

# 무기체계 소프트웨어 품질 개선을 위한 ISO/IEC 25023의 소프트웨어 개발 프로세스 적용 방안

윤경환\*, 류지선  
국방기술품질원

## Applying ISO/IEC 25023 to Software Engineering Process in Weapon System for Quality Improvement

Gyeonghwan Yoon\*, Jiseon Yu  
Defense Agency for Technology and Quality

**요약** 본 연구에서는 기존의 무기체계 소프트웨어 개발 프로세스에 ISO/IEC 25023의 소프트웨어 품질 평가 모델을 적용하여 기존보다 무기체계 소프트웨어의 품질을 향상시킬 수 있는지 확인하였다. 현재 무기체계 소프트웨어 개발 프로세스는 소프트웨어 품질 요구사항을 관리하고 있지만 소프트웨어 품질은 제한적으로만 향상시키고 있었다. 소프트웨어 품질 요구사항에서 품질 특성을 정의하고 관리하였지만 개별적인 사업에 따라 품질 특성이 다르게 정의되었기 때문이다. 따라서 소프트웨어 품질 요구사항의 품질 특성이 동일하여도 산출물이 다르고 사업에 따라 품질 관리의 차이가 컸다. 무기체계 소프트웨어 개발 프로세스 내에서 소프트웨어 품질 관리 제한점의 원인은 소프트웨어 품질 특성에 대한 일관된 정의의 부족과 정량적으로 측정되지 않는 산출물로 파악하였다. 국제표준 소프트웨어 품질평가 메트릭인 ISO/IEC 25023은 소프트웨어를 8가지 주특성 및 31가지 부특성으로 구분하여 품질 특성을 정의하고 정량적으로 소프트웨어 제품 품질을 평가할 수 있는 기준을 제시한다. ISO/IEC 25023을 무기체계 소프트웨어 개발 프로세스에 적용하여 소프트웨어 품질을 향상시키는 방안을 검토하고 소프트웨어 품질 향상 효과를 분석하였다.

**Abstract** This study aimed to improve the software quality in weapon systems by applying ISO/IEC 25023 to the software engineering process. The software engineering process in weapon systems manages the Software Quality Requirement but has restrictively improved the software quality. Software quality items are already defined and used for software engineering processes in weapon systems, but they are defined inconsistently according to individual software development cases. Thus, the methods of quantifying software quality are different, even though the same software quality is defined. In addition, the software quality has been managed differently. The major problems of software engineering processes in weapon systems are the unclear definition of the software quality requirements and the absence of quantifying software quality standards. ISO/IEC 25023 is composed of eight quality characteristics and thirty-one sub-characteristics. ISO/IEC 25023 provides measurement functions that can quantify the software product quality based on its characteristics. To improve the software quality further, the measurement functions in ISO/IEC 25023 are suggested to quantify and manage software quality for software engineering processes in weapon systems. The expected effects of this study were analyzed.

**Keywords** : Software Quality, ISO/IEC 25023, Weapon System, Quality Assurance, Software Engineering Process

---

\*Corresponding Author : Gyeonghwan Yoon(DTaQ)

email: sunshine@dtaq.re.kr

Received April 5, 2021

Accepted May 7, 2021

Revised May 3, 2021

Published May 31, 2021

## 1. 서론

방위력 개선사업으로 무기체계와 함께 획득되는 소프트웨어는 방위사업청에서 제정한 「무기체계 소프트웨어 개발 및 관리 매뉴얼」의 표준 프로세스에 따라 소프트웨어가 개발된다[1]. 그렇지만 개발 프로세스 내에서 소프트웨어 품질은 제한적으로 검증되어 높아진 품질 요구사항을 만족하지 못하고 있다.

무기체계 소프트웨어의 품질을 향상시키기 위해 소프트웨어의 개발단계 품질보증 방안이 연구되기도 했다[2]. 개발 프로세스의 품질개선 방안, 소프트웨어 제품의 품질평가 방안, 개발단계별 소프트웨어의 신뢰도평가 방안 등이다. 그렇지만 개선방안과 평가모델을 기존의 무기체계 개발 프로세스에 적용하는 구체적인 방법은 미비했다.

또한 무기체계의 운용유지단계의 소프트웨어 품질평가를 위해 ISO/IEC 25010의 품질 주특성, ISO/IEC 25023의 품질 부특성의 중요도 및 품질특성 우선순위 연구도 있었다[3]. 그렇지만 운용유지 단계에서 필요한 품질특성으로 제한되어 개발단계의 품질 향상 연구로 확장될 필요가 있었다. 기존의 무기체계 소프트웨어 품질 개선을 위한 제도적, 기술적 연구사례는 Table 1과 같다.

본 연구에서는 무기체계 소프트웨어 품질이 제한적으로 검증되는 이유를 소프트웨어 품질 측정의 기준 부재와 개발 프로세스 문제로 파악하였다. 이를 해결하기 위

해 먼저 소프트웨어 품질측정 기준으로 국제 소프트웨어 품질 측정 기준인 ISO/IEC 25023이 활용될 수 있는지 확인하였다.

또한 무기체계 소프트웨어 개발 프로세스를 개선하여 소프트웨어 품질을 향상시킬 수 있는지 검토하였다. ISO/IEC 25023의 소프트웨어 품질측정 기준을 무기체계 개발 프로세스 초기 단계에 적용하여 소프트웨어 품질을 효율적이고 근본적으로 향상시키는 방안을 제시하였다.

현재 무기체계 소프트웨어 개발 프로세스에서 소프트웨어 품질 관리가 제한적으로 이뤄지고 있음을 먼저 사례로 파악할 것이다. 그리고 ISO/IEC 25023의 제품 품질측정 기준이 무기체계 소프트웨어의 품질 측정 기준으로 활용되는 방안을 제시한다. 소프트웨어 개발 프로세스 전반적으로 품질 측정기준이 활용됨으로써 무기체계 소프트웨어 품질이 향상되는 것을 보일 것이다. 그리고 결론적으로 예상되는 품질 개선 효과를 확인한다.

## 2. 관련 연구

### 2.1 국제 소프트웨어 제품 품질 측정 기준ISO/IEC 25023

Table 1. Prior research on quality improvement method in weapon system software

View	Institutional Method	Technical Method
Process	○ Quantification of Software Development Process to evaluate involved groups.	○ Select Metrics, Standards for Software Development Phase
	○ Integrating Software Reliability Testing Procedure to Software Development Process in weapon system.	○ Introduce Software Reliability Testing Procedure which modified Michael R. Lyu's Software Reliability Analysis Process for quality improvement of weapon system software. ○ Using RADC (Rome Air Development Center) model, Industry/Application type model, Shortcut model to get Software Reliability prediction values.
Product	○ Developing software quality assessment model for Operation/Support Phase.	○ Evaluate importance of quality characteristics in ISO/IEC 25010 and subcharacteristics in ISO/IEC 25023 using AHP analysis method for software quality in weapon system.
	○ Specify three quality characteristics (Maintainability, Clarity, Testability) to SDP (Software Development Plan), SQAP (Software Quality Assurance Plan) to evaluate software quality in Development Test	○ Recommended Quality Metrics in weapon system software. - $Maintainability = \frac{100 \times \sum_1^{11} All\ maintainability\ metrics\ that\ pass}{11}$ $- Clarity = \frac{100 \times \sum_1^{14} All\ Clarity\ metrics\ that\ pass}{14}$ $- Testability = \frac{100 \times \sum_1^{13} All\ Testability\ metrics\ that\ pass}{13}$

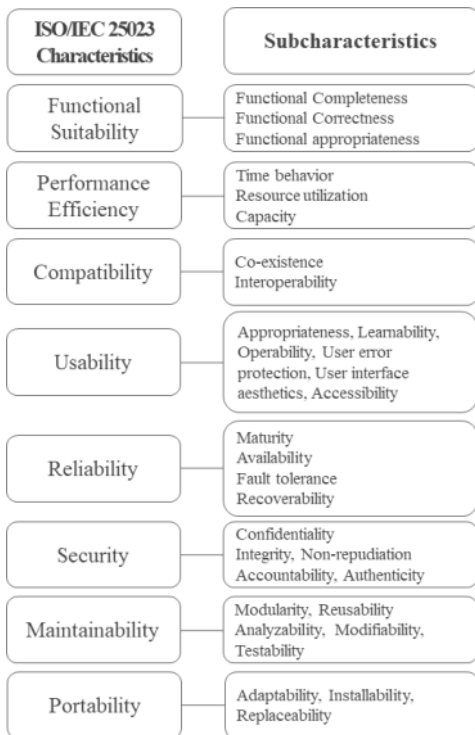


Fig. 1. ISO/IEC 25023 eight quality characteristics and thirty-one subcharacteristics

기존의 소프트웨어 품질 측정의 국제표준이었던 “ISO/IEC TR 9126-2 : 2003”과 “ISO/IEC TR 9126-3 : 2003”을 완전히 대체하는 국제표준으로 ISO/IEC 25023은 Fig. 1과 같이 8개의 주특성과 31개의 부특성으로 구성된다[4]. 이렇게 주특성과 부특성으로 구분하는 품질측정 기준은 그대로 인용되어 한국산업표준 KS X ISO/IEC 25023으로도 제정되어 있다[5].

소프트웨어 제품 주특성은 기능 적합성, 성능 효율성, 호환성, 사용성, 신뢰성, 보안성, 유지보수성, 이식성 등 8가지로 나누고 각각의 주특성은 Fig. 1과 같은 부특성의 조합으로 구성된다. 각각의 부특성의 정량적인 품질 특성 구성이 상위 주특성의 품질 척도가 된다.

ISO/IEC 25023은 소프트웨어가 개발되는 전체 산업 분야에서 활용 가능하다. 사용자, 개발자, 시험 담당자, 품질보증 담당자 등 모두에게 소프트웨어 제품 특징에 대한 공통된 기준을 제공한다.

## 2.2 개발 프로세스와 사용자 품질

Fig. 2는 소프트웨어 개발 프로세스의 품질이 내부 품질과 외부 품질, 사용자의 품질까지 영향을 주는 것을 의

미한다[4]. 프로세스의 품질 개선은 제품의 내부 품질, 외부 품질 향상을 가져오고 결국 사용자가 느끼는 품질 향상까지 가져온다. 따라서 개발 프로세스를 개선하는 것이 제품 품질 및 사용 품질의 개선 방안이 된다.

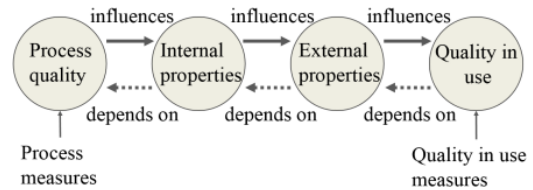


Fig. 2. Relationship between types of quality measures

## 2.3 소프트웨어 테스트 3 원칙: Early testing

소프트웨어 품질을 높이기 위한 테스트는 7가지 원칙이 있다[6]. 그 중 3원칙은 소프트웨어 개발 주기 중 초기에 테스트 하는 것을 말한다. 버그의 조기 제거로 소프트웨어 품질을 높이는 원칙이며[7] 이는 결함이 초기 단계에서 발견될수록 해결비용이 줄어드는 원리 때문이다. IBM의 The Systems Sciences Institute는 소프트웨어의 설계 과정에서 발생한 결함 해결비용을 1로 가정할 경우, 운용유지 단계에서 발생한 결함 해결비용을 100으로 환산하고 있다.

제한된 시간과 자원으로 개발되는 무기체계 소프트웨어 역시 위의 원칙이 적용된다. 무기체계 소프트웨어의 품질 향상을 위하여 개발 프로세스의 초기부터 소프트웨어 결함을 찾기 위한 테스트가 고려되어야 할 것이다. 또한 ISO/IEC 25023의 소프트웨어 제품 품질 기준과 품질 측정 함수를 무기체계 소프트웨어 개발 초기부터 고려한다면 더 효율적으로 소프트웨어 품질 향상을 이끌어 낼 수 있을 것이다.

## 2.4 주요 연구절차

먼저 관련 연구를 통해 국제표준 ISO/IEC 25023의 품질특성이 전체 산업분야에서 활용되며 무기체계 소프트웨어 품질 측정 기준이 될 수 있음을 확인했다. 그리고 개발 프로세스의 개선이 제품 및 사용 품질을 향상시키는 것도 알 수 있었다. 이때 개발 프로세스 초기 단계의 검증이 품질 향상에 더 효과적이었다.

위의 관련 연구를 참고하여 본 연구에서는 ISO/IEC 25023의 품질 측정 기준을 활용하고 무기체계 개발 프로세스의 초기단계를 개선하는 품질관리 개선 방안을 제

시할 것이다. 본 논문의 주요 연구절차는 Fig. 3과 같다.

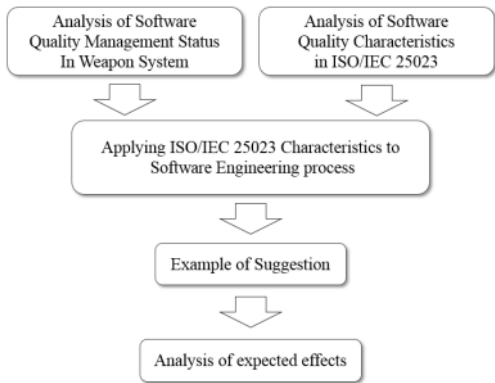


Fig. 3. Schematic diagram of research process

### 3. 현 상황 및 한계점

#### 3.1 프로세스 내 소프트웨어 품질관리

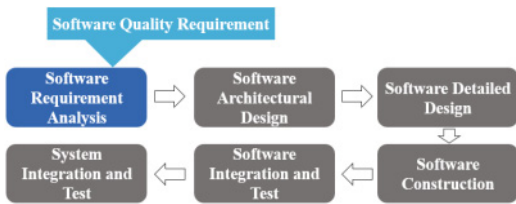


Fig. 4. Software engineering process in weapon system and Software Quality Requirement

무기체계와 함께 획득되는 소프트웨어는 Fig. 4의 프로세스에 따라 단계적으로 개발된다. 각각의 개발 단계는 요구사항 분석, 구조 설계, 상세 설계, 구현, 통합테스트, 시스템 통합테스트 단계를 거친다. 그 중 요구사항 분석 단계에서 소프트웨어 품질 요구사항을 정의하고 프로세스 내에서 소프트웨어 품질을 추적하여 관리하게 된다. 결국 소프트웨어 품질은 최초의 소프트웨어 품질 요구사항 정의에 따라 결정된다.

#### 3.2 소프트웨어 품질 요구사항 사례 및 한계

2020년도 규격화된 무기체계 소프트웨어 282건을 분석한 결과 Fig. 5와 같이 245건의 소프트웨어 품질 요구사항이 식별자로 구분되어 사용 된 것을 확인하였다. 하나의 품질 요구사항 식별자에는 여러 가지의 품질 항목

들이 중복적으로 포함되어 관리되기도 했다. 품질 항목 중 신뢰성 항목이 216건으로 총 88%를 차지했으며 가장 많이 활용된 항목이었다. 다음으로 분류체계 항목이 44건(18%)이었고 기능성 24건(10%), 유지보수성 15건(6%), 사용성 11건(4%), 기타 45건(18%) 등이었다.

신뢰성 항목의 경우 정적 시험이나 동적 시험의 수행을 요구사항으로 정의하고 있었고 특정 사업의 경우 「무기체계 소프트웨어 개발 및 관리 매뉴얼」에 의해 시험 수행이 규정되어 있었다. 신뢰성 항목은 테스트 결과물이 산출물로 관리되어 소프트웨어 품질을 제한적으로 향상시키고 있었다.

그렇지만 다음으로 활용된 분류 체계, 기능성, 유지보수성, 사용성, 기타 항목들은 신뢰성 항목과 다르게 테스트 결과물이나 정량적으로 측정된 산출물이 관리되지 않았다. 따라서 소프트웨어 품질을 향상시키는지 검증할 수 없었다.

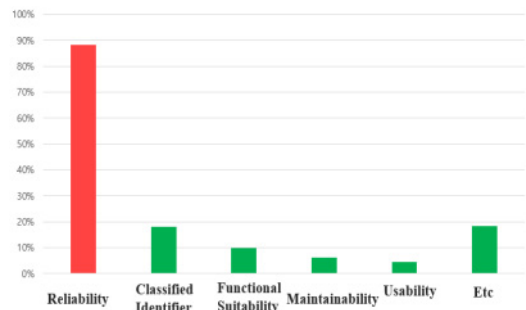


Fig. 5. Frequency of Software Quality Requirement Items in standardized weapon system in 2020

소프트웨어 품질 관리를 위한 프로세스가 있음에도 불구하고 현재 소프트웨어 품질 관리의 가장 큰 제한점은 품질 요구사항이 정량적으로 측정될 수 있는 정의 및 판단 기준이 없다는 것이다. 따라서 품질 요구사항을 정의해도 정량적으로 측정할 수 없고 품질 향상활동이나 산출물을 프로세스 내에서 관리할 수 없었다.

### 4. ISO/IEC 25023 품질 기준 적용 방안

#### 4.1 품질 요구사항 표준화

무기체계 소프트웨어 품질 향상을 위해 소프트웨어 품질 요구사항의 항목들을 국제표준과 일치하도록 Table 2와 같이 ISO/IEC 25023의 주특성 및 부특성으로 재정

Table 2. Standardization of Quality Characteristics

View	Current Quality Characteristics	Suggested Quality Characteristics
Quality Characteristics	Functional suitability Performance efficiency Compatibility Usability Reliability Security Maintainability Portability Classified Identifier Availability Adaptability Testability Log analysis Reusability etc.	Functional suitability Performance efficiency Compatibility Usability Reliability Security Maintainability Portability
Characteristics Configuration	None	31 Quality Subcharacteristics
Quantification Method	None	86 Measurement Functions

의한다. 그러면 기존과 다르게 ISO/IEC 25023 품질 부특성의 측정 함수를 활용하여 소프트웨어 품질을 정량적으로 측정할 수 있다. 기존의 신뢰성 항목, 분류체계 등 ISO/IEC 25023에서 다루지 않는 항목들은 품질 요구사항에서 제외하여 품질 요구사항 항목들과 명확하게 구분한다.

기존의 품질관리 프로세스 내에서 소프트웨어 품질 요구사항이 ISO/IEC 25023의 주특성과 부특성으로 품질 측정 기준이 정의되면 프로세스의 모든 단계에서 소프트웨어 품질 향상 활동이 관리될 수 있다.

#### 4.2 정량화 가능한 품질 요구사항의 확장

소프트웨어 품질 요구사항은 프로세스 내에서 식별자로 구분되고 있다. 기존에는 신뢰성 항목만이 정량적인 방식으로 측정되었다면 개선 방안에서는 8개의 주특성 및 31개 부특성이 정량적으로 측정 가능한 소프트웨어

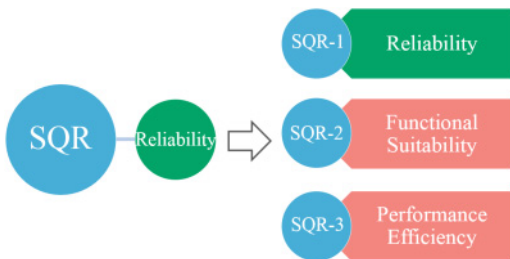


Fig. 6. Expansion of Software Quality Requirement by applying ISO/IEC 25023 characteristics

요구사항으로 관리된다. 또한 86개의 측정함수가 활용되기 때문에 Fig. 6과 같이 주특성의 수준이나 부특성의 수준으로 소프트웨어 요구사항 식별자가 세분화되어 사용될 것이다.

#### 4.3 품질 요구사항 관리방안

소프트웨어 요구사항 분석 이후 소프트웨어 기술 문서인 「소프트웨어 요구사항 명세서」 3.9절에 소프트웨어 품질 요구사항이 명시되고 관리된다. 현재는 품질요소로 신뢰성(정적 시험, 동적 시험), 보안성, 기능적합성, 유지보수성, 사용성, 효율성, 이식성, 호환성 등의 사용을 권고하고 있다.

그렇지만 개선 방안을 적용하면 ISO/IEC 25023의 소프트웨어 제품 품질 특성을 품질 요구사항 기준으로 표준화하고 정량화하는 측정함수 기준으로 관리될 것이다. 식별자로 구분된 소프트웨어 품질 요구사항은 이후의 개발 프로세스인 소프트웨어 구조설계, 상세설계, 구현, 소프트웨어 통합 테스트, 시스템 통합 테스트 등 각각의 단계에 반영된다. 개발 전 과정의 프로세스에서 정량적으로 측정될 수 있는 산출물이 고려되면 소프트웨어 품질을 구조적으로 향상시킬 수 있다.

#### 4.4 적용 방안 예시

기존의 OOO 무기체계 소프트웨어 요구사항 명세서의 경우 아래의 Table 3과 같이 품질 요구사항이 정의되어 있다.

Table 3. One example of Software Quality Requirement Identifier

Identifier	R-OOO-SQR-001
Requirements	○ Classified Identifier: According to guideline of DAPA, SW classified identifier should be applied when software is designed
	○ Functional suitability: All the functional requirement described in Software Requirement Specification should be constructed.
	○ Reliability: Static test and Dynamic test should be performed according to Weapon System Development and Management Manual.

Table 3의 경우 하나의 식별자에 세 가지 품질 요인이 정의되어 있다. 첫 번째 분류체계 항목의 경우 소프트웨어 품질의 정량적인 측정이 어렵고 ISO/IEC 25023의 제품 품질 속성과는 관계가 없다.

Table 3의 두 번째 항목인 기능적합성의 경우 기능 요구사항의 충족만으로 만족되는 품질 항목으로 볼 수 있다. 산출물과 정량적인 품질 기준이 없어 독립적인 품질 활동이 이뤄지지 않았다. Table 3의 세 번째 항목인 신뢰성의 경우 규정에서 요구하는 테스트를 정의하고 있지만 소프트웨어 품질을 정량적으로 측정하지 못하고 있다.

본 연구에서 제안하는 방안을 적용하면 Table 4와 같이 세분화될 수 있다. 분류체계 항목은 ISO/IEC 25023의 속성이 아니므로 제외해야 한다. 또한 기능적합성과 신뢰성은 ISO/IEC 25023의 정의에 따라 기준을 재정립하고 정량화 함수 역시 소프트웨어 품질 요구사항에 정의한다. 뿐만 아니라 각각의 품질 특성에 따라 품질요소 식별자도 분리한다.

Table 4. Modified example of Software Quality Requirement Identifier

Identifier	Quality characteristics	Quality subcharacteristics	Measurement function name
R-OOO-SQR-001	Functional suitability	Functional completeness	Functional coverage
R-OOO-SQR-002	Reliability	Fault tolerance	Failure avoidance

ISO/IEC 25023의 제품 품질 기준에 따라 소프트웨어 품질 요구사항이 명확히 정의되고 구분되었다. 각각의 품질 요구사항은 후속 프로세스에서 각각 추적 관리될 것이다.

#### 4.5 품질 정량화 예 - 고장회피율

ISO/IEC 25023에서 신뢰성의 부특성인 결함 허용성은 고장회피율을 Table 5의 측정함수로 정량화하고 있다. 강건한 소프트웨어 품질을 요구할 때 고장회피율이 사용되며 추천 레벨도 HR(Highly Recommended)로 무기체계 소프트웨어 품질 검증에 활용성이 높을 것으로 예상된다.

Table 5. One example of Measurement function for Fault Tolerance in ISO/IEC 25023

Sub characteristics	Quality measure name	Measurement function
Fault tolerance	Failure avoidance	$X = A/B$ A = Number of avoided critical and serious failure occurrences (based on test cases) B = Number of executed test cases of fault pattern (almost causing failure) during testing

고장회피율은 Table 5의 측정함수와 같이 치명적인 고장을 유발하는 테스트 케이스를 만들어 시험한다. 무기체계 특성을 고려하여 테스트 케이스를 만들고 소프트웨어와 시스템을 함께 테스트한다. 총 테스트 개수와 고장이 미검출된 테스트 개수를 변수 연산으로 정량화하여 고장회피율을 측정한다.

측정함수의 결과 값은 0과 1사이를 정규화 하며 고장회피율처럼 대부분의 X는 1에 가까워지는 것이 품질 향상을 나타낸다. 고장회피율이 소프트웨어 품질 요소로 선택된다면 측정함수의 결과 값인 X가 1에 가까워지도록 소프트웨어의 구조 설계 단계부터 코드가 강건하게 설계되거나 방어코드가 추가로 고려될 것이다. 결과적으로 측정함수가 소프트웨어 구조 설계 단계부터 고려되어 소프트웨어 품질이 향상되는 것이다.

#### 4.6 품질 개선 효과

개선방안의 도입으로 인한 제도적, 기술적 효과는 Table 6과 같다. 2020년도 규격화된 무기체계 소프트웨어 282건 중 주장비의 제어를 담당하는 내장형 소프트웨어가 177건으로 총 63%를 차지했다. 이러한 무기체계 소프트웨어를 대상으로 ISO/IEC 25023의 품질특성이 품질 요구사항으로 정의되어 개발된다면 2020년도 무기체계 소프트웨어 기준으로 최대 63%의 소프트웨어 품질 향상을 기대할 수 있을 것이다.

Table 6. Suggested method and Expected effects in weapon system software

View	Institutional Method	Technical Method	Expected effects
Process	○ Standardization of quality characteristics.	○ Selecting quality characteristics for Software Quality Requirements using ISO/IEC 25023 characteristics	○ Clarify Software Quality Requirements.
Product	○ Expansion of measuring method ○ Quantification of software quality	○ Using 86 Measurement function to Quantify characteristics	○ Up to 63% of software quality improvement as Quality characteristic applicable business accounts for 63%.

## 5. 결론

기존에 사용되던 품질 요구사항은 일관된 기준으로 정의되지 않거나 정량적으로 측정될 수 없는 품질 요구사항 항목들이 있었다. 그래서 품질 요구사항이 개발 프로세스 내에서 산출물로서 명확히 관리되기 어려웠다.

이번 연구를 통해 제시한 ISO/IEC 25023의 소프트웨어 품질특성과 품질 측정 기준은 무기체계 소프트웨어의 품질 요구사항으로 활용될 수 있을 것이다. 또한 무기체계 소프트웨어의 개발 프로세스에서 소프트웨어 품질 요구사항이 다루지는 방식을 수정하면 더 효율적으로 소프트웨어 품질을 향상시킬 수 있을 것이다.

그렇지만 무기체계 소프트웨어의 중요도가 높다하더라도 ISO/IEC 25023의 부특성과 측정합수를 모두 활용하는 것은 실질적으로 불가능하다. 따라서 개별적 사업의 특성과 사업에 할당된 자원, 중요도, 사업의 품질관리수준 등에 따라 품질 항목을 선택하는 방안에 대한 추가 연구가 필요할 것으로 생각된다.

## References

- [1] Defense Acquisition Program Administration(DAPA), "Weapon System development and management manual", Notice, DAPA, Korea, 2020.
- [2] In Soo Ryu, Byeong-Hun Park , "A Study on improvement method of software quality assurance in development phase", *Communications of the Korean Institute of Information Scientists and Engineers*, Vol.35, No.12, pp.53-60, 2017.
- [3] Hong Ju Kim, Chan Hyeon Park, Euiwhan Kim, "A Study on the Evaluation Method of the Quality of the Weapons System Software Using ANP in the Operation/Support Phase", *Journal of the Korean Association of Defense Industry Studies*, Vol.25, No.2, pp.1-14, 2018.
- [4] International Standard Organization(ISO), Systems and software engineering — Systems and software Quality Requirements and Evaluation(SQuaRE) — Measurement of system and software product quality, *ISO/IEC 25023*, Geneva, Switzerland, 2016.
- [5] Korean industrial Standards(KS), Systems and software engineering — Systems and software Quality Requirements and Evaluation(SQuaRE) — Measurement of system and software product quality, *KS X ISO/IEC 25023*, Seoul, Korea, 2019.
- [6] Dorothy Graham, Erik van Veenendaal, Isabel Evans, Rex Black, "Foundations of Software Testing: ISTQB Certification", p.288, International Software Testing

Qualifications Board, 2019, p.288.

- [7] Scott W. Ambler, Ron Jeffries, Ron Jeffries, "Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process", p.400, John Wiley & Sons, 2002, p.400.

윤 경 환(Gyeonghwan Yoon)

[정회원]



- 2013년 2월 : 서울시립대학교 전 자전기컴퓨터공학부(학사)
- 2013년 1월 ~ 2016년 8월 : 현대오트론 연구원
- 2018년 12월 ~ 현재 : 국방기술품질원 연구원

<관심분야>

무기체계 소프트웨어, 소프트웨어 품질관리, 테스트

류 지 선(Jiseon Yu)

[정회원]



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

<관심분야>

무기체계 소프트웨어, 소프트웨어품질