공개소프트웨어 R&D 결과물 품질 검증방안 연구

한정훈
한국정보통신기술협회
jhhan@tta.or.kr

A Study on Quality Validation Measures of Open Source Software R&D Result

Jeong-Hoon Han
Telecommunications Technology Association

요 약

그 동안 통합발주로만 공공 분야에 도입됐던 공개소프트웨어가 2022년 9월부터 분리발주를 통한 조달 등록이 가능해짐에 따라, 공공 분야에서의 공개소프트웨어 도입 및 활용이 확대될 것으로 기대된다. 특히 최근에는 기존에 많이 사용되던 운영체제, 데이터베이스 외에도 인공지능, 클라우드, 블록체인 기술 분야에서의 활용도가 증가하고 있다. 공개소프트웨어는 개발기간 단축, 개발비용과 라이선스 비용 절감 등 다양한 장점이 존재하나, 실제로 이를 도입하려는 기관에서는 상용소프트웨어와 대비하여 품질을 보장할 수 있는지에 대한 불확실성이 부담으로 작용할 수 있다. 따라서 본 논문에서는 도입기관이 안심하고 공개소프트웨어를 도입할 수 있도록 공개소프트웨어 R&D 결과물에 대한 품질을 보증할 수 있는 검증방안을 제안한다.

Ⅰ. 서 론

조달청은 2021년 말에 우수한 국산 소프트웨어의 공공조달 진입을 위해 공개소프트웨어 다수공급자계약제도(MAS, Multiple Award Schedule)[1] 추진을 결정하였으며, 2022년 9월부터 분리발주를 통해 공개소프트웨어의 조달 등록이 가능해졌다. 이를 통해 공개소프트웨어의 공공 시장의 진입 장벽이 낮아져, 공개소프트웨어 R&D가 활성화 될 것으로 기대된다.

공개소프트웨어란 저작권이 있으면서 저작권자가 소스 코드를 공개해 누구나 특별한 제한 없이 해당 코드를 복제, 사용, 배포, 수정, 활용할 수 있는 소프트웨어를 말한다[2]. 공개소프트웨어는 기존에 운영체제(OS), 데이터베이스(DBMS) 제품군 중심으로 개발 및 판매가 이루어졌으나, 최근에는 인공지능, 클라우드, 블록체인 기술 분야에서의 활용도가 증가하고 있어 지속적으로 활용 분야가 확대될 것으로 예상된다[3].

그러나, 실제로 공개소프트웨어를 도입하려는 기관에서는 상용소프트웨어와 대비하여 기능 및 성능 품질을 보장할 수 있는지에 대한 불확실성이 여전히 존재하고, 특히 오픈소스 기반으로 개발된 소프트웨어라는 특성 때문에 보안 취약점(백도어 등)이 충분히 제거되었는지도 도입을 결정하기 위해 고려해야 하는 매우 중요한 요소이다.

본 논문에서는 도입기관이 안심하고 공개소프트웨어를 도입하여 운영할 수 있도록 공개소프트웨어 R&D 결과물에 대한 기능, 성능 및 보안성에 대한 제3자 검증방안을 제안한다.

Ⅱ. 본론

공개소프트웨어가 조달청의 디지털서비스몰에 등록되기 위해서는 GS인증(한국정보통신기술협회 등), 공개소프트웨어확인서(정보통신산업진흥원) 발급 등의 절차를 거쳐야 한다. 이러한 과정을 통해 소프트웨어의 기본적인 품질, 유지관리 지원체계 등의 항목은 제도적으로 검증될 수 있다. 따라서 본 논문에서는 제도적인 장치를 통해 검증하지 못하는 부분으로 예상되는 ① 기능 및 성능 검증(공개소프트웨어 R&D 결과물의 정량적 기능 및 성능 목표치 달성 여부), ② 보안성검증(보안취약점 제거 여부)과 같이 두 가지 분야에 대한 품질 검증 방안을 제시한다.

1. 기능 및 성능 검증 방안

시험 대상 제품의 특성에 따라 다양한 시험항목이 도출될 수 있으나, 본 논문에서는 국제 표준에 근거하여 일반적인 소프트웨어 제품에 적용하여 기능 및 성능을 검증할 수 있는 방법을 다음과 같이 제시한다.

ISO/IEC 25023:2016[4]에는 소프트웨어 제품의 기능 및 성능을 평가할 수 있는 다양한 지표를 명세하고 있으며, 공개소프트웨어 R&D 결과물의 정량적인 기능 및 성능 목표 달성 여부를 평가하기 위한 기준으로 활용할 수 있다. 세부 지표에 대한 내용은 다음과 같다.

구분	기준 및 산정식
기능 정확성	* 기준: 8.2.2 Functional correctness measures (FCr-1-G: Functional correctness) * 산정식: X = 1 - A/B - A: 정확한 결과를 제공하지 않은 기능의 수 - B: 제품에서 제공된 기능의 수
기능 응답시간	* 기준: 8.3.1 Time behaviour measures (PTb-1-G: Mean response time) * 산정식: $\sum_{i=1}^{n}(A_i)/n$ - A_i: i번째 측정에서 특정 사용자 작업이나 시스템 작업에 응답하기 위해 시스템이 소요한 시간 - n: 측정된 응답의 수
기능 소요시간	* 기준: 8.3.1 Time behaviour measures (PTb-3-G: Mean turnaround time) * 산정식: $\sum_{i=1}^{n}(B_i - A_i)/n$ - A_i: 작업 i의 시작 시간

	- B_i: 작업 i의 완료 시간 - n: 측정 횟수
자원 효율성 (프로세서)	* 기준: 8.3.2 Resource utilization measures (PRu-1-G: Mean processor utilization) * 산정식: X = $\sum_{i=1}^{n}(A_i/B_i)/n$ - A_i: 작업을 실행하는데 사용되는 프로세서 시간 - B_i: 작업을 수행하는데 걸린 운영 시간 - n: 관찰 횟수
자원 효율성 (메모리)	* 8.3.2 Resource utilization measures (PRu-2-G: Mean memory utilization) * 산정식: X = $\sum_{i=1}^{n}(A_i/B_i)/n$ - A_i: 작업을 수행하는데 사용되는 실제 메모리 크기 - B_i: 작업을 수행하는데 사용될 수 있는 가용 메모리 크기 - n: 관찰 횟수
트랜잭션 처리 용량	* 8.3.3 Capacity measures (PCa-1-G: Transaction processing capacity) * 산정식: X = A/B - A: 관찰 시간 동안 완료되는 트랜잭션의 수 - B: 관찰 시간
작업 처리율	* 8.3.1 Time behaviour measures (PTb-5-G: Mean throughput) * 산정식: X = $\sum_{i=1}^{n}(A_i/B_i)/n$ - A_i: i번째 관찰 시간 동안 완료된 작업의 수 - B_i: i번째 관찰 지속 시간 - n: 관찰 횟수
사용자 접속 용량	* 8.3.3 Capacity measures (PCa-2-G: User access capacity) * 산정식: X = $\sum_{i=1}^{n}(A_i)/n$ - A_i: i번째 관찰에서 시험대상제품에 동시 접근할 수 있는 최대 사용자 수 - n: 관찰 횟수

[표 1] 공개소프트웨어 기능 및 성능 검증 지표(ISO/IEC 25023:2016)

2. 보안성 검증 방안

공개소프트웨어 R&D 결과물은 직접 개발한 소스 코드와 외부의 공개 SW를 활용한 소스 코드가 함께 사용되어 외부로 배포되는 특징 때문에 일반적인 보안성 검증 방법으로 시큐어코딩 적용 여부를 확인하는 “소프트웨어 보안약점 진단”이 가장 많이 활용되고 있다[2][5]. 해당 진단 방식은 소프트웨어의 분석/설계/구현 단계에서 기능별 보안요구사항 만족 여부, 안전하지 않은 코드 사용 여부를 확인할 수 있어 소프트웨어의 보안성을 향상시키는데 큰 도움을 줄 수 있다.

그러나, 개발자가 악의적인 목적으로 숨겨놓은 백도어와 같은 취약점은 소프트웨어 보안약점 진단 방식으로는 발견하기 어려우며, 이를 정확히 점검하기 위해서는 충분한 시간을 투자하여 소스 코드 수준에서의 점검 및 이상행동 분석을 수행해야 한다. 본 논문에서 제안하는 공개소프트웨어 R&D 결과물의 보안성 검증 방안은 다음과 같다.

구분	상세 내용
소스코드 리뷰	* 소프트웨어 보안약점 진단 수행
	* 소프트웨어의 전체 소스 코드 리뷰를 통해 비정상 행동을 수행하는 코드를 식별하는 것은 현실적으로 많은 시간과 비용이 소요되므로, 데이터 송/수신 기능이 구현된 코드를 샘플링하여 악의적인 코드 삽입 여부 확인
이상행동 분석	* 공개소프트웨어가 도입 및 운영되는 환경과 동일한 수준의 시험환경을 구축하고, 설치된 소프트웨어의 비정상 행동 수행 여부 확인 - 통신 데이터 모니터링(1주/1개월/1년)을 통해 허용된 통신 주소 및 포트가 아닌 경로로 중요 데이터를 유출하는 등의 비정상 행동 수행 여부 확인 - 은닉채널(Covert Channel)을 이용하여 중요 데이터를 유출하는지 검증하기 위해 통신프로토콜 및 업무 데이터 규격과 상이한 구조의 통신 데이터가 존재하는지 확인(데이터 규격이 명확한 경우, 자동화 분석 도구 개발 가능)

[표 2] 공개소프트웨어 보안성 검증 방안

Ⅲ. 결론

본 논문에서는 공개소프트웨어 R&D 결과물에 대한 품질 검증방안으로 기능 및 성능 검증 방안과 보안성 검증 방안을 제시하였다. 기능 및 성능 검증은 정량적 목표가 명확한 경우, 단 기간 내에 시험을 통해 확인이 가능하다. 그러나 보안성 검증은 최종 수행년도에 결과물이 완성되는 일반적인 R&D 과제 진행방식에는 적용하기 어려운 부분이 존재한다. 특히 국가에서 발주하는 R&D 과제는 마지막 수행년도에 결과물이 반드시 산출되어야 하고, 별도의 신뢰된 제3자 전문시험기관을 통해 검증하여 결과물을 보완할 수 있는 충분한 시간적 여유가 없다.

따라서 본 논문에서 제안한 방식을 통해 공개소프트웨어 R&D 결과물의 품질을 검증하기 위해서는, 시험기관의 역량도 중요하나 기존 국가 R&D 과제의 수행기간에 별도의 제3자 검증기간을 추가로 부여하여 충분한 검증이 이루어지도록 정책을 마련할 필요가 있다. 한국정보통신기술협회에서는 보다 효율적으로 공개소프트웨어 R&D 결과물을 검증할 수 있도록, 지속적으로 검증 방안을 연구할 계획이다.

참 고 문 헌

[1] “다수공급자(MAS) 계약제도 이해”, 조달교육원, 2021.11.

[2] “공개SW R&D 실무 수행 가이드라인”, IITP, 2022. 7.

[3] “2021 오픈소스SW(OSS) 실태조사 보고서”, NIPA, 2021.11.

[4] “ISO/IEC 25023:2016 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality, ISO/IEC JTC 1/SC 7, 2016. 6.

[5] “소프트웨어 보안약점 진단가이드”, KISA, 2021.11.