯이 초기 몇 구간에서는 그 이후의 구간보다 더 많은 고장이 발생한다. 초기에 발생하는 고장은 세 가지 유형으로 나뉘질 수 있다. 첫 번째는 테스트 초기에 시스템이 불안정한 상황에서 발생하는 고장으로 시스템이 안정화된 이후의 고장 발생에 영향을 주지 않는 경우이다. 두 번째는 시스템이 안정화된 이후에 발생한 고장이지만 미래에 발생할 고장에 영향을

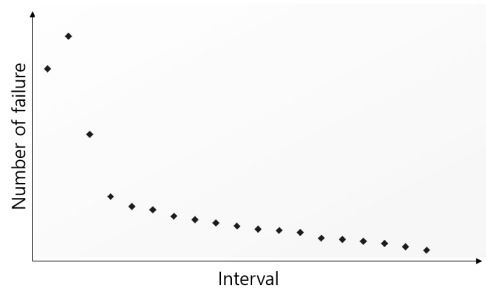


Fig. 1. Example of Failure Data Distribution

주지 않는 고장이다. 세 번째는 시스템이 안정화된 이후에 발생한 고장이며 이 고장은 미래에 발생할 고장에 영향을 주는 경우이다.

이 논문에서는 수집된 고장 데이터가 첫 번째 또는 두 번째에 해당하는 초기 고장 데이터를 포함할 때, 정확한 신뢰도 예측을 위해서 고장 데이터에서 해당하는 데이터를 제외한다. T-Point는 전체 고장 데이터에서 유효한 데이터(고장 발생에 영향을 주는 데이터)와 유효하지 않은 데이터(고장 발생에 영향을 주지 않는 데이터)를 구분하는 지점을 의미한다.

4.1 고장 데이터의 분할

전체 고장 데이터를 사용하는 방법에서는 Fig. 2와 같이 수집된 데이터 전체를 사용한다. 즉 수집된 고장 데이터의 처음부터 끝까지를 사용하여 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다.



Fig. 2. Range of Full Data Set

이 논문에서는 Fig. 3과 같은 고장 데이터 분할을 사용한다. T-Point를 결정한 후 T-Point 이후의 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다.

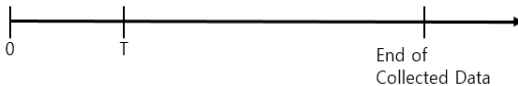


Fig. 3. Partitions of Data Set with T-Point

마지막으로 T-Point를 찾기 위해 사용하는 고장 데이터의 분할은 Fig. 4와 같다. Fig. 4는 수집된 고장 데이터를 3개의 구간으로 나누고 있다. 첫 번째 구간인 [0-T] 구간은 소프트웨어 신뢰도 성장 모델에 적용할 고장 데이터에서 제외할 구간이다. 두 번째 구간인 (T-S] 구간은 소프트웨어 신뢰도 성장 모델에 적용할 고장 데이터이다. 이 구간의 고장 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다. 마지막으로 (S-End] 구간은 소프트웨어 신뢰도 성장 모델의 예측 능력을 평가하기 위해서 사용된다. (S-End] 구간의 고장 데이터는 실제 수집된 값이다. 하지만 소프트웨어 신뢰도 성장 모델의 파라미터를 추정하는 과정에서는 (S-End] 구간의 고장 데이터를 수집되지 않은 미래의 고장으로 가정하고 이 구간의 데이터를 제외한 (T-S] 구간의 고장 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다. (S-End] 구간을 사용하는 이유는 과거 데이터가 미래 고장 예측에 끼치는 영향을 확인하

기 위함이다. Fig. 3처럼 T-Point 이후의 모든 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정하는 경우에는 소프트웨어 신뢰도 성장 모델의 예측 능력을 파악하기 위해 추가적인 실제 고장 데이터가 필요하고 이를 위해서는 추가 테스트를 수행해야 한다. 하지만 추가 테스트는 신뢰도를 평가하여 목표치에 도달하지 못한 경우에만 수행되어야 한다. 소프트웨어 신뢰도 성장 모델을 적용해서 얻은 (S-End] 구간의 고장에 대한 추정 값과 그 구간에서 실제 수집된 고장 데이터 값을 비교하면 각 소프트웨어 신뢰도 성장 모델의 예측 능력을 비교할 수 있다. 이 논문에서는 여러 실험 결과를 바탕으로 고장 데이터의 특징, 분포를 헤치지 않는 범위에서 S를 수집된 고장 데이터 전체 대비 80% 위치로 정한다. 즉 (S-End] 구간은 (0.8-1.0] 구간에 해당한다.

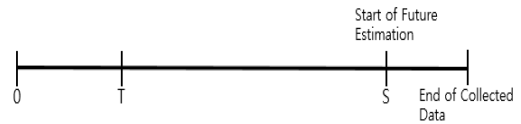


Fig. 4. Partitions of Collected Failure Data

4.2 T-Point를 찾기 위한 절차

T-Point를 찾기 위한 절차는 Fig. 5와 같다. Fig. 5에서 사용되는 변수 T는 고장 데이터 중에서 소프트웨어 신뢰도 성장 모델의 파라미터 추정에 사용되지 않는 데이터 영역을 가리키기 위해 사용된다. T가 0일 때는 (0-S]의 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정하며, T가 2이면 처음 두 구간을 제외한 나머지 고장 데이터 즉, (2-S] 구간을 사용함을 의미한다. S는 예측 구간의 시작 지점이며 End는 수집된 데이터의 끝을 나타낸다.

T-Point를 찾기 위한 자세한 절차는 다음과 같다.

1) T의 단위를 결정한다. T-Point를 찾기 위해서 사용할 단위를 결정하는 단계이다. T의 단위는 고장 개수 단위, 또는 구간 단위가 될 수 있다. 또한 한 단위에 한 개의 고장 또는 한 구간을 포함할지 그 이상을 포함할지를 결정한다. T의 단위가 작을수록 더 정확한 T-Point를 결정할 수 있지만 T-Point 결정에 더 많은 시간이 소모된다.

2) T는 고장 데이터의 시작에서 S까지 단계 1)에서 정한 T의 단위만큼 증가시킨다. 각 T에 대해서 (T-S] 구간의 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다. 그리고 이 소프트웨어 신뢰도 성장 모델을 사용해서 (S-End] 구간의 고장을 예측한다. T가 변함에 따라서 미래 고장 예측의 정확성이 변하는 정도를 확인하기 위해서 각 T에 대해서 미래 고장을 예측하는 것이다. 미래 고장 예측의 정확성은 적합도를 이용하여 평가한다. 즉 수집된 실제 고장 데이터 중 (S-End] 구간의 고장 데이터와 동일 구간에서 소프트웨어 신뢰도 성장 모델을 사용해서 추정

한 고장 데이터를 사용해서 평가한다. T의 적합도가 이전 반복에서 측정된 적합도보다 커지는 경우(T-1의 적합도보다 T의 적합도가 나빠진 경우를 의미)가 발생하면 반복을 중단하고 T-Point 결정 단계로 넘어간다.

3) T-Point를 결정한다. T-Point를 결정하기 위해서 T별로 평가된 적합도를 사용한다. T가 증가할수록 적합도가 좋아지는 것은 고장을 예측하는 데 불필요한 데이터를 제외함으로써 정확성이 높아지기 때문이다. 반대로 적합도가 낮아지는 것은 고장 예측에 필요한 데이터가 제외됨으로써 올바른 예측을 하지 못하기 때문이다. 적합도의 변화를 사용해서 T-Point를 결정한다.

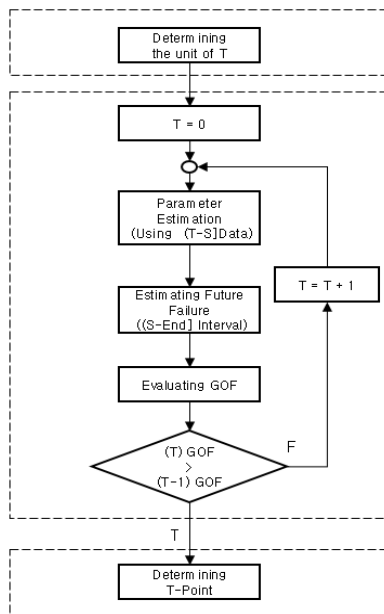


Fig. 5. Procedure for Finding T-Point

5. 소프트웨어 신뢰도 성장 모델 선택 방법

소프트웨어 신뢰도 성장 모델은 소프트웨어 테스트 단계에서 남아 있는 고장의 수, 다음 고장 발견까지의 시간, 고장 강도(Failure Intensity), 목표 신뢰도 성취 여부 등을 통해서 추가 테스트 유무를 결정하고, 테스트가 필요한 경우 추가 테스트에 필요한 자원을 계산하는 데 사용된다. 소프트웨어 신뢰도 성장 모델을 사용할 때 정확한 결과를 얻기 위해서는 테스트 프로세스, 디버깅 프로세스, 고장 데이터의 분포 등에 따라서 여러 소프트웨어 신뢰도 성장 모델 중에서 해당 소프트웨어에 가장 적합한 소프트웨어 신뢰도 성장 모델을 선택해야 한다. 이 논문에서 제안하는 소프트웨어 신뢰도 성장 모델의 선택 방법은 Fig. 6과 같다.

소프트웨어 신뢰도 모델 선택 방법은 세 단계로 구성된다. 첫 번째는 소프트웨어 신뢰도 성장 모델 선택에 사용될 기준을 결정하는 단계이다. 결정할 기준으로 소프트웨어 신뢰

도 성장 모델 식별 기준과 T-Point 및 소프트웨어 신뢰도 성장 모델 결정에 사용할 적합도 평가 기준이 있다. 두 번째는 첫 번째 단계에서 결정한 기준을 사용해서 적절한 소프트웨어 신뢰도 성장 모델 후보들을 식별하고 수집된 고장 데이터와 식별된 소프트웨어 신뢰도 성장 모델 후보를 사용해서 T-Point를 결정한다. 마지막 단계는 식별된 소프트웨어 신뢰도 성장 모델 후보 중에서 최적의 소프트웨어 신뢰도 성장 모델을 선택하는 단계이다. 이를 위해 T-Point가 적용된 고장 데이터를 사용하여 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다. 계속해서 실제 값(수집된 고장 데이터)과 추정 값(신뢰도 모델을 사용해서 추정한 값)의 적합도(GOF)를 평가한다. 평가 결과 가장 좋은 적합도를 갖는 모델을 소프트웨어에 가장 잘 맞는 소프트웨어 신뢰도 성장 모델로 결정한다.

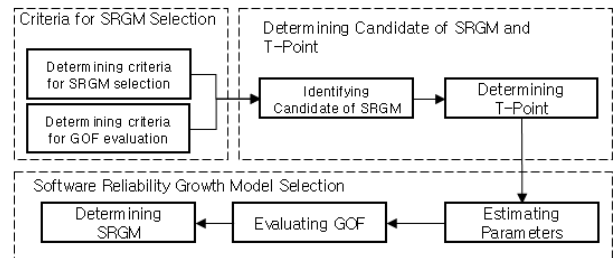


Fig. 6. Procedure for Selecting Software Reliability Growth Models

5.1 소프트웨어 신뢰도 성장 모델 선택 기준 결정

이 단계에서는 소프트웨어 신뢰도 성장 모델 선택 과정에서 사용할 두 종류의 기준을 결정한다.

1) 소프트웨어 신뢰도 성장 모델 후보 식별 기준 결정

소프트웨어 신뢰도 성장 모델들은 각 모델에 맞는 테스트 프로세스, 디버깅 프로세스, 고장 데이터의 분포 등과 같은 몇 가지 가정을 갖는다. 여기서는 이런 정보를 기반으로 소프트웨어 신뢰도 성장 모델 후보를 식별할 기준을 결정한다. 사용할 수 있는 기준으로는 고장 데이터의 분포(누적 고장 데이터 분포 유형 : 오목(Concave) 분포, S형(S-Shaped) 분포), 테스트·디버깅 프로세스, 고장 데이터의 유형, 소프트웨어 신뢰도 성장 모델을 통해서 얻을 수 있는 값의 종류 등이 있다.

2) 적합도 평가 기준 결정

소프트웨어 신뢰도 성장 모델 선택 과정에서 적합도는 T-Point 결정 과정과 고장 데이터의 분포에 가장 적합한 소프트웨어 신뢰도 성장 모델을 결정할 때 사용된다. Table 1은 적합도를 평가하기 위한 기준들이다. 사실 이외에도 더 많은 기준들이 존재한다.

Fig. 6의 두 번째 단계에서 T-Point 결정을 위해 사용되는 적합도 평가 기준은 소프트웨어 신뢰도 성장 모델 내에

서 T의 변화에 따른 적합도의 변화를 확인하는 데 사용된다. 하지만 Fig. 6의 세 번째 단계에서 소프트웨어 신뢰도 성장 모델 선택을 위해 사용되는 적합도 평가 기준은 신뢰도 모델 간에 비교에 사용되기 때문에 R-Square 같이 특정 범위 내의 값으로 확인할 수 있는 기준을 사용한다.

5.2 소프트웨어 신뢰도 성장 모델 후보 및 T-Point 결정

소프트웨어 신뢰도 성장 모델 선택에 필요한 입력물을 준비하는 단계이다. 소프트웨어 신뢰도 성장 모델을 선택하기 위해서 필요한 소프트웨어 신뢰도 성장 모델 후보군을 식별하며, 소프트웨어 신뢰도 성장 모델에 적용할 고장 데이터를 결정하기 위해서 T-Point를 결정한다.

1) 소프트웨어 신뢰도 성장 모델 후보군 식별

소프트웨어 신뢰도 성장 모델 식별 기준에 따라 적절한 신뢰도 모델을 파악한다. 예를 들어, 누적 고장 데이터의 분포가 오목 분포인 경우에는 NHPP(Non-Homogeneous Poisson Process) 모델, Musa/Okumoto Logarithmic 모델, Jelinski/Moranda 모델 등이 적용될 수 있으며 S형 분포의 경우에는 Gompertz 모델, Yamada Delayed S-shaped 모델, Yamada S-Shaped 모델 등이 적용될 수 있다. 고장 값의 종류는 고장 사이의 평균 시간(Mean Time To Failure : MTTF)과 구간 내 고장의 수(Failure Count)로 나뉜다. 고장 사이의 평균 시간의 경우에는 NHPP 모델, Musa/Okumoto Logarithmic 모델, Jelinski/Moranda 모델 등이 적용될 수 있고 구간 내 고장의 수의 경우에는 Schneidewind 모델, Yamada S-Shaped 모델, Schick/Wolverton 모델 등이 적용될 수 있다.

Table 1. Goodness-of-fit evaluation criteria

Criterion	Definition
Bias	$\sum_{i=1}^k \frac{\hat{m}(t_i) - m_{t_i}}{k}$
MSE (Mean Square Error)	$\sum_{i=1}^k \frac{(m_{t_i} - \hat{m}(t_i))^2}{k}$
MAE (Mean Absolute Error)	$\sum_{i=1}^k \frac{ m_{t_i} - \hat{m}(t_i) }{k}$
PRR (Predictive-ratio Risk)	$\sum_{i=1}^k \frac{\hat{m}(t_i) - m_{t_i}}{\hat{m}(t_i)}$
Variance	$\sqrt{\frac{1}{k-1} \sum_{i=1}^k (m_{t_i} - \hat{m}(t_i) - Bias)^2}$
RMSPE (Root Mean Square Prediction Error)	$\sqrt{Variance^2 + Bias^2}$
R-Square	$\frac{\sum_{i=1}^k (\hat{m}(t_i) - \bar{m})^2}{\sum_{i=1}^k (m_{t_i} - \bar{m})^2}$

$\hat{m}(t_i)$: number of failure estimated at time t_i
 m_{t_i} : number of failure observed at time t_i
 k : Total number of failure

전체 소프트웨어 신뢰도 성장 모델 중에서 소프트웨어에 사용가능한 소프트웨어 신뢰도 성장 모델 후보군을 식별함으로써 이후 적합도 평가 단계에서 불필요한 노력을 줄인다.

2) T-Point 결정 및 고장 데이터 식별

테스트 단계에서 수집된 고장 데이터 중에서 소프트웨어 신뢰도 성장 모델 선택에 사용할 데이터를 구분하기 위해서 T-Point를 결정한다. T-Point를 결정하기 위한 방법은 4절의 절차를 따른다. T-Point를 결정 후 T-Point 이후의 데이터를 고장 데이터로 식별한다.

5.3 소프트웨어 신뢰도 성장 모델 선택

식별된 소프트웨어 신뢰도 성장 모델 후보 중에서 최적의 소프트웨어 신뢰도 성장 모델을 결정한다.

1) 소프트웨어 신뢰도 성장 모델 파라미터 추정

소프트웨어 신뢰도 성장 모델을 사용하기 전에 소프트웨어 신뢰도 성장 모델이 갖는 파라미터를 추정한다. (T-E) 구간의 고장 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한다.

2) 적합도 평가

실제 값(수집된 고장 데이터)과 추정 값(신뢰도 모델을 사용해서 추정한 값)을 사용해서 소프트웨어 신뢰도 성장 모델 후보들의 적합도(GOF)를 평가한다.

3) 소프트웨어 신뢰도 성장 모델 결정

소프트웨어 신뢰도 성장 모델 후보들의 적합도(GOF)를 비교해서 실제 고장 데이터와 가장 적합한 소프트웨어 신뢰도 성장 모델을 결정한다.

6. 사례 연구

6.1 실험 환경

실험 환경은 아래와 같이 구성하였다. 고장 데이터는 Musa 데이터 셋[13]에서 Table 2와 같이 프로젝트 1, 프로젝트 2, 프로젝트 3에 해당하는 3개의 고장 데이터 셋(SET)을 사용한다.

Table 2. Failure Data Set

PROJECT	APPLICATION	Number of Failures
PROJECT 1	Real Time Command & Control	136
PROJECT 2		54
PROJECT 3		38

실험에 사용한 도구는 SMERFS[14], SRTPro[15]이다. 소프트웨어 신뢰도 성장 모델은 NHPP 모델[16], Musa/

Okumoto Logarithmic 모델[17], Musa Basic 모델[18]을 사용한다.

6.2 실험 결과

실험은 5절의 소프트웨어 신뢰도 성장 모델 선택 방법의 단계를 따라 진행하였다.

1) 소프트웨어 신뢰도 성장 모델 선택 기준

A) 소프트웨어 신뢰도 성장 모델 후보 식별 기준 결정

Musa 데이터 셋의 고장 데이터에 대한 테스트 프로세스, 디버깅 프로세스 등을 파악할 수 없기 때문에 고장 데이터의 분포 및 소프트웨어 신뢰도 성장 모델 적용 가능 여부를 사용해서 소프트웨어 신뢰도 성장 모델 후보를 식별한다. 고장 데이터의 분포는 오목 분포, S형 분포로 구분하며 소프트웨어 신뢰도 성장 모델 적용 가능 여부는 신뢰도 평가 도구의 지원 여부를 통해서 판단한다.

B) 적합도 평가 기준 결정

적합도 평가 기준은 Table 1에 나열된 기준들 외에 여러 기준이 있지만 어떤 기준이 가장 좋다고 결정할 수 없다. 이 논문에서는 T-Point를 결정하기 위한 적합도 평가 기준으로 MAE(Mean Absolute Error)를 사용한다. 그리고 고장 수(Failure Count) 모델과 고장 사이 평균 시간(MTTF)모델을 함께 사용하기 때문에 모델 간 비교를 위해서 적합도 평가 기준으로 R-square를 사용한다.

2) 소프트웨어 신뢰도 성장 모델 후보 및 T-Point 결정

A) 소프트웨어 신뢰도 성장 모델 후보군 식별

프로젝트 1의 고장 데이터 분포는 Fig. 7과 같다. 고장 데이터 분포에 의해서 NHPP 모델, Musa/Okumoto Logarithmic 모델, Musa Basic 모델을 식별한다. 프로젝트 1 고장 데이터는 식별된 모든 모델에 적용 가능하다.

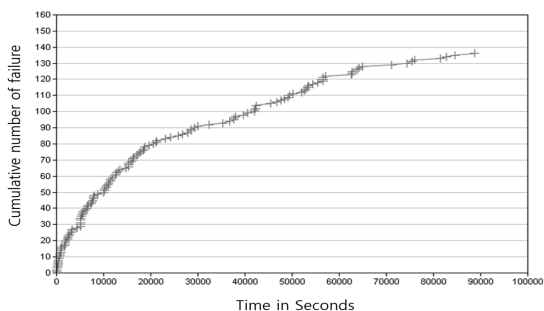


Fig. 7. Failure Data Distribution of Project1[13]

프로젝트 2의 고장 데이터 분포는 Fig. 8과 같다. 고장 데이터 분포에 의해서 NHPP 모델, Musa/Okumoto Logarithmic 모델, Musa Basic 모델을 식별한다. 프로젝트 2 고장 데이터는 식별된 모든 모델에 적용 가능하다.

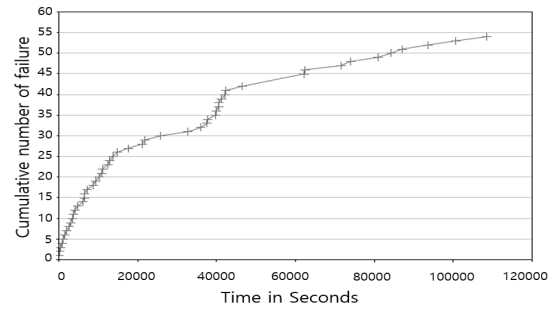


Fig. 8. Failure Data Distribution of Project2[13]

프로젝트 3의 고장 데이터 분포는 Fig. 9와 같다. 고장 데이터 분포에 의해서 NHPP 모델, Musa/Okumoto Logarithmic, Musa Basic 모델을 식별한다. 프로젝트 3 고장 데이터는 식별된 모든 모델에 적용 가능하다.

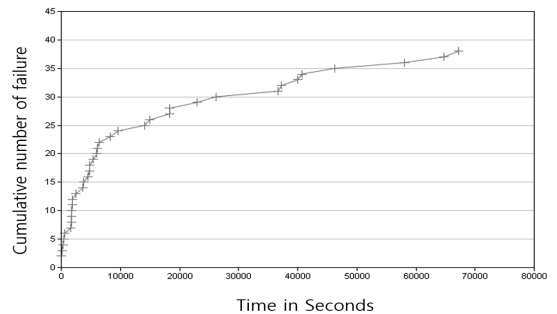


Fig. 9. Failure Data Distribution of Project3[13]

고장 데이터별로 적용할 소프트웨어 신뢰도 성장 모델을 정리하면 Table 3과 같다.

Table 3. Candidates of Software Reliability Growth Model

PROJECT	Candidates of SRGM
PROJECT 1	Musa/Okumoto Logarithmic Model NHPP Model Musa Basic Model
PROJECT 2	Musa/Okumoto Logarithmic Model NHPP Model Musa Basic Model
PROJECT 3	Musa/Okumoto Logarithmic Model NHPP Model Musa Basic Model

B) T-Point 결정 및 고장 데이터 식별

T-Point는 고장 데이터 별로 결정하며 Table 3의 소프트웨어 신뢰도 성장 모델과 앞에서 결정한 MAE 기준을 사용해서 결정한다.

- T의 간격은 구간(Interval) 단위로 한다.
- Table 4, 5, 6은 각각의 프로젝트에 대해서 T를 구간 단위로 증가시키면서 미래 고장 예측 능력을 평가하기 위한 적합도(GOF)를 계산한 것이다.

Table 4. T-Point in PROJECT 1 Data Set

T	Musa/Okumoto	Musa Basic	NHPP
0	2107.746	2516.863	1.060
1	1887.382	1514.768	0.556
2	1845.563	1395.821	0.473
3	1810.276	1341.192	0.276
4	13516.7	13909.5	0.392

Table 4, 5, 6에서 사용된 고장 데이터는 프로젝트 1, 2, 3의 고장 데이터를 Fig. 4처럼 나눠서 사용한다. T(0)은 Fig. 4의 [0-S] 구간의 고장 데이터를 사용한 경우이고 T(1)은 첫 번째 구간을 제외한 (1-S] 구간의 고장 데이터를 사용한 경우이다. 나머지도 처음부터 T까지의 구간을 제외한 고장 데이터를 사용한 경우에 해당한다. 표에서 모델별로 계산된 값들은 (S-End] 구간에서의 실제 값(수집된 고장 데이터)과 추정 값(신뢰도 모델을 사용해서 추정한 값)의 적합도 평가 결과이다. 프로젝트별로 T를 증가시켜가며 적합도 (Goodness-of-fit)를 평가한 결과를 바탕으로 T-Point를 선택한다. 적합도 평가 기준으로 MAE를 사용했기 때문에 0에 가까울수록 좋다. Table 6에서 Musa/Okumoto 모델의 경우 첫 번째 구간을 제외했을 때의 적합도가 전체 고장 데이터를 사용했을 때보다 약 67%가량 좋아진다. 즉, 전체 고장 데이터를 사용한 경우보다 T-Point 이후의 고장 데이터를 사용했을 때 미래 고장 예측의 결과의 정확도가 향상됨을 볼 수 있다. 세 번째 구간까지 제외했을 때 가장 좋은 미래 고장 예측 능력을 보이고 있다.

Table 5. T-Point in PROJECT 2 Data Set

T	Musa/Okumoto	Musa Basic	NHPP
0	4351.741	4127.572	0.813
1	1613.8	3866.93	0.298
2	1928.36	5421.875	0.548

Table 6. T-Point in PROJECT 3 Data Set

T	Musa/Okumoto	Musa Basic	NHPP
0	35899.033	5606.191	10.85
1	11683.722	2750.120	8.252
2	8889.870	2416.966	8.071
3	6237.390	2104.453	7.757
4	12166.258	2791.928	8.966

위 표에서 Musa/Okumoto 모델과 Musa Basic 모델은 입력과 출력으로 고장 사이 시간(MTTF)을 사용하는 모델이므로 초(sec)단위의 고장 데이터를 사용하였고 NHPP 모델은 MTTF와 고장 수(Failure Count)를 모두 사용할 수 있지만 입력 데이터가 고장의 개수로 주어졌기 때문에 각 모델의 적합도 평가 결과 값의 크기가 다르게 나타난 것을 볼 수 있다. 하지만 소프트웨어 신뢰도 성장 모델 내에서의 적

합도 변화를 기반으로 T-Point를 결정하기 때문에 소프트웨어 신뢰도 성장 모델 간에 값의 크기 차이는 허용 가능하다.

Table 7은 프로젝트별 T-Point와 소프트웨어 신뢰도 성장 모델 선택의 세 번째 단계에서 사용할 고장 데이터의 범위를 나타낸다.

Table 7. Failure Data Used for SRGM Selection

PROJECT	T-POINT	FAILURE DATA
PROJECT 1	3	(3,END]
PROJECT 2	1	(1,END]
PROJECT 3	3	(3,END]

3) 소프트웨어 신뢰도 성장 모델 선택

Table 8. Results of Parameter Estimation

PROJECT	Musa/Okumoto		Musa Basic		NHPP
	β_0	β_1	β_0	β_1	b
PROJECT 1	41.694	7.63E-05	94.867	2.48E-05	2.50E-05
PROJECT 2	18.104	6.43E-05	42.391	2.04E-05	2.06E-05
PROJECT 3	12.358	3.10E-05	18.909	1.99E-05	2.34E-05

Table 9. Results of Goodness-of-fit Evaluation

PROJECT	Musa/Okumoto	Musa Basic	NHPP
PROJECT 1	0.965	0.985	0.983
PROJECT 2	0.910	0.984	0.975
PROJECT 3	0.815	0.974	0.975

Table 10. Results of SRGM Selection

PROJECT	Result of SRGM Selection
PROJECT 1	Musa Basic Model
PROJECT 2	Musa Basic Model
PROJECT 3	NHPP Model

Table 8은 Table 7의 고장 데이터를 사용해서 소프트웨어 신뢰도 성장 모델의 파라미터를 추정한 결과이다. 앞서 언급하였듯이 Table 4, 5, 6에서는 모델 내에서의 미래 예측 결과를 비교하기 때문에 MAE와 같은 적합도 평가 기준을 사용할 수 있지만, 각 모델의 입력 값과 출력 값의 양식이 다를 경우에 소프트웨어 신뢰도 성장 모델들을 비교하기 위해서는 R-square와 같은 적합도 평가 기준을 사용해야 한다. Table 9는 Table 8의 파라미터 추정 결과를 적용해서 소프트웨어 신뢰도 성장 모델이 계산한 추정 값과 수집된 실제 고장 데이터 값 사이의 적합도를 평가한 결과이다.

Table 10은 각 프로젝트별로 선택된 소프트웨어 신뢰도 성장 모델이다. 적합도 평가 결과가 가장 좋은 소프트웨어

신뢰도 성장 모델을 선택한 결과이다. Table 9에서 사용한 적합도 평가 기준은 R-Square이기 때문에 1에 가까울수록 적합한 모델이다.

7. 결론 및 향후 연구

이 논문에서는 소프트웨어 신뢰도 성장 모델 선택 방법과 소프트웨어 신뢰도 성장 모델 선택에 사용할 부분 고장 데이터를 찾기 위한 T-Point 결정 방법을 제안하였다. 이 논문에서 제안하는 방법은 소프트웨어 신뢰도 성장 모델 식별 기준과 고장 데이터를 사용한 소프트웨어 신뢰도 성장 모델 적합도 비교 방법을 포함하고 있다.

소프트웨어 신뢰도 성장 모델의 식별 기준과 고장 데이터를 사용한 적합도 평가 두 방법을 함께 적용한 점과 전체 고장 데이터가 아닌 부분 고장 데이터를 사용한 점이 기존 소프트웨어 신뢰도 성장 모델 선택 방법과 차이가 있다. 이 연구의 장점은 고장 데이터를 적용하기 전에 식별 기준을 사용해서 선택 가능한 소프트웨어 신뢰도 성장 모델 후보군을 파악함으로써 적합도 평가에 필요한 시간을 줄일 수 있다는 것이다. 또한 전체 고장 데이터 중에서 실제 미래 고장 예측에 필요한 부분 고장 데이터를 결정하고 사용함으로써 신뢰도 평가의 정확도를 높일 수 있다는 것이다.

향후 연구로는 좀 더 다양한 고장 패턴과 그에 맞는 다양한 소프트웨어 신뢰도 성장 모델에 적용하여 이 연구 방법의 보편성을 높일 것이다.

References

[1] J. D. Musa, et al., "Software Reliability : Measurement, Prediction, Application", MacGraw-Hill, NewYork, 1987.

[2] K. Sharma, et al., "Selection of optimal software reliability growth models using a distance based approach", IEEE Transactions on Reliability, Vol.59, No.2, pp.266-276, 2010.

[3] Ullah N., Morisio M., Vetro A., "A comparative analysis of software reliability growth models using defects data of closed and open source software", In Software Engineering Workshop (SEW), 2012 35th Annual IEEE, Heraclion, pp. 187-192, 2012.

[4] C. A. Asad, et al., "An approach for software reliability model selection", In Proc. 28th Annual International Computer Software and Applications Conference, Vol.1, pp.534-539, 2004.

[5] A. Iannino, J. D. Musa, et al., "Criteria for Software Reliability Model Comparisons", IEEE Transactions on Software Engineering, Vol.se-10, No.6, 1984.

[6] T. M. Khoshgoftaar and T. G. Woodcock, "Software reliability model selection : A case study", In Proc. International Symposium on Software Reliability Engineering,

Austin, pp.183-191, 1991.

[7] Okamura. H, Dohi, T., "SRATS: Software reliability assessment tool on spreadsheet (Experience report)," 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), Pasadena, pp.100-107, 2013.

[8] Zeng. J, Li. J, Zeng. X, Luo. W., "A prototype system of software reliability prediction and estimation", Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, pp.558-561, 2010.

[9] M. R. Lyu and A. Nikola, "Applying reliability models more effectively", IEEE Software, Vol.9, No.4, pp.43-52, 1992.

[10] C. Stringfellow and A. A. Andrews, "An empirical method for selecting software reliability growth models", Empirical Software Engineering, Vol.7, No.4, pp.319-343, 2002.

[11] M. Garg, et al., "A method for selecting a model to estimate the reliability of a software component in a dynamic system", 2013 22nd Australian Conference on Software Engineering, Melbourne, pp.40-50, 2013.

[12] IEEE Std 1633-2008, "IEEE Recommended Practice on Software Reliability", IEEE Reliability Society.

[13] J. D. Musa, The Software Reliability Dataset [Internet], <https://sw.csiac.org/databases/sled/swrel.php>.

[14] William Farr, Statistical Modeling and Estimation of Reliability Functions for Systems [Internet], <http://www.slingcode.com/smerfs>.

[15] Myungmuk Kang, Software Reliability Tool professional [Internet], <http://spiral.kaist.ac.kr/wp/research/reliability-tools>.

[16] A. L. Goel and K. Okumoto, "Time-Dependent Error-Detection Rate Model for Software and Other Performance Measures", IEEE Transaction on Reliability, Vol.R-28, pp.206-211, 1979.

[17] J. D. Musa and K. Okumoto, "A logarithmic Poisson execution time model for software reliability", Proceedings of the Seventh International Conference on Software Engineering, Orlando, pp.230-238, 1984.

[18] J. D. Musa, "A theory of software reliability and its application", IEEE Transaction on Software Engineering, Vol.1, Issue 1, pp.312-327, 1975.



박 용 준

e-mail : dearpyj@cnu.ac.kr

2014년 충남대학교 컴퓨터공학과(학사)

2014년~현재 충남대학교 컴퓨터공학과 석사과정

관심분야 : 소프트웨어 신뢰성, 소프트웨어 테스트



민 법 기

e-mail : bkmin@cnu.ac.kr

2009년 공주대학교 산업정보학과(학사)

2012년 충남대학교 컴퓨터공학과(석사)

2012년~현 재 충남대학교 컴퓨터공학과
박사과정

관심분야: 소프트웨어 신뢰성, 소프트웨어
테스팅, 분산 시스템



김 현 수

e-mail : hskim401@cnu.ac.kr

1988년 서울대학교 계산통계학과(학사)

1991년 한국과학기술원 전산학과(공학석사)

1995년 한국과학기술원 전산학과(공학박사)

1995년~1995년 한국전자통신연구원 Post
Doc.

1996년~2001년 금오공과대학교 조교수

2001년~현 재 충남대학교 컴퓨터공학과 교수

관심분야: 소프트웨어공학(소프트웨어 테스팅, 소프트웨어 유지
보수), 분산 컴퓨팅