

# 무기체계를 위한 소프트웨어의 신뢰성 공학 프로세스

정회원 김기백\*, 종신회원 이재천\*\*

## On Software Reliability Engineering Process for Weapon Systems

Ghiback Kim\* *Regular Member*, Jae-Chon Lee\*\* *Lifelong Member*

### 요 약

무기체계가 첨단 복잡화 형태로 진화하면서 물리적 구현에서 소프트웨어가 차지하는 비중과 중요성은 급격히 증대되고 있다. 특히 네트워크 중심전(NCW)으로 대변되는 현대의 전쟁 상황에서 신뢰성 있게 운용되는 무기체계를 개발하기 위해서는 신뢰성 있는 소프트웨어의 개발이 필수 불가결한 요소이다. 국내 무기체계의 연구개발 과정에서 하드웨어 구성품과는 다르게 소프트웨어 요소에 대한 신뢰성 관련 기술 활동은 일부 구현된 소프트웨어에 대해 신뢰도 평가를 수행하는 수준이다. 그러나 연구개발 기간에 소프트웨어의 목표 신뢰도를 확보하기 위해서는 무기체계 개발 초기 단계부터 체계적인 신뢰성 공학 활동이 요구된다. 본 연구에서는 범위가 제약되어 있던 그동안의 소프트웨어 신뢰성 관련 연구 활동을 무기체계 연구개발 수명주기 동안 목표 신뢰도를 만족하는 소프트웨어를 개발할 수 있도록 시스템 공학 관점에서 접근법을 제시하였다. 그 결과 무기체계의 연구개발 수명주기 동안 적용할 수 있는 프로세스를 도출하였는데 이를 무기체계의 소프트웨어 신뢰성 공학 프로세스 (Software Reliability Engineering Process for Weapon Systems: SREP-WS)로 명명하였다. 각종 첨단무기체계 개발에 도출된 SREP-WS를 적용함으로써 소프트웨어 신뢰도를 정량적 관리할 수 있고, 또한 목표 신뢰도를 개발기간 중에 확보할 수 있게 된다.

**Key Words** : Software Reliability, Software Reliability Allocation, Prediction, Improvement, Estimation, Software Reliability Engineering Activities, Software Reliability Engineering Process

### ABSTRACT

As weapon systems are evolving into more advanced and complex ones, the role of the software is becoming heavily significant in their developments. Particularly in the war field of today as represented by the network centric warfare(NCW), the reliability of weapon systems is definitely crucial. In this context, it is inevitable to develop software reliably enough to make the weapon systems operate robustly in the combat field. The reliability engineering activities performed to develop software in the domestic area seem to be limited to the software reliability estimations for some projects. To ensure that the target reliability of software be maintained through the system's development period, a more systematic approach to performing software reliability engineering activities are necessary from the beginning of the development period. In this paper, we consider the software reliability in terms of the development of a weapon system as a whole. Thus, from the systems engineering point of view, we analyze the models and methods that are related to software reliability and a variety of associated activities. As a result, a process is developed, which can be called the software reliability engineering process for weapon systems (SREP-WS). The developed SREP-WS can be used in the development of a weapon system to meet a target reliability throughout its life-cycle. Based on the SREP-WS, the software reliability could also be managed quantitatively.

\* 국방과학연구소 2본부 4부(ghiback@yahoo.com), \*\* 아주대학교 시스템공학과(jaelee@ajou.ac.kr)  
논문번호 : KICS2011-01-028, 접수일자 : 2011년 1월 10일, 최종논문접수일자 : 2011년 4월 4일

## I. 서 론

1차, 2차 세계대전을 치르면서 무기체계의 신뢰성 확보에 대한 필요성이 제기되었고, 이에 따라 이 분야에 대한 연구가 지속적으로 수행되어 하드웨어에 대한 신뢰도 분석 및 검증에 대한 다양한 기법과 국방 표준<sup>[1-3]</sup>이 정립되었다.

기능의 구현이 용이하고 하드웨어의 변경으로부터 적응성이 뛰어난 소프트웨어의 특징으로 인하여, 1950년대 이후 무기체계의 중요 제어 및 연산 기능이 하드웨어에서 소프트웨어로 점차 옮겨오면서 소프트웨어의 비중이 점차 커지지 시작했다. 비중이 커진 소프트웨어로 인하여 1960년대 후반에 소프트웨어 위기가 도래하게 되었으며, 보다 신뢰성 있는 소프트웨어를 고민해 왔던 많은 연구자들이 1970년에 이르러 소프트웨어 신뢰도 평가 모델<sup>[4]</sup>을 제시하기 시작하였다.

1980대에는 미군을 중심으로 소프트웨어 신뢰도 예측 모델<sup>[8-10]</sup>이 개발되었고, 1990년대에는 소프트웨어 신뢰도 할당 모델<sup>[11-15]</sup>이 학계에서 제시되었다. 1990년대 후반에 소프트웨어 신뢰도 활동이 구체적으로 정리된 미군 표준<sup>[16]</sup>이 제시되어 무기체계 개발에 적용되기 시작하였다. 2000년대에 소프트웨어 신뢰성을 검증하기 위한 소프트웨어 신뢰성 공학 프로세스(SREP: Software Reliability Engineering Process)<sup>[17-18]</sup>가 학계에서 정의되었다.

하드웨어와는 다르게 국내 무기체계 연구개발 과정에서 소프트웨어에 대한 신뢰성 관련 공학적 활동은 일부 소프트웨어에 대하여 신뢰도 평가를 수행하는 수준이다. 그러나 연구개발 기간 내에 소프트웨어의 목표 신뢰도 확보를 위해서는 개발 초기 단계부터 개발이 종료될 때까지 체계적인 신뢰성 공학 활동이 요구된다. 그런데 기존에 정의된 SREP는 소프트웨어 신뢰도의 평가에 중점을 두고 있으며, 소프트웨어 개발 과정에서 초기 신뢰도 예측 및 신뢰도 개선에 대한 활동이 미흡하여 무기체계 연구개발에 적용하기에 어려움이 있다.

본 논문에서는 기존에 제시된 SREP의 단점을 보완하여 무기체계 소프트웨어의 연구개발에 적용 가능한 무기체계의 소프트웨어 신뢰성 공학 프로세스(SREP-WS: Software Reliability Engineering Process for Weapon Systems)를 제시한다. 기존의 SREP의 단점을 보완하기 위하여 다양한 관점에서 소프트웨어 신뢰성 공학 활동(SREA: Software Reliability Engineering Activities)을 비교, 분석하여 SREA를

도출하고, 도출된 소프트웨어 신뢰성 공학 활동을 종합하여 무기체계 소프트웨어에 적합한 신뢰성 공학 프로세스를 정의한다. 그리고 SREP-WS 각각의 활동에 대해 선행 연구를 통해 식별한 각 신뢰성 공학 활동의 세부 활동과 관련 모델 및 기법, 입출력 등을 정의한다.

본 논문의 연구 결과인 SREP-WS에 대한 검증은 세가지 방법으로 수행한다. 먼저, SREP-WS를 시스템 공학 모델링 기법으로 모델링함으로써 프로세스 자체에 논리적인 오류가 없음을 검증한다. 두 번째로 무기체계 연구개발 프로세스와 SREP-WS의 비교 분석을 통해 무기체계 연구개발 프로세스 내에서 SREP-WS가 적합함을 검증하고, 마지막으로 제안된 SREP-WS와 기존 SREP 간의 상호 비교를 통해 SREP-WS가 기존 SREP보다 개선되었음을 확인한다.

본 논문에서 제시한 SREP-WS는 그 동안 무기체계 연구개발 과정에서 소홀했던 소프트웨어 신뢰성 공학 분야에 체계적인 공학적 활동을 제시한다. 무기체계 연구개발 과정에 SREP-WS를 적용함으로써, 소프트웨어 신뢰도의 정량적 관리가 가능하고, 목표 신뢰도를 개발기간 내에 확보할 수 있다.

본 논문의 II장에는 소프트웨어 신뢰성 공학에 대한 기존의 연구 내용을 기술하였고, III장에서 다양한 프로세스를 비교 분석하여 무기체계 소프트웨어 신뢰성 공학 프로세스에 적용할 소프트웨어 신뢰성 공학 활동을 식별하였다. SREP-WS를 IV장에서 정의하였으며, V장에서는 제안한 SREP-WS에 대한 평가를 수행하였고, VI장에 결론을 기술하였다.

## II. 소프트웨어 신뢰성 공학

### 2.1 소프트웨어 신뢰도

하드웨어의 고장은 주로 장비의 운용 과정에 장비에 가해지는 열, 마찰, 진동, 충격과 같은 스트레스로 인한 구성품의 마모, 열화에 따라 발생하게 된다. 운용환경에 영향을 받는 하드웨어와는 달리 소프트웨어는 주로 소프트웨어의 요구사항, 설계 및 구현 오류(Error)에 의해 결함(Defect)이 내재되고 그 결함이 있는 코드가 실행될 때 고장(Failure)이 발생한다.

장비의 수명주기 동안 하드웨어의 고장률은 그림 1과 같이 육조 곡선과 같은 형태를 갖는다. 수명 초기의 안정화 시기를 지나면 수명 후기 전까지는 일정한 값의 고장률을 유지하게 된다.

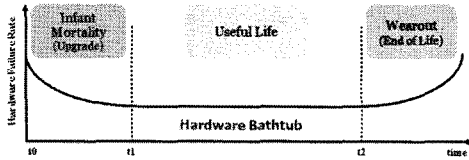


그림 1. 수명주기에 따른 하드웨어 고장률의 변화

신뢰도 관점에서 소프트웨어는 목표 신뢰도가 확보되는 시점에 개발이 완료되어 사용자에게 배포되며, 이후 소프트웨어에 대한 형상 변경이 없으면 고장률은 일정하게 유지된다. 그러나 그림 2에서 보여 주듯이 소프트웨어를 운용하는 동안 소프트웨어의 오류 수정, 기능 개선 및 추가 등의 활동으로 소프트웨어의 형상이 변경되며, 이에 따라 소프트웨어의 고장률은 변화되며 일정 시점이 되면 소프트웨어를 폐기하게 된다.

하드웨어 신뢰도와 유사하게 소프트웨어 신뢰도는 주어진 조건에서 지정된 시간 동안 소프트웨어가 고장을 일으키지 않을 확률<sup>[19]</sup>로 정의된다.

$$R(t) = P(T > t) \quad T: \text{소프트웨어 실행시간}$$

소프트웨어 고장의 의미는 사용자가 의도하지 않은 결과를 소프트웨어의 실행 결과로 얻는 것을 의미하며 소프트웨어 신뢰도를 평가할 때 이 고장의 기준을 정의하는 것이 매우 중요한 의미를 갖게 된다.

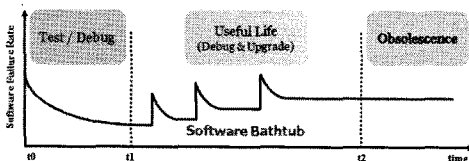


그림 2. 수명주기에 따른 소프트웨어 고장률의 변화

## 2.2 소프트웨어 신뢰도 할당

소프트웨어 신뢰도 할당(Reliability Allocation)은 사용자 요구 분석을 통해 식별된 소프트웨어의 목표 신뢰도를 개발 초기에 각각의 소프트웨어 형상 항목(CSCI: Computer Software Configuration Item)에 할당하는 활동이다.

하드웨어의 경우 일반적으로 구성품(Components) 단위까지 신뢰도를 할당하여 목표 신뢰도를 관리하는데, 소프트웨어는 CSCI를 기준으로 목표 신뢰도를 할당하여 관리한다. 일반적으로 소프트웨어의 신뢰도 목표값은 CSC(Computer Software Component)

또는 CSU(Computer Software Unit)에 할당하지 않는데, 신뢰도 예측하거나 평가할 때 CSCI를 기준으로 하기 때문이다.

소프트웨어 신뢰도 할당에 대한 연구는 소프트웨어의 구조, 이용률, 비용 등에 기반하여 소프트웨어 신뢰도를 할당하는 기법<sup>[11]</sup>이 제시된 이후 다중 사용자를 고려한 모듈러 소프트웨어의 신뢰도 할당 기법<sup>[12]</sup>, 불확실한 Operational Profile 하의 소프트웨어 신뢰도 할당<sup>[13]</sup>, 비용통제 기반의 소프트웨어 신뢰도 할당<sup>[14]</sup>, 다계층 구조에서 신뢰도 할당 및 분석<sup>[15]</sup> 등 1990년대 이후 많은 연구가 이루어졌다.

실무에 적용 가능한 신뢰도 할당 기법<sup>[20]</sup>은 표 1에 정리되어 있다.

표 1. 소프트웨어 신뢰도 할당 기법

	Allocation Methods	Characteristics
1	Equal apportionment applied to sequential software components	Use early in the SW development process when the software components are executed sequentially
2	Equal apportionment applied to concurrent software components	Use early in the SW development process and the software components are executed concurrently
3	Mission or Operational Profile allocation	Use when the operational profile of the CSCIs are known
4	Allocation based on operational criticality factors	Use when the operational criticality characteristics of the software is known
5	Allocation based on complexity factors	Use when the complexity factors of the software components are known
6	Allocation based on achievable failure rates	Use when CSCI utilization varies significantly
7	Re-allocation based on predicted failure rates	Use to re-balance the SW reliability allocations

## 2.3 소프트웨어 신뢰도 예측

소프트웨어 신뢰도 예측(Reliability Prediction)은 개발 초기 단계부터 소프트웨어 단위 시험이 수행되기 전까지 소프트웨어에 대한 경험적 또는 과거의 자료를 기반으로 신뢰도를 예측하는 방법이다.

소프트웨어의 위기 이후 많은 학자들이 예측 기법을 연구하였는데, 주로 소프트웨어의 특징(Texture)을 기반으로 신뢰도 예측을 시도하였다.

McCabe는 프로그램의 제어 흐름에 기반하여 소프트웨어의 논리적 복잡도를 예측<sup>[8]</sup>하였고, Halstead는 연산자와 피연산자의 개수 그리고 프로그램의 크기를 이용하여 프로그램의 결함수를 예측<sup>[9]</sup>하였다.

1992년 미국의 Rome Lab에서 소프트웨어 신뢰도 예측을 위한 RL-TR-92-52<sup>[10]</sup> 모델을 제시하였는데, 소프트웨어 개발 초기의 제안 시점부터 구현이 종료되는 시점까지 개발 순기에 따라 확보할 수 있는 데이터를 이용하여 소프트웨어의 결함 밀도를 예측할 수 있다. 이 모델은 내장형 소프트웨어의 신뢰도 예측에 최근까지도 많이 활용되고 있다.

개발 초기에 적용할 수 있는 소프트웨어 신뢰도 예측에 효과적인 모델로 Industry Data Collection 모델<sup>[21]</sup>이 있다. 이 모델은 개발기관 또는 부서의 CMM(Capability Maturity Model) 수준과 개발 대상 소프트웨어의 기능 점수(Function Point)를 결합한 모델로서, 개발부서의 CMM 수준에 따라 결함의 발생 정도가 통계적으로 유의하다는 점을 이용한 기법이다. 과거의 자료를 기반으로 정의된 CMM 수준에 따른 기능점수별 평균 결함 수를 이용하여 개발하고자 하는 소프트웨어의 결함 개수를 예측하는 모델이다.

#### 2.4 소프트웨어 신뢰도 평가

소프트웨어 신뢰도 평가(Reliability Estimation)는 소프트웨어 시험을 통해 수집되는 데이터를 분석하여 소프트웨어의 신뢰도를 평가하는 활동이다. 시험을 통해 수집되는 고장 데이터의 추세 분석을 통하여 소프트웨어의 신뢰도를 추정한다.

시험과정을 통해 수집되는 데이터는 Time Domain Data와 Interval Domain Data로 구분할 수 있는데, 필요에 따라 상호 호환하여 분석 가능하다. 소프트웨어 신뢰도 평가 모델은 비추계적(Non-stochastic) 모델과 추계적(Stochastic) 모델로 구분할 수 있다. 비추계적 모델로 결함 삽입을 통해 전체 결함을 추정하는 Mills의 Error Seeding 모델<sup>[4]</sup>과 결함 노출률을 이용하여 신뢰도 성장을 평가하는 Rome Lab의 RL-TR-92-15 모델<sup>[5]</sup>이 있다.

추계적 모델은 소프트웨어 신뢰도 분야에서 가장 많은 연구가 이루어진 분야로서, 1972년 Jelinski와 Moranda가 소프트웨어 내의 총 결함수를 추정하는 모델<sup>[6]</sup>을 제시한 이후, 다양한 형태의 모델이 연구

되어 제시되었다. 1979년 Goel과 Okumoto의 NHPP (Non-Homogeneous Poisson Process) 기반의 신뢰도 평가의 기반 모델<sup>[7]</sup>이 개발되었고 이것의 개선 모델<sup>[22,23]</sup>이 제시되었다. IEEE Std 1633<sup>[24]</sup>에 적용 가능한 여러 신뢰도 평가 모델이 제시되어 있다.

일반적인 소프트웨어 신뢰도 평가 절차는 고장 정의로부터 출발하여 시험데이터를 수집하고, 선정된 모델의 모수를 선정하고, 모델을 검증한 뒤, 신뢰도를 평가하는 단계로 수행된다.

#### 2.5 소프트웨어 신뢰성 공학 활동

1970년대 소프트웨어 신뢰성에 대한 연구가 시작되어 신뢰도 예측, 평가, 할당에 대한 다양한 기법이 제시된 이후, 이를 시스템 개발에서 공학적인 활동으로 정립하기 위한 노력이 지속되었다. 미군에서는 1998년 MIL-HDBK-338B<sup>[16]</sup>에 이와 같은 소프트웨어 신뢰성 공학 활동을 하드웨어의 신뢰성 공학 활동과 같이 정의하였으며, 2005년 Rome Lab을 위해 발간된 System and Software Reliability Assurance Notebook<sup>[25]</sup>에서 이러한 공학적 활동을 정의하였다.

MIL-HDBK-338B에서는 소프트웨어 신뢰도 예측, 평가 및 할당 활동을 제시하고 있는데, 다섯 종류의 신뢰도 예측 모델과 네 종류의 신뢰도 평가 모델, 그리고 네 종류의 신뢰도 할당 모델을 제시하고 있다. System and Software Reliability Assurance Notebook에서는 소프트웨어 신뢰성 공학 활동을 하드웨어와 동등한 관점에서 식별하고 있는데, 소프트웨어 신뢰도 모델링, 할당, 예측, 성장과 데모시험 등의 활동을 제시하고 있다.

#### 2.6 소프트웨어 신뢰성 공학 프로세스

1990년 후반에 소프트웨어 신뢰성 공학 활동이 정의되었고 소프트웨어 신뢰성 공학 프로세스의 개념이 2000년 이후 제시되었다. 신뢰성 공학 활동이 소프트웨어 개발에 부분적으로 적용되어 목표 신뢰도를 평가하고, 그에 따라 적절한 소프트웨어의 배포 시기 등을 예측하게 되면서 소프트웨어 신뢰성 공학 프로세스가 소프트웨어 개발의 한 영역으로 자리 잡게 되었다. Norman F. Schneidewind는 소프트웨어 신뢰성 공학 프로세스(SREP: Software Reliability Engineering Process)를 다음과 같이 정의하고 있다.

*“A life-cycle approach to Software Reliability Engineering that takes into account all aspects of a quantitative approach to achieving high-quality*

software”

SREP은 소프트웨어 개발 순기에 걸쳐 소프트웨어의 목표 신뢰도를 확보하기 위하여 수행하는 전반적인 공학적 활동으로 정의할 수 있다. 2007년 ICSE(International Conference on Software Engineering)에서 The Chinese University of Hong Kong의 Michael R. Lyu 교수는 Software Reliability Engineering: A Roadmap<sup>9)</sup>에서 그림 3과 같은 SREP를 제시하였다.

Lyu가 제시한 SREP는 주로 소프트웨어 평가 및 운용을 통한 신뢰도 검증에 중점을 두고 있다. 초기에 설정한 목표 신뢰도를 시험을 이용하여 확보하고 운용을 통하여 검증하는 과정으로 정의하고 있다.

2008년 배포된 IEEE Std 1633<sup>1)</sup>은 소프트웨어

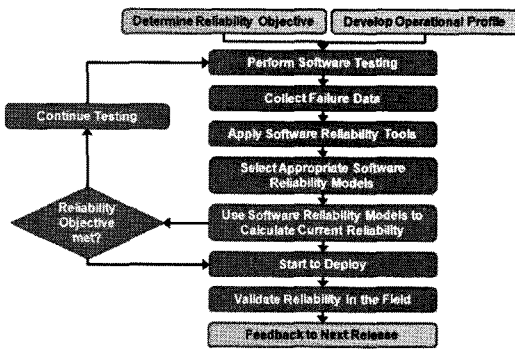


그림 3. Michael R. Lyu가 제시하고 있는 SREP

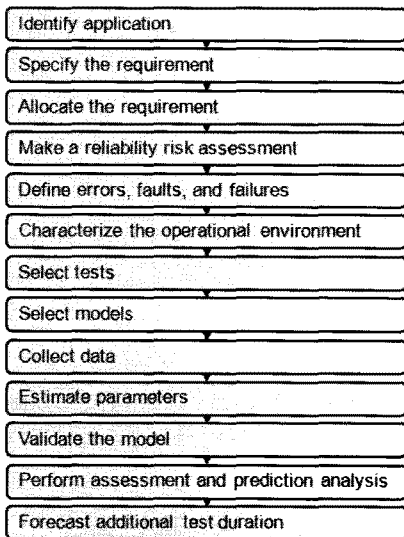


그림 4. IEEE Std 1633의 SR Procedure

신뢰성 절차(Software Reliability Procedure)를 제시하고 있는데, 소프트웨어의 전 개발 순기에 걸쳐 적용할 수준의 신뢰성 공학 활동을 상세히 정의하고 있다. Lyu가 제시하고 있는 SREP와 비교할 때, 신뢰도 할당, 리스크 평가, 고장 정의 등의 활동이 추가로 식별되어 있다

2.7 문제 정의 및 연구 목표

무기체계의 연구개발 과정에서 점차 소프트웨어의 비중이 커지고 그 중요성이 증대되면서 소프트웨어의 개발 절차와 기술 문서를 정형화하기 위한 노력이 지속되어 왔다. 방위사업청에서는 체계개발 단계의 내장형 소프트웨어 개발관리 업무 편람<sup>126)</sup> 등을 제정하여 소프트웨어의 개발절차, 개발단계별 주요 활동 점검목록, 소프트웨어 기술문서의 작성방법 등을 제시하고 있다.

국내 무기체계의 연구개발 과정에서 소프트웨어에 대한 신뢰성 관련 공학적 활동은 일부 소프트웨어에 대해 신뢰도 평가를 수행<sup>127,28)</sup>하는 수준으로 무기체계 소프트웨어의 목표 신뢰도를 확보하기 위한 공학적 프로세스는 없는 상태이다. 그러나 연구개발 기간 내에 소프트웨어의 목표 신뢰도 확보를 위해서는 개발 순기에 적용할 체계적인 소프트웨어 신뢰성 공학 프로세스가 요구된다.

Lyu가 제시한 SREP는 주로 소프트웨어 평가 및 운용을 통한 신뢰도 검증에 중점을 두고 있으며, 초기 신뢰도 할당, 할당을 통한 설계 및 구현 과정에서 소프트웨어의 신뢰도 확보 활동에는 다소 미흡한 부분이 있다. 또한 운용을 통해 획득되는 데이터 환류를 통해 목표 소프트웨어의 신뢰도 확보를 제시하고 있다. 그런데, 업그레이드가 용이한 상용 소프트웨어와는 달리 무기체계 소프트웨어의 경우 개발 이후 운용 중에 소프트웨어의 수정 및 형상변경 그리고 배포가 쉽지 않아서 개발 과정에 소프트웨어의 목표 신뢰도를 확보할 수 있도록 최대한 공학적 기법을 적용하는 것이 필요하다.

IEEE Std 1633에서 정의하고 있는 소프트웨어 신뢰성 절차는 개발 전 순기에 걸쳐 적용 가능하도록 목표신뢰도 설정, 신뢰도 할당, 신뢰도 평가 등의 활동이 정의되어 있으나, 목표 신뢰도를 확보하기 위하여 개발 초기 단계에 신뢰도 예측 및 이를 통한 신뢰성 개선과 같은 공학적 활동이 부족하다

이와 같은 문제점을 해결하기 위하여 본 논문에서는 연구개발 중인 무기체계에 적합한 소프트웨어 신뢰성 공학 프로세스를 제안한다. 무기체계 소프트

웨어 신뢰성 공학 프로세스(SREP-WS)를 정의하기 위하여 다음과 같은 절차로 수행하였다.

먼저, 기존의 식별된 소프트웨어 신뢰성 공학 활동을 기반으로 여러 프로세스 관점에서 소프트웨어 신뢰성 공학 활동을 식별하여 소프트웨어 신뢰성 공학 활동을 도출하였다. SREA 도출은 시스템 개발 수명주기 관점, 소프트웨어 개발 수명주기 관점, 하드웨어 신뢰도 프로그램 관점, RAM<sup>1)</sup> 분석 프로세스 관점에서 이루어졌다. 이와 같이 도출된 SREA를 분석, 종합하여 무기체계 소프트웨어 신뢰성 공학 프로세스(SREP-WS)를 정의하였고, 선행 연구를 통해 식별한 각 신뢰성 공학 활동의 세부 활동과 관련된 모델 그리고 입력력 요소를 SREA로 정의하였다.

### III. 무기체계 소프트웨어 신뢰성 공학 활동

SREP-WS를 정의하기 위하여 먼저 다양한 관점에서 SREA를 식별하였다. 시스템 개발 수명주기 관점, 소프트웨어 개발 수명주기 관점, 하드웨어 신뢰도 프로그램 관점, RAM 분석 프로세스 관점에서 SREA를 도출하였다.

#### 3.1 시스템 개발 수명주기 관점에서 SREA

SREA 식별을 위하여 시스템 개발 수명주기 관점에서 두가지 모델을 고려하였다. 시스템 개발의 프로세스의 기본 모델이 되는 Vee 모델과 시스템 수명주기 프로세스를 정의하는 IEEE Std 15288<sup>[29]</sup> 관점에서 SREA를 검토하였다.

Vee 모델은 사용자 요구사항 분석으로부터 사용자 요구사항을 검증하기까지 9단계로 시스템 개발 활동을 정의하고 있는 시스템 개발 모델이다. 표 2에 Vee 모델의 시스템 개발 활동과 관련된 SREA가 식별되어 있다.

IEEE Std 15288은 시스템 수명주기 동안 요구되는 프로세스를 정의하고 있다. 시스템 수명주기 프로세스에는 Agreement Process, Enterprise Process, Project Process, Technical Process를 포함하고 있는데, 시스템 개발 프로세스와 관련이 깊은 Technical Process 관점에서 SREA를 식별하여 표 3에 정리하였다.

표 2. Vee 모델 관점에서 SREA 식별

	시스템 개발 활동	SREA
1	Understand User Requirements and Develop System Concept & Validation Plan	-
2	Develop System Performance Spec & Verification Plan	-
3	Expand Performance Spec into CI design-to Spec & CI Verification Plan	Operational Profile 정의 SW 신뢰도 할당
4	Evolve Design-to Spec into Build-to Documentation & Inspection Plan	SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
5	Fabricate, Assemble and Code to Build-to Documentation	SW 신뢰도 예측 SW 신뢰도 개선
6	Inspect to Build-to Documentation	SW 신뢰도 개선
7	Assemble CIs and Perform CI Verification to CI Design-to Spec	SW 신뢰도 평가
8	Integrate System and Perform System Verification to Performance Spec	SW 신뢰도 평가
9	Demonstrate and Validate System to User Validation Plan	SW 신뢰도 평가

표 3. IEEE Std 15288 관점에서 SREA 식별

Process	Sub-processes	SREA
Technical processes	Stakeholder Requirements Definition Process	-
	Requirements Analysis Process	Operational Profile 정의
	Architectural Design Process	SW 신뢰도 할당 SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
	Implementation Process	SW 신뢰도 예측 SW 신뢰도 재할당
	Integration Process	SW 신뢰도 평가
	Verification Process	SW 신뢰도 평가
	Transition Process	SW 신뢰도 평가
	Validation Process	SW 신뢰도 평가
	Operation Process	-
	Maintenance Process	-
	Disposal Process	-

1) RAM: Reliability, Availability and Maintainability, 신뢰도 가용도 정비도

3.2 소프트웨어 개발 수명주기 관점에서 SREA

소프트웨어의 개발 수명주기 관점에서 SREA를 식별하기 위하여 ISO/IEC 12207<sup>[30]</sup>과 방위사업청의 체계개발단계 내장형 소프트웨어 개발관리 업무 편람을 적용하였다.

ISO/IEC 12207은 소프트웨어 수명주기 프로세스

표 4. ISO/IEC 12207 관점에서 SREA 식별

Process	Activities	SREA
Development Process	Process Implementation	-
	System Requirements Analysis	-
	System Architectural Design	Operational Profile 정의
	Software Requirements Analysis	SW 신뢰도 할당
	Software Architectural Design	SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
	Software Detailed Design	SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
	Software Coding And Testing	SW 신뢰도 예측
	Software Integration	SW 신뢰도 평가
	Software Qualification Testing	SW 신뢰도 평가
	System Integration	SW 신뢰도 평가
	System Qualification Testing	SW 신뢰도 평가
	Software Installation	-
	Software Acceptance Support	-

표 5. 내장형 SW 개발관리 업무 편람 관점에서 SREA 식별

	소프트웨어 개발 활동	SREA
1	소프트웨어 개발계획	Operational Profile 정의
2	소프트웨어 요구사항분석	SW 신뢰도 할당 SW 신뢰도 예측
3	소프트웨어 기본설계	SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
4	소프트웨어 상세설계	SW 신뢰도 예측 SW 신뢰도 개선 SW 신뢰도 재할당
5	소프트웨어 코딩 및 단위 시험	SW 신뢰도 예측
6	소프트웨어 통합	SW 신뢰도 평가
7	소프트웨어 개발시험	SW 신뢰도 평가

를 정의하고 있는 표준으로 Primary Life Cycle Processes, Supporting Processes, Organizational Life Cycle Processes 등 세 가지 프로세스를 정의하고 있다. 이 중에서 소프트웨어 개발 활동과 연관성이 깊은 Primary Processes의 Development Process의 활동을 기반으로 SREA를 식별하였다.

내장형 소프트웨어 개발관리 업무 편람은 2008년부터 국내 연구개발 무기체계의 내장 소프트웨어 개발에 적용되는 기준으로 표 5에 제시된 개발 활동을 식별하고 있으며 관련 SREA는 다음과 같다.

3.3 하드웨어 신뢰도 프로그램 관점에서 SREA

국내 무기체계 연구개발시 MIL-STD-785B 표준을 하드웨어 신뢰도 프로그램으로 적용하고 있는데,

표 6. MIL-STD-785B 관점에서 SREA 식별

Tasks	SREA
Task 100 : Program Surveillance and Control	
Task 101 : Reliability Program Plan	-
Task 102 : Monitor/ Control of Subcontractors and Suppliers	-
Task 103 : Program Reviews	-
Task 104 : FRACAS	-
Task 105 : Failure Review Board	-
Task 200 : Design and Evaluation	
Task 201 : Reliability Modeling	SW 신뢰도 구조 모델링
Task 202 : Reliability Allocations	SW 신뢰도 할당
Task 203 : Reliability Predictions	SW 신뢰도 예측
Task 204 : FMECA	SW 신뢰도 개선
Task 205 : Sneak Circuit Analysis	SW 신뢰도 개선
Task 206 : Electronic Parts/Circuits Tolerance Analysis	-
Task 207 : Parts Program	-
Task 208 : Reliability Critical Items	-
Task 209 : Effects of Functional Testing, Storage, Handling, Packing, Transportation, and Maintenance	-
Task 300 : Development and Production Testing	
Task 301 : Environmental Stress Screening (ESS)	-
Task 302 : Reliability Development/Growth Test (RDGT) Program	SW 신뢰도 평가
Task 303 : Reliability Qualification Test (RQT) Program	SW 신뢰도 평가
Task 304 : Production Reliability Acceptance Test (PRAT) Program	-

이 표준에서 정의하고 있는 활동(Task)을 비교하여 SREA를 식별할 수 있다. MIL-STD-785B는 세계의 주요 활동을 식별하고 있는데, Task 100을 제외한 Task 200과 Task 300을 SREA 식별에 적용할 수 있다. 분석 결과 식별된 SREA가 표 6에 제시되어 있다. MIL-STD-785B의 관점에서는 개발 과정에 요구되는 신뢰도 개선 활동이 식별되었으며, 이는 개발 과정에서 신뢰도 개선을 위한 노력이 필요함을 보여준다.

### 3.4 RAM 분석 업무 프로세스 관점에서 SREA

사용자 관점에서 무기체계의 가동률은 개발자 관점에서 RAM 인수(Parameter)로 구체화된다. RAM 분석 업무는 무기체계 개발과정에서 RAM 인수를 정의하고, 예측하고, 평가해 가는 과정으로 신뢰도는 RAM 인수의 일부분을 차지한다.

방위사업청은 종합군수지원 개발 실무지침서<sup>[31]</sup>를 통하여 표 7과 같은 RAM 분석 활동을 제시하고 있는데, 이 활동 관점에서 SREA를 식별하면 표 8과 같다. 신뢰도 목표값 설정, 신뢰성 구조 모델링, 고장 정의 등이 다른 관점과 유의한 활동으로 식별되었다.

표 7. RAM 분석 업무 프로세스 관점에서 SREA 식별

	RAM 분석 활동	SREA
1	OMS/MP	Operational Profile 정의
2	RAM 목표값 설정	SW 신뢰도 목표값 설정
3	신뢰성 구조 모델링	SW 신뢰성 구조 모델링
4	RAM 할당	SW 신뢰도 할당
5	RAM 예측	SW 신뢰도 예측
6	고장정의/판단기준 설정	SW 고장 정의
7	FMECA/FTA	SW 신뢰도 개선
8	RAM 시험평가	SW 신뢰도 평가

## IV. 무기체계 소프트웨어 신뢰성 공학 프로세스

### 4.1 무기체계 소프트웨어 신뢰성 공학 프로세스

무기체계의 소프트웨어 신뢰성 공학 프로세스는 다양한 개발 프로세스 관점에서 식별된 무기체계의 소프트웨어 신뢰성 공학 활동을 기반으로 정의되었다. 표 8에 제시된 바와 같이 먼저, 기 식별된 SREA를 바탕으로 다섯 개의 주요 활동을 식별하였다.

SW 신뢰성 요구분석 활동은 신뢰도 할당, 예측, 평가에 소요되는 기초 데이터를 준비하는 활동으로

표 8. 무기체계의 소프트웨어 신뢰성 공학 활동 식별

Activity	무기체계 SREA	SREA
100	SW 신뢰성 요구분석	Operational Profile 정의
		SW 신뢰도 목표값 설정
		SW 신뢰성 구조 모델링
		SW 고장 정의
200	SW 신뢰도 할당	SW 신뢰도 할당
300	SW 신뢰도 예측	SW 신뢰도 예측
400	SW 신뢰도 개선	SW 신뢰도 개선
		SW 신뢰도 재할당
500	SW 신뢰도 평가	SW 신뢰도 평가

네 개의 하부 활동을 갖는다. SW 신뢰도 할당 활동은 목표 신뢰도를 CSCI에 합리적으로 배분하는 활동이며, SW 신뢰도 예측 활동은 CSCI에 할당된 신뢰도가 달성 가능한지를 확인하는 활동이다. SW 신뢰도를 예측한 결과 신뢰도 목표값이 만족되지 않을 경우 SW 신뢰도 개선 활동을 수행한다. 소프트웨어 개발 단계가 시험평가 단계에 이르면 SW 신뢰도 평가를 수행한다. 시험평가를 통해 획득한 데이터를 이용하여 SW 신뢰도를 평가하고 목표 신뢰도가 확보될 때까지 시험평가를 진행하도록 기준을 제시한다. 그림 5는 SREP-WS의 흐름을 보여준다.

SREP-WS를 구성하고 있는 각각의 활동에 대해 관련 모델 및 기법, 입출력 데이터 그리고 하부 활동에 대하여 정의한다.

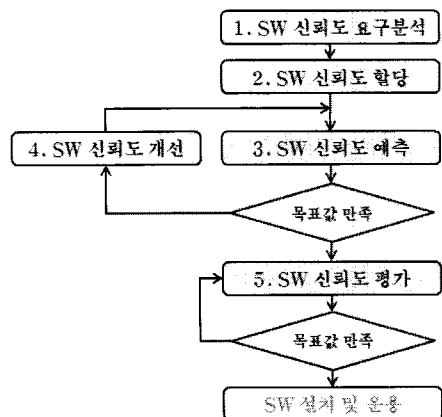


그림 5. 무기체계 소프트웨어 신뢰성 공학 프로세스

### Activity 100 SW 신뢰성 요구분석 활동

SW 신뢰성 요구분석 활동은 신뢰도 할당, 예측, 개선 및 평가에 활용되는 기초 데이터를 준비하는



활동으로 네 개의 하부 활동으로 구성된다.

- Activity 101 Operational Profile 정의
- Activity 102 SW 신뢰도 목표값 설정
- Activity 103 SW 신뢰성 구조 모델링
- Activity 104 SW 고장 정의

이 활동의 입력은 OMS/MP(Operational Mode Summary/Mission Profile), 소프트웨어 아키텍처 및 시스템의 기능분석 자료이고, 출력은 Operational Profile, SW 목표 신뢰도, SW 신뢰성 구조, SW 고장 정의자료이다.

Activity 101 Operational Profile 정의

Operational Profile 정의 활동은 OMS/MP로부터 Operational Profile을 정의하는 활동으로, 이 활동의 입력은 OMS/MP이고 출력은 Operational Profile 이다.

Activity 102 SW 신뢰도 목표값 설정

SW 신뢰도 목표값 설정 활동은 SW 신뢰도 할당의 기준이 되는 목표 신뢰도를 정의하는 활동이다.

소프트웨어가 무기체계의 구성품으로 개발될 때 소프트웨어의 목표 신뢰도는 시스템 신뢰도로부터 할당되어 설정된다. 그러나 소프트웨어가 시스템의 대부분을 차지할 경우 목표 신뢰도를 여러 가지 기법을 이용하여 정의할 수 있다.

초기 목표 신뢰도 설정 방법은 유사 시스템의 신뢰도를 참조하여 결정<sup>[32]</sup>하거나, OMS/MP에 따라 신뢰도를 정의<sup>[33,34]</sup>하는 방법 등이 있다.

이 활동의 입력은 OMS/MP이며, 출력은 소프트웨어의 목표 신뢰도이다.

Activity 103 SW 신뢰성 구조 모델링

SW 신뢰성 구조 모델링 활동은 목표 신뢰도를 할당하기 위한 CSCI의 구조를 신뢰도 관점에서 정의하는 활동이다. 대부분의 무기체계에 적용되는 소프트웨어는 중복구조를 갖지 않기 때문에 N-version 프로그래밍과 같이 특별한 경우를 제외하고 SW 신뢰성 구조는 직렬 구조로 정의된다.

그러나 한 소프트웨어가 여러 구성품에서 동시에 운용될 경우 Operational Profile에 해당 소프트웨어의 가중치를 조정하여 그 소프트웨어의 중요도를 식별할 수 있다.

이 활동의 입력은 소프트웨어 아키텍처이며, 출력은 SW 신뢰성 구조이다.

Activity 104 SW 고장 정의

SW 고장 정의 활동은 향후 SW 신뢰도 평가시 적용될 소프트웨어의 고장 여부를 판단하는 기준을 정의하는 것이다. SW의 고장은 SW의 기능의 결함으로 발생하는데, 이는 시스템의 기능 분석을 통하여 식별될 수 있다. 시스템에 식별된 기능 중에서 CSCI에 할당되는 기능을 기준으로, 그 기능의 결함이 해당 CSCI의 고장임을 식별하게 된다.

이 활동의 입력은 시스템 기능분석 자료이며, 출력은 소프트웨어 고장정의 자료이다.

Activity 200 SW 신뢰도 할당 활동

SW 신뢰도 할당은 목표 신뢰도를 만족하기 위하여 소프트웨어를 구성하고 있는 CSCI에 CSCI의 목표 신뢰도를 할당하는 것이다. 소프트웨어 신뢰도 할당은 CSCI의 실행방법, 실행시간 비율, 치명도, 복잡도, 등에 따라 목표 신뢰도를 할당할 수 있는데, 신뢰도 할당 기법과 할당기법은 표 1에 정리되어 있다.

이 활동의 입력은 Operational Profile, SW 목표 신뢰도, SW 신뢰성구조, CSCI의 실행방법, 실행시간 비율, 치명도, 복잡도와 같은 CSCI 속성이며, 출력은 CSCI에 할당된 목표 신뢰도이다.

Activity 300 SW 신뢰도 예측 활동

SW 신뢰도 예측 활동은 CSCI의 신뢰도를 예측하는 활동이다. 신뢰도 예측은 개발 순기의 초기 단계부터 구현 단계까지 수행될 수 있다. 소프트웨어 개발의 각 단계에서 획득되는 데이터에 따라 신뢰도를 예측할 수 있는데, 개발이 진행될수록 점차 신뢰성 있는 많은 데이터가 수집되는데 이를 이용하여 신뢰도를 점점 더 정확하게 예측하게 된다.

신뢰도 예측 모델은 II장 3절에 제시되어 있는데, 이 중에서 무기체계 신뢰도 예측에는 RL-TR-92-15 모델과 Industry Data Collection 모델을 적용할 수 있다. 신뢰도 예측 모델에 적용되는 입력 인수(Parameter)는 소프트웨어를 개발하는 기관이나 부서의 과거 데이터에 매우 중속적이기 때문에 정확한 신뢰도 예측을 위해서는 많은 양의 과거 데이터 수집과 분석이 요구된다.

이 활동의 입력은 SW 신뢰도 예측 모델, 신뢰도 예측 모델의 입력 인수이며, 출력은 CSCI 신뢰도 예측값이다.

Activity 400 SW 신뢰도 개선 활동

SW 신뢰도 개선 활동은 개발 초기 단계부터 구현단계까지 소프트웨어의 신뢰도를 개선하기 위한 활동이다. 이 활동은 CSCI의 신뢰도 예측결과가 CSCI에 할당된 신뢰도를 만족하지 못할 때, CSCI에 대해 신뢰도 예측 모델 관점에서 수행할 수 있으며, 또한, 일반적인 소프트웨어의 신뢰도 개선 측면에서 수행할 수 있다.

신뢰도 예측 모델 관점에서 신뢰도 개선 기법은 소프트웨어를 RL-TR-92-15 모델로 신뢰도 예측한 경우, 개발환경과 개발 언어, 모듈화, 복잡도와 같은 소프트웨어의 특성을 변경하여 신뢰도를 개선할 수 있다. 그리고 Industry Data Collection 모델을 적용한 경우 개발조직의 CMM 수준이나 CSCI의 기능 점수를 조정하여 신뢰도를 향상시킬 수 있다.

일반적인 관점에서 신뢰도 개선 기법은 설계 단계에서 FMEA(Failure Mode and Effects Analysis)<sup>[35]</sup>, FTA(Fault Tree Anlysis)<sup>[36]</sup> 기법 등이 적용, 구현 단계에서 MISRA<sup>[37]</sup>에서 제시하는 코딩 표준의 적용, 시험 단계에서 DO-178B/ED-12B에 정의된 시험 커버리지(Test Coverage)<sup>[38]</sup>의 적용 등이 고려될 수 있다. 개선활동을 수행한 CSCI에 신뢰도가 개선 되었으면 목표 신뢰도를 고려하여 CSCI에 신뢰도를 재할당한다.

이 활동의 입력은 CSCI의 신뢰도 예측 모델의 입력 인수, 할당된 목표 신뢰도, 신뢰도 예측값이며, 출력은 신뢰도가 개선된 CSCI에 재할당된 신뢰도 목표값이다.

Activity 500 SW 신뢰도 평가 활동

SW 신뢰도 평가 활동은 시험평가 단계에서 시험을 통해 수집되는 데이터를 이용하여 소프트웨어의 신뢰도를 평가하는 활동이다. 신뢰도 평가는 시험단계 및 운용단계에 수집되는 데이터를 이용하여 수행할 수 있는데, 무기체계의 경우 운용단계에 수집되는 데이터를 이용한 신뢰도 평가가 쉽지 않기 때문에 시험평가 단계인 개발시험 및 운용시험시 수집되는 데이터를 이용하여 신뢰도를 평가할 수 있다.

신뢰도 평가는 그림 6에 제시된 절차에 따라 수행되며 II장 4절에 제시된 다양한 종류의 신뢰도 평가 모델이 신뢰도 평가에 적용 가능하다. 신뢰도 평가에는 반드시 신뢰도 평가 모델이 개발되는 소프트웨어에 적합한지 검증한 뒤 신뢰도 평가에 이용하여야 한다.

이 활동의 입력은 SW 고장 정의자료, SW 신뢰

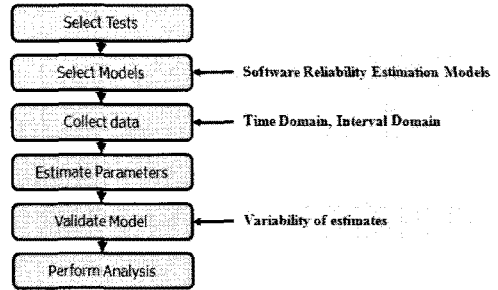


그림 6. 무기체계 소프트웨어 신뢰도 평가 절차

도 평가 모델, 신뢰도 평가를 위해 수집된 고장관련 데이터이며 출력은 신뢰도 평가값이다.

V. SREP-WS 평가

제안된 무기체계의 소프트웨어 신뢰성 공학 프로세스를 평가하기 위하여 세 가지 방법을 적용하였다.

먼저, SREP-WS의 논리적 무결성을 검증하기 위하여 시스템 공학 모델링 도구를 이용하여 모델링을 수행함으로써, 프로세스를 구성하는 활동의 타당성과 활동간 입출력 연관성을 검증하였다. 두 번째로 무기체계 연구개발에 적용 기준인 국방과학연구소의 연구개발 프로세스 표준화 규정<sup>[39]</sup>의 연구개발 프로세스 내에서 SREP-WS가 적용할 때 적합한지를 검토하여 무기체계 개발시 SREP-WS의 적용의 타당성을 확인하였다. 끝으로, 제안된 SREP-WS와 기존 SREP를 비교 검토함으로써 제안된 SREP-WS가 기존의 SREP 보다 무기체계 소프트웨어 신뢰성 공학에 적합함을 보였다.

5.1 모델링에 의한 SREP-WS 평가

SREP-WS의 모델링에 활용된 체계공학 모델링 기법은 IDEF0로서 프로세스를 구성하는 활동을 명료하게 표현하고 활동 간의 입출력을 명확히 식별할 수 있는 장점이 있다. 모델링의 전산지원도구로 Vitech사의 CORE 7.0을 활용하였다.

그림 7은 CORE를 이용하여 모델링된 SREP-WS를 보여준다. 프로세스를 구성하는 각각의 활동들과 활동의 입출력이 일관성 있게 구성되어 있어 프로세스가 논리적임을 알 수 있다.

5.2 SREP-WS를 무기체계 연구개발에 적용성 평가

국방과학연구소의 연구개발 프로세스 표준화 규정은 무기체계 개발단계의 프로세스에 대해 정의하고 있다. 이 규정에 정의된 주요 개발 활동은 운용

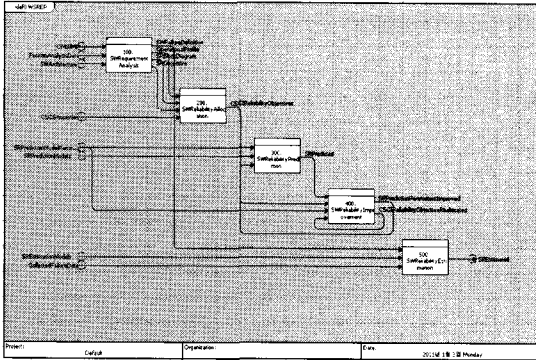


그림 7. 무기체계 소프트웨어 신뢰성 공학 프로세스 모델링

개념 도출, 사용자 요구사항 개발, 체계 요구사항 개발, 체계기능 분석, 기본설계, 상세설계, 제작 및 구현, 체계통합, 시험평가 및 규격화인데 그림 8에서 제시하듯이 SREP-WS의 활동이 규정의 개발 단계에 잘 부합되어 각각의 단계에 적합한 활동을 수행할 수 있다. 무기체계 소프트웨어 개발에 SREP-WS를 적용함으로써 무기체계 개발순기에 소프트웨어 신뢰도를 정량적으로 관리할 수 있으며 소프트웨어의 목표 신뢰도를 개발기간 중에 확보할 수 있다.

SREP-WS는 무기체계 연구개발 프로세스 내에서 중요한 공학적 활동으로, 무기체계 개발 과정의 다른 공학적 활동과 융합되어 신뢰성 있는 소프트웨어의 개발을 보증할 수 있는 공학적 활동으로 수행될 수 있다.

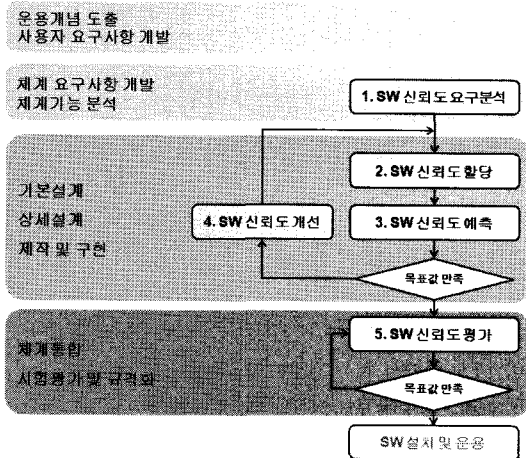


그림 8. 무기체계 개발활동과 SREP-WS 관련성

5.3 SREP-WS를 기존의 SREP와 비교 평가

SREP-WS를 기존의 SREP과 비교를 통하여 SREP-WS를 평가한다. SREP-WS와 비교 대상 SREP

로 Lyu의 SREP와 IEEE Std 1633의 SR Procedure를 선정하였다.

5.3.1 SREP-WS와 Lyu의 SREP의 비교

Lyu가 제시한 SREP는 신뢰도 평가에 초점을 두어 개발 초기에 신뢰도 목표값 설정과 Operational Profile 정의를 하고 시험평가 단계 이후에 신뢰도 평가에 중점을 두고 있다. 그러나 설계 및 구현과정에 신뢰성 확보 관련한 활동은 정의되어 있지 않다.

무기체계의 경우 양산 배치되어 운용 단계에 접어들면 신뢰도 평가를 위한 고장 데이터의 수집이 어렵고, 소프트웨어의 변경과 그에 따른 형상변경 그리고 배포가 쉽지 않은 점을 고려하면, 개발기간 중에 목표 신뢰도를 확보하기 위하여 많은 노력을 기울여야만 한다.

이러한 관점에서 표 10에 제시된 비교 내용을 보면 SREP-WS는 Lyu의 SREP보다 무기체계 소프트웨어 개발에 보다 적합한 프로세스라고 할 수 있다.

표 9. SREP-WS와 Lyu의 SREP의 비교

SREP-WS		Lyu의 SREP
100 SW 신뢰도 요구분석	101 Operational Profile 정의	Develop Operational Profile
	102 SW 신뢰도 목표값 설정	Determine Reliability Objective
	103 SW 신뢰성 구조 모델링	-
	104 SW 고장 정의	-
200 SW 신뢰도 할당		-
300 SW 신뢰도 예측		-
400 SW 신뢰도 개선		-
500 SW 신뢰도 평가	Perform Software Testing	
	Collect Failure Data	
	Apply Software Reliability Tools	
	Select Appropriate Software Reliability Models	
		Use Software Reliability Models to Calculate Current Reliability
		Start to Deploy
		Validate Reliability in the Field

5.3.2 SREP-WS와 IEEE Std 1633의 SR Procedure의 비교

IEEE Std 1633의 SR Procedure는 소프트웨어

개발시 목표 신뢰도 확보를 위한 절차를 비교적 구체적으로 식별하고 있다. 그러나 소프트웨어에 할당된 신뢰도가 설계 및 구현 단계에서 확보 가능한지의 예측과 그 결과에 따른 소프트웨어의 개선 노력 그리고 합리적인 수준에서 신뢰도의 재할당 등의 활동이 부족하다.

개발기간 중에 목표 신뢰도 확보가 반드시 필요한 무기체계 소프트웨어의 특성상 개발 초기 및 중기에 목표 신뢰도 확보를 위한 노력이 요구되며 이러한 활동이 소프트웨어 신뢰성 공학 프로세스에 포함되어야 한다. 이와 같은 관점에서 표 10에 제시된 SREP-WS와 1633 SR Procedure의 비교자료를 통하여 SREP-WS가 1633 SR Procedure보다 무기체계 소프트웨어의 신뢰성 공학에 적합한 프로세스임을 알 수 있다.

표 10. SREP-WS와 IEEE Std 1633 SR Procedure의 비교

SREP-WS		IEEE Std 1633 SR Procedure
100 SW 신뢰도 요구분석	101 Operational Profile 정의	a) Identify application
	102 SW 신뢰도 목표값 설정	b) Specify the requirement
	103 SW 신뢰성 구조 모델링	a) Identify application
	104 SW 고장 정의	e) Define errors, faults, and failures
200 SW 신뢰도 할당		c) Allocate the requirement
-		d) Make a reliability risk assessment
300 SW 신뢰도 예측		-
400 SW 신뢰도 개선		-
500 SW 신뢰도 평가		f) Characterize the operational environment
		g) Select tests
		h) Select models
		i) Collect data
		j) Estimate parameters
		k) Validate the model
		l) Perform assessment and prediction analysis
-		m) Forecast additional test duration

#### 5.4 평가 결과에 대한 고찰

논문에서 제안한 SREP-WS를 논리적 무결성, 무기체계 연구개발에 적용성, 기존 SREP와 비교를 통

한 개선점 검토 등 세가지 관점에서 평가하였다. 첫 번째 평가 결과 프로세스를 구성하는 활동들과 하부 활동들 간의 수행 순서 및 입출력 자료간의 연계성 검토를 통하여 SREP-WS가 프로세스 관점에서 논리적으로 오류가 없음이 확인되었다.

두 번째로 SREP-WS가 무기체계 연구개발 과정에 적용될 때의 적절성을 검토하였다. SREP-WS는 연구개발 전 순기에 걸쳐 적용하도록 개발하였는데, 그림 8을 살펴보면 체계 요구사항 개발 단계부터 시험평가 및 규격화 단계까지 SREP-WS의 활동이 연구개발 프로세스 표준화 규정의 개발 활동과 잘 부합되어 각각의 개발 활동에 적합한 SREA를 수행할 수 있음을 알 수 있다.

기존 SREP를 무기체계 소프트웨어 개발에 적용할 때 식별된 문제점을 SREP-WS를 이용하여 해결할 수 있는가를 세 번째 평가를 통해서 확인하였다. Lyu의 SREP 단점으로 식별된 설계 및 구현과정에서 신뢰성 확보와 관련되어 누락되었던 신뢰도 할당, 예측 및 개선과 같은 활동을 SREP-WS를 통해 수행할 수 있다. 그리고 IEEE Std 1633의 SR Procedure에서 부족했던 목표 신뢰도가 설계 및 구현 단계에서 확보 가능한지의 예측과 그 결과에 따른 소프트웨어의 신뢰도 개선 및 재할당 등의 활동을 SREP-WS를 통해 수행 가능하다.

SREP-WS의 평가 결과 무기체계 소프트웨어의 신뢰성 공학 업무에 적합한 프로세스로 판단되었다.

## VI. 결 론

첨단 복합 무기체계 형태로 진화하는 최근의 무기체계의 발전 동향을 고려할 때, 향후 무기체계에서 소프트웨어가 차지하는 비중과 중요성은 앞으로 점점 증가될 것이다. 이와 같은 상황에서 신뢰성 있는 소프트웨어를 개발하지 않고서 전장에서 신뢰성 있는 무기체계를 운용하는 것은 불가능하다. 그런데 국내의 무기체계 연구개발 과정에서 소프트웨어 신뢰성 공학 활동은 일부 소프트웨어에 대해 신뢰도 평가를 하는 수준으로, 개발 기간 내에 목표 신뢰도를 담보할 수 있는 신뢰성 공학 프로세스가 요구되는 상황이다.

본 연구에서는 그동안의 소프트웨어 신뢰성 연구 결과를 체계공학 관점에서 분석하여, 무기체계의 연구개발 전 수명주기 동안 적용할 수 있는 SREP-WS를 제시하였다. 무기체계 소프트웨어 개발에 제안된 SREP-WS를 적용함으로써, 개발순기 동

안 소프트웨어 신뢰도를 정량적으로 관리할 수 있고 소프트웨어의 목표 신뢰도를 개발기간 중에 확보할 수 있다.

신뢰도 예측의 경우 소프트웨어의 개발 기관의 경험적 데이터를 기반으로 하는데, SREP-WS에서 참조하는 신뢰도 예측 모델은 외국에서 개발된 모델로서 모델에서 제시하는 인수(Parameter) 값에서 국내의 환경과의 많은 차이가 있을 수 있다. 국내에서는 이러한 데이터의 수집이나 분석이 부족한 상태이며, 향후 신뢰도 예측 결과의 신뢰성을 담보하기 위해서 이 분야에 대한 연구와 데이터 축적이 요구된다.

신뢰도 평가의 경우 국내 무기체계 소프트웨어 개발에 신뢰도 평가를 적용 사례가 일부 있지만 이것이 소프트웨어 개발에 본격적으로 적용되고 있지는 않다. 향후 무기체계 소프트웨어 개발에 지속적인 신뢰도 평가 적용을 통하여 획득된 데이터를 기반으로 국내 무기체계 소프트웨어 연구개발 환경에 적합하도록 신뢰도 평가 모델의 개선 및 보완이 필요하다.

### 약 어 정 의

- SR: Software Reliability, 소프트웨어 신뢰성
- SRE: Software Reliability Engineering, 소프트웨어 신뢰성 공학
- SREA: Software Reliability Engineering Activity, 소프트웨어 신뢰성 공학 활동
- SREP: Software Reliability Engineering Process, 소프트웨어 신뢰성 공학 프로세스
- SREP-WS: Software Reliability Engineering Process for Weapon Systems, 무기체계의 소프트웨어 신뢰성 공학 프로세스

### 참 고 문 헌

- [1] *Reliability Program for Systems and Equipment Development and Production (MIL-STD-785B)*, DoD U.S, Sep. 1980.
- [2] *Reliability Modeling and Prediction (MIL-STD-756B)*, DoD U.S, May 1963.
- [3] *Reliability Prediction of Electronic Equipment (MIL-HDBK-217F)*, DoD U.S, Dec. 1991.
- [4] H.D. Mills "On the statistical validation of computer program," *IBM FSD*, Jul. 1970.

- [5] *Introduction to Software Reliability: A state of the Art Review*, Reliability Analysis Center, pp. 100, Dec. 1996.
- [6] Z. Jelinski, P.B. Moranda, "Software reliability research" In *Statistical Computer Performance Evaluation*, W. Freiberger (ed.), Academic Press, New York, 1972.
- [7] A.L. Goel, K. Okumoto, "Time-dependent error-detection rate model for software and other performance measures," *IEEE Transaction on Reliability* 28, 206-211, 1979.
- [8] T.J. McCabe, "A complexity measure," *IEEE Trans. Software Engineering* Vol.SE-2(4), 1976.
- [9] M.H. Halstead, *Elements of Software Science*, Elsevier, New York, 1997.
- [10] *Introduction to Software Reliability: A state of the Art Review*, Reliability Analysis Center, pp 99.
- [11] Fatemeh Zahedi and Noushin Ashrafi, "Software Reliability Allocation Based on Structure, Utility, Price, and Cost," *IEEE transactions on software engineering*, Vol.17, No.4, pp.345-356, 1991.
- [12] Y.W. Leung, "Optimal Reliability Allocation for Modular Software System Designed for Multiple Customers," *IEICE transactions on information and systems*, Vol.79, No.12, pp.1655-1662, 1996.
- [13] Y.W. Leung, "Software reliability allocation under an uncertain operational profile," *The journal of the Operational Research Society*, Vol.48, No.4, pp.401-411, 1997.
- [14] C. Huang, R.Z. Xu, L.P. Zhang, "Software Reliability Allocation Model Based on Cost-controlling," *Asian international workshop on advanced reliability modeling*, AIWARM 2004, pp.173-180, 2004.
- [15] Norman Schneidewind, "Allocation and analysis of reliability: multiple levels: system, subsystem, and module," *Innovations in Systems and Software Engineering: a NASA journal*, Vol.2, No.3/4, pp.121-136, 2006.
- [16] *Electronic Reliability Design Handbook (MIL-HDBK-338B)*, pp.9.1-9.69, Oct., 1998.
- [17] Michael R. Lyu, "Software Reliability

Engineering: A Roadmap," ICSE2007, 2007.

[18] *IEEE Std 1633 Recommended Practice on Software Reliability*, IEEE Reliability Society, pp.4, Jun 2008.

[19] Hoang Pham, *Software Reliability*, Springer, 2000

[20] Peter B. Lakery, Ann Marie Neufelder, *System and Software Reliability Assurance Notebook*, pp.4.14, 2005.

[21] Peter B. Lakery, Ann Marie Neufelder, *System and Software Reliability Assurance Notebook*, 7.10-7.15, 2005.

[22] S. Yamada, M. Ohba, S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on Reliability* 12, 475-484, 1983.

[23] S. Yamada, K. Tokuno, S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *International Journal of Systems Science* 23(12), 2253-2264, 1992.

[24] *IEEE Std 1633 Recommended Practice on Software Reliability*, IEEE Reliability Society, pp.16-36, Jun 2008.

[25] Peter B. Lakery, Ann Marie Neufelder, *System and Software Reliability Assurance Notebook*, pp.4.1-4.21, 2005.

[26] *체계개발단계 내장형 소프트웨어 개발관리 업무 편람*, 방위사업청, 2008.

[27] 신주환, 김용국, 이성은, *검독수리 A급 전투체계 소프트웨어 신뢰도 분석 (ADDR-415-081426)*, 국방과학연구소, 2008.

[28] 김홍수의 6, *소프트웨어 신뢰성 향상 방안 및 모델 개발 (ADDR-412-100046)*, 국방과학연구소, 2009.

[29] *Systems Engineering-System Life Cycle Processes (IEEE Std 15288)*, IEEE, pp.5-39, 2004.

[30] *Software Life Cycle Processes (ISO/IEC 12207)*, ISO/IEC, 1998.

[31] *중합군수지원 개발 실무지침서*, 방위사업청, pp.65, 2009.

[32] 김효창, 조용석, *M-SAM 체계의 신뢰도 목표값 설정 연구 (MADC-516-030530)*, 국방과학연구소, 2003.

[33] 최광조, 노병래, *차기전차 RAM-D 요구조건/설계*

목표 및 예측결과 (GSDC-519-050291), 국방과학연구소, 2005.

[34] 배상수, *생물독소분석 식별기 OMS/MP에 따른 RAM 설정(안) (ADDR-523-080178)*, 국방과학연구소, 2008.

[35] *Procedures for Performing a Failure Mode, Effects and Criticality Analysis (MIL-STD-1629A)*, DoD U.S, Dec 1980.

[36] *The DACS Software Reliability Sourcebook*, DACS, pp.7.2, Sep., 2001.

[37] Guidelines for safety analysis of vehicle based programmable systems, MISRA, Nov., 2007.

[38] <http://www.misra.org.uk/>

[39] *연구개발 프로세스 표준화 규정(안)*, 국방과학연구소, 2010.

김기백 (Ghiback Kim)

정회원



1989년 2월 충북대학교 전자계산기공학과 공학사  
 1991년 2월 충북대학교 전자계산기공학과(전산기공학) 공학석사  
 1991년 3월~현재 국방과학연구소 선임연구원

2008년 3월~현재 아주대학교 시스템공학과 박사과정  
 <관심분야> 시스템공학, 소프트웨어 신뢰성공학, 군수공학

이재천 (Jae-Chon Lee)

종신회원



1977년 2월 서울대학교 전자공학과 공학사  
 1979년 2월 KAIST 전기 및 전자공학과 (통신시스템) 공학석사  
 1983년 8월 KAIST 전기 및 전자공학과 (통신시스템) 공학박사

1994년 9월~현재 아주대학교 시스템공학과 교수  
 <관심분야> 시스템공학, 모델기반 시스템공학, 모델링 및 시뮬레이션, 안전시스템공학