

무기체계의 고장 이력 데이터를 활용한 소프트웨어 신뢰도 분석 모델 적용 사례 연구

조일훈[†] · 황성국 · 이익도 · 박연경 · 이정훈 · 신창훈

LIG넥스원 ILS연구소

The Case Study on Application of Software Reliability Analysis Model by Utilizing Failure History Data of Weapon System

Ilhoon Cho[†] · Seongguk Hwang · Ikdo Lee · Yeonkyeong Park · Junghoon Lee · Changhoon Shin

ILS(Integrated Logistics Support) R&D Lab, LIG NEX1

Purpose: Recent weapon systems in defense have increased the complexity and importance of software when developing multifunctional equipment. In this study, we analyze the accuracy of the proposed software reliability model when applied to weapon systems.

Methods: Determine the similarity between software reliability analysis results (prediction /estimation) utilizing data from developing weapon systems and system failures data during operation of weapon systems.

Results: In case of a software reliability prediction model, the predicted failure rate was higher than the actual failure rate, and the estimation model was consistent with actual failure history data.

Conclusion: The software prediction model needs to adjust the variables that are appropriate for the domestic weapon system environment. As the reliability of software is increasingly important in the defense industry, continuous efforts are needed to ensure accurate reliability analysis in the development of weapon systems.

Keywords: Software Reliability, RIAC-HDBK-217Plus, Schneidewind's Model

1. 서론

기존 국방 분야의 무기체계들은 대부분 하드웨어 기반의 시스템으로 개발되어 하드웨어 성능 및 신뢰도에 중점을 두었다. 그러나 무기체계는 점차 다양한 환경에서 운용되기를 요구받고 있으며 복합적 기능

을 갖는 무기체계 개발이 주를 이루고 있다. 이러한 기능을 구현하기 위해 무기체계 소프트웨어는 더욱 복잡, 정교해지고 있으며 기 개발된 하드웨어 기반에 소프트웨어 성능 개선을 통한 성능 개량 사업도 이루어지고 있다.

무기체계의 소프트웨어 개발이 중요해지면서 소프

[†] 교신저자 ilhoon.cho@lignex1.com

2017년 8월 8일 접수; 2017년 10월 12일 수정본 접수; 2017년 10월 13일 게재 확정.

트웨어에 대한 신뢰성 보증 또한 국내/외적으로 이슈가 되고 있다. 국외적으로는 소프트웨어 시간 계산 오류로 인한 미사일 방어 시스템 미작동, 항공 레이더 시스템 결함으로 인한 민간 항공기 격추 등과 같이 막대한 인적, 금전적 손해를 일으키는 사고가 늘어나고 있다[1]. 국내에서도 우주 발사체 나로호 발사 시 압력 측정 소프트웨어 결함으로 발사가 중지되는 일이 발생하였다. 이처럼 단순 하드웨어 고장이 아닌 소프트웨어 신뢰성 저하로 인한 사고가 다수 일어나고 있으며 신뢰성 보증 활동은 무기체계 개발에서 점차 중요한 요소가 되어가고 있다.

국내 개발된 무기체계의 경우도 소프트웨어 신뢰성이 점차 중요한 부분이 되고 있다. 실제 업체로 정비 요청이 들어오는 사항을 분석한 결과는 <Fig. 1>과 같으며 전체 고장의 약 13%가 소프트웨어에 대한 성능개선, 결함 수정 요청사항이다. 또한 연도 별 업체 정비 항목 중 소프트웨어에 대한 비율은 <Fig. 2>와 같으며 이를 통해 점차 무기체계 운용 중 소프트웨어가 차지하는 비중이 늘어남을 알 수 있다.

소프트웨어 정비 원인을 세부적으로 살펴보면 <Fig. 3>과 같다. 성능개량 및 기술발전에 따른 수정 사항이 약 69%를 차지하며 고장으로 인한 재설치 사항이 약 22%를 차지하고 있다. 이처럼 소프트웨어로 인한 무기체계 고장은 점차 늘어나고 있으며 소프트웨어 신뢰성 확보를 통해 무기체계의 가동률을 높이기 위한 노력이 필요하다.

이러한 흐름에 맞춰 방위사업청은 방위사업규정 [2] 및 무기체계 소프트웨어 개발 및 관리 매뉴얼[3]을 통해 연구개발주기관은 무기체계 개발 시 소프트웨어 종합군지원요소를 개발하도록 명시하고 있다. 이를 바탕으로 현재 일부 무기체계 개발 시 소프트웨어 신뢰도 분석을 수행하고 있다. 신뢰도 분석은 무기체계 개발동안 데이터를 수집하여 공학적 절차와 기법에 따라 분석이 이루어지고 시스템의 목표 신뢰도에 만족하는지 여부를 판단하게 된다. 이 결과를 바탕으로 소프트웨어에 대한 추가 시험이 필요성과파인도 시점을 결정하는데 활용되고 있다.

하지만 아직까지 국내 무기체계의 경우 소프트웨어의 신뢰도 분석에 대한 구체적 기준과 규정은 미정립되어 있으며 국외 연구기관에서 개발한 소프트웨어 신뢰도 분석 모델을 적용하고 있다. 그에 따라 분석 결과 값이 국내 무기체계에 어느 정도 적합한지에

대해서는 연구된 바가 없다. 하드웨어의 경우 그동안 꾸준한 연구들을 근간으로 분석 모델 검증 방안, 신뢰성 시험 등 상당 부분 발전한 반면 소프트웨어의 신뢰도는 이제 분석 단계에 접어든 상태다.

본 연구에서는 먼저 소프트웨어 신뢰도를 분석하는 대표적인 예측/추정 모델을 사용하여 개발 간 데이터를 통해 신뢰도 분석을 수행한다. 추가적으로 해당 무기체계가 개발 이후 운용 간의 고장 이력 데이터를 활용하여 예측/추정 분석 값과 일치성을 갖는지 알아보고자 한다. 이를 통해 소프트웨어 신뢰도 분석 모델이 국내 무기체계 환경에서 직접적으로 적용 가능한지 분석하였다.

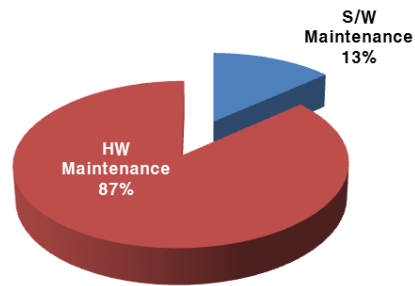


Fig. 1 HW/SW depot maintenance ratio

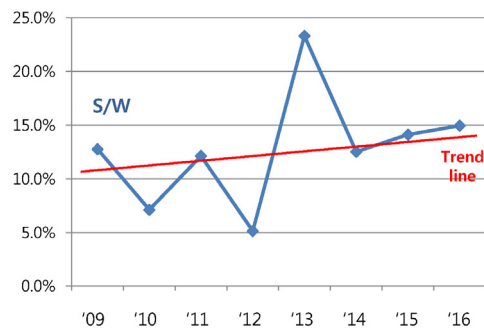


Fig. 2 Ratio of SW failure per year

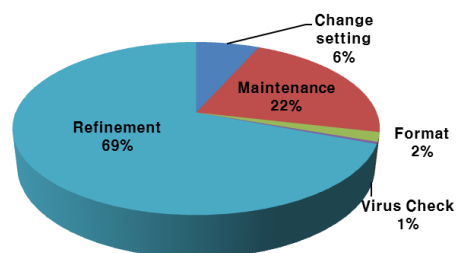


Fig. 3 Ratio of SW fault causes

2. 소프트웨어 신뢰도

본 장에서는 신뢰도 분석 전에 소프트웨어가 하드웨어와 비교하여 갖는 특징을 알아보고 본 연구에서 적용 할 소프트웨어 신뢰도 분석 모델에 대해서 정리하였다.

2.1 소프트웨어 신뢰도 특성

소프트웨어 신뢰도 분석에 앞서 소프트웨어가 가지고 있는 특성을 이해하여야 신뢰도 분석을 수행할 수 있다. MIL-HDBK-338B[4]에서 제시하는 소프트웨어 신뢰도 특성은 다음과 같다.

- 마모 메커니즘 없음(not Rust or Wearout mechanisms)
- 사용 빈도와 소프트웨어 신뢰도는 무관(동일한 입력값/하드웨어/소프트웨어/인터페이스 적용 시)

- 소프트웨어 수정 시 통상 또 다른 새로운 결함 주입
- 탑재되는 하드웨어와 매칭되어 동작함

<Fig. 4>는 소프트웨어 고장률의 특징과 하드웨어 고장률과의 차이를 나타낸다. 소프트웨어는 Upgrade를 통해 새로운 버전 개발 시 새로운 결함이 주입되며 지속적으로 고장률이 감소하는 형태를 나타낸다. 하드웨어의 경우 초기 안정화 단계에서 고장률이 감소한 이후 안정화 시험에 일정한 고장률을 보이다 노후화가 이루어지면 다시 고장률이 증가하는 형태를 보인다. 그에 반해 소프트웨어는 마모 메커니즘이 없으며 지속적으로 감소하는 고장률을 갖게 된다.

하드웨어와 소프트웨어 신뢰도 비교 사항은 <Table 1>에 정리하였다. 소프트웨어 신뢰도 분석 시 주요 입력 데이터는 코드라인 수, 성숙도 레벨, 시험을 통한 결함 수 등과 같다. 소프트웨어 크기 뿐만 아니라 개발/운용 환경에 따라서 신뢰도 값은 변화하게 된다.

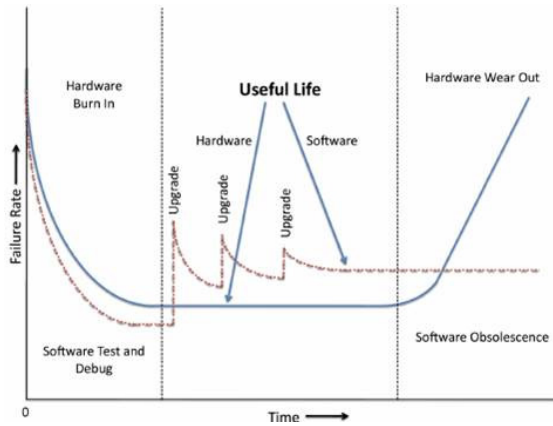


Fig. 4 HW/SW Failure Rate during Life Cycle

Table 1 Comparison between HW and SW reliability

Classifi-cation	MIL standard	Reliability model	Input data	Wear out Environment	Failure rate curve
HW Reliability	MIL-STD-785B[5]	MIL-HDBK-217F, NPRD-2011, NSWC98	Type, Quantity, Stress, Temperature etc.	Application	Constant Function λ
SW Reliability	MIL-HDBK-338B	RL-TR-92-52, RIAC-HDBK-217Plus, Schneidewind's Model	Lines of Source Code, Level of maturity etc.	Non Application	Exponential Function λ

2.2 소프트웨어 신뢰도 분석 모델

소프트웨어 신뢰도 분석 모델은 분석 시점에 따라 예측 모델과 추정 모델로 구분할 수 있다. 예측 모델은 설계 초기에 과거 경험 데이터를 기반으로 목표 신뢰도를 예측하며 개발 환경과 관련된 소프트웨어 척도를 기반으로 분석한다. 추정 모델은 개발 간 소프트웨어 시험 데이터를 활용하여 최종 신뢰도를 추정하는 방법으로 시험과 운용 과정에서 수집되는 결합데이터를 기반으로 최종 신뢰도를 분석한다. 예측 모델과 추정 모델에 대한 비교는 <Table 2>에 작성하였다[4].

대표적 소프트웨어 신뢰도 예측 모델에는 Musa's Execution Time Model, RIAC-HDBK-217Plus, RL-TR-92-52 Model, COQUALMO 등이 있다. 신뢰도 추정 모델에는 General Exponential Model, Schneidwind's

Model, NHPP Model 등이 있다. 각 모델 별 주요 입력 팩터 및 산출값은 <Table 3>을 참고한다[4, 7].

모델의 무기체계 적용 시 적합성을 알아보하고자 하는 본 연구에서는 예측 모델 중 217 Plus Model 과 추정 모델 중 Schneidwind's Model을 선정하였다. 예측 모델 중 217 Plus Model을 선정한 이유는 객관적인 데이터를 기반으로 수리적 모델을 통해 분석이 이루어지기 때문이다. RL-TR-97-52, COQUALMO의 경우 개발자의 주관적인 판단하에 체크리스트를 작성하고 그 결과로 소프트웨어 신뢰도를 예측하므로 결과 값의 객관성이 다소 떨어진다. 추정 모델 중 Schneidwind's Model을 선정한 이유는 주어진 데이터가 Falut Count 기반 모델에 적합하고 기존 무기체계 신뢰도 추정 시 적용 사례가 있는 모델이기 때문이다. 두 모델의 특징과 상세 입력 팩터는 다음과 같다.

Table 2 Comparison between prediction and estimation model

Classification	Analysis data	Application time	Time parameter	Purpose
Prediction Model	Historical data	Initial design phase (before development or test)	Future random point	Prediction of latent failure at initial design phase
Estimaiothn Model	SW Test data	After SW test (after data collection)	Present or Future random point	Estimation of latent fault for decision of testing time

Table 3 Description of SW reliability analysis models

Classification	Analysis model	Computed result	Input data
Prediction Model	217 Plus Model	<ul style="list-style-type: none"> Failure rate of any point after deployment 	<ul style="list-style-type: none"> Lines of source code Fault density Duty cycle
	RL-TR-97-52 (Rome laboratory Prediction Model)	<ul style="list-style-type: none"> Failure density at time of deployment 	<ul style="list-style-type: none"> SW development environment, language SW complexity, traceability
	COQUALMO	<ul style="list-style-type: none"> Failure density at time of deployment 	<ul style="list-style-type: none"> Factor of injection flaw (SW complexity etc.) Factor of defect removal (SW testability etc.)
Estimaiothn Model	Exponential Distribution Model	<ul style="list-style-type: none"> Total number of fault Mean time to failure 	<ul style="list-style-type: none"> Number of fault detected/corrected Test time
	Schneidwind's Model	<ul style="list-style-type: none"> Total number of fault 	<ul style="list-style-type: none"> Number of fault detected/corrected Test time
	NHPP Model (Nonhomogeneous Poisson Process)	<ul style="list-style-type: none"> Estimated number of fault at next time Failure rate at any point 	<ul style="list-style-type: none"> Number of fault detected/corrected Test completion time

2.2.1 RIAC-HDBK-217Plus Model

RIAC-HDBK-217Plus Model은 DoD(Department of Defense) Information Analysis Center에서 제시한 모델로 기존 MIL-HDBK-217F[8]를 발전시켜 하드웨어뿐만 아니라 소프트웨어에 대한 신뢰도 분석 모델을 제시하였다. 217Plus 모델은 무기체계 배치 시점부터 폐기시점까지 고장률 산출이 가능하며 잔존결함 및 고장률 함수가 시간에 따른 지수분포 형태를 갖고 있다. RIAC - HDBK-217Plus 세부 모델은 식은 다음과 같으며 분석을 위한 입력 팩터에 대한 설명은 <Table 4>를 참고한다.

$$\lambda_{sw} = \left(\frac{F_0 e^{-k(t_{i-1})} - F_0 e^{-kt_i}}{730} \right) \quad (1)$$

$$(DC \times FL \times FA \times AS) \times 10^6$$

$$F_0 = KSLOC \times FD \quad (2)$$

$$F_{t_i} = F_0 e^{-kt_i} \quad (3)$$

$$k = \frac{\ln\left(\frac{1}{DSL}\right)}{t_s} \quad (4)$$

- λ_{sw} : SW 고장률(t_i 시점),
- F_0 : 초기결함밀도,
- F_{t_i} : t_i 시점 잔존 결함수,
- t_i : SW 배포 후 경과시간(월 단위)

2.2.2 Schneidewind's Model

Schneidewind's Model은 Handbook of Reliability Engineering[7]에서 제시한 모델로 Falut Count 기반의 신뢰도 추정 모델이다. Schneidewind's Model은 동일한 시간 간격동안의 결함수를 기반으로 과거의 결함률을 이용하여 현재의 결함률을 추정함으로써 앞으로의 결함발생을 보다 정확하게 예측하고자 하는 모델이다. 산출값은 임의 시점의 고장률 SW 총 결함수와 일정한 시간 간격으로 구분한 특정구간의 예상 결함수이다. Schneidewind's Model의 세부 모델은 다음과 같다.

$$\lambda(t) = \alpha \exp(-\beta t) \quad (5)$$

$$D_i = \mu(t_i) = \frac{\alpha}{\beta} [1 - \exp(-\beta i)] \quad (6)$$

$$m_i = D_i - D_{i-1} = \frac{\alpha}{\beta} [\exp(-\beta(i-1)) - \exp(-\beta i)] \quad (7)$$

Table 4 Description of input factor(RIAC-HDBK-217Plus Model)

Parameter Symbol	Name	Description	Default
KSLOC	Lines of Source Code(in thousands)	Lines of source code(in thousands), not including comments	None
FD	Fault Density	Initial quality as measured by fault density at item shipment	Refet to 217Plus
FL	Fault Latency	Average number of times a failure is expected to reoccur before its underlying fault is corrected	2.0
FA	Fault Activation	Fraction (in decimal form) of population exhibiting fault activation	1.0(100%)
AS	Average % Severity	Fraction (in decimal form) of fault that are disruptive, or critical to the customer	0.5(50%)
ts	Time to Stabilization	-	Initial = 48 Subsequent = 24
DSL	Defect Stabilization Level	The level at which the software failure rate stabilizes relative to F0	Refet to 217Plus
DC	Duty Cycle	Fraction of calendar time the software is in operation	-

- α : 초기 고장률,
- β : 고장률에 영향을 미치는 비례상수,
- t_i : 시작 시점부터 i 번째 기간까지 총 기간
- D_i : i 번째 기간까지의 예상 누적 결함 수,
- m_i : i 번째 기간 동안의 예상 결함 수

위의 식에서 α, β 를 추정하기 위한 방안으로 Handbook of Reliability Engineering[7]에서는 3가지 모델을 제시한다. 그 중 본 연구에서는 모든 데이터를 활용하는 모델을 선정하였으며 매개변수 추정식은 다음과 같다.

$$\hat{\alpha} = \frac{\hat{\beta} F_n}{1 - \exp(-\hat{\beta} n)}, \tag{8}$$

$$\frac{1}{\exp(\hat{\beta}) - 1} - \frac{n}{\exp(\hat{\beta} n) - 1} = \sum_{k=0}^{n-1} k \frac{f_{k+1}}{F_n}$$

- f_i : i 번째 기간에 발견된 결함 수,
- F_n : n 번째 기간까지 누적 결함 수

3. 소프트웨어 신뢰도 분석

본 장에서는 무기체계 개발 간 분석 자료를 활용하여 소프트웨어 신뢰도를 예측/추정 모델을 통해 분석한다. 그리고 해당 무기체계의 개발 이후 운용 중 고장들을 분석하여 어느 정도 일치성을 갖는지 알아보도록 한다.

3.1 소프트웨어 신뢰도 예측

먼저 소프트웨어 신뢰도 예측을 위해 대상 사업을 선정하였다. 개발 간 데이터를 통한 분석 값과 무기체계 운용 간 고장 이력 데이터와의 일치성 검증을 위하여 개발이 완료되어 최소 3년 이상 운용된 무기체계

를 선정하였다. 또한 소프트웨어 개발 간 분석 자료를 활용할 수 있는 사업을 고려하여 본 연구에서는 OO 사업을 대상으로 선정하였다.

OO사업은 2011년 개발을 완료한 사업으로써 소프트웨어 코드라인 수(SLOC)는 개발 간 작성한 소프트웨어 산출물 명세서(SPS, Software Product Specification)를 통해 확인할 수 있다. OO사업에서 개발되는 모든 소프트웨어의 라인수는 총 314,691이었으며 이중 컴파일 작업에서 제외되는 주석을 제외하면 총 277,309였다.

OO사업의 초기 결함밀도(FD)와 결함 안정화 등급(DSL)은 RIAC-HDBK-217Plus RIAC[6]에서 개발 업체의 CMM 레벨에 따라 값을 제시하고 있다. OO사업의 소프트웨어 개발을 담당한 업체는 CMMI 5레벨을 인증받았으며 그에 해당하는 값을 얻을 수 있다.

제고장 횟수(FL), 결함 활성비(FA), 치명결함 발생비(AS)의 경우 소프트웨어 고장에 대한 분석 및 관리 데이터가 없으므로 Default 값을 넣었으며 안정화 시간(ts)은 해당 소프트웨어가 최초 개발 버전이므로 48(months)를 적용하였다.

마지막 작동시간(DC)은 해당 무기체계의 연간 운용시간(AOR)을 적용하였으며 연 시간(8,760시간)에 대한 비율로 환산하여 적용하였다.

OO사업의 소프트웨어 신뢰도 예측을 위한 최종 입력 값들은 <Table 5>와 같다.

개발 간 자료를 통해 산출한 소프트웨어의 고장률 및 잔존 결함 함수는 다음과 같다.

$$\lambda_{sw} = \left(\frac{138.65e^{-0.09(t_i-1)} - 138.65e^{-0.09t_i}}{730} \right) (0.83 \times 10^6)$$

$$F_0 = 138.65, F_{t_i} = 138.65e^{-0.09t_i}$$

잔존 결함 함수 통해 OO사업 종료 이전에 개발 소프트웨어가 전력화 후 2년 간 몇 개의 결함이 발생할지 분석하였다. 단위 기간은 3개월 단위로 정리하였으며 그 결과는 <Table 6> 과 같다.

Table 5 Data for software reliability prediction

Factor	Value	Basis	Factor	Value	Basis
KSLOC	277.309	SPS	AS	0.5	Default
FD	0.5	217Plus	ts	48(개월)	Initial
FL	2.0	Default	DSL	0.01	217Plus
FA	1.0	Default	DC	7,300/8,760	AOR

Table 6 Predicted number of SW fault after developmnet(OO system)

Unit time	Number of SW Fault	Unit time	Number of SW Fault
0~3 months	34.97	12~15 months	10.97
3~6 months	26.01	15~18 months	8.22
6~9 months	19.50	18~21 months	6.16
9~12 months	14.62	21~24 months	4.62
Total		124.78	

분석 결과 OO사업의 개발 소프트웨어는 개발 이후 초기에 약 138개의 결함을 갖으며 2년 간 약 125개의 결함이 발견되고 조치될 것이라 예측할 수 있다. 분석 과정에서 알 수 있듯이 소프트웨어 신뢰도 예측은 코드 라인 수에 크게 영향을 받으며 Default 값이 적용된 팩터가 많은 부분을 차지하고 있다.

다음은 OO사업 종료 후 전력화되어 운용 중 실제 발생한 고장, 결함을 분석하였다. 분석 자료는 개발 업체에서 관리하는 데이터베이스를 활용하였으며 해당 무기체계의 소프트웨어 정비는 모두 업체에서 수행하기 때문에 운용 간 고장 이력을 정확하게 분석할 수 있다.

Table 7 Number of SW fault after developmnet (OO system)

Unit time	Number of SW Fault	Unit time	Number of SW Fault
0~3 months	3	12~15 months	0
3~6 months	6	15~18 months	1
6~9 months	4	18~21 months	2
9~12 months	4	21~24 months	0
Total		20	

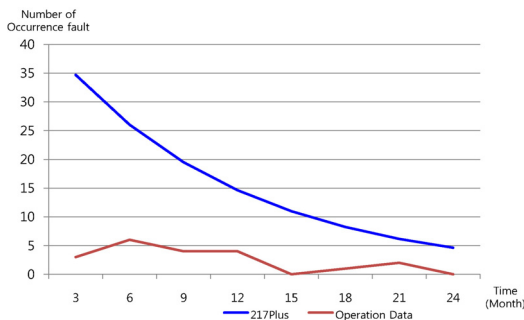


Fig. 5 Comparison between 217Plus and Software Failure Histories

OO사업 무기체계의 운용 간 결함 자료를 단위 시간 3개월로 나누어 분석하였고 이는 <Table 7>와 같다.

분석 결과 OO 무기체계 소프트웨어는 운용 단계에서 2년 간 20건의 고장이 발생하였다. 이는 개발 자료를 활용하여 분석한 신뢰도 예측 값인 124건과는 많은 차이가 있음을 알 수 있다. 해당 데이터를 단위 시간 별 발생 결함 수 그래프로 나타내면 <Fig. 5>와 같다.

지금까지 개발 초기에 수행하는 217Plus Model을 통한 소프트웨어 예측 값과 실제 국내 무기체계 운용 결과를 비교하였다. 그리고 그 결과 두 데이터간의 차이가 있음을 확인하였다. 이는 모델 적용을 위한 기초 데이터 부족으로 보정 계수 팩터(결함 잠재도, 결함 활성화도, 치명결함발생도)를 Default 값으로 적용한 결과로 판단할 수 있다. 그러므로 217Plus 모델의 국내 무기체계 적용을 위하여 보다 적합한 보정 계수 산출 방법을 개발하여 정확한 신뢰도 예측이 수행되어야 할 것이다.

3.2 소프트웨어 신뢰도 추정

소프트웨어 신뢰도 추정을 위한 대상 사업은 별도로 선정하였다. 신뢰도 예측과는 다르게 개발 간 소프트웨어에 대한 시험 데이터를 필요로 하기 때문에 개발 완료 후 최소 3년 이상 운용된 무기체계 중 시험 데이터를 보유하고 있는 사업을 선정하였다. 이러한 과정을 거쳐 소프트웨어 통합 시험, 시험 평가 데이터 확보가 가능한 XX사업을 대상으로 선정하였다.

XX사업은 2011년 개발을 완료한 사업으로써 먼저 개발 간 소프트웨어 통합 시험 및 시험평가를 통해 발견된 소프트웨어 결함 자료를 수집하였다. 수집된 결함 자료는 단위 시간 1개월로 나누어 결함 수를 체크하였고 이는 <Table 8>과 같다.

Table 8 Number of SW test fault during developmnet(XX system)

Unit time	Number of SW Fault	Unit time	Number of SW Fault
1 month	20	4 months	9
2 months	7	5 months	13
3 months	8	6 months	0
Total		57	

수집된 소프트웨어 결함 데이터를 Schneidewind's Model에 적용한 결과는 다음과 같으며 이를 그래프로 나타내면 <Fig. 6>과 같다.

$$\hat{\alpha} = 18.47, \hat{\beta} = 0.25$$

예상 총 결함수: $\hat{\alpha}/\hat{\beta} = 72.98$

결함률: $\lambda(t) = 18.47 \times \exp(-0.25 \times t)$

분석 결과 해당 소프트웨어는 총 약 73개의 결함을 가지고 있으며 개발 간 시험을 통해 57개의 결함을 찾아 수정하였다. 그에 따라 해당 무기체계가 전력화 이후 약 16개의 결함이 추가로 발생 할 수 있음을 알 수 있다.

다음은 XX사업 종료 후 해당 무기체계가 전력화되어 운용 중에 발생간 고장을 검토하였다. 해당 자료는 신뢰도 예측과 마찬가지로 개발 업체의 정비 이력 관리 데이터베이스를 활용하였다.

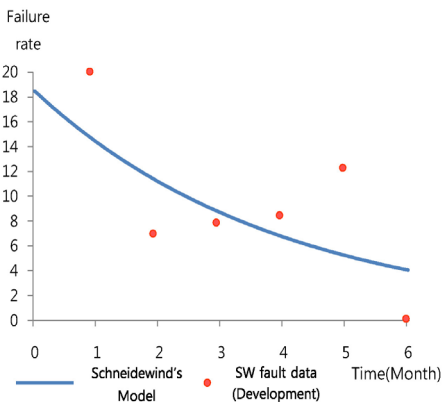


Fig. 6 Result of SW failure rate estimation (Development phase)

Table 9 Number of SW fault after development (XX system)

Unit time	Number of SW Fault	Unit time	Number of SW Fault
1 month	2	7 months	0
2 months	4	8 months	0
3 months	1	9 months	0
4 months	2	10 months	1
5 months	0	11 months	1
6 months	0	12 months	0
Total		11	

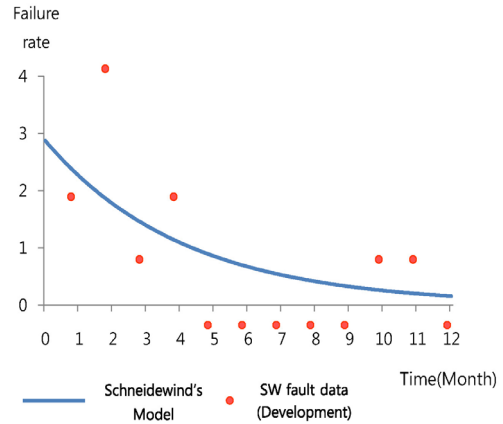


Fig. 7 Results of SW reliability estimation during development and operation phase

XX사업 무기체계의 운용 간 결함 자료를 개발 간 자료와 마찬가지로 단위 시간 1개월로 나누어 체크하였고 이는 <Table 9>와 같다.

수집된 소프트웨어 결함 데이터를 Schneidewind's Model에 적용한 결과는 다음과 같으며 이를 그래프로 나타내면 <Fig. 7>과 같다.

$$\hat{\alpha} = 2.87, \hat{\beta} = 0.24$$

예상 총 결함수: $\hat{\alpha}/\hat{\beta} = 11.81$

결함률: $\lambda(t) = 2.87 \times \exp(-0.24 \times t)$

분석 결과 해당 소프트웨어는 개발 이후 약 12개의 결함을 갖고 있으며 배치 후 1년 간 11개의 결함을 찾아 수정하였다. 추가로 이후 약 1개의 결함이 발생할 수 있음을 알 수 있다.

이제는 개발 간 데이터와 운용 간 데이터 간의 상관 관계를 알아보도록 한다. 개발 간 데이터를 활용한 소프트웨어 추정을 통해 개발 이후 약 16개의 결함을 추정하였다. 이는 실 운용 간 데이터를 통해 추정한 약 12개의 값과 어느 정도 일치성을 보인다. 추가적으로 Schneidewind's Model을 통해 산출된 형상 모수 β 를 비교하면 두 데이터간의 일치성을 알 수 있다. 해당 결과는 <Table 10>에 정리하였으며 결함률 곡선에 대한 결과는 <Fig. 8>에 그래프로 작성하였다.

최종적으로 개발 기간과 운용 기간의 소프트웨어 고장을 추정값을 비교한 결과, 개발 간 잔존 결함 수와 운용 간 초기 결함 수가 유사성을 보이며 두 구간

Table 10 Results of SW reliability estimation during development and operation phase

Classification	Development data	Operation data	Remark
$\hat{\beta}$	0.25	0.24	Comparison of shape parameter
Nunmer of fault at deployment	15.98	11.81	Comparision remaining fault after development and initial fault before operation

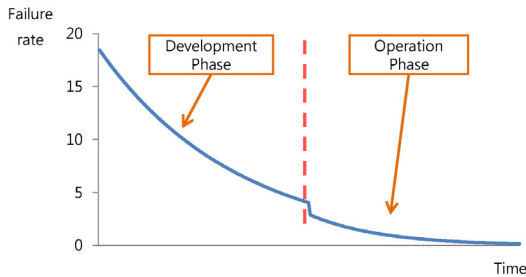


Fig. 8 Results of SW reliability estimation during development and operation phase

의 고장률 곡선이 유사하게 생성되었음을 확인할 수 있었다. 이는 곧 개발 간 데이터를 통해 운용 간 발생할 소프트웨어 잠재 결함 수 추정이 가능하다는 것을 의미한다.

4. 결론

본 연구에서는 소프트웨어 결함 특징을 분석하고 신뢰도 분석을 위한 예측 및 추정 모델을 검토하였다. 이어 대상 사업을 선정하여 개발 간 작성된 소프트웨어 관련 자료 및 시험 데이터를 활용하여 소프트웨어 신뢰도를 예측 및 추정하였다. 마지막으로 분석된 결과와 무기체계 전력화 이후 운용 간 결함 이력을 비교하여 모델 적합성을 검증해 보았다.

RIAC-HDBK-217Plus Model을 통한 소프트웨어 신뢰도 예측은 무기체계 운용 간 발생 결함수와 큰 차이를 보였다. 이는 예측의 정확도를 높이기 위해서는 모델 입

력 값에 대한 신뢰도를 높여야 하며 국내 무기체계에 적합한 보정 계수 산출 방법을 개발해야 할 것이다.

Schneidewind’s Model을 통한 추정 모델은 개발/운용 기간으로 나누어 비교 분석을 수행한 결과, 개발 기간동안의 시험데이터를 활용한 신뢰도 추정 값이 무기체계 운용 간 발생 결함 수와 유사한 경향을 보였다. 또한 개발 기간과 운용 기간의 고장률 곡선이 유사하게 생성되어 개발 간 데이터를 통해 운용 간 발생할 SW 잠재 결함 수 추정이 가능함을 확인할 수 있었다.

지금까지 본 논문을 통해 소프트웨어 신뢰도 분석 모델이 국내 무기체계 환경에 적합성을 가지는 지 분석하였다. 점차 국방 산업에서도 소프트웨어에 대한 신뢰성이 중요시되는 만큼 무기체계 개발 시 정확한 신뢰도 분석이 이루어지도록 지속적인 노력이 필요하다.

References

- [1] Kim, Y. T., Park, J. H., Jeon, Y. M., Baek, J. M., and Bae, D. H. (2015). “An Approach for Software Process Improvement based on Software Defect Analysis of UAS”. KCSE 2015, Vol. 17, No. 1, pp. 278-281.
- [2] Defense Acquisition Program Administration (2016). “Defense Acquisition Management Regulation”. DAPA LOI 361, Article 36, Vol. 311, pp. 14-159.
- [3] Defense Acquisition Program Administration (2016). “Manual of Software Development and Management in Weapon System Software Development”. DAPA Manual 2016-4, pp. 100.
- [4] DOD (1998). “Electronic Reliability Design Handbook”. MIL-HDBK-338B.
- [5] DOD (1980). “Reliability Program for System and Equipment”. MIL-STD-785B.
- [6] RIAC (2006). “Reliability Prediction models”. RIAC-HDBK-217Plus.
- [7] Michael R. L. (2006). “Handbook of Software Reliability Engineering”. IEEE Computer Society Press and McGraw-Hill Book Company, Ch3, pp. 71-115.
- [8] DOD (1991). “Reliability prediction of electronic equipment”. MIL-HDBK-217F.