

국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화 고도화 사례

박보경*¹, 장우성*², 문소영*³, 김영철*⁰, 김영수**

*홍익대학교 소프트웨어공학연구소, **정보통신산업진흥원

{*¹park, *²jang, *³msy}@selab.hongik.ac.kr, *⁰bob@hongik.ac.kr, **ysgold@nipa.kr

Advanced Practices on Software Visualization for Improving Code Quality of the Military Defense Software

Bo Kyung Park*¹, Woo Sung Jang*², So-Young Moon*³, R. Young Chul Kim*⁰, Young s. Kim**

*SE Lab, Dept. of Software and Communication Engineering, Hongik University, **NIPA

요 약

현대 소프트웨어는 첨단 무기체계의 핵심요소이며, 그 역할과 기능이 점차 확대되고 있다. 하지만 국내 소프트웨어의 특성인 비가시성, 복잡도 증가 등으로 인해 품질 관리가 어려운 실정이다. 국방 소프트웨어의 품질 개선을 위한 국NIPA 소프트웨어공학센터의 소프트웨어 가시화 고도화 방안을 제안한다. 이 개선된 방법은 프로세스 가시화, 코드 가시화, 문서 가시화로 구성된다. 개선된 소프트웨어 가시화를 통해, 국방 소프트웨어 전체 개발의 추적성(역공학 자동화), 내부 복잡도, 품질 지표 적용, 개발 문서 자동화로 고품질(시간, 비용, 인력) 소프트웨어 개발 용이성을 기대할 수 있다.

1. 서 론

국방 소프트웨어는 대형화, 복잡화됨에 따라 최종제품에서 소프트웨어의 역할과 기능이 점차 확대되고 있다. 이처럼 소프트웨어는 최종 제품의 경쟁력을 좌우하며, 부가가치를 높인다. 하지만 소프트웨어의 특성인 비가시성, 복잡도 증가, 국내의 소프트웨어 개발 환경은 소프트웨어 품질 관리를 어렵게 한다[1,2]. 1) 소프트웨어 개발 관리에서 가장 중요한 문제점은 가시성 확보가 어렵다는 것이다. 현재의 개발 과정은 소프트웨어 개발 후 출시된 최종 제품을 테스트한다. 가시성 확보는 개발 후반에 발생하는 문제가 요구사항 정의에 관한 문제인지, 개발 문제인지에 대해 파악할 수 있다. 이를 위해서, 소프트웨어 개발 전체가 유기적으로 연계되어야 한다. 2) 소프트웨어의 복잡도 증가는 잠재되어 있는 버그나 예상치 못한 위험요소에 의해 제품 납기 지연, 비용 초과, 품질 저하 등의 문제가 발생한다. 이를 해결하기 위해, 체계적이고 정량적인 방법이 필요하다. 3) 국내 소프트웨어 개발 환경은 기존에 제시된 다양한 방법론을 적용하는데 많은 어려움이 있다. 소프트웨어 품질향상 방법은 소프트웨어 테스트, 정형 검증, 소프트웨어 품질 인증 모델 등이 있다[1,2]. 하지만 이러한 방법론들은 추가 인력 및 비용이 발생한다. 또한 오류 검출은 가능하지만 소프트웨어 개선 방법을 제공하지 않아, 쉽게 적용하지 못한다. 그러므로 좀 더 원천적인 소프트웨어 품질 개선 방법이 필요하다.

본 논문에서는 국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화 방법을 제안한다. 이 방법은 프로세스 가시화, 아키텍처 가시화, 문서 가시화로 구성된다. 소프트웨어 가시화를 통해 소프트웨어 개발 과정에서의

문제를 쉽게 발견할 수 있으며, 적은 시간, 비용, 인력으로 품질을 향상시킬 수 있다.

2. 관련연구

소프트웨어 가시화에 대한 연구는 SW Visualization, CodeCrawler 등이 있다. SW Visualization은 효율적인 소프트웨어 개발 관리를 통하여 기업의 소프트웨어 경쟁력 확보를 지원하는 서비스이다[2]. SW Visualization은 1) 품질지표에 의한 명확한 목표수집, 2) 시스템 기반의 효율적인 개발 활동, 3) 시각화를 통한 지속적인 모니터링과 통제가 가능한 소프트웨어 개발 관리 기반을 제공한다. SW Visualization은 공개 소프트웨어 기반 가시화 개발 측면에서 본 연구와 유사하지만, 요구사항 리팩토링 모델 가시화, 테스트케이스 자동 생성 방법 등은 제공하지 않는다. 또한 소프트웨어공학센터가 2016년 11월에 폐지됨으로써, 지속적인 서비스가 어렵다.

CodeCrawler는 경량급 소프트웨어 시각화 도구로 1998년에 Lanza 박사 팀에 의해 구현되었다[3]. CodeCrawler는 언어 독립적이며, 매트릭과 소프트웨어 시각화의 조합을 통해 역공학을 지원한다. 또한 개발자의 생산성을 높이고, 소스 코드 학습 시간을 줄이기 위해 사용 편의성, 성능 및 지능적 검색 기능을 결합하였다. 하지만 CodeCrawler는 공개 소프트웨어 기반의 Plug&Play 기술을 적용하고 있지 않으며, 2002년 10월에 마지막으로 릴리스 되었다.

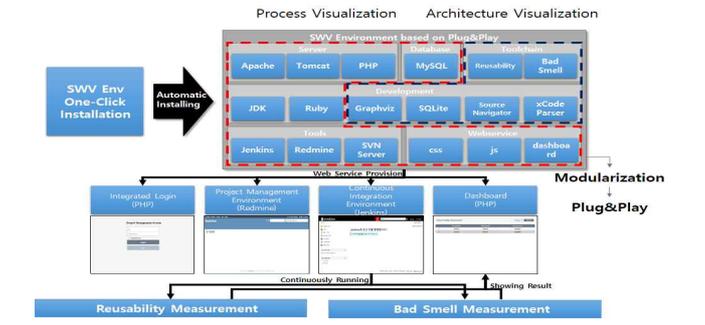
3. 국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화

국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화 방법은 프로세스 가시화, 아키텍처 가시화를 통해 소프트웨어의 비가시성을 극복함으로써 소프트웨어 개발의

전체 과정을 파악할 수 있고, 소프트웨어 개발의 품질 관리가 가능하다. 또한 문서 가시화는 기업의 개발 노하우 관리 및 내부 인력간의 업무 이해도 향상과 특정 상황에서 외부와의 의사소통이 가능하다.

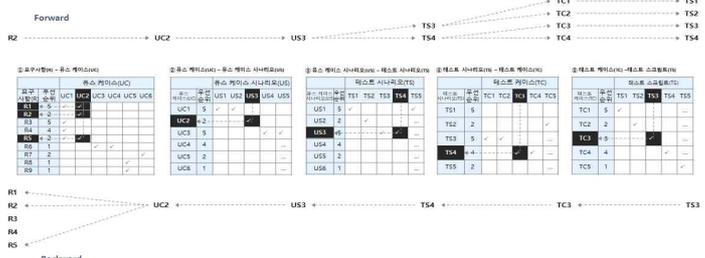
3.1 프로세스 가시화

프로세스 가시화는 소프트웨어 개발 단계에 따른 각각의 방법론을 시스템으로 구성하고, 시스템에서 발생하는 데이터를 종합하여 시각화한다[3]. 프로세스 가시화는 소프트웨어 전체 과정의 상태를 파악할 수 있으며, 유기적인 연계를 통해 현재 개발 상태의 원인을 파악할 수 있다. 또한 품질 지표를 통해 현재 상태의 품질 정보를 제공할 수 있다. 만약 기업에서 소프트웨어 프로세스를 보유하고 있다면, 요구사항 추적성, 스케줄 관리, 품질 지표 측정, 이해 관계자의 신속한 동의, 프로세스, 활동 및 결과물의 효율적인 관리가 가능하다. 하지만 소프트웨어 프로세스는 프로그래머와 전문가의 부족, 불충분한 프로젝트 기간, 비용 문제 등으로 구성하기 매우 어렵다. 성공적인 프로젝트를 위해서는 프로세스 가시화 시스템 자동화가 필요하다. 프로세스 가시화 자동화 시스템은 Plug&Play 가능하며, 자동으로 구축된다. 그림 2는 소프트웨어 프로세스 자동화 시스템의 구조도이다.



(그림 2) 프로세스 자동화 시스템 구조도[3]

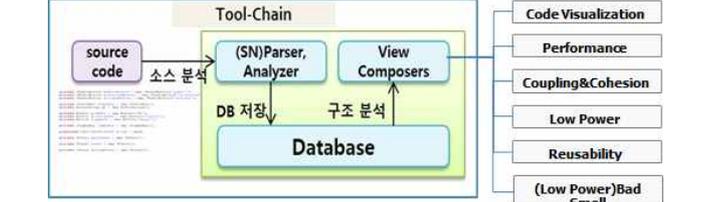
소프트웨어 프로세스를 위해 프로젝트 관리도구, 지속적 통합도구, 버전 관리 도구 및 실행 환경 제공을 위한 서버, 데이터베이스 등을 정의하였다. 프로젝트 관리 도구는 Redmine, 소스코드 자동 빌드와 품질 측정 도구는 Jenkins, 버전관리 Server는 VisualSVN, Client는 TortoiseSVN을 적용하였다. 또한 서버와 데이터베이스 구축을 위해 Apache, Tomcat, PHP, Java, MySQL 등을 구축하였다. 프로세스 가시화 시스템을 통해 요구사항의 변경 관리, 프로젝트 일정 관리, 이슈 관리, 지속적 통합 및 자동 빌드 등이 가능하며, 소프트웨어 요구사항 정의에서부터 테스트까지 전체 과정을 가시화할 수 있다.



(그림 3) 요구사항 추적 매트릭스[3]

그림 3은 소프트웨어 요구사항 추적 매트릭스이며, 프로세스 가시화를 통해 분석된 가시화 정보이다.

3.2 코드 가시화



(그림 4) 코드 가시화 구조도(Tool-Chain)

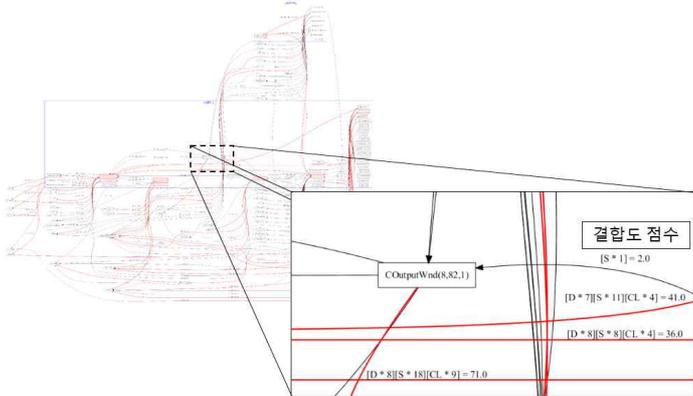
코드 가시화는 소스 코드 내부 구조를 가시화하고 품질을 개선할 수 있다. 그림 4는 코드 가시화 구조도이다. 코드 가시화는 역공학을 기반으로 소스 코드를 분석할 파서(Parser), 분석한 정보를 저장하는 데이터베이스, 분석한 정보를 이미지화 할 수 있는 뷰 컴포저(View Composer)로 구성된다. 구성된 도구를 툴체인(Tool-Chain)이라 하며, 코드 가시화 과정은 다음과 같다[1].

- 1단계: SNDB 파일 추출은 툴 체인 분석을 위해, 먼저 소프트웨어에 대한 정보가 필요하다. 소프트웨어의 정보는 프로그램의 소스 코드를 소스 네비게이터에 입력하고 프로그램 코드에 함수 정보, 지역변수, 전역변수, 파라미터 등의 정보를 SNDB 파일로 추출한다.
- 2단계: 소프트웨어 정보 추출 단계는 소스 네비게이터에서 추출한 모든 SNDB 파일(.cl, .con, .e, 등)이 바이너리 형태로 출력된다. 바이너리 형태는 직접 읽을 수 없기 때문에 소스 네비게이터의 DBdump 실행파일을 이용하여 텍스트 정보로 변환한다.
- 3단계: 이전 단계에서 추출된 모든 소프트웨어 정보를 데이터베이스의 각 테이블 별로 각 요소에 맞는 정보를 선택해 저장한다.
- 4단계: 품질지표 적용단계에서는 고품질화를 위해서 결합도에 추출을 위한 쿼리 등을 정의하고, 정량적 측정 점수를 정의한다.
- 5단계: 가시화는 소스 코드를 시각적으로 표현하는 단계로써, 4단계에서 정의된 품질 지표에 대한 쿼리 문 등으로 정보를 추출하고, 품질지표에 대한 정량화된 수치를 Dot 스크립트로 가시화한다.

코드 가시화 과정을 통해 개발 중인 소스 코드의 시각화 및 정량적 수치를 제공하여 소프트웨어의 품질을 측정할 수 있다. 따라서 개발 중인 소프트웨어의 진척 상황을 바로 확인 가능하며, 소스 코드의 정량적 분석이 가능하다. 또한 개발 중인 코드에서 어느 부분이 복잡하고 잘못되었는지 시각화 그래프를 통해 확인 가능하기 때문에 개발자의 나쁜 습관을 개선할 수 있다. 그 외에도 코드 가시화는 소스 코드의 성능 측정, 재사용성, 저전력 코드 분석 등이 가능하다.

그림 5는 COutputWnd 모듈의 성능 측정 가시화 결과이다. (1)은 “모듈이름” (“호출횟수”, “수행시간 [ms]”, “성능저해요소 개수”)로 구성된다. COutputWnd의 괄호 안의 숫자는 8,82,1이다. 즉 이 모듈은 8번 호출

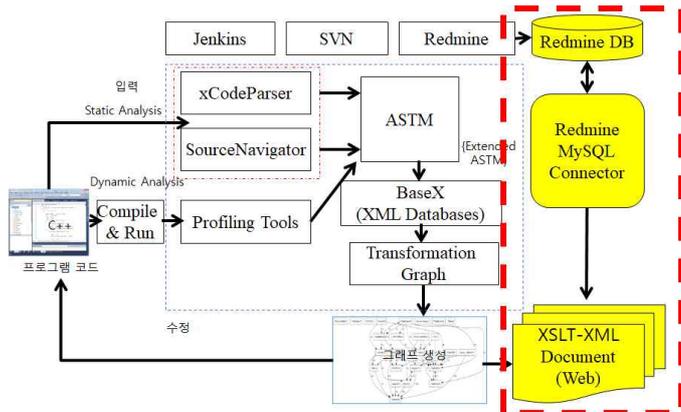
되었고, 수행 시간은 82[ms]이며, 성능저해요소의 개수는 1개이다. 또한 화살표에 있는 [S*1], [D*7], [CL*4]등의 정보는 다른 모듈과의 결합도 점수를 측정하는 것이다.



(그림 5) 성능 가시화 결과

3.3 문서 가시화

산출물 관리, 발주자 요구사항 등은 문서형태로 작성된다. 하지만 현재의 소프트웨어 개발 문서들은 소프트웨어 개발 이후에 문서 작업을 수행하기 때문에 소프트웨어 개발 활동에 부담으로 작용한다[4]. 본 연구에서는 개발 과정에서 시스템 내부에 있는 다양한 산출물을 문서화할 수 있는 방안을 제시한다. 문서 가시화는 소프트웨어의 개발 효율성과 추적성 확보가 필요하며, 이는 소프트웨어 개발의 전체 활동을 시스템화하여 연계하는 것이 중요하다.

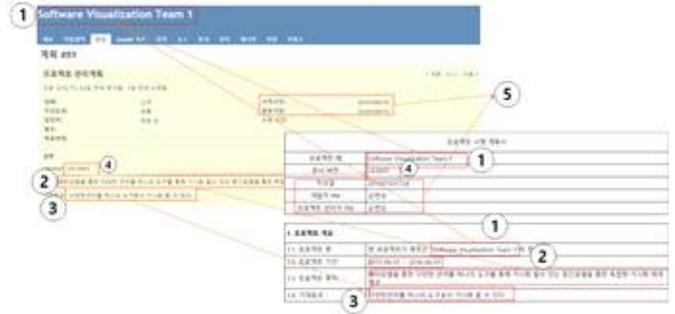


(그림 6) 문서 가시화 시스템 구성도[4]

그림 6은 소프트웨어 프로젝트 관리를 위한 문서 가시화 시스템 구성도이다. 이 시스템은 효과적인 프로젝트 관리를 위한 프로젝트 관리 계획서, 소프트웨어 구조 분석서를 자동으로 추출한다. 프로젝트 관리계획서를 추출하기 위해서 레드마인의 일감에서 직접 추출할 수 없는 개요, 범위, 계획상세 등을 일감의 설명항목에 일정한 패턴(특수 문자)으로 입력한다. 이와 더불어 직접 추출할 수 있는 정보(시스템 구축환경, 프로젝트 명 등) 등을 데이터베이스에서 불러와 XML파일로 출력한다. 추출된 XML 파일은 XSLT와 연동되어 웹에서 출력할 수 있다.

그림 7은 레드마인의 일감에 있는 내용을 프로젝트 관리 계획서로 추출한 결과이다. ①은 프로젝트 명이다. ②

는 프로젝트의 목적이며, ③은 프로젝트의 기대효과이다. ④는 문서 버전이며 ⑤는 작성일(일감의 날짜), 프로젝트의 관리자 정보를 조회하여 작성한 것이다.



(그림 7) 프로젝트 관리 계획서 결과[4]

4. 결론

국방 소프트웨어는 대형화, 복잡화됨에 따라 최종제품에서 소프트웨어의 역할과 기능이 점차 확대되고 있다. 하지만 현재의 소프트웨어는 비가시성, 복잡성, 변경성 등의 특징이 있고, 다양한 요구와 개발자에 대한 개인의 의존도가 높다. 소프트웨어 개발 관리의 가장 중요한 문제점은 가시성 확보가 어렵다는 것이다. 이에 대한 해결책으로 여러가지 프로세스와 개발 방법론들이 제안되고 있지만 소프트웨어의 복잡성만 증가시킬 뿐 본질적인 문제는 해결되지 않고 있다. 본 논문에서는 국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화 방법을 제안하였다. 프로세스 가시화는 소프트웨어 전체 과정의 상태를 파악할 수 있으며, 유기적인 연계를 통해 현재 개발 상태의 원인을 파악할 수 있다. 코드 가시화는 소프트웨어 개발 과정에서의 문제점을 쉽게 발견할 수 있으며, 적은 시간, 비용, 인력으로 품질 개선이 가능하다. 문서 가시화는 소프트웨어의 개발 효율성과 추적성을 확보할 수 있다. 향후 연구로 본 연구의 방법을 국방 소프트웨어에 적용하고자 한다.

ACKNOWLEDGEMENT

본 논문은 2019년도 산업통상자원부의 ‘창의산업융합 특성화 인재양성사업’ (과제번호 N0000717)과 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017RID1A3B03035421).

참고 문헌

[1] 박보경, 문소영, 서채연, 김영철, “국방 소프트웨어 자원 관리시스템의 소스 코드 품질 향상을 위한 소프트웨어 가시화 적용 구축 사례”, 한국정보과학회, pp.161-163, 2017.12
 [2] NIPA 소프트웨어공학센터, “SW개발 품질관리 매뉴얼”, 2013.12.
 [3] 이진협, “소프트웨어 프로세스 자동화 기반의 요구사항 추적성 매트릭스 연구”, 홍익대학교 대학원 석사학위 논문, 2018.
 [4] 강건희, 손현승, 이근상, 김영철, 이상은, “효과적인 프로젝트 관리 계획서 위한 프로젝트 문서 생성 자동화”, 한국정보처리학회, 제22권, 제2호, pp.959-961, 2015.