

무기체계 소프트웨어 기술변경 원인 및 결함 유입단계 분석

김종규*, 윤재형, 류지선
국방기술품질원

Analysis of Engineering Change Cause and Defect Inflow Stage of Weapon System Software

Jong-Kyu Kim*, Jae-Hyeong Yun, Jiseon Yu
Defense Agency for Technology and Quality

요약 무기체계 개발단계에서 무기체계의 복잡성, 소요군과 개발기관의 의사소통 문제 등 다양한 이유로 결함이 체계로 유입된다. 유입된 결함은 양산 및 운영유지 단계에서 문제점으로 발견되어 이를 해결하기 위한 기술변경이 증가하고 있으며, 특히 무기체계의 소프트웨어 비중이 증가함에 따라 무기체계 소프트웨어 기술변경에 대한 별도의 관리가 필요한 상황이다. 국방기술품질원에서는 무기체계 소프트웨어 기술변경에 대한 체계적인 추적성 관리를 위하여 기술변경 원인 및 소프트웨어 기술지원 결과에 대한 데이터베이스를 구축한 상태이다. 본 논문에서는 국방기술품질원에서 구축한 데이터베이스의 자료를 바탕으로 2017, 2018년도에 기술변경이 이루어진 소프트웨어들에 대한 결함 원인을 분석하였다. 결함 원인 유입단계는 무기체계 소프트웨어 개발 프로세스를 기반으로 요구사항분석 단계, 설계 단계, 구현 단계, 시험 단계, 규격화 단계, 운영유지 단계로 구분하였으며, 각 유입 단계에 따른 결함의 분류 방법을 정의하였다. 최종적으로 8대 무기체계 분류에 따라 결함이 빈번하게 유입되는 개발단계를 식별하였다. 본 논문의 분석결과를 바탕으로 향후 개발 단계 소프트웨어 품질관리를 수행할 때 결함이 빈번하게 유입되는 개발단계를 식별하고 집중적으로 품질관리를 수행하여 효율적인 무기체계 소프트웨어 품질 확보에 기여할 수 있음을 보인다.

Abstract Defects flow into weapon system development for various reasons, such as the complexity of the weapon system and the communication problems between the requirement-raising armed forces and developers. These defects also emerge as problems in the operation and maintenance stage. Hence, engineering change to make up for the defects is increasing. In particular, as the proportion of software in the weapon system increases, separate management of software engineering change is needed. Hence, DTaQ has established a database for the systematic traceability management of software engineering changes. This research used the database built by DTaQ and analyzed the cause of defects in weapon system software that was changed in 2017 and 2018. As a result, the cause of the defect was identified to be from requirement analysis, design, implementation, testing, standardization, or the operation and maintenance stage, depending on the software development process of the weapon system. Finally, defect classification was carried out according to the eight weapon system types. Based on the analysis results of this paper, this research is expected to contribute to the efficient securement of the quality of weapon system software. In particular, the quality of weapons system software is achieved by intensively managing the development stage in which defects frequently flow during software quality management.

Keywords : Software, Software Quality Assurance, Software Engineering Change, Weapon System, Software Quality

*Corresponding Author : Jong-Kyu Kim(DTaQ)

email: rnseorka528@dtaq.re.kr

Received December 13, 2021

Accepted March 4, 2022

Revised January 3, 2022

Published March 31, 2022

1. 서론

무기체계를 연구개발 획득함에 있어 제한된 개발기간 및 예산으로 모든 운용환경 및 특수한 예외 상황에 대한 운용성 평가가 이루어지기 어려운 현실이다. 이렇게 획득된 무기체계는 개발 단계에서 유입된 결함 및 운용 간 사용자의 예기치 못한 실수로 인하여 운영유지 단계에서 문제점이 발생되고 있다. 국방기술품질원(이하 기품원)에서는 하자 처리, 소요군 기술지원 등의 업무를 수행함으로써 앞선 문제점을 해결하고 규격화된 무기체계의 형상관리 업무를 통해 소요군의 전력공백을 줄이고 있다. 국방에서의 형상관리는 품목의 기능적 또는 물리적 특성을 식별하고 통제하여 전체 수명주기 동안 경제적인 운영을 목적으로 하며, 형상식별, 형상통제, 형상확인, 형상자로 유지로 구분된다. 그 중 형상통제는 형상과 형상식별서의 변경을 통제하는 활동으로 양산 및 운영유지 단계의 형상통제는 기술변경, 규격완화, 면제로 분류된다. 2013년부터 형상통제의 분류 중 하나인 기술변경으로 인한 소프트웨어 기술문서 및 컴퓨터 파일의 개정은 꾸준히 증가하고 있다[1].

기품원은 방위사업청(이하 방사청) 훈령 제502호 방위사업품질관리규정(2019.4.18. 제정)에 따라 기존 양산 단계 소프트웨어 품질관리 업무에 추가로, 그동안 제한적으로 수행하던 개발단계 소프트웨어 품질관리 업무를 품질관점에서 적극적으로 기술지원을 수행하고 있다. 이에 따라 제한된 기간, 인원에 늘어난 업무를 원활하게 수행하기 위해서는 업무의 효율화가 필요하며, 양산 및 운영유지 단계에서 지속적으로 발생하는 문제점을 식별하고 데이터베이스화함으로써, 실전적 결함 데이터를 환류하여 유사체계 개발 시 결함을 사전에 방지할 필요성이 대두되고 있다[1]. 본 논문에서는 기품원이 구축한 데이터베이스를 활용해 기술변경 발생원인 즉, 결함의 원인을 분류 하였으며, 이를 바탕으로 무기체계 소프트웨어 개발 프로세스 중 어떤 단계에서 어떤 원인으로 결함이 유입되었는지를 분석하였다. 본 논문의 2장에서는 이론적 배경에 대해 3장에서는 분석의 대상 및 분석방법, 4장에서는 분석 결과를 마지막 5장에서는 결론에 대하여 작성하였다.

적 배경에 대해 3장에서는 분석의 대상 및 분석방법, 4장에서는 분석 결과를 마지막 5장에서는 결론에 대하여 작성하였다.

1.1 연구 배경 및 목적

무기체계 첨단화가 이루어지면서 소프트웨어가 탑재되는 무기체계의 비율이 높아지고 있다. 이에 따라 운영유지 단계에서 소프트웨어 결함에 의한 기술변경 수 또한 증가하고 있으며 기술변경은 무기체계를 운용함에 있어 비용과 전력공백을 야기한다.

무기체계 소프트웨어의 개발이 완료된 후 발견되는 소프트웨어 결함은 운영유지 단계에서 기술변경을 통해 조치하지만 근본적인 원인은 개발단계에서 해당 결함을 발견하지 못한 채 운영유지 단계로 넘어온 것에 있다. 그러므로 운영유지 단계의 비용, 전력공백의 최소화를 위해서는 효율적인 개발단계 소프트웨어 품질관리가 필요한 상황이다[1].

그렇기에 본 논문에서는 기품원이 구축한 기술변경 데이터베이스를 분석하여 무기체계 소프트웨어의 개발단계 결함 유입 최소화 및 품질관리를 효율화 방안을 제시하고자 한다.

2. 이론적 배경

2.1 무기체계 소프트웨어의 정의

무기체계는 유도무기·항공기·함정 등 전장에서 전투력을 발휘하기 위한 무기와 이를 운영하는 데 필요한 장비·부품·시설·소프트웨어 등 제반요소를 통합한 것을 말한다[2]. 소프트웨어란 컴퓨터, 통신, 자동화 등 장비와 그 주변장치에 대하여 명령·제어·입력·처리·저장·출력·상호작용이 가능하도록 하게 하는 지시·명령의 집합과 이를 작성하기 위하여 사용된 기술서나 그 밖의 관련 자료를 말한다[3].

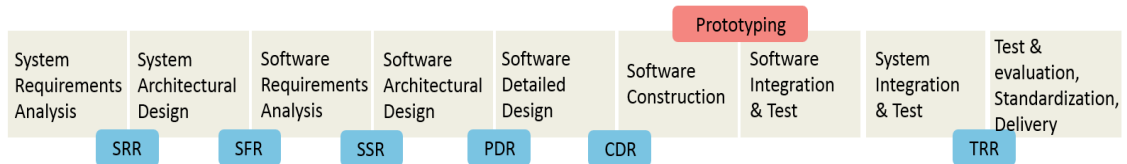


Fig. 1. The process of Weapon System Software Development

2.2 무기체계의 세부분류

현재 무기체계 세부분류는 지휘통제·통신, 감시정찰, 기동, 함정, 항공, 화력, 방호, 사이버, 우주, 그 밖의 무기체계로 분류하고 있다[4].

본 논문은 사이버, 우주 무기체계 분류가 추가되기 이전의 데이터베이스를 기반으로 분석을 실시하였기에 위의 두 분류를 제외한 8대 무기체계에 한정하여 분석·작성하였다.

2.3 무기체계 소프트웨어 개발 프로세스

무기체계 소프트웨어 개발 프로세스는 체계 개발 준비, 체계 요구사항 분석(체계요구사항 검토회의 수행), 체계 구조설계(체계기능 검토회의 수행), 소프트웨어 요구사항 분석(SW 요구사항 검토회의 수행), 소프트웨어 구조설계(기본설계 검토회의 수행), 소프트웨어 상세설계(상세설계 검토회의 수행), 소프트웨어 구현, 소프트웨어 통합 및 시험, 체계통합 및 시험(시험 준비 검토회의 수행), 개발시험평가, 운용시험평가, 소프트웨어 설치, 구축화, 인도과정을 거쳐서 진행되며, Fig. 1과 같다[2].

2.4 기술변경

기술변경은 방사청 훈령 '표준화 업무규정'에서 정의한 형상통제의 분류 중 하나로 국방규격 제정 이후에 발생하는 물품의 형상특성 및 기능 등의 변경을 말하며 기술변경은 해당 기술자료 묶음의 수정을 필요로 한다. 기술변경은 작전운용성능에 영향을 미치는 사항 또는 전력화 일정에 지장이 가거나 비용변동 등 수정계약이 되는 사항에 대해서는 I급 기술변경, I급 외의 사항에 대해서는 II급 기술변경으로 등급을 분류한다. I급 기술변경은 방사청, II급 기술변경은 기품원 및 국방기술진흥연구소의 승인을 통해 규격 개정이 이루어진다. 기술변경의 절차는 형상통제 제안기관(각 군, 업체, 기품원 등)에서 국방규격 개정 제안서를 제출하면 기술변경의 업무 절차가 시작 된다. 제출된 국방규격 개정 제안서를 통해 기술변경이 이루어지는 대상품의 도면번호, 기술변경 관련근거, 국방규격 개정사유 및 내용, 세부항목 내역서, 시험평가 결과 등의 내용을 포함하고 있다. 제안서 접수 이후 제안서 타당성 검토, 관련기관 검토, 형상통제심의회 운영, 형상통제 결과 통보, 후속조치의 절차를 통해 규격의 개정이 이루어진다[5].

2.5 소프트웨어 기술변경 데이터베이스

소프트웨어 결함으로 인한 기술변경의 내용은 심의제안서의 일부로 작성되고 별도의 시스템으로 분석내용이 종합되지는 않기 때문에 소프트웨어의 문제원인 분석, 개선방안 내용의 관리가 체계적으로 이루어지지 못하고 각 기술변경 건마다 단발성으로 관리되었다. 기품원은 이러한 문제점을 해결하여 향후 무기체계 소프트웨어 품질 확보를 위해 소프트웨어 기술변경 데이터베이스를 구축하였다. 데이터베이스는 기술변경 내용 추적성 테이블, 기술변경 자료 검토 결과 이력, 검토 요청 정보/처리 정보 등이 있으며, 이 중 기술변경 내용 추적성 테이블을 바탕으로 산재되어 있던 소프트웨어 형상단위의 추적성 확보를 했다[1].

3. 분석 대상 및 방법

3.1 분석 대상

본 논문은 기품원에서 구축한 무기체계 소프트웨어 기술변경 데이터베이스 중 데이터 입력 초기 장비명, 변경 전/후 등의 기본적인 데이터 입력은 하였으나 추가적인 기술변경 분석 데이터가 입력되지 않은 '17~'18년(2년간) 소프트웨어 기술변경 302건을 대상으로 기술변경 발생 원인을 분석하였다. 기술변경세부적인 분석항목은 Table 1과 같다.

Table 1. Engineering Change Analysis Item

| Analysis Item | |
|---------------------------------------|----------------------------------|
| · Equipment Name | · The Year of Engineering Change |
| · Proposal Number | · Weapon System Classification |
| · Drawing Number (SW Document Number) | · Changes of Software |
| · Cause of Engineering Change | · Improvement Plan |

3.2 분석 방법

본 논문은 기품원에서 구축한 데이터베이스의 자료를 바탕으로 분석하였다. 세부적인 기술변경 내용이 필요한 경우 기술변경 검토 요청 시 첨부된 국방규격 개정 제안서 및 소프트웨어 기술자료를 바탕으로 분석을 수행하였다. 단, 첨부 자료가 누락된 경우 국방표준종합정보시스템(KDSIS: Korea Defense Standard Information System, 이하 KDSIS)에 등재된 자료를 바탕으로 분석

하였다.

분석 내용을 기품원의 기존 데이터베이스와 연계하기 위하여 기품원에서 데이터베이스 구축에 사용한 데이터 관리·분석도구인 지멘스 Polarion ALM을 활용해 분석하였다[1]. 분석의 기준은 기술자료에 필요 내용의 작성 유무이며, 작성이 되어 있다면 충분히 작성 되어있는지가 주된 기준이다. 내용 및 설계의 충분성, 적절성은 체계의 특성, 개발 프로그램의 특성을 고려하여 판단을 하였으며 판단자료는 동료검토를 통하여 검증하였다.

3.2.1 기술변경 발생원인 분석 방법

데이터베이스 및 국방규격 개정 제안서 분석을 통해 식별한 기술변경 발생 원인이 개발 단계의 어떤 단계에서 결함으로 유입되었는지를 분석하고자 한다. 결함 유입시점은 무기체계 소프트웨어 개발 프로세스 기반에 운영유지 단계를 추가하여 요구사항 분석 단계, 설계 단계, 구현 단계, 시험평가 단계, 규격화 단계, 운영유지 단계 총 6가지 단계로 분류하였다.

분류 중 요구사항 분석 단계는 소요군의 요구사항이 소프트웨어요구사항명세서(SRS: Software Requirement Specification, 이하 SRS)에 누락된 경우, 요구사항이 반영 되어있어도 내용이 불충분한 경우와 같이 요구사항 분석이 미흡한 결함에 대해 분류하였다. 단, 통상적인 기능은 SRS에 기재되어 있지 않아도 요구사항 누락으로 간주하지 않으며, 여기서 통상적인 기능은 '목록 번호는 1, 2, 3... 순서로 순차적으로 부여한다.', '입력에 필요한 팝업창이 앞에 전시될 수 있도록 한다.'와 같이 요구사항으로 정의되어 있지 않더라도 경험기반으로 설계·구현이 가능한 기능들을 의미한다. 설계 단계는 SRS의 요구사항이 소프트웨어설계기술서(SDD: Software Design Description, 이하 SDD)에 누락된 경우, 설계에 오류가 있는 경우, 오류가 없더라도 내용이 불충분한 경우와 같이 설계가 미흡한 결함에 대해 분류하였다. 구현 단계는 SRS, SDD에서 정의한 요구사항, 설계 내용이 프로그램(코드)에 누락된 경우, 구현에 있어 오기, 통상적인 기능, 분기 처리를 누락한 경우와 같이 구현이 미흡한 결함에 대해 분류하였다. 시험평가 단계는 소프트웨어 시험 수행에 있어 기준 및 계획이 불충분한 경우와 같은 시험계획이 미흡한 결함에 대해 분류하였다.

규격화 단계는 요구사항, 설계, 시험계획을 제외한 소프트웨어 기술자료의 내용이 누락·미흡한 경우, KDSIS에 소프트웨어 기술자료 등재가 누락·미흡한 경우와 같은 규격 자료가 미흡한 결함에 대해 분류하였다. 마지막

으로 운영유지 단계는 체계 운용 중 소프트웨어가 참조하는 자료의 최신화 또는 상용 소프트웨어의 업데이트와 같은 소프트웨어의 최신성이 미흡한 결함에 대한 분류이다(Table 2).

Table 2. Defect Classification of Development Stage

| Development Stage | Classification | Detailed Classification |
|-----------------------|----------------------------------|---|
| Requirement Analysis | Requirement Defect | Requirement Missing |
| | | Requirement Insufficiency |
| Design | Design Defect | Design Missing |
| | | Design Insufficiency |
| | | Design Error |
| Implementation | Implementation Defect | Implementation Missing |
| Testing | Test Defect | Software Test Plan Insufficiency |
| | | Software Technical Document Insufficiency |
| Standardization | Standardization Data Defect | KDSIS Upload Insufficiency |
| | | Software Update |
| Operation Maintenance | Software Performance Improvement | Compatibility Insufficiency |
| | | |

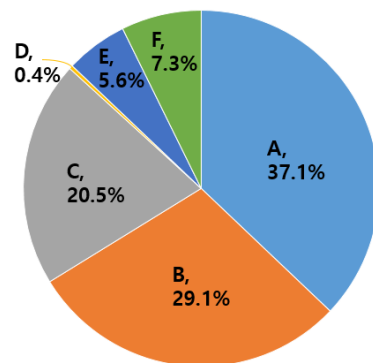


Fig. 2. SW Engineering Change Cause Analysis Result

Table 3. SW Engineering Change Cause Analysis Result

| Development Stage | Amount | Ratio |
|--------------------------|--------|-------|
| A. Requirement Analysis | 112 | 37.1% |
| B. Design | 88 | 29.1% |
| C. Implementation | 62 | 20.5% |
| D. Testing | 1 | 0.4% |
| E. Standardization | 17 | 5.6% |
| F. Operation Maintenance | 22 | 7.3% |

4. 분석 결과

‘17년~’18년(2년간) 총 64개의 기술변경 심의 제안서를 바탕으로 총 302건의 소프트웨어 기술변경 을 검토한 결과는 표 Fig. 2, Table 3과 같다.

분석결과 요구사항 분석 단계에서 112건의 결함이 식별되었다. 상세분류로는 요구사항 누락이 91건으로 구체적인 요구사항이 SRS에 반영되지 못하였거나, 운용환경을 고려 못해 소요군도 요구사항으로 제시하지 못한 내용들이 주를 이루었다. 설계 단계에서는 88건의 결함이 식별되었으며, 상세분류로는 설계 오류가 46건으로 설계 내용을 SDD에 기재 하였으나 알고리즘 결함 및 오류발생 경우가 설계 단계 결함 중 과반을 차지하였다. 구현 단계의 경우 62건의 결함이 식별되었으며, 해당 내용은 전부 상세분류 중 구현 미흡으로 식별하였다. 이는 SDD에 설계된 내용들은 구현에 있어 누락된 것이 없음을 의미한다. 시험 단계에서의 결함은 총 1건으로 식별되었다. 시험 단계 결함 검출 건수가 현저하게 낮은 원인은 무기체계 개발프로세스를 통해 개발 소프트웨어에 대해 다양한 시험·검증을 수행하는 절차가 있다는 점과 분석을 수행함에 있어 실제 개발단계에 참여하여 분석한 것이 아닌 소프트웨어 기술문서를 바탕으로 분석을 수행하여 실

제와 차이가 있을 수 있다는 점이 있다. 규격화 단계의 결함은 17건, 운용유지 단계의 결함은 22건 식별되었으며, 규격화 단계의 경우 소프트웨어 기술문서 연계성 미흡, 운용유지 단계의 경우 상용 소프트웨어 최신화와 그에 따른 호환성 보장이 주를 이루었다. Table 4는 8대 무기체계 분류(화력, 기동, 방호, 항공, 감시정찰, 함정, 지휘통제, 그 외)에 따른 결함 유입단계를 분석한 결과이다. 상세결과로 각 체계분류별 가장 많은 결함이 유입된 원인을 살펴보면 화력, 방호, 감시정찰, 지휘통제 무기체계는 상세분류 중 요구사항 누락이 각각 69건, 10건, 5건, 2건 으로 가장 많은 비중을 차지하였다. 항공과 기동 무기체계는 상세 분류 중 설계오류가 각각 20건, 9건으로 가장 많은 비중을 차지하였으며, 특히 항공 무기체계의 결함 44건 중 대부분의 결함이 설계오류로 나타났다. 항공, 기동 무기체계의 경우 개발단계 품질관리를 수행하는 경우 SDD에 알고리즘이 체계 운용환경에 적합하게 설계되었는지 검토가 필요하다. 기술변경 검토 요청이 없는 함정과 1건의 기술변경이 검토된 그 외 무기체계를 제외한 8대 무기체계는 분류 무관하게 요구사항 분석 단계와 설계 단계에서 과반 이상의 결함이 식별되었다. 이는 소요군이 요구사항을 제시하는 단계와 개발기관에서 요구사항을 설계로 구체화하는 단계에서 체계 운용환경

Table 4. SW Engineering Change Cause Analysis Result by Weapon Classification

| Stage \ Classification | Firing System | Aeronautical System | Land System | Protection System |
|--------------------------|---------------|---------------------|-------------|-------------------|
| A. Requirement analysis | 82 (50%) | 3 (7%) | 6 (14%) | 12 (37%) |
| B. Design | 28 (17%) | 27 (61%) | 18 (44%) | 12 (37%) |
| C. Implementation | 36 (22%) | 10 (23%) | 6 (15%) | 5 (15%) |
| D. Testing | 0 (0%) | 1 (2%) | 0 (0%) | 0 (0%) |
| E. Standardization | 5 (3%) | 2 (5%) | 5 (12%) | 2 (6%) |
| F. Operation maintenance | 12 (8%) | 1 (2%) | 6 (15%) | 2 (6%) |
| Total | 163 | 44 | 41 | 33 |

| Stage \ Classification | Surveillance Patrol System | Command and Control System | Other Systems | Vessels |
|--------------------------|----------------------------|----------------------------|---------------|---------|
| A. Requirement analysis | 6 (37%) | 3 (75%) | 0 (0%) | 0 (0%) |
| B. Design | 3 (19%) | 0 (0%) | 0 (0%) | 0 (0%) |
| C. Implementation | 5 (31%) | 0 (0%) | 0 (0%) | 0 (0%) |
| D. Testing | 0 (0%) | 0 (0%) | 0 (0%) | 0 (0%) |
| E. Standardization | 2 (13%) | 0 (0%) | 1 (100%) | 0 (0%) |
| F. Operation maintenance | 0 (0%) | 1 (25%) | 0 (0%) | 0 (0%) |
| Total | 16 | 4 | 1 | 0 |

및 예외사항을 전부 식별하지 못한 것에 있다.

분석결과를 종합하면 요구사항 분석 단계와 설계 단계에서 결함이 총 200건으로 전체의 66.2%의 결함이 개발 단계 초기에 발생하고 있음을 알 수 있다. 그렇기에 개발 단계 소프트웨어 품질관리를 수행함에 있어 요구사항 분석 단계, 설계 단계와 같은 개발의 초기 단계에 소프트웨어 개발계획이 체계구조와 상충되지 않게 적절하게 반영되었는지, 소프트웨어의 요구사항과 설계 내용이 체계 특성을 고려하여 작성되었는지에 대한 검토를 집중적으로 수행한다면 한정된 자원에서 효율적인 업무 수행을 통해 결함 방지가 가능하다. 또한 이 결과를 환류해서 다양한 기관에서 개발 초기단계에 소프트웨어 품질관리를 중점적으로 수행해야 하는 근거로 활용될 수 있다고 판단된다.

5. 결론

소프트웨어는 개발 완료 이후 양산 및 운영유지로 인한 오류, 시간에 따른 변형이 발생하지 않기 때문에 타 소프트웨어와의 호환성, 버전 최신성 등의 문제를 제외한 대부분의 소프트웨어 결함은 개발단계에서 발생한다. 즉 운영유지 단계에서 소프트웨어 결함을 방지하기 위해서는 개발단계 소프트웨어 품질관리가 필요하다. 본 논문에서는 개발단계 소프트웨어 품질관리를 수행하며 결함을 효율적으로 식별하기 위하여 기품원에서 '17-'18년에 수행한 소프트웨어 기술변경 기술지원 결과에 대해 분석하였다. 분석한 결과 대부분의 소프트웨어 기술변경 결함들이 개발단계의 초기에서 유입되는 것을 확인할 수 있었다. 결함이 다수 발생하는 체계개발 초기에 집중적으로 품질관리를 수행하면 기간, 인력 등 한정된 자원에서도 개발단계 소프트웨어 품질관리를 보다 효율적으로 수행할 수 있으며, 추후 실제 개발단계 소프트웨어 품질관리를 수행하면서 각 결함의 분류 구체화 및 결함을 완화할 해결방법에 대한 데이터를 축적하면 소프트웨어 품질관리 업무의 효율 및 무기체계 소프트웨어 품질이 더욱 향상 될 것이다.

References

[1] J. W. Oh, J. K. Kim, J. S. Yu, J. H. Yun, C. H. Song, "Research on DB Construction and Utilization Measure to Analyze the Cause of Weapon System Software Engineering Change and Derive Improvement Plan"

The Korea Academia-Industrial cooperation Society, Vol.22, No.4, pp.331-337, April. 2021.

DOI : <https://doi.org/10.5762/KAIS.2021.22.4.331>

- [2] Defense Acquisition Program Administration(DAPA) Manual "Weapon System Software Development and Management Manual", DAPA, Korea, pp.6~8, 2020.
- [3] Ministry of Science and ICT(MSIT), "Software Industry Promotion Act", MSIT, Korea, Article 2(Definition), 2020.
- [4] Department of Defense(DoD) Instruction 2568, "Defense Forces Development Work Instruction", DOD, Korea, pp.197-205,2021
- [5] Defense Acquisition Program Administration(DAPA) Instruction 683, "Supporting the Development of Weapon System Software Regulation", DAPA, Korea, pp.1, 2021.
- [6] Defense Acquisition Program Administration(DAPA) Instruction 687, "Standardization Business Regulation", DAPA, Korea, pp.15-21, 2021.

김 종 규(Jong-Kyu Kim)

[정회원]



- 2019년 8월 : 동국대학교 정보통신공학과(정보통신공학학사)
- 2019년 12월 ~ 현재 : 국방기술품질원 연구원

<관심분야>

국방, 무기체계 소프트웨어, 소프트웨어공학

윤 재 형(Jae-Hyeong Yun)

[정회원]



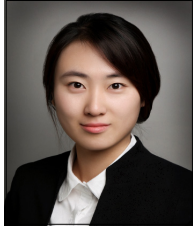
- 2017년 2월 : 건국대학교 전자공학부 (전자공학학사)
- 2017년 2월 ~ 현재 : 국방기술품질원 연구원

<관심분야>

국방, 무기체계 소프트웨어, 소프트웨어 품질

류 지 선(Jiseon Yu)

[정회원]



- 2018년 8월 : 고려대학교 정보보호대학원 정보보호학과 (정보보호학석사)
- 2018년 12월 ~ 현재 : 국방기술품질원 연구원

〈관심분야〉

무기체계 소프트웨어, 소프트웨어품질