

한국형 합동 전술데이터링크체계의 소프트웨어 신뢰성 시험수행 방안에 관한 연구

송영중* · 이윤정** · 구영훈***

A Study on the Weapon System Software Reliability Testing for the Joint Tactical Data Link System Project Case

Young-Joong Song* · Youn-Jeong Lee** · Young-Hoon Goo***

요 약

본 연구는 한국형 합동 전술데이터링크 체계 사업의 사례를 통해 무기체계 소프트웨어 개발 및 관리 매뉴얼에서 규정하고 있는 무기체계 소프트웨어 신뢰성 시험이 정의되기 전에 획득한 소프트웨어 자산을 안정적으로 무기체계에 통합하는 방안을 제시한다. 이를 위하여 무기체계 소프트웨어 신뢰성 시험에 대한 개요와 한국형 합동 전술데이터링크체계 사업의 특성에 따른 신뢰성시험 적용의 문제점을 기술하고, 각 문제점에 대한 사업의 결정안과 이를 통한 기대 효과를 기술한다.

ABSTRACT

This study presents the method for integrating the software assets acquired before the weapon system software reliability test was not formed in the weapon system software development and management manual through JTDLS Batch II case. This paper describes the problems for applying manual's direction to JTDLS Batch II project, and decisions and expected effects.

키워드

Joint Tactical Data Link System Batch II, Software Testing, Software Quality Management
한국형 합동 전술 데이터링크, 무기체계 소프트웨어 신뢰성 시험, 소프트웨어 품질 관리

1. 서 론

무기체계 소프트웨어는 일반적인 소프트웨어와 달리 사소한 결함으로도 발생하는 막대한 인명피해 및 경제적인 피해로 높은 수준의 품질이 요구된다. 특히

최근 무기체계에서 소프트웨어가 차지하는 비중이 증가함에 따라, 소프트웨어는 무기체계의 기능과 성능을 결정하는 요인이 되었다 [1].

이에 따라 방위사업청은 무기체계 소프트웨어의 신뢰성 확보를 위하여 무기체계 소프트웨어 개발 및 관

* 한화시스템 전문연구원(yungjoong.song@hanwha.com)

** 국방과학연구소 지휘통제체계단 (youn@add.re.kr)

*** 교신저자 : 국방과학연구소 지휘통제체계단

• 접수일 : 2022. 06. 22

• 수정완료일 : 2022. 07. 17

• 게재확정일 : 2022. 08. 17

• Received : Jun. 22, 2022, Revised : Jul. 17, 2022, Accepted : Aug. 17, 2022

• Corresponding Author : Young-Hoon Goo

Dept. Command and Control System PMO Team 1, Agency for Defense Development,

Email : gyh0808@add.re.kr

리 매뉴얼을 통하여 무기체계 소프트웨어의 체계적인 개발 및 관리를 위한 세부 기준, 절차, 방법과 함께 프로세스를 제시하고 있다 [2].

다만 해당 매뉴얼은 지속적인 개정을 통해 그 내용이 보강되고 있음에도 불구하고 각 무기체계 획득사업의 특성이 반영되지 못한 부분이 존재한다.

본 논문은 한국형 합동 전술데이터링크체계(JTDLS)를 구체적인 사례로 선정하여 매뉴얼이 제시하고 있는 소프트웨어 신뢰성 시험 방안에 대해 기술하고 이에 대해 해당 사업에서 결정된 적용 안을 기술하고 검토한다. 이를 통해 추후 진행될 유사 사업의 소프트웨어 신뢰성 시험의 정책 결정에 참고 사례로 활용될 수 있도록 하고자 한다.

이를 위하여 본 논문은 2장에서 무기체계 소프트웨어 신뢰성 시험에 대한 개요, 한국형 합동 전술데이터링크체계 사업의 개요를 기술한다. 3장은 한국형 합동 전술데이터 링크체계 사업에 대한 무기체계 소프트웨어 신뢰성 시험 적용의 어려움과 이를 해결하기 위한 방안을 기술한다. 4장은 제한된 시험자원을 보다 효과적으로 활용하여 무기체계 소프트웨어 신뢰성시험을 수행하는 방안을 기술한다.

II. 관련 연구

2.1 무기체계 소프트웨어 신뢰성 시험

무기체계 소프트웨어 신뢰성 시험이라는 용어는 무기체계 분야에서 사용되는 고유한 용어로 소프트웨어 테스트 분야에서 쓰이는 정적·동적 시험 개념을 무기체계 분야에 적용하면서 생성된 신조어이다. 무기체계 소프트웨어 개발 및 관리 매뉴얼에 따르면, 무기체계 소프트웨어 신뢰성 시험은 소프트웨어가 동작할 수 있는 다양한 경우의 수를 확인함으로써 소프트웨어가 일으킬 수 있는 결함을 식별하는 시험을 말한다.

시험은 프로그램을 실행하지 않고 코딩 규칙 준수 여부 및 실행시간 중에 발생할 수 있는 오류를 예방하는 정적시험과 실제 하드웨어(Target)에 탑재한 상태에서 소프트웨어 통합시험 절차서에 기술된 시험절차에 따라 요구사항 기반으로 소프트웨어 코드 실행

를 점검하는 동적 시험으로 나눈다. 이들 시험은 각각 SW 구현/통합 및 시험 단계, 시험평가 단계, 규격화 단계, 양산 단계별로 자동화 도구를 사용하여 실시한다. 시험 대상언어는 현시점 자동화 도구가 지원하는 C, C++, Java, C#을 원칙으로 하고 있다. 특히, 16년 7월 개정은 체계의 특성에 가장 적합한 국제 표준 준수를 원칙으로 하여 정적시험 및 동적시험의 요구 수준 및 사항이 강화되었다. 그 결과 C와 C++ 언어를 사용할 경우, 기존의 방위사업청에서 선정한 코딩 규칙(65개)을 준수하던 수준을 넘어 MISRA-C(143개 규칙), MISRA-C++(228개 규칙)의 규칙에 대한 점검이 요구된다. 또한 소프트웨어의 품질 측정 지표를 선정하여 코드의 형상을 매트릭 내에서 유지하도록 요구하고 있다.

동적시험은 단순한 구조적인 실행를 점검이 아닌 요구 기능과 결합하여 결합의 발생빈도, 영향성 및 제어 가능성을 평가한 후, 그 결과를 바탕으로 코드 실행률 측정 방법(문장, 분기, MC/DC 커버리지)을 결정하고 시험 대상 소프트웨어의 시험케이스를 시험 후, 측정된 실행률을 점검한다. 이때, 미충족 부분에 대해 소명서를 작성하도록 요구되고 있다.

2.2 한국형 합동 전술데이터링크체계(JTDLS)

한국형 합동전술데이터링크체계(완성형)는 2014년 완료된 무기체계 간 실시간 전술정보 교환 및 상황인식을 공유하기 위해 국내 독자 개발한 전술데이터링크 체계 획득사업이다. 한국형 합동전술데이터링크체계(기본형) 보다 전송능력을 향상하고, 전송 자료, 문자 이외에도 정지영상 및 음성의 전송 기능과 대전자전 기능이 갖춰진 완성형 체계를 개발하는 사업이다.

본 사업은 소프트웨어 측면에서 선행사업 완료 후, 한국형 합동 전술데이터링크체계가 지원하는 각 전술데이터링크(Link-16 등)의 메시지 프로토콜 표준문서와 서로 다른 데이터링크 간의 메시지를 서로 교환할 수 있도록 규정한 중계 표준문서의 최신 개정 본을 적용하며, Link-11의 교체를 대비한 신규 데이터링크(Link-22)의 체계 내 통합을 위한 개발을 진행하는 사업이라고 할 수 있다.

본 사업은 국방과학연구소 주관으로 한국형 합동전술데이터링크(Link-K) 단말기 시제품을 개발하여 지상, 함정, 차량, 항공기 및 헬기 등의 플랫폼과 통합하여 시험평가 및 규격화를 통해 전력화를 목표로 하고 있다. 현재 해당 사업은 요구사항 검토, 주요 설계, 상세설계를 지나 소프트웨어 및 하드웨어의 제작 단계에 들어서 있다 [3~5].

III. 무기체계 소프트웨어 신뢰성 시험 적용 방안

3.1 사업의 특성과 제한사항

앞서 기술한 바와 같이 한국형 합동 전술데이터링크체계 사업은 지상/해상/공중 플랫폼에 탑재되는 한국형 합동 전술데이터링크체계를 국내 개발하여 전력화하는 시제로, 한국형 합동 전술데이터링크체계사업(기본형)의 성능 개량이 기본이 되는 사업이다. 따라서 성능 향상과 기능 확장을 위하여 변화된 최신 사양의 하드웨어를 신규로 개발 및 교체할 필요가 있다. 반면, 소프트웨어는 획득하게 되는 무기체계의 신뢰성 호환성을 고려할 때, 선행사업에서 축적된 소프트웨어 자산의 적극적인 활용이 필요하다. 그에 따라 기본형 사업과 선도형 사업을 통해 축적된 개발 인프라와 소프트웨어 자산을 활용하여 진행되고 있다.

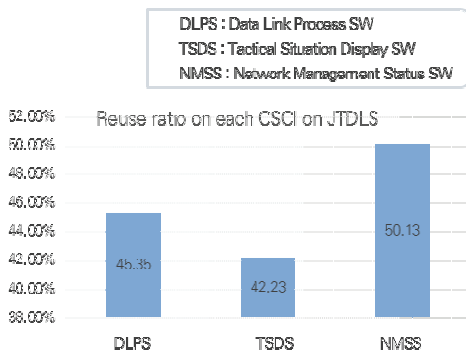


그림 1. 한국형 합동 전술데이터링크체계 소프트웨어 재사용률

Fig. 1 Software reuse ratio on JTDLS

3.2 시험 대상 선정

본 사업에 활용된 다수의 소프트웨어 자산은 이미 선행사업의 개발시험 평가 및 소요 군의 운용을 통해 객관적으로 기능과 성능이 입증된 신뢰성 높은 소프트웨어들이다. 하지만 해당 소프트웨어 중 일부는 무기체계 소프트웨어 신뢰성 시험이 규정되기 이전에 사업이 종료되어, 해당 시험을 시행한 이력이 없다 [6].

그 결과 매뉴얼에 따르면, 해당 소프트웨어들은 소프트웨어의 수정 없이 그대로 사용하더라도 신규 소프트웨어와 같이 신뢰성시험 대상 소프트웨어 대상으로 간주한다. 이 경우 우선 소프트웨어 신뢰성 시험 항목 중 정적시험을 수행할 경우, 다수의 규칙 위반이 발생하며, 코드 매트릭 결함도 발생하게 된다.

이때 결함이 검출된 소프트웨어의 수정이 지역적으로 한정되어 있다면, 충분한 검토를 통해 해당 오류를 수정하고 이를 적용하는 것이 가능할 것이다. 하지만 대표적으로 코드 매트릭과 같이 수정을 위해 전체 소프트웨어에 대한 구조 변경 및 재설계가 요구된다면 수정 후, 해당 소프트웨어에 대해 기존에 검증되었던 기능과 성능이 같은 것으로 보장할 수 없을 것이다. 이것은 매뉴얼이 규정한 소프트웨어 신뢰성 시험 중, 정적시험이 해당 소프트웨어에 대한 기능과 성능을 확인하는 것이 아니기 때문이다. 그 결과 정적시험 결과를 반영한 소프트웨어의 안정성은 소프트웨어에 대한 운용성 시험 및 평가(OT&E)를 통한 검증이 진행되지 않은 신규 개발 소프트웨어와 같은 수준으로 떨어진다고 볼 수 있다.

따라서 본 사업에서는 이러한 불합리한 사항을 고려하여 규격화 및 전력화되어 운용 중인 관급 소프트웨어도 기개발 소프트웨어로 분류하여 재활용(수정되는)되는 부분에 대해서만 신뢰성시험을 수행하는 것으로 방침을 정하였다.

3.3 시험 수준 정의

시험의 대상을 선정하는 문제 외에도 대상이 되는 소프트웨어의 시험 수준을 결정하는 것 역시 해소해야 할 문제이다. 이에 대해 무기체계 소프트웨어 개발 및 관리 매뉴얼은 소프트웨어 상세설계 단계의 산출

물인 소프트웨어 개발계획서에 개략적인 소프트웨어 정적 시험 계획 및 동적 시험 계획과 그 절차를 작성하게 되어 있다. 그리고 그 결과를 최종적으로 사업관리 회의를 통하여 결정토록 정의하고 있다.

매뉴얼이 요구하는 동적시험의 수준은 적용 커버리지를 결정하는 것으로 정의된다. 이는 설계 단계에서 요구사항을 바탕으로 설계된 소프트웨어의 위험 수준을 영향도, 발생빈도, 제어 가능성을 기준으로 예상 및 분석하여 결정할 수 있다 [표 1].

표 1. (일부) 동적시험 수준
Table 1. (Partial) Dynamic software test coverage

CSCI	CSC	CSU	Severity	Exposure	Controllability	Coverage Type
TSDS I	Tactical Situation Display	Map Display	S1	E1	C3	S
		Grid Display	S1	E1	C2	S
		Track Display	S2	E2	C2	S
	Track Mgmt.	Track Mgmt.	S2	E3	C2	S
		PPLI Mgmt.	S2	E3	C2	S
		Target Mgmt.	S2	E3	C2	S

정적시험의 수준은 적용 규칙 및 코드 매트릭 값으로 결정된다. 매뉴얼 역시 이 값을 사업의 특성에 맞추어 테일러링 하도록 요구하고 있다. 해당 값의 테일러링을 위해서는 다음의 사항을 고려해야 한다.

첫째, 소프트웨어 신뢰성 시험에 사용하는 도구 및 버전에 따라 시험 결과가 다르다. 현재 소프트웨어 신뢰성 시험도구에 대한 품질과 성능은 각 제조사에서 개별적으로 진행된다. 또한 지속적인 버전 상향에 따

라 그 성능이 변경된다. 따라서 같은 대상 소프트웨어도 어떤 신뢰성 시험도구의 어떤 버전을 사용했는가에 따라 검출되는 결함의 범위 및 수치가 달라진다. 즉, 실제 측정을 통한 확인 외에는 예측할 수 없다. 재활용, 재사용 소프트웨어는 기존 사업을 통해 완성된 형상이 존재하므로 시험도구를 이용하여 실제로 측정된 후, 이를 바탕으로 기준값을 정할 수 있다. 반면 신규 개발 소프트웨어는 이러한 형상이 존재하지 않기 때문에 실제 구현된 소프트웨어가 없는 설계 단계에서 테일러링을 위한 기준 및 근거를 제시하는 것이 매우 어려운 일이다. [7~8]

둘째, 사용자의 요구사항이 지속해서 변경되고 추가된다는 점이다. 특히 신규 개발하는 소프트웨어의 경우 큰 요구사항의 틀이 변경되지 않더라도 구현을 통해 구체적인 소프트웨어의 형상이 확인되면 이에 대한 사용자의 추가 요구사항(피드백)이 발생하게 된다. 그에 따라 개발 및 시험평가 기간에도 소프트웨어의 기능을 변경하는 요소가 되며 이에 따라 사전에 예측/계획하였던 규칙과 기준이 맞지 않게 되는 결과로 이어진다. 이를 방지하고자 이러한 추가 식별된 요구 기능을 반영하지 않는 것은 무기체계의 기능을 제약하는 결과로 이어지게 되므로 대부분 이를 반영하게 된다. 따라서 이처럼 설계를 통해 계획을 세운 후라도 소프트웨어 구현의 필요에 따라 해당 규칙 및 매트릭을 초과해야 하는 경우가 종종 발생할 수 있다.

이를 종합하여 본 과제는 다음과 같은 소프트웨어 신뢰성 시험의 정적시험 시행 기준을 세웠다. 재사용 및 재활용이 예상되는 모듈은 실제 검증에 사용하고자 하는 도구 및 버전을 명시하고 이를 이용하여 소프트웨어의 수정 전에 측정된 결과를 바탕으로 기준값을 지정하였다. 이를 바탕으로 테일러링이 필요한 규칙과 매트릭 항목에 관해서는 구체적인 사례와 수정 필요에 대한 근거를 제시한 후 이에 대한 검토를 통해 최종 적용 규칙을 도출하였다. 신규 개발 소프트웨어는 설계 단계에서 기준값 및 테일러링 값을 제시하는 것이 불가능하므로 원칙적으로 매뉴얼에 제시된 소스 코드 매트릭 수치를 준수하여 구현하고 검증 규칙도 예외 되는 규칙이 없도록 개발하는 것을 목표로

하되, 구현이 완료되는 시점에 신뢰성 시험도구를 이용하여 시험 결과를 도출하여 조정이 필요한 항목을 확인하여 이에 대해서 재활용 재사용 소프트웨어와 같이 기준을 조정하는 절차를 수행한다.

3.4 소프트웨어 수정에 대한 정의

엄밀히 따진다면, 기존 소스 코드를 바탕으로 어떤 변경 사항이 있다면 해당 소프트웨어는 재사용이 아닌 재활용 항목으로 신뢰성시험을 수행할 필요가 있다. 하지만 본 사업은 확보된 소프트웨어 자산의 재사용 및 재활용을 쉽게 할 수 있도록 소스 내의 주석을 추가하고, 변수명을 좀 더 명확히 하는 등의 “소프트웨어 기능과 성능에 영향을 주지 않는 것으로 판단되는 소프트웨어의 수정”은 변경되지 않은 것으로 판단하여 해당 수정으로 인한 신뢰성시험은 수행하지 않도록 하였다. 이를 통하여 기존 소프트웨어의 유지 보수성을 향상할 수 있을 것이다 [6].

IV. 무기체계 소프트웨어 신뢰성 시험 수행 방안

4.1 개요

이번 장은 체계의 성격을 고려하여 결정된 소프트웨어 신뢰성 시험에 대한 정책을 바탕으로 구축한 시험수행 환경과 각 신뢰성시험을 더욱 효과적으로 수행하기 위하여 이를 개선한 내용을 살펴보도록 하겠다.

4.2 시험 환경 구성

신뢰성시험은 자동화 도구를 사용하여 결함을 검출하도록 규정되어 있다. 본 사업에서는 소스 코드 형상관리 시스템과 신뢰성시험 수행 서버를 연계하여 시험의 진행이 자동으로 진행된 후, 개발자가 이를 확인할 수 있도록 시스템을 구축하였다.

4.3 정적시험 수행 고려사항

과거의 매뉴얼이 요구하던 시험 규정과 달리 각 체계의 특성에 적합한 국제 표준 규칙을 적용하는 것으로 방침이 변경되며, 시험에 드는 시간이 늘어났다.

물론 신뢰성시험의 시간 대부분은 여전히 분석이 아닌 코드의 수정에 들지만, 분석에 걸리는 시간 역시 크게 늘어 이에 대응하는 방안이 요구된다.

본 사업에 적용되는 정적시험의 코딩 규칙은 C 및 C++ 언어로 작성된 소프트웨어가 점검해야 하는 MISRA-C(143개 규칙), MISRA-C++(228개 규칙) 규칙이다. 이들 규칙은 코드의 스타일 내지 작성 규칙을 점검하는 것이다 [9~10].

코드의 스타일에 대한 규칙은 런타임 오류를 찾는 것에 비하여 분석과 수정이 상대적으로 간단하다. 일반적으로 수초 내지 수 분 내에 즉각 분석이 완료되므로 개발자가 소스 코드를 수정한 즉시 시험을 진행하여 결과를 확인할 수 있다. 하지만 본 체계 사업은 기존 체계 사업으로 획득한 소프트웨어 자산을 재활용하기 때문에 분석 대상 소프트웨어의 규모가 상대적으로 방대하여 (98만 SLOC) 모든 규칙을 동시에 검증할 경우, 분석의 완료까지 오랜 시간 개발자가 대기할 필요가 있어, 전체 규칙 중 분석과 수정이 쉬운 일부(C/C++ 용 규칙 각 19개)를 선별하여 상시 진행하는 것이 바람직한 것으로 판단하였다. 그 외의 분석 및 수정이 오래 걸리는 것으로 판단한 규칙은 개발자들이 도구를 사용하지 않는 심야 시간을 이용하여 자동으로 분석을 수행하고, 그 결과를 확인할 수 있도록 시스템을 구성하였다 [표 2] [그림 2].

표 2. 상시 점검 규칙 (MISRA C++ 2008, MISRA C 2012)

Table 2. Selected MISRA C++ 2008, MISRA C 2012 rules for regular testing

	Rules
MISRA C++ 2008 (19)	0-1-3, 0-1-7, 2-7-3, 2-10-1, 2-13-4, 3-2-3, 3-4-1, 3-9-2, 4-5-2, 5-0-4, 5-0-13, 5-2-1, 5-2-4, 5-3-1, 6-4-1, 6-4-2, 8-0-1, 8-4-2, 9-3-3
MISRA C 2012 (19)	2-3, 2-4, 2-5, 2-6, 2-7, 3-1, 3-2, 7-2, 7-3, 8-1, 12-3, 14-4, 15-6, 15-7, 16-4, 16-7, 20-1, 20-7, 20-8

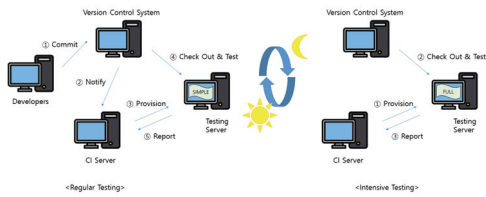


그림 2 신뢰성시험 시스템

Fig. 2 Weapon system software reliability testing system on JTDLS

코드 매트릭의 경우는 단순히 하나의 함수 및 기능에 가해지는 수치가 아닌 전체 소프트웨어의 구조에 영향을 미치는 값이다. 개발이 완료된 시점에서는 해당 규칙을 위하여 코드의 구조를 변경하는 것이 몹시 어렵다. 따라서 이 역시 앞서 기술한 시스템을 통해 자동화하여 지속해서 측정하여 그 결과를 개발자에게 전달하여 현재 작성하고 있는 코드의 품질 및 결함 요소에 대한 피드백을 제공하도록 하였다.

V. 결 론

본 연구는 한국형 합동 전술데이터링크 체계 사업의 사례를 통해 무기체계 소프트웨어 개발 및 관리 매뉴얼에서 규정하고 있는 소프트웨어 신뢰성 시험이 시행되지 않은 소프트웨어에 대한 재활용, 재사용 소프트웨어를 어떻게 안정적으로 무기체계 사업에 적용할 수 있을 것인지 고찰하였다. 부가적으로 사업을 통해 개발된 소프트웨어의 품질을 높이기 위하여 적용한 시험정책과 세부 방안에 대한 효과를 기술하였다.

이를 통해 기존 사업에서 시험 및 평가로 검증된 소프트웨어 자산을 바탕으로 안정적이고 신속한 소프트웨어 개발이 가능하다. 또한 한정된 개발 자원을 신규 및 수정이 이루어지는 소프트웨어에 대해 소프트웨어 테스트를 집중하여 한정된 자원으로 높은 수준의 신뢰성을 확보할 가능성을 만들었다.

본 연구 결과는 이외에도 한국형 합동 전술데이터링크 체계 사업과 같이 각 부대에 배포되어 안정적으로 사용되고 있던 다수의 소프트웨어를 기존의 많은 예산을 투입하여 운용 시험 및 평가 등을 통해 검증

된 소프트웨어의 안정성을 헤치지 않고 현재 개발하는 체계 사업에 적용할 수 있는 사례를 남김으로써, 많은 사업에서 무기체계 소프트웨어의 재시험 및 검증으로 중복 소요되는 비용 요소를 제거할 근거가 되어, 무기체계 소프트웨어의 획득 비용을 낮출 수 있을 것으로 전망한다.

Acknowledgement

본 논문은 국방과학연구소 주관 “20-’26 한국형 합동 전술데이터링크체계(완성형) 체계개발 결과로 작성된 논문임.

References

- [1] T. Lee, O. Baek, and T. Kim, “The suggestion on the software reliability testing for weapon system,” *Korea Institute of Information Security & Cryptology review*, vol. 28 no. 6, 2018, pp. 76-82.
- [2] Defense Acquisition Program Administration, “The manual of development and management for weapon systems Software,” *Manual*, Feb. 2020.
- [3] S. Jung, Y. Lee, J. Hwang, and J. Park, “A Study on the Expansion and Improvement of the Tactical Data Link Processing Structure for Link-22 SNC Interface,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 6, 2021, pp. 1045-1052.
- [4] J. Hwang, K. Lee, and S. H. Jung, “A Study On the Multi-tactical Data Link Data Management,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 15, no. 3, 2020, pp. 457-464.
- [5] Agency for Defense Development, “CDR Agenda of JTDLS Batch II program,” *Report*, Nov. 2021.
- [6] Defense Acquisition Program Administration, “plans of software reliability testing for weapon system on JTDLS Batch II program,” *Report*, Dec. 2021.

- [7] S. Herbold, J. Grabowski, and S. Waack "Calculation and optimization of thresholds for sets of software metrics," *Empirical Software Engineering*, vol. 6, Issue 6, 2011, pp. 812-841.
- [8] T. L. Alves, C. Ypma, and J. Visser, "Deriving metric thresholds from benchmark data," *2010 IEEE International Conference on Software Maintenance*, Timisoara, Romania, 2010, pp. 1-10.
- [9] Motor Industry Software Reliability Association, "MISRA C++ 2008: Guidelines for the use of the C++ language in critical systems," *Report*, June 2008.
- [10] Motor Industry Software Reliability Association, "MISRA C 2012: Guidelines for the use of the C language in critical systems," *Report*, Mar. 2013.

구영훈(Young-Hoon Goo)



2016년 고려대학교 컴퓨터정보학과 (공학사)

2020년 고려대학교 컴퓨터정보학과 (이학박사)

2020년 7월 ~ 2021년 11월 한국과학기술정보연구원 박사후연구원

2021년 12월~현재 국방과학연구소 선임연구원

※ 관심분야 : 전술데이터링크, Link-K, Link-16, Link-22, 소프트웨어, 네트워크

저자 소개



송영중(Young-Joong Song)

2009년 연세대학교 전기전자공학과 졸업(공학사)

2009년 ~현재 한화시스템 재직

※ 관심분야 : 전술데이터링크, Link-K, Link-16, Link-22, 소프트웨어

이윤정(Youn-Jeong Lee)



2001년 숭실대학교 정보통신공학과 (공학사)

2003년 숭실대학교대학원(공학석사)

2006년 숭실대학교대학원(공학박사)

2015년 7월 ~ 2016년 6월 미국 University Southern California

박사후 연구원

2006년~2022년 국방과학연구소 책임연구원

※ 관심분야 : 전술데이터링크, Link-K, Link-16, Link-22, 소프트웨어

