

소프트웨어 ILS 적용방안 연구 : 소프트웨어와 하드웨어의 유지보수 특성을 고려한 비교 분석을 중심으로

이관영¹, 김상문¹, 박은심¹, 박재은¹, 김근형^{1*}

¹LIG 넥스원 ILS 연구센터

Qualitative study of software ILS application: Comparison analysis of maintenance types in software and hardware

Kwan-Young Lee¹, Sang-Moon Kim¹, Eun-Shim Park¹, Jae-Eun Park¹,
Geun-Hyung Kim^{1*}

¹ILS(Integrated Logistics Support) R&D Lab, LIG Nex1

요약 무기체계에서의 소프트웨어에 대한 비중이 커짐에 따라, 소프트웨어 ILS(Integrated Logistics Support)에 대한 관심이 커지고 있다. 그러나 기존 무기체계에서 ILS 개발은 하드웨어 중심으로 적용하고 있어 소프트웨어에 그대로 적용하기 힘든 실정이다. 본 논문은 소프트웨어 ILS의 실무 적용 어려움이 무엇인지 원인을 파악하고 문제점을 분석하기 위해 국내/외 규정 및 사례들을 통해 현 실태를 파악하였다. 또한, 소프트웨어의 유지보수 특성을 고려하여 소프트웨어와 하드웨어 ILS의 차이점을 검토하고, 소프트웨어 ILS 요소개발 범위 설정을 통해 국내에 적용이 가능한 소프트웨어 ILS 개발방안을 제안하였다. 이는 소프트웨어 ILS 개발을 위한 구체적인 실무 적용방안을 수립하는 데 도움이 될 것으로 기대한다.

Abstract The increasing importance of software has highlighted the need for the proper application of software ILS. On the other hand, the development of ILS has been somewhat limited to the area of hardware development. Therefore, this study examined the potential difficulties in applying software ILS to practical uses, and analyzed the drawbacks by reporting several domestic or foreign regulation cases. In addition, the differences between hardware and software ILS were examined by considering their characteristics of maintenance. Eventually, by establishing the proper range, this study proposes a development plan that is suitable for domestic weapon systems. The proposed research is expected to be helpful for offering specific plans for designing software ILS.

Key Words : FMECA(Failure Modes Effects and Criticality Analysis), ILS(Integrated Logistics Support), LSA(Logistics Support Analysis), RAM(Reliability, Maintainability, Availability), Software ILS

1. 서론

21세기 무기체계는 네트워크 중심화 및 무인화를 바탕으로 발전하고 있다. 무기체계의 첨단화가 진행됨에 따라 소프트웨어에 대한 중요성이 증가하고 있으며, 무기체계 성능의 핵심 요소로 대두되고 있다[1].

과거 무기체계는 핵심 기능 구현 시 하드웨어를 기반으로 개발되었다. 하지만 최신 무기체계는 하드웨어의 물리적 한계, 기능 구현 시 제한되는 유연성과 확장성, 개

발기간 단축에 대한 요구 등으로 소프트웨어 의존도가 증가하고 있다.

Fig. 1은 미 공군에서 개발된 전투기 무기체계의 기능적인 역할에서 소프트웨어가 차지하는 비중의 변화 추이다. 소프트웨어 비중은 꾸준히 증가하고 있으며, 하드웨어 비중을 넘어 전투기 무기체계 기능의 대부분을 차지하고 있다.

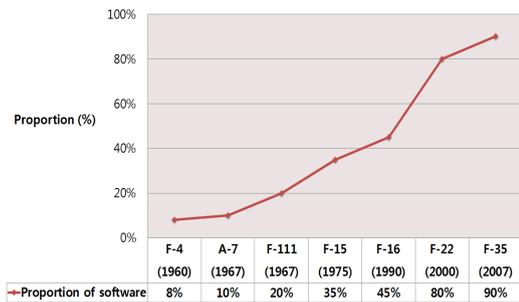
수치상으로 1960년대 개발된 F-4의 소프트웨어 비중은 8%에서 2007년 F-35는 90%로 증가하였으며, 전투기

*Corresponding Author : Geun-Hyung Kim(ILS R&D Lab, LIG Nex1.)

Tel: +82-31-8026-4695 email: geunhyung.kim@lignex1.com

Received August 7, 2014 Revised September 10, 2014

Accepted September 11, 2014



[Fig. 1] Change of proportional importance on the softwares equipped in US Air forces

의 소프트웨어 비중은 11배 이상 증가하였다[2].

무기체계에서 소프트웨어는 높은 비중을 차지하고 무기체계를 제어하는 핵심 요인으로 자리매김하였다. 그러므로 소프트웨어의 결함 발생은 무기체계 전체에 치명적인 영향을 미친다. 게다가 하드웨어 고장 발생은 다양한 방법(육안, 자체 점검 기능, 점검 장비 등)을 통해 고장 탐지 및 고장 조치를 할 수 있는 반면에 소프트웨어 결함에 따른 무기체계 오작동은 발생 원인, 조치 방안 및 결함에 인한 무기체계에 미치는 여파를 정확하게 감지할 수 없다.

실제 소프트웨어 결함은 항공기 추락, 로켓 폭발 및 미사일 방어 시스템 미작동 등 인명 및 비용 손실에 치명적인 결과를 초래할 수 있다[3]. 따라서 무기체계의 핵심요소로 자리 잡은 소프트웨어는 개발 시 결함을 최소화시키기 위한 신뢰성 확보 활동, 최적 운용 상태를 유지하기 위한 군수지원 계획을 중요시 다루고 있다. 즉, 소프트웨어 ILS에 대한 중요성이 매우 강조되고 있다.

또한, 미 육군에서는 소프트웨어 유지보수를 위해 무기체계 소프트웨어에 지출되는 비용의 2/3 이상을 투자하고 있다[4]. 그리고 국내 방위사업관리규정에서의 무기체계 소프트웨어 ILS 요소개발은 IPT(Integrated Project Team) 팀장과 연구개발 주관기관의 협의를 통해 개발 범위 및 세부내용을 결정하도록 명시하고 있다. 특히, 연구개발 주관기관은 무기체계 ILS 개발 규정에 따라 SW ILS를 수행하도록 제시하고 있다[5]. 하지만 소프트웨어 ILS에 대한 적용 방안은 아직도 구체화되어 있지 않은 실정이다. 따라서 소프트웨어는 하드웨어 ILS와 같이 구체화된 소프트웨어 ILS의 적용 방안 개발을 위한 국내/외 민간·정부·학술기관 등 다양한 연구기관들의 지속적인 연구가 필요하다.

하드웨어 ILS는 명확한 개발 기준, 규정 및 수십 년의 개발 경험을 바탕으로 적용 방안이 구체화되어 있다. 그러나 소프트웨어는 하드웨어와 달리 실제 형상 자체가 존재하지 않고, 기계적 고장에 의해 결함이 발생하는 항목이 아니므로 하드웨어 ILS와 동일한 방법으로 개발하는 것은 적절하지 않다.

따라서 본 연구에서는 소프트웨어 ILS에 대한 이론적 배경을 검토하고, 현 무기체계에서 적용된 소프트웨어 ILS 국내 개발 실태를 중점적으로 분석하였다. 나아가 분석된 사례를 토대로 문제점을 제시하고, 소프트웨어 ILS에 대한 적용 방안을 세부적으로 제안한다.

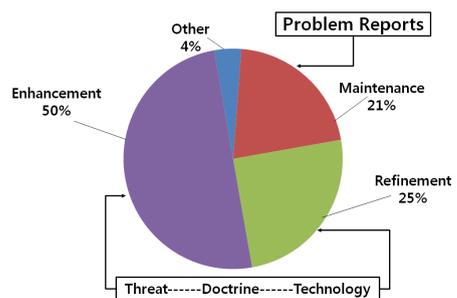
본 논문의 구성은 6장으로 구성되어 있으며, 1장은 연구의 배경 및 목적, 2장은 소프트웨어 ILS 개요, 3장은 소프트웨어 ILS 개발 실태, 4장은 소프트웨어 ILS 요소 개발 범위 및 수준, 5장은 소프트웨어 ILS 적용 방안, 6장은 결론으로 작성하였다.

2. 소프트웨어 ILS 개요

소프트웨어 ILS는 하드웨어 ILS와 유사하지만 동일하게 보긴 어렵다. 본 장에서는 소프트웨어 ILS와 관련된 분야의 특성을 설명하고, 하드웨어 ILS와 차이에 대해 비교하였다.

2.1 소프트웨어 유지보수

소프트웨어 유지보수는 무기체계 개발 이후, 시스템의 야전 배치 후 결함 정비, 성능 향상 및 적용 환경 변경에 따라 소프트웨어를 수정하는 활동이다[6]. 연구 결과에 따르면 유지보수를 위한 소프트웨어 수정 원인의 비율은 Fig. 2와 같다[6].



[Fig. 2] Cases of Software Changes

사용자가 새로운 기능을 필요로 함에 따른 성능 개선(Enhancement) 및 기술 발전에 따른 수정(Refinement) 활동, 즉 고장에 의한 정비가 아니라 위협 수준의 변동, 고리 발전, 기술발전 등 무기체계 개발 업무와 맞물려 수행되어야 하는 소프트웨어 수정 활동이 75%를 차지하고 있음을 알 수 있다. 이것은 하드웨어 ILS가 장비 사용에 따른 마모 및 고장에 대한 최적의 유지보수 활동을 수행할 수 있도록 ILS 요소개발을 하는 것과는 다른 방향(방법)으로 소프트웨어 개발 시 수정 활동을 고려하여, 소프트웨어 지원 방향을 검토하고, 수립해야 한다는 것을 의미한다.

2.2 소프트웨어 지원업무의 구분

소프트웨어의 지원 업무는 크게 시스템 운용과 정비 지원 업무(System operations and maintenance support tasks)와 소프트웨어 수정 업무(Software modification tasks)로 구분할 수 있다[11]. Table 1은 소프트웨어의 지원 업무 활동별 세부 업무에 대해 설명하고 있다.

[Table 1] Software Support Task

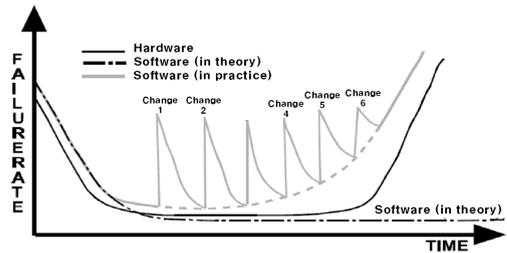
Task	Activity	
System operations and maintenance support task	Operations Support Tasks	<ul style="list-style-type: none"> Action to reload / restart software Record information about the system condition / configuration at the time of failure
	Post-Mission Tasks	<ul style="list-style-type: none"> Fault investigation Data extraction for operational / engineering analysis
	Software Embodiment and Mission Preparation Tasks.	<ul style="list-style-type: none"> Reinstalling software
Software modification tasks	<ul style="list-style-type: none"> Changes to the design of a configured software item 	

지원 업무 중 소프트웨어 수정 업무(Software modification tasks)를 제외한 시스템 운용 및 정비지원 업무는 소요군의 정비 정책에 따라 적용될 수 있다[11]. 그러나 소프트웨어 수정 업무는 소프트웨어 개발 업

무와 유사한 절차에 따라 수행되는 특성과 장기간의 지원업무시간 등의 특성으로 인해 시스템 운용과 정비지원 업무와 달리 여러 가지 측면의 검토가 필요하다. 따라서, 본 장에서 제시된 지원 업무를 바탕으로 4장에서는 현실적으로 적용 가능한 지원 업무 수준에 대해 추가 검토를 수행하였다.

2.3 소프트웨어의 고장률 특성

Fig. 3은 소프트웨어와 하드웨어의 시간의 흐름에 따른 고장률을 표현한 그림이다. 이상적인 개발과정을 거친 소프트웨어는 이론적으로 개발 단계에서 설계 완료 후 오류수정이 완료되면 고장률이 매우 낮은 상태를 유지한다. 그러나 실제 문제에서는 유지보수 활동에 의해 고장률이 급격히 증가하는 양상이 번번이 발생한다. 하드웨어는 안정화 단계에 들어가면 일정한 고장률을 유지하기 때문에 일반적으로 bathtub 곡선을 그린다. 하지만 소프트웨어는 수정 활동을 수행함에 따라 오히려 추가적인 결함이 주입되어 고장률이 상승하는 경향이 있다[6]. 즉, 성능 개선(Enhancement) 및 기술 발전에 따른 수정(Refinement) 활동뿐만 아니라 결함 정비를 위한 오류 수정 활동에서도 동일한 고장률 특성을 가진다.



[Fig. 3] Change of failure rates by software and hardware maintenances

2.4 하드웨어 ILS와 소프트웨어 ILS의 관계

하드웨어 분야에서는 장비 사용에 따른 마모 및 고장에 대한 최적의 유지보수 활동을 수행할 수 있도록 ILS 요소개발을 수행한다. 하지만 소프트웨어의 고장은 개발상의 오류뿐만 아니라 고객의 요구 사항 및 기능 추가, 하드웨어 변경 등 다양한 요소에 의해 발생할 수 있다. 그러므로 소프트웨어 ILS는 소프트웨어의 특성을 충분히 고려하여 요소 개발 방향을 검토하고, 수립해야 한다.

Table 2는 하드웨어와 소프트웨어의 고장과 관련된 특성을 비교한 표이다.

[Table 2] Comparisons of ILS specification between hardware and software

	Hardware	Software
Failure cause	Failure occurrence with designing, manufacturing, operating	1. Error on the development 2. Customers needs 3. Addition of function, change of hardware
Failure symptoms	Detectable	Non detectable
Prediction on failure	Predictable through analysis of RAM	Difficulties in predicting for error Only possible to grasp by addition of function, change of hardware
Maintenance task	Restoration to the original state with exchange of component	1. System operations and maintenance support task 2. Software modification tasks
Failure recurrence	Replicable same failure	1.Modified error do not appear again. 2.High probability of occurrence with improper maintenance

3. 소프트웨어 ILS 개발 실태

소프트웨어 ILS 적용 방안을 수립하기 위해서는 소프트웨어 ILS에 대한 현 실태에 대한 검토가 필요하다. 따라서, 본 장에서는 국내/외 소프트웨어 ILS 관련 규정을 조사했고, 현재 규정에서 제시하고 있는 소프트웨어 ILS의 개발 범위를 정리하였다. 또한, 국내 3군의 소프트웨어 ILS 적용 사례를 조사하였고, 현재 실무에서 소프트웨어 ILS가 적용되고 있는 실태를 정리하였다.

3.1 소프트웨어 ILS 관련 규정 및 자료

서론에서 언급한 무기체계의 소프트웨어 비중 증가에 따라 관련 기관들은 다양한 정책 연구를 수행하고 있다. 관련 기관은 연구 결과를 토대로 규정 및 보고서에 소프트웨어 ILS와 관련된 사항을 추가하고 있다[1,8,9]. 본 장에서는 소프트웨어 ILS 관련 규정에서 제시하고 있는 업무 범위, 수행 방법 및 각 기관에서 수행된 연구결과를 통해 소프트웨어 ILS에 대한 적용 방안을 검토한다.

소프트웨어 ILS와 관련하여 제시하고 있는 국내/외 규정 및 관련 자료는 Table 3과 같다.

현재 검토된 국내/외 소프트웨어 ILS 관련 규정과 자료에서는 소프트웨어 ILS에 대한 필요성 및 지원 시 고려 사항에 대한 일반적인 내용에 대해서만 기술하였다. 하지만 각 항목별 수행 방법에 대하여 구체적으로는 제시하고 있지 않은 실정이다. Table 3의 국/내외 규정 및 지침에서 제시되어 있는 소프트웨어 ILS 관련 내용을 RAM, LSA, ILS 요소로 구분하면 다음과 같다.

[Table 3] List of domestic and foreign regulations

	Main contents
Defense Acquisition Program Administration Regulations[5]	· Acquirement of weapon system software & regulations for developments plans
ILS study on weapon system software [8]	· Actual conditions of ILS application · Analysis on the range of ILS application · Plans of reliability, maintainability and LSA · Establishment of concepts and steps of software maintenance · Application plans on the 11 factors of ILS
Weapon system software development & management guidelines [10]	· Changes in regulations and developments of ILS
DEF-STAN 00-60 Part3 [11]	· Definition, managements and plans of software ILS · Description of ILS factors · Applications on software LSA
MIL-HDBK -338B [12]	· Definition of software reliability · Predictions on software reliability and estimation model · Software reliability allocation method and analysis method
ILS Handbook Chapter 9 [13]	· Software development process · Description of 11 factors for software ILS
AR 700-127 [14]	· Description of software characteristic & ILS
PAM 700-56 [4]	· Necessity & considerations of software ILS · Case study on software application considering levels of maintenances

GSAM 3.0 Chapter12 [6]	<ul style="list-style-type: none"> · Software ILS on TLCSM (Total Life Cycle Systems Management) · Considerations on successful application of Software ILS
---------------------------	---

3.1.1 RAM

국내/외 대부분의 규정 및 관련 자료에서는 소프트웨어에 대한 신뢰도 예측에 대하여 구체적으로 제시되어 있지 않다. 하드웨어 신뢰성은 그동안 꾸준한 연구들로 구체화한 분석 방안들이 제시되어 있지만, 소프트웨어는 신뢰성에 대한 인식 부족으로 체계적인 연구 및 관리가 이루어지지 않았다. 이것은 소프트웨어 신뢰성에 대한 중요성이 부족하다기보다는 비가시적인 소프트웨어의 속성 때문이다[6]. 하드웨어는 고장률을 이용한 MTBF (Mean Time Between Failure) 값을 통해 신뢰도 분석이 수행되지만, 소프트웨어의 경우 기구물이 아닌 프로그래밍 언어로 이루어진 결과물이므로 하드웨어와 동일한 방법으로 고장률 값을 연산하는 것은 제한된다.

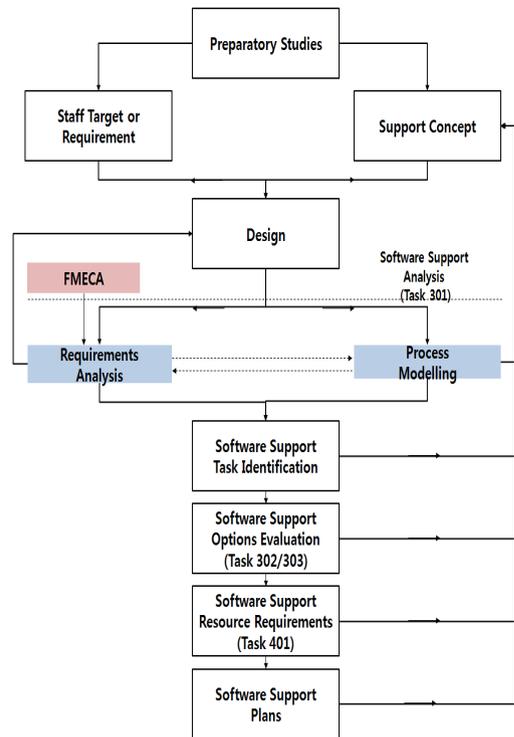
MIL-HDBK-338B에서는 소프트웨어 신뢰도 예측 및 추정 모델, 신뢰도 할당 및 신뢰도 테스트에 대하여 설명하고 있다. 특히, 신뢰도 예측은 개발하고 있는 소프트웨어에 내재한 결함 요소를 예측하여, 릴리즈 시점까지 소요되는 시험비용과 기간을 고려하여 최적의 소프트웨어 릴리즈 시점을 결정하기 위하여 수행하는 것으로 제시되어 있다[12]. 따라서, 하드웨어의 신뢰도 예측과는 상이한 개념으로 소프트웨어 신뢰도 예측의 결과를 하드웨어와 동일하게 LSA 및 ILS 요소개발에 활용하는 것은 추가 검토가 필요하다.

또한, 안보경영연구원에서는 “소프트웨어 신뢰성 자료를 활용하여 고장을 예측하거나 정비업무량을 산정하는 것은 적합하지 않으므로 RAM 분석은 수행하지 않는다.”라고 제시하고 있다[8].

3.1.2 LSA

소프트웨어에 대한 LSA는 DEF-STAN 00-60 에서 하드웨어 LSA 절차와 유사한 형태로 제시하고 있다. 특히 소프트웨어의 LSA 프로세스에서의 FMECA(Failure Modes Effects and Criticality Analysis)는 소프트웨어 고장유형의 분석을 통해 필요한 지원업무를 식별하는 역할을 한다. 고장유형이 확인되면, 즉각적인 지원 업무와 장기간의 소프트웨어 수정의 지원업무를 식별할 수 있다.

FMECA를 통한 소프트웨어 고장 원인은 인터페이스 변경, 위험 환경 변화 및 상호 운용 중인 장비의 변경 등의 요인들을 포함한다. 그러나 하드웨어 예방정비를 위하여 주요하게 분석되는 RCM (Reliability Centered Maintenance)의 경우는 운용 시간에 따른 마모 및 파손이 발생하지 않는 소프트웨어에서는 적용되지 않는 것으로 검토되어 있다. Fig. 4는 소프트웨어 LSA 프로세스를 표현한 그림으로 하드웨어 유사하게 제시되어 있다[11].



[Fig. 4] Process of software LSA

3.1.3 ILS 요소

AR 700-127에서는 ILS 요소에 하드웨어와 소프트웨어 관련 요소를 포함하고 있다. 또한, ILS 핸드북에서는 전체 무기체계가 획득 규격서의 운용 및 지원성 요구조건을 만족할 수 있게 하도록 소프트웨어 개발 시 ILS 요소도 고려해야 한다고 제시하고 있다[14]. 또한 PAM 700-56에서 ILS 개발 방안을 수립하는 데 필요한 소프트웨어 정비개념에 대한 사례를 제시하고 있다[4]. Table 4는 영국 국방성, ILS 핸드북 등에서 제시한 소프트웨어 ILS 요소를 정리하였다.

[Table 4] The main factors of ILS

Factors	
Hardware ILS	<ul style="list-style-type: none"> · Research & applying design · Maintenance plan · Technical manual · Logistics manpower operation · PHS&T · Technical data management · Standardization & compatibility · Support equipment · Supply support · Logistics support training · Maintenance and supply
Software ILS (DEF-STA N 00-60)	<ul style="list-style-type: none"> · Change Traffic · Safety Integrity · Expansion · Capability · Fleet Size and Disposition · Modularity · Size · Security · Skills · Standardization · Technology · Tools and Methods · Documentation
Software ILS (ILS Handbook)	<ul style="list-style-type: none"> · Reliability · Maintainability · Maintenance plan · Stability · Ergonomics · Support equipment · Personnel · Facilities · Technical manual · Training · Supply support

Table 4에서 볼 수 있듯이, 소프트웨어 ILS 요소는 제시하고 있는 규정 및 자료에 따라 상이하고, 다양함을 알 수 있다. 예를 들면 미 육군 규정은 소프트웨어 지원 요소를 소프트웨어 개발 단계에서 고려해야 하는 항목에 초점이 맞춰져 있다면, ILS 핸드북은 하드웨어와 유사하게 운용 및 정비에 중점을 두고 요소가 제시되어 있다.

다양한 소프트웨어 지원 요소 중에서 개발하는 무기 체계에 대한 소프트웨어 ILS 요소 개발의 방향 및 범위는 Fig. 4에서 초기에 지원개념(Support Concept)이 설정된 바와 같이 무기체계 소프트웨어의 정비개념에 의해 결정된다. 즉 초기에 지원개념 수립 시 정비개념도 포함하여 시스템의 설계와 개발에 앞서 필수적으로 정의되기 때문이다[7]. 따라서, SW의 초기 정비개념을 어떻게 세울 것인가 요소개발의 시작점이 된다. 따라서 본 장에서 검토된 소프트웨어 ILS 요소를 바탕으로 5장에서 소프트웨어에 대한 정비개념을 제안하고, 정비개념에 따른 ILS 요소 개발 방안을 제시하였다.

3.2 국내 무기체계의 소프트웨어 ILS 개발 사례

국내 무기체계의 소프트웨어 ILS에 대한 개발은 어떤 항목에 대하여 어느 수준으로, 어떠한 방법을 적용하여 개발하고 있는지를 검토하였다. Table 5는 국내 3군의 소

프트웨어 ILS 개발 사례를 검토한 결과를 나타낸 표이다.

[Table 5] The recent developments of softwares on domestic weapon systems

	Weapon system	Development item
Army	OO Rocket System	· Reinstalling software
	OO control system	· Development of factor related in software
Navy	OO radar system	· Reliability prediction of software · Reinstalling software
	OO submarine system	· Reliability prediction of software
	OO combat system	· Reliability prediction of software
	OO combat system	· Reliability prediction of software
Air force	OO system	· Development factor of training equipment · reinstalling software and Applying a technical manual related in network status check

국내에서 개발되고 있는 무기체계의 소프트웨어 ILS는 주로 소프트웨어 시험 결과를 활용한 신뢰도 예측, 운용 소프트웨어 재설치에 따른 일부 ILS 요소개발 업무를 수행하는 것으로 검토되었다.

예를 들어, 육군 OO 통제장비 체계 사업에서는 연구 및 설계반영, 표준화 및 호환성, 정비계획, 군수지원 교육, 기술교범, PHS&T(Packaging, Handling, Storage & Transportation), 기술자료 관리에만 한정하여 소프트웨어 ILS 개발방안을 제시하였다. 그리고 소프트웨어 신뢰도 예측은 주로 결함밀도 값을 이용한 분석을 통해 크게 세 가지 목적을 두고 수행하였다.

- 1) 개발품의 완성도 예측 및 시험을 통한 제품의 신뢰도 예측
- 2) 소프트웨어 시험의 종료 시점 결정(특정 신뢰도를 가지려고 수행해야 하는 시험시간 결정)
- 3) 시스템 운영 중 소프트웨어 패치 주기 결정

LSA는 앞서 관련 규정 및 자료 검토 시 일부 수행방안이 제시되었으나, 국내 무기체계 개발 시 실제로 수행된 사례는 없었다.

4. 소프트웨어 ILS 개발 범위 및 수준

3장에서 국내/외 규정 및 관련 자료 검토 결과와 같이 소프트웨어 ILS의 중요성 및 필요성이 증대함을 확인하였지만, 실제 국내에서 개발된 무기체계의 소프트웨어 ILS 요소 개발에 대한 사례 분석 결과, 소프트웨어 ILS 개발 범위, 적용 방법 및 분석 기준들이 무기체계별로 상이하였고, 적용 범위도 한정적이었다. 특히, 소프트웨어 ILS 요소의 개발방향과 범위를 결정하는 정비개념이 각 규정과 사례에서 상이함을 확인할 수 있다.

정비개념은 LSA와 요소개발의 입력 자료로 활용되어 소프트웨어 ILS 요소의 개발 범위를 결정짓는 핵심요소이기 때문에, 시스템 개발 초기에 정의되어야 한다. 2장에서 정의한 바와 같이 소프트웨어의 지원업무는 소프트웨어의 수정업무도 포함되어 있다. 그러나, 야전에 하드웨어의 고장정비와 같이 소프트웨어 수정을 통해 결함을 수정하는 정비개념을 그대로 적용하는 것은 소프트웨어 특성과 현재 국내 지원체계의 여건상 제한점이 있다.

4장에서는 하드웨어의 정비개념을 동일하게 적용할 수 없는 사유를 기술적, 구조적 측면에서 검토하여 소프트웨어 정비개념을 정립하기 위한 근거를 마련하고자 한다.

4.1 기술적 측면

4.1.1 소프트웨어 수정에 따른 고장률 증가

하드웨어는 Fig. 3과 같이 초기 제조과정에서 고장률이 높게 나타나고, 안정화 단계에 접어들면서 하드웨어 수명이 다할 때까지 낮아진다. 이후 내구수명이 도래하는 시점에 고장률은 다시 증가한다. 이처럼 시간에 따른 하드웨어의 고장률 변화는 'Bathtub curve' 형태로 나타난다. 반면, 소프트웨어는 하드웨어와 달리 물리적 또는 환경적인 이유로는 문제가 발생하지 않는다. 즉, 소프트웨어는 이론적으로 낡거나 마모되지 않는다.

그럼에도 불구하고, 소프트웨어 수정 과정에서 소프트웨어의 고장률 변화는 하드웨어의 'Bathtub curve' 형태와 유사하면서도 다른 복합적인 형태로 나타난다. 소프트웨어는 개발과정을 거쳐 안정화 단계가 되면 낮은 고장률을 그대로 유지해야 한다. 하지만 소프트웨어를 운용하기 시작하면 잠재적인 결함을 수정하거나, 사용자 요구에 따른 소프트웨어 변경 혹은 수행 능력을 향상하기 위한 소프트웨어 업그레이드 등 소프트웨어 수정 활동이 일어난다[6].

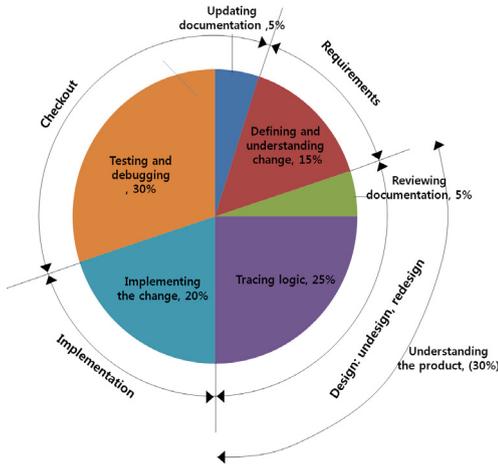
하드웨어는 유지보수가 제대로 이루어지지 않아 문제가 발생하지만, 소프트웨어는 유지보수로 인해 문제가 발생한다. 즉, 소프트웨어를 변경함으로써 'side-effects'가 생긴다. 하드웨어는 어느 한 부분이 문제가 발생하면 이를 대신할 만한 다른 부품으로 교환할 수 있지만, 소프트웨어는 전체적인 설계를 수정해야 하는 문제가 발생한다. 이 경우에는 소프트웨어 설계 수정에 따라 기존의 소프트웨어 구조가 약해지는 문제가 발생할 수 있다. 하지만 이러한 문제를 바로잡고자 소프트웨어 구조에 변경을 가하면 기존 소프트웨어와의 호환 문제나 개발자와 정비자의 로직에 대한 이해의 상이함으로 소프트웨어 결함이 발생할 수 있다. 따라서 소프트웨어는 야전 수준에서 소프트웨어 수정 시 부작용이 발생할 수 있다.

4.1.2 소프트웨어 수정에 따른 체계 간 연동

21세기 무기체계는 네트워크 중심의 NCW (Network Centric Warfare)를 기본으로 한다. 즉, 전투 공간에서 파악 가능한 모든 요소를 네트워크로 종합 및 연계하여 정보를 확보하고 전투력으로 전환하는 것이다. 이러한 상황에서 특정 무기체계를 임의로 변경 또는 수정하는 경우, 연동된 다른 장비나 체계에 영향을 미치게 된다[9]. 특히, 동일한 무기체계임에도 불구하고 한쪽 소프트웨어의 소스코드를 변경하여 다른 버전이 된 경우, 통신규약 또는 명령어 충돌에 인한 네트워크 상의 문제가 발생할 수 있다. 대개 무기체계는 한 번에 양산되는 것이 아니라 길게는 수년간의 양산 기간이 있으며, 이를 배치하는 데도 일정 기간이 소요된다. 따라서 기 배치되어 운용 중인 무기체계의 소프트웨어를 변경하는 경우, 기존 양산중인 체계와 다른 버전이 되는 것은 큰 문제를 야기할 수 있다. 게다가 더 큰 문제는 이러한 영향을 정확히 분석하는 데 제한이 있다는 것이다. 그러므로 소프트웨어를 수정하는 것은 기술적 검토를 통하여, 필요 시 개발하는 것과 동일한 수준의 기능 및 구조에 대해 체계적으로 시험하고 검증하는 절차를 거쳐야 한다.

4.1.3 소프트웨어 개발과 유지보수와의 관계

소프트웨어 유지보수는 소프트웨어 개발 업무와 유사한 절차에 따라 수행된다. Fig. 5는 소프트웨어 개발 및 유지보수 절차 간 관계를 나타낸 것이다[6].



[Fig. 5] Relationship between software development & maintenances activities

이는 소프트웨어 수정 프로세스가 하드웨어의 정비와는 상이하게 소프트웨어 재개발(Software Re-engineering)이라는 것을 의미한다. 이것은 기 개발된 설계 및 구현 현황과 시스템 측면의 연관성을 파악하지 못한다면 유지보수가 불가능하며, 유지보수가 진행된다 하더라도 추가적인 결함이 주입될 수 있어 오히려 잠재적인 고장 및 오류를 증가시킬 수 있다[6].

따라서 야전에서 수행하는 무기체계 운용유지 간 소프트웨어의 수정은 최소화하는 것이 바람직하다고 판단된다. 따라서, 무기체계 운용유지 시 유지보수 요인이 최대한 발생하지 않도록 개발 단계에서 소프트웨어 시험 등을 통한 최적화 설계에 주안점을 두어야 하며, 이러한 개념을 기반으로 소프트웨어 ILS 방안을 수립하여야 한다.

4.2 구조적 측면

4.2.1 소프트웨어 전문 조직 부재

미 공군은 지난 1987년부터 소프트웨어 기술 지원 센터(A.F.STSC)를 설립하여 무기 및 네트워크 체계의 소프트웨어 품질 향상, 효율성 제고, 시스템 개선 등 소프트웨어 전문 조직을 운영하고 있다. 또한, 소프트웨어 기술 회의(STC)를 개최하고, IEEE와 연계하여 지속적인 교육/훈련을 수행하고 있다. 그러나 우리 군은 육군전산소, 국군지휘통신사령부, 정보체계관리단을 운영하면서 소프트웨어 기술 지원보다는 인사, 군수 프로그램, 국방망, 관제시스템 등 네트워크 보안 및 DB 운용/관리 임무를 수행하고 있다. 또한 전문적인 소프트웨어 관련 조직이

없어 무기체계 소프트웨어의 개발부터 폐기까지 형상관리, 기술지원 등이 제한되고 있다.

4.2.2 인력양성 기관 및 체계 부재

현재 우리 군의 소프트웨어와 관련 있는 인원은 정보보호기술병과 소프트웨어 개발병이 있다. 정보보호기술병의 임무는 육군본부, 군단급 이상 부대에서 컴퓨터 정보보호 관련 임무를 수행하는 것이고, 소프트웨어 개발병의 임무는 전산프로그램 개발(프로그램 관리 및 시스템 설계, 정보보호, 인터넷 디자인)이다. 그러나 프로그래밍 교육이 기초 수준에 국한되어 있고, 교육과정 이수 후에도 홈페이지 관리, 인터넷 보안 요원 등의 프로그래밍과는 무관한 임무를 수행하고 있다. 따라서 무기체계 소프트웨어를 다루기에는 적합하지 않다. 육군 통신학교의 전산특기 인원 또한 주로 운영체제, DB 서버관리 및 운용, 정보체계 운용/관리 등에 대한 교육을 받고 있다. 즉, 프로그래밍이 아닌 네트워크 보안 및 DB 운용/관리 교육 중심이다. 결론적으로, 무기체계 소프트웨어를 보수하거나 유지보수 가능한 인력양성 기관 및 체계 등의 인프라가 부족한 실정이다.

5. 소프트웨어 ILS 적용 방안

3장에서 검토한 바와 같이, 무기체계 개발 시 소프트웨어 ILS는 개발의 필요성에 대해서 인식하고 있으나, 국내 환경에 부합되는 소프트웨어 ILS 요소개발 방안이 정립되어 있지 않다. 따라서 5장에서는 앞서 제시한 기존 사례검토와 제한 사항 검토 결과를 통해 국내 여건에 맞는 소프트웨어에 RAM, LSA, ILS 요소 개발의 현실적인 개발방안을 제시하고자 한다.

5.1 RAM 개발방안

전통적으로 RAM 분석은 OMS/MP에 근거하여 수립된 정량적 RAM 목표값 달성 여부로 그 결과를 판단하게 된다. 또한 RAM 분석 결과는 주장비 설계단계에서 신뢰성, 정비성 측면에서의 설계개선 근거가 되고, LSA의 기초 입력 자료가 되어 ILS 요소의 정량화 개발의 근거 자료로 활용된다.

하드웨어에 대한 RAM 분석 기준, 절차 등은 기정립되어 수행되고 있다. 하지만 소프트웨어 RAM 분석에 대

해서는 그 방법론에 대한 연구가 미흡한 실정이다. 따라서 소프트웨어에 대한 RAM 분석은 현재 국내에 활용되고 있는 하드웨어에 대한 RAM 업무절차와 동일하게 수행할 수는 없다.

또한, MIL-HDBK-338B에 제시된 소프트웨어 신뢰도 예측은 하드웨어 신뢰도 예측과는 상이하며, 현재 일부 무기체계에서 관련 내용을 적용하여 개발한 사례가 있다 [12]. 그러나 실질적인 소프트웨어 신뢰도 분석을 위해서는 소프트웨어 신뢰도 예측 및 추정 모델, 목표값 설정, 적용 방법 등에 대한 추가적 연구가 향후에 지속적으로 수행되어야 한다.

또한 소프트웨어 신뢰도 향상은 무기체계 규모와 사업 특성을 고려, SW를 실행하지 않고 코드 수준에서 결함을 검출하는 정적 시험과 Test Case별 SW를 실행하면서 결함을 검출하는 동적 시험을 수행하는 과정에서 이루어지고 있으며, 이를 통하여 소프트웨어의 품질을 향상할 수 있다.

무기체계 소프트웨어 개발 및 관리 매뉴얼에 의하면 각 사업단계별 소프트웨어 신뢰성 확보 활동을 수행하고, 결과에 대하여 시험평가 시 확인하도록 되어 있다[15]. 또한 무기체계 소프트웨어 개발 및 관리 실무지침서에는 소프트웨어 신뢰성 시험 수행에 필요한 항목(시험 종류, 절차, 시험기준, 시험도구 및 평가 기준)을 세부적으로 제시하고 있다[10].

따라서, RAM 분석에 대한 현시점에서의 실질적 방안은 소프트웨어 시험을 통한 신뢰도 향상 활동에 주력하여, 실질적인 소프트웨어 신뢰도를 확보하는 것이다.

5.2 LSA 개발방안

LSA는 무기체계의 ILS 요소를 통합하고 배치 및 운영유지 단계에서 군수지원 요소를 식별하며 이를 구체화하는 데 필요한 분석 자료를 제공한다. 3장에서 검토한 바와 같이 소프트웨어 LSA 대한 절차와 필요성이 제기되고 있으며, 소프트웨어 ILS 요구사항 분석, 소프트웨어 ILS 요소 식별 등의 지원계획 수립을 위해 하드웨어 LSA의 절차에 포함되어 진행되어야 할 것이다.

LSA 활동에서 가장 먼저 수행해야 할 활동은 군수지원 대상 품목을 선정하는 것이다. 소프트웨어의 군수지원 대상 품목 선정은 해당 소프트웨어의 구조, 초기 정비 개념 분석을 통해서 가능하다. 또한, 선정된 품목에 대한 LCN(LSA Control Number) 부여는 기존의 하드웨어

LCN에 포함되어 부여하되, LSA영국 규정(Ministry of Defence Standard 00-60)에서 제시된 바와 같이 소프트웨어의 타입에 따라, Lodable(일반적인 탑재 가능)과 Resident(내장형)를 구분하여 부여해야 한다[11].

현재 국내에서 적용 중인 LSA 주요활동 중, 소프트웨어 특성상 RCM은 LSA활동에서 제외되어야 하며, FMECA는 제한적으로 적용되어야 한다. 영국 규정(Ministry of Defence Standard 00-60)에서도 FMECA의 수행의 필요성에 대해서 제시되어 있고, 미국 신뢰성 관련 지침서인 MIL-HDBK-338B에서도 FMECA에 대해서 간략하게 수록되어 있다[12]. 그러나 FMECA를 그대로 적용하기에는 분명 제한사항이 존재한다. FMECA의 목적의 하나인 치명도 분석을 통해 치명 품목을 선정하여 신뢰성 향상(설계변경)하는 활동은 소프트웨어 설계 완료시점과 소프트웨어의 복잡성의 문제로 제한되기 때문이다[6].

즉, 설계 과정에서 FMECA를 통해 취약한 설계 부분을 보완하는 목적의 분석이 아닌, 소프트웨어로 인한 고장유형을 식별하고, 발생한 고장에 대해 어떠한 지원계획을 수립할 것인지에 대한 분석과정으로서의 LSA가 수행되어야 할 것이다. 또한, 소프트웨어로 인한 고장유형이 식별되면, 이에 대한 지원방법들을 분석하고, 이를 실제로 확인하는 정비업무분석을 수행하여 지원계획의 적절성에 대한 검토가 이루어져야 한다.

하지만, LSA 수행을 위한 전제조건은 LSA 수행 시 적용되는 정량적 고장 소요의 기초인 신뢰도 산출 방안이 먼저 수립되어야 한다는 것이다. 따라서, 정량적 신뢰도 산출 방안에 대한 연구에 가시적 성과가 나타난다면, 이를 기반으로 LSA 수행은 기존 수행 프로세스 내에 통합하여 수행이 가능할 것이다.

5.3 ILS 요소 개발방안

ILS 요소는 무기체계 수명주기 간 무기체계를 효과적, 경제적으로 운용 유지할 수 있도록 군수지원을 보장해주는 제반 사항으로 LSA 결과를 토대로 개발된다.

4장에서 검토한 바와 같이 현시점에서 소프트웨어의 성능개선과 소프트웨어 결함 수정을 위한 야전에서 소프트웨어 수정은 제한됨을 확인하였다. 따라서 소프트웨어의 야전에서 유지보수수준은 소프트웨어가 포함된 하드웨어의 정비개념을 기준으로 분석되어야 하며, 무기체계의 특성에 따라 소프트웨어의 보안성, 연동성, 복잡성 등

을 고려하여 소프트웨어의 제거/설치, 업로드/다운로드 까지 고려하여 분석되어야 한다.

소프트웨어의 정비개념을 고려한 소프트웨어 ILS 요 소개발 방안은 아래와 같다.

- 연구 및 설계반영
 - 신뢰성, 정비용이성, 군수지속성, 운용성, 보안성 및 안정성 등을 고려한 소프트웨어 설계 반영 활동
- 표준화 및 호환성
 - 표준화된 개발 프로세스 및 코딩 지침 준수 확인
 - 확장성(이식성)을 고려한 소프트웨어 설계반영 활동
- 정비계획
 - 소프트웨어 정비지원을 위한 정비계단 구분, 단계별 정비방침 및 책임 등의 정비개념 수립 및 지원요소 분석/개발
 - 소프트웨어 정비능력 확보를 위한 정비개념 수립
- 지원장비
 - 정비단계별 소프트웨어 유지보수(재설치 등)를 위한 지원장비 및 치구 개발
 - 소프트웨어 오류 및 정비 실적 등 데이터 관리 소프트웨어 개발
- 기술교범
 - 소프트웨어 운용 및 유지를 위한 소프트웨어 유지보수 방법 수록
- 보급지원
 - 소프트웨어 정비개념을 고려한 보급 품목 식별
 - 소프트웨어 재설치 프로그램(CD, USB 등), 정비장비(SW 재설치 프로그램 탑재), 치구 등
- 군수인력운용
 - 소프트웨어 정비를 위한 정비인력 소요 산출
- 군수지원교육
 - 소프트웨어 정비 관련 교육을 하드웨어에

포함하여 교육 계획 수립

- PHS&T
 - SW의 포장, 취급, 저장 및 수송 기준 수립
 - 소프트웨어 재설치 프로그램(CD, USB 등) 보급 기준 포함
- 정비 및 보급시설
 - 소프트웨어 정비 및 보급시설 소요 검토
- 기술자료관리
 - 기술자료 묶음(소프트웨어 규격화 자료 포함) 제공
 - ILS 기술자료(SW ILS 내용 포함) 제공

6. 결론

본 연구는 소프트웨어 ILS의 실무적 적용분야에서 문제가 발생하고 있는 현재의 실태를 제시하고, 하드웨어 ILS와의 비교를 통해 문제점들을 정리하였다.

기존 연구에서는 소프트웨어 ILS 개발 방안을 제시하였지만, 기존의 하드웨어의 ILS 절차와 산출물을 그대로 적용하는 한계점이 있었다. 따라서 선행연구 사례를 검토하고 현실적인 소프트웨어 ILS를 위해 정비개념을 검토하여, 소프트웨어에 대한 실질적인 RAM, LSA, ILS 요 소개발 범위와 방안을 제시하였다.

이를 통해 하드웨어와 함께 소프트웨어 분야에서도 ILS 개발 시 표준화된 틀을 형성하여 개발자와 정비자 모두가 효율적으로 소프트웨어에 대한 지원을 수행할 수 있을 것이다. 무엇보다 현재 소프트웨어 ILS의 중요성을 인지하고 있지만, 실무에서 적용되지 못하고 있는 문제를 현실적으로 해결하여 빠른 시일 내에 실무에 적용하는데 기여할 수 있을 것으로 기대한다.

향후에는 현재의 연구결과를 기초로 실무에 적용할 수 있는 더욱 구체화된 방안을 제시하고, 이를 토대로 소프트웨어 ILS 항목들을 실질적으로 검증하는 작업이 필요하다.

References

- [1] Security Management Institute, Analysis of Reality and Research for the Expansion Plan on Localizing Software, 2011. 12.
- [2] DAPA, How to do acquisition management of weapon embedded software, 2007. 8.
- [3] S. Ghasemi, Shooting Down Iran Air Flight 655, http://www.iranchamber.com/history/articles/shootingdown_iranair_flight655.php, 2004.
- [4] Department of the Army, Pamphlet 700-56, Logistics Supportability Planning and Procedures in Army Acquisition, 2006. 4. 21.
- [5] DAPA, DAPA Anweisung No. 270, Defense Acquisition Regulation, 2014. 2.
- [6] Department of The Air Force Software Technology Support Center, Guidelines for Successful Acquisition and Management of Software-Intensive Systems, Version 3.0, Chapter 12, 2000. 5.
- [7] BENJAMIN S. BLANCHARD, Logistics Engineering and Management, Sixth Edition, 2003.10.31
- [8] Security Management Institute, Integrated Logistics Support Study on Weapon System Software, 2012.12.
- [9] T. I. Jung, G. Y. Lee, I. S. Ryu, C. S. Jung, Policy Approaches to Development Management for Software in Weapon Systems. *Conference on the Korean Society for Aeronautical and Space Sciences*, 1120-1123, 2013. 4.
- [10] Defense Acquisition Program Administration, A guideline for development and management in Weapon system software, 2013. 12. 27.
- [11] Ministry of Defence Standard 00-60, Part 3, Guidance for Application of Software Support, 1998. 3. 31.
- [12] MIL-HDBK-338B, Electronic Reliability Design Handbook, 1998. 10. 1.
- [13] JAMES V. JONENS, Integrated Logistics Support Handbook, Third edition, Chapter 9, 2006.
- [14] Department of the Army, Army Regulation 700-127, Integrated Logistics Support, 2008. 7. 17.
- [15] Defense Acquisition Program Administration, A manual for development and management in Weapon system software, 2014. 2. 19.

이 관 영(Kwan-Young Lee)

[정회원]



- 2002년 2월 : 성균관대학교 산업시스템경영학과 (공학학사)
- 2011년 1월 ~ 현재 : LIG넥스원 선임연구원

<관심분야>
중합군수지원, 신뢰성 공학

김 상 문(Sang-Moon Kim)

[정회원]



- 1995년 2월 : 성균관대학교 산업시스템경영학과 (공학학사)
- 1995년 6월 ~ 현재 : LIG넥스원 수석연구원

<관심분야>
중합군수지원, 신뢰성 공학

박 은 심(Eun-Shim Park)

[정회원]



- 2002년 2월 : 중앙대학교 전자전기공학부 (공학학사)
- 2002년 4월 ~ 현재 : LIG넥스원 선임연구원

<관심분야>
중합군수지원, 신뢰성 공학

박 재 은(Jae-Eun Park)

[정회원]



- 2008년 8월 : 연세대학교 경영정보학과 (경영학사)
- 2013년 2월 : 고려대학교 정보경영전문대학원 정보경영공학과 (공학석사)
- 2013년 1월 ~ 현재 : LIG넥스원 연구원

<관심분야>

종합군수지원, 인간공학

김 근 형(Geun-Hyung Kim)

[정회원]



- 2009년 8월 : 인하대학교 산업공학과 (공학학사)
- 2012년 2월 : 고려대학교 정보경영전문대학원 정보경영공학과 (공학석사)
- 2013년 1월 ~ 현재 : LIG넥스원 연구원

<관심분야>

종합군수지원, 신뢰성 공학, 생산관리