

# 육군 정보체계 소프트웨어 개발 관리 프로세스 개선 사례

(Best Practices on Software Development and Management  
Process for the Republic of Korea Army Information System)

박 보 경 <sup>†</sup>    서 채 연 <sup>\*\*</sup>    김 기 두 <sup>\*\*\*</sup>    이 중 훈 <sup>\*\*\*\*</sup>    김 영 철 <sup>\*\*\*\*\*</sup>  
(Bo Kyung Park)    (Chae Yun Seo)    (Ki Du Kim)    (Jong Hoon Lee)    (R.Young Chul Kim)

**요 약** 육군 정보체계관리단은 고품질 소프트웨어 개발을 위해 소프트웨어 개발 관리 프로세스의 도입에 대한 관심이 높다. 그래서 상업용 소프트웨어 및 운영체제를 기반으로 육군 정보체계 소프트웨어 개발 시스템을 제한적으로 구축하였다. 하지만 육군의 현안은 공개 소프트웨어 기반의 소프트웨어 개발과 관리 시스템을 어떻게 완전하게 구축하느냐 하는 것이다. 이러한 문제를 해결하기 위해, 육군 정보체계 소프트웨어의 품질 개선을 위한 개방형OS 기반의 소프트웨어 개발 관리 프로세스로 기존의 육군 정보 시스템을 개선하는 방법을 제안한다. 이 방법은 CentOS 기반의 모든 공개 소프트웨어를 통해 독립적이고 확장성에 유연한 소프트웨어 개발 관리 프로세스를 구축할 수 있는 해결책을 제공한다. 이 프로세스는 소프트웨어 전체 개발 과정에 재정의된 품질 지표 적용, 개발 문서 자동 생성, 코드 복잡도 식별과 같은 메커니즘에 쉽게 적용할 수 있으며, 고품질 소프트웨어 개발을 용이하게 한다. 향후 수명주기 동안 소프트웨어 프로세스 교육, 설계, 개발, 유지보수, 구축과 같은 각 단계별 절차를 확대하고자 한다.

**키워드:** 소프트웨어 품질, 코드 가시화, 설계 문서 자동화, 프로세스 가시화, 소프트웨어 개발 관리

**Abstract** Korea Army HQ Information System Management Group is in absolute need to construct a software development management process to develop high-quality software. Therefore, they tried to partially construct the Army information software development system based on commercial software and operating systems. But the current issue is the complete construction of the open source-based software development and management system. To solve this problem, we propose the means to enhance the previous Army Information System with an open OS-based software development management process for quality improvement of Army software. This approach brings a solution to build an independent, salable and flexible software development management process with all open sources based on CentOS. This process can be easily adapted with some mechanisms such as

· 이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2020R111A1A01072928)과 2019년도 산업통상자원부의 '창의산업융합 특성화 인계양성사업'의 지원을 받아 수행된 기초연구사업(과제번호 N0000717)

· 이 논문은 2019국방소프트웨어연구회 추계워크숍에서 '국방 소프트웨어의 품질 개선을 위한 소프트웨어 가시화 고도화 사례'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 비 회 원 : 홍익대학교 전자전산공학과 학생  
parkse@hongik.ac.kr

<sup>\*\*</sup> 비 회 원 : 홍익대학교 소프트웨어융합학과 교수  
chaeyun@hongik.ac.kr

<sup>\*\*\*</sup> 비 회 원 : 한국정보통신기술협회 팀장  
kdkim@tta.or.kr

<sup>\*\*\*\*</sup> 비 회 원 : 홍익대학교 소프트웨어융합학과 교수  
mizzouri@naver.com

<sup>\*\*\*\*\*</sup> 정 회 원 : 홍익대학교 소프트웨어융합학과 교수(Hongik Univ.)  
bob@hongik.ac.kr  
(Corresponding author임)

논문접수 : 2020년 3월 6일

(Received 6 March 2020)

논문수정 : 2020년 7월 20일

(Revised 20 July 2020)

심사완료 : 2020년 7월 24일

(Accepted 24 July 2020)

Copyright©2020 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지 제47권 제10호(2020. 10)

application of the redefined quality metrics, automatically generating development documentation, and identifying the code complexity on the entire software lifecycle. Also, the process possibly facilitates the development high-quality software. In the future, we will need to extend the process by involving software process training, designing, development, maintenance and establishment at each phase of the lifecycle.

**Keywords:** software quality, code visualization, design document automation, process visualization, software development management

## 1. 서론

최근 IT 융복합화가 진행되면서, 세계 IT 시장은 하드웨어에서 소프트웨어 중심으로 변화하고 있다. 또한 소프트웨어는 기존 산업(조선, 항공, 국방, 의료 등)과 융합은 물론이고 최근 주목받고 있는 클라우드 컴퓨팅, 빅데이터 등에 활용되어 기업 경쟁력의 원천으로 자리 잡고 있다. 소프트웨어는 최종 제품의 경쟁력을 좌우하며, 부가가치를 높이는 역할을 하고 있지만, 소프트웨어의 특징인 비가시성, 복잡도 증가, 국내 기업의 소프트웨어 개발환경은 소프트웨어 품질 관리를 어렵게 한다[1,2].

현재의 소프트웨어 개발 과정은 소프트웨어의 최종 제품을 테스트한다. 따라서 개발 후반에 발생하는 문제가 요구사항 정의에 관한 문제인지, 개발 문제인지에 대해 파악하기 어렵다. 소프트웨어의 복잡도 증가는 잠재되어 있는 버그나 예상치 못한 위험요소에 의해 제품 납기 지연, 비용 초과, 품질 저하 등의 문제가 발생한다[3]. 국내 기업의 소프트웨어 개발 환경은 인력, 시간, 비용 등으로 기존에 제시된 다양한 방법론을 적용하는데 많은 어려움이 있다[4].

이러한 문제를 해결하기 위해, 육군에서는 오래 전부터 소프트웨어 개발 관리 프로세스를 적용하고 있다. 육군 정보체계관리단은 육군 본부 및 예하 부대의 정보체계 유지보수, 군이 필요로 하는 정보체계의 자체 개발, 외주 개발 관리 임무를 수행하는 기관으로, 기존의 소프트웨어 개발 관리 프로세스에 윈도우 기반으로 소프트웨어 가시화 체계가 도입되었다. 윈도우 기반 소프트웨어 가시화 체계는 일반 사용자들에게 매우 익숙하며, Bitnami, Docker 등과 같은 쉬운 설치, GUI기반의 환경 설정, 호환되는 응용 프로그램이 많다는 장점을 가진다. 하지만 윈도우 기반의 문제점인 고가의 구입비용 등으로 인해 적극적인 도입 및 확산을 하지 못하는 실정이며, 체계 구성이 어렵다. 또한, 윈도우 운영체제는 잦은 버전 업(평균 2년)과 사용자와 관계없이 일방적인 기술지원의 종료 등도 큰 문제점이 되고 있다. 특히, 군의 특성 상 무기체계 및 소프트웨어의 잦은 변경이 어렵다. 만약 운영체제 단종 및 변경이 발생한다면, 무기체계 소프트웨어의 정비 및 성능 개선에 많은 문제점들이 나타날 것이다.

본 논문에서는 육군 정보체계 소프트웨어의 품질 개

선을 위한 개방형OS기반의 소프트웨어 가시화 방법을 제안한다. 이 개방형OS기반의 소프트웨어 가시화 체계는 국가의 산업 기술 활성화를 위해서 소프트웨어 품질 향상이 가능한 새로운 방법이다. 이 방법은 공개소프트웨어를 적극 활용하여 추가적인 인원 및 비용이 발생되지 않고 기업에서는 쉽게 적용할 수 있다. 또한 독립적인 소프트웨어 의존도를 줄이기 위해 독립적이고 확장에 유연한 체계적인 방법을 사용하였으며, 기존의 공개 소프트웨어를 쉽게 자산화 할 수 있기 때문에 외산 소프트웨어에 대한 국가 경쟁력을 갖출 수 있다.

## 2. 관련 연구

소프트웨어 가시화(Software Visualization)의 목적은 소프트웨어 시스템(구조) 및 알고리즘의 이해, 소프트웨어 시스템의 분석 및 탐색, 품질 지표(결합도, 응집도, 클래스 등), 개발과 개선 방법을 제공한다. 소프트웨어 가시화의 강점 중 하나는 요구사항, 설계, 코드 변경에 대한 것을 추적하는 것과 본질적으로 연결되지 않은 소프트웨어 시스템의 정보를 결합하고 관련시켜 보이지 않는 정보를 볼 수 있도록 하는 것이다. 이 가시화는 소프트웨어 시스템 정보를 탐색 및 분석하는 도구와 기술로 사용될 수 있다. 또한 코드 품질이나 팀 활동과 같은 활동을 모니터링 하는데 사용된다. 본 절에서는 국내외 소프트웨어 가시화 기술을 소개한다.

국내의 소프트웨어 가시화 기술은 눈비소프트의 피니언(Pinion)[11], 정보통신산업진흥원 소프트웨어공학센터의 SW Visualization[1]이 있다.

피니언은 설계된 산출물을 서버의 운영 모듈에 보내주고, 이것을 실시간으로 인터프리팅(Interpreting)을 통해 서비스하는 구조이다[11]. 하지만 피니언은 완성된 개발 코드의 가시화 서비스는 제공하고 있지 않으며, 공개소프트웨어기반의 Plug & Play 기술을 적용하고 있지 않다.

정보통신산업진흥원의 소프트웨어공학센터에서 제공하는 SW Visualization은 효율적인 소프트웨어 개발 관리를 통하여 기업의 소프트웨어 경쟁력 확보를 지원하기 위한 서비스로, 1) 품질지표에 의한 명확한 목표수립, 2) 시스템 기반의 효율적인 개발 활동 3) 시각화를 통한 지속적인 모니터링과 통제를 가능토록 하여 성공적인 소프트웨어 개발 관리 기반을 제공한다[1]. 하지만

정보통신산업진흥원 소프트웨어공학센터가 2016년 11월에 폐지됨으로써, 소프트웨어 가시화 서비스가 더 이상 지원되지 않고 있다.

국외 소프트웨어 가시화 도구는 CodeCrawler, Architecture Explorer, Class Visualizer, Code\_Swarm, JIVE, X-Ray 등이 있다. 표 1은 국외 소프트웨어 가시화 도구를 비교한 것이다.

Architecture Explorer는 JVM 기반 응용프로그램의 아키텍처를 분석하는 브라우저 기반의 가시화 프로그램이다[5]. 이 기술은 자바 바이트 코드로 작성된 그래프

가 브라우저에 표시되며, 다양한 뷰(jar, 패키지, 클래스, tangles) 및 필터가 지원되며 모듈화 문제와 관련된 여러 반 패턴 및 매트릭을 산출한다. 하지만 웹브라우저에 따라 속도 차이가 크다. Class Visualization은 자바 바이트 코드에서 클래스 다이어그램을 생성한다[6]. 하지만 매트릭스 추출 기능이 존재하지 않는다. Code\_Swarm은 소프트웨어 프로젝트의 commit 이력을 시각화하는 프로그램이지만 commit 이외에 소프트웨어 개발 지원 기능이 없다[7]. JIVE는 자바 프로그램을 시각화하는 Eclipse 플러그인이다[8]. 이 도구는 객체 구조와 메소

표 1 국외 소프트웨어 가시화 도구 비교  
Table 1 Comparison of International Software Visualization Tool

Tool Feature	Architecture Explorer[5]	Class Visualizer[6]	Code_Swarm[7]	JIVE[8]	X-Ray[9]	CodeCrawler [10]
Function	<ul style="list-style-type: none"> <li>- JVM based architecture analysis</li> <li>- Generating Multiple repeating patterns and metrics</li> </ul>	<ul style="list-style-type: none"> <li>- Creating class diagram</li> <li>- Analyze all relationships, including nested classes, dependencies, and exception handling</li> </ul>	<ul style="list-style-type: none"> <li>- Visualizing Commit history of software project</li> </ul>	<ul style="list-style-type: none"> <li>- State diagram provided</li> <li>- Visualize object structure and method interaction</li> </ul>	<ul style="list-style-type: none"> <li>- Visualization of system complexity, including class and package dependencies for Java projects</li> </ul>	<ul style="list-style-type: none"> <li>- Metric extraction</li> <li>- Software visualization based on reverse engineering technology</li> </ul>
Strength	<ul style="list-style-type: none"> <li>- Design defect detection and system maintenance improvement</li> <li>- Convenient to build plug-ins and extensions</li> <li>- product-Line engineering support</li> </ul>	<ul style="list-style-type: none"> <li>- No storage present, no sync issues</li> <li>- exploring available</li> </ul>	<ul style="list-style-type: none"> <li>- Easy to manage commitment</li> <li>- Contains previous Commit information in histogram format</li> </ul>	<ul style="list-style-type: none"> <li>- Possibility of runtime verification</li> <li>- Enable large system visualization with filters and dynamic slicing</li> </ul>	<ul style="list-style-type: none"> <li>- Highly scalable open source software for the Eclipse framework</li> <li>- User can manipulate to increase result visibility</li> </ul>	<ul style="list-style-type: none"> <li>- Light weighted tools</li> <li>- Language independent</li> <li>- Combination of convenience, superior performance and intelligent search</li> </ul>
Weakness	<ul style="list-style-type: none"> <li>- Speed difference depending on web browser</li> <li>- No Internet Explorer support</li> </ul>	<ul style="list-style-type: none"> <li>- Extracted data cannot be used for individual use</li> <li>- No matrix extraction functioning</li> </ul>	<ul style="list-style-type: none"> <li>- No software development support other than Commit</li> </ul>	<ul style="list-style-type: none"> <li>- No matrix extraction functioning</li> <li>- Lack of object convenience in user diagram</li> </ul>	<ul style="list-style-type: none"> <li>- Limited matrix support</li> <li>- Does not provide dynamic visualization</li> </ul>	<ul style="list-style-type: none"> <li>- Does not apply Plug &amp; Play technology based on open software</li> </ul>
Commercial/Op ensource	Opensource	Opensource	Opensource	Opensource	Opensource	Opensource
2D/3D	2D	2D	2D	2D	2D	2D/3D
Class Diagram	✓	✓				
Dynamic Visualization				✓		
Commit Visualization			✓			
Metrics	✓					✓
System Complexity					✓	✓
Reverse Engineering						✓
Plug in Support	✓			✓	✓	

드 상호작용에 대한 시각화뿐만 아니라 상태 다이어그램을 기반으로 런 타임 검증을 목표로 한다. 하지만 매트릭스 추출 기능이 없다. 마지막으로 X-Ray는 Eclipse 프레임워크 오픈 소스 소프트웨어 시각화 플러그인으로 자바 프로젝트에 대한 시스템 복잡도, 클래스 및 패키지 종속성을 가시화한다[9]. 하지만 동적 가시화를 제공하지 않으며 매트릭스 기능이 제한적으로 제공된다. Code-Crawler는 경량급 소프트웨어 시각화 도구로, 1998년에 Lanza 박사 팀에 의해 처음으로 구현되었다[10]. 이 도구는 매트릭 추출이 가능하고, 역공학 기반의 소프트웨어 시각화를 지원한다. 하지만, 공개 소프트웨어 기반의 Plug & Play 기술을 적용하고 있지 않다.

**3. 육군 정보체계 소프트웨어의 품질 고도화 방안**

본 절에서는 육군 정보체계 소프트웨어의 품질 고도화를 위한 개방형 OS 기반의 소프트웨어 가시화 방법을 설명한다. 그림 1은 육군 정보체계 소프트웨어의 품질 고도화 방법이다. 먼저, 객체지향 프로그램의 품질 측정을 위해, 소프트웨어의 품질지표를 재정의하고 측정 방법을 정의한다. 재정의된 품질지표의 정량적 측정을 위해 코드 가시화(Tool-Chain)를 개발한다. 또한 프로젝트 수행 시, 효율적인 업무 수행을 위해 설계 문서의 자동화 기술을 개발한다. 마지막으로, 육군 정보체계 소프트웨어 개발 프로세스의 시각화를 위한, 프로세스 가시화 환경 프레임워크를 개발한다.

**3.1 소프트웨어의 품질지표 종류 및 측정 방법 정의**

기존 절차적 소프트웨어 개발에서 신뢰성과 유지보수 측면에서의 품질 척도 중 하나는 응집도(Cohesion)와 결합도(Coupling)이다[12,13]. 응집도는 모듈 내 요소들의 관계를 측정하는 척도이고, 결합도는 모듈 간 의존도를 측정한다. 하지만 객체지향 소프트웨어는 객체의 추상화를 통해 시스템 모듈을 쉽게 재사용하고, 이해가 쉬운 소스코드 구조의 소프트웨어를 만드는데 중점을 둔다. 그렇기 때문에 기존 절차지향 측면의 응집도와 결합도를 객체지향 소프트웨어에 단순 적용하기에는 맞지

않는 부분이 존재했다. 하지만 객체지향 방법론이 적용되어 개발되더라도 모든 소프트웨어가 좋은 품질을 갖는 것은 아니기 때문에, 객체지향 소프트웨어의 품질 측정 방법은 매우 필요하다.

객체지향 시스템에서 중요 요소는 클래스로 모델링되고, 클래스는 멤버 변수와 메소드로 이루어진다. 시스템은 클래스로부터 생성된 객체들로 구성되고, 클래스 간의 상호 작용으로 동작된다. 기존 결합도와 응집도 측정 방법을 확장하여 객체지향 언어 기반 프로그램의 품질을 측정한다면, 기존의 검증된 품질 척도를 객체지향 프로그램에도 적용할 수 있는 기준이 될 수 있다. 본 논문에서는 결합도에 관해서 언급한다.

**3.1.1 품질지표 재정의**

결합도는 모듈 사이의 상호 연관성의 복잡도를 나타내는 척도이다. 높은 결합도를 가진 모듈은 하나의 모듈이 다른 모듈의 기능을 많이 사용한다는 것을 의미한다[14]. 결합도에는 자료 결합도, 스탬프 결합도, 제어 결합도, 외부 결합도, 공통 결합도, 내용 결합도가 있다. 결합도가 높을수록 다른 모듈과 많은 연관이 있기 때문에 구조 변경, 유지보수, 품질이 나빠짐을 의미한다. 가장 높은 결합도는 내용 결합도이고, 가장 낮은 결합도는

표 2 재정의된 결합도  
Table 2 The Redefined Coupling

Coupling	Description
Data	Parameters of primitive types are exchanged between classes and objects
Stamp	Array or object type parameters are exchanged between class and object
Control	Instructs the class and object control where the calling class and object variable are called
External	External files and data between classes and objects are transferred
Common	Global variables are used as parameters in calls between classes and objects
Content	Directly change the internal properties of other objects between classes and objects

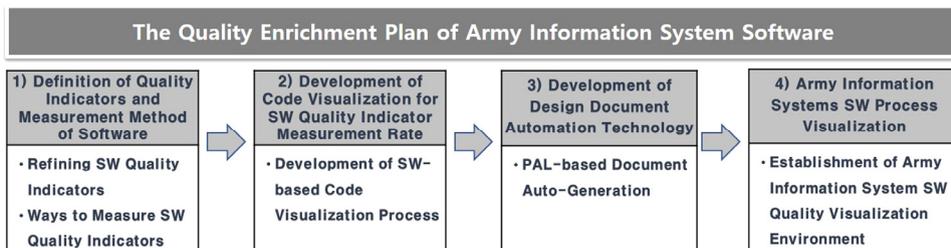


그림 1 육군 정보체계 소프트웨어 품질 고도화 방법

Fig. 1 The Method to Enhance Software Quality of Army Information System

자료 결합도이다. 본 논문에서는 객체지향적으로 결합도를 재정의 하였다. 표 2는 재정의된 결합도이다.

3.1.2 품질지표 측정 방법

본 절은 재정의된 품질지표를 기반으로 결합도 측정 방법을 제시한다.

(1) 자료 결합도

객체지향 언어에서의 자료 결합도는 클래스/객체 간에 기본 타입의 매개 변수가 교환되는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 자료 결합도의 예시를 보여준다. int형 데이터를 매개변수로 전달한다.

```
Circle c = new Circle();
c.setRadius(3);
```

(2) 스탬프 결합도

스탬프 결합도는 클래스/객체 간에 배열 또는 오브젝트 타입의 매개 변수가 교환되는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 스탬프 결합도의 예시를 보여준다. 배열 형 데이터가 매개변수로 전달된다.

```
int array[3] = {1,2,3};
MathUtil mu = new MathUtil();
mu.getSum(array);
```

(3) 제어 결합도

제어 결합도는 호출하는 클래스/객체의 변수가 호출되어지는 클래스/객체의 제어를 지시하는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 제어 결합도의 예시를 보여준다. manageFile 메소드의 b 변수에 의해 RW 메소드의 제어가 결정된다.

```
class SubPro {
    public void manageFile() {
        boolean b = true;
        ListOb lo = new ListOb();
        lo.RW(b);
    }
}
class ListOb {
    public void RW(boolean isRead) {
        if(isRead){
            System.out.println("read");
        } else {
            System.out.println("write");
        }
    }
}
```

(4) 외부 결합도

외부 결합도는 클래스/객체 간의 외부 파일 및 데이터가 전송되어 사용되는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 외부 결합도의 예시를 보여준다. 외부 파일이 Dto 클래스에 전송된다.

```
Dto dto = new Dto();
File f = new File("text.txt");
dto.readFile(f);
```

(5) 공통 결합도

공통 결합도는 클래스/오브젝트 간 호출에서 전역 변수가 매개 변수로 사용되는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 공통 결합도의 예시를 보여준다. PI 변수가 AreaCalc 클래스의 여러 메소드에서 호출되어 사용된다.

```
class AreaCalc {
    float circleArea(float r){
        return r*r*PI.get();
    }
    float shpereArea(float r){
        return 4*r*r*PI.get();
    }
}
class PI {
    private static float PI = 3.14;
    float get() {
        return PI;
    }
}
```

(6) 내용 결합도

내용 결합도는 클래스/객체가 다른 객체의 내부 속성을 직접적으로 변경하는 경우를 측정한다. 아래의 소스코드는 객체지향 언어에서 내용 결합도의 예시를 보여준다. Calc 클래스가 Circle 클래스의 변수를 직접적으로 변경한다.

```
class Circle {
    public float PI = 3.14;
}
class Calc {
    public void setPI(){
        Circle c = new Circle();
        c.PI = 3.1415;
    }
}
```

3.2 소프트웨어의 품질지표 측정을 위한 코드 가시화 개발

3.2.1 툴체인 개발 환경

그림 2는 품질지표 측정을 위한 코드 가시화(툴체인)이다. 툴체인은 Parser/Semantic Analyzer, Information Base, View Composers로 구성된다[15]. C/C++ 또는 Java로 된 소스코드가 Parser에 입력된다. Parser는 소스코드의 정보(클래스/메소드들의 이름, 타입, 연관관계 등)를 분석 및 추출하여 Information Base에 저장한다. View Composers는 Information Base에 저장된 정보를 불러와서 그래프 형태로 시각화한다. New Views of

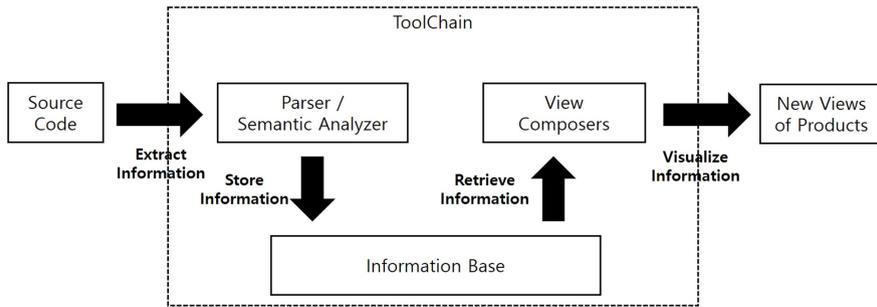


그림 2 품질지표 측정을 위한 코드 가시화(툴체인)

Fig. 2 Code Visualization(Tool-Chain) for Measuring Quality Indicators

Product는 외부 사용자가 조회할 수 있는 그래프 형태의 데이터이다. 툴체인은 기존 상용 툴을 가져와 사용하거나, 새롭게 개발될 수 있다. 툴체인의 각 요소별 특징은 다음과 같다.

(1) Source Navigator

소스 네비게이터는 오픈 소스 정적 분석기로서 분석하고 싶은 소스코드 프로젝트를 SNDB파일들로 출력해준다[16]. 즉, 소스코드를 분석하여 메소드의 리턴 타입, 클래스 이름, 메소드 이름, 메소드 간 호출 관계 등의 다양한 정보를 추출한다. 각 SNDB 파일들은 hash와 btree 형태로 binary 코드로 작성되어 있다. 그렇기 때문에 파일 그대로 툴체인에서 사용하기에 불편함이 존재한다. 하지만 C/C++, Java를 모두 지원하기 때문에 툴체인의 활용범위가 넓어진다.

(2) SQLite

SQLite는 MySQL이나 PostgreSQL과 같은 데이터베이스 관리 시스템이다[17]. 하지만 다른 데이터베이스에 비해서 비교적 가벼운 데이터베이스이므로 서버가 아닌 응용 프로그램에 넣어 사용하기 편리하다. 일반적인 RDBMS에 비해 가볍기 때문에 중소 규모라면 속도에 손색이 없지만 대규모 작업에는 적합하지 않다. 또 API는 단순히 라이브러리를 호출하는 것만 있으며, 데이터를 저장하는데 하나의 파일만을 사용하는 특징이 있다.

(3) Graphviz

Graphviz는 DOT 언어 스크립트로 지정된 그래프를 그리기 위해 1991년 AT&T Labs Research에서 시작한 오픈 소스 도구 패키지이다[18]. 또한 소프트웨어 응용 프로그램에 도구를 사용하기 위한 라이브러리도 제공한다. 리눅스, MacOS, 윈도우 등 다양한 운영체제를 지원하며 Eclipse Public License에서 라이선스가 등록된 무료 소프트웨어이다. Graphviz를 사용하는 다양한 어플리케이션이 존재한다. Graphviz는 DOT 언어로 명명된 그래프 기술 언어와 DOT 파일을 생성 및 처리할 수 있는 도구 집합으로 구성된다.

3.2.2 툴체인 수행 프로세스

육군 정보체계 소프트웨어는 전자정부 프레임워크를 기반으로 개발되어야 한다. 이를 위해, 본 연구는 전자정부 프레임워크에서도 실행 가능한 툴체인을 개발하였다. 툴체인 수행 프로세스는 다음 과정을 따른다.

(1) Annotation 제거

전자정부 프레임워크 소스코드는 Spring 프레임워크를 기반으로 작성되어 있다. 즉, @ (Annotation) 마크로 코드가 작성되어 있다. Annotation이란 주석이라는 사전적 의미가 있으며 컴파일 혹은 런타임에 해석된다. 메타데이터로서 실제 데이터가 아닌 Data를 위한 데이터라고도 불리며 JDK5부터 등장한다. 하지만 소스 네비게이터가 분석하지 못한다는 문제가 있다. 그로 인해 Annotation이 달린 부분은 분석에 오류가 생긴다. 또한, Annotation을 지운다고 해도 런타임에 해석되기 때문에 정적 분석에는 문제가 없다. 따라서 이러한 Annotation을 삭제하여 소스 네비게이터가 소스코드의 분석을 용이하게 만들어야 한다.

먼저 오픈 소스인 JavaParser를 이용하여 소스코드를 AST구조로 분석한다. JavaParser는 Java 코드를 AST로 파싱해주는 정적 분석도구이다[19]. API를 제공하기 때문에 API를 이용하여 AST를 조작할 수 있다. Annotation이 존재하는 노드를 삭제함으로써 전체 코드에서 Annotation을 삭제하여 수정할 수 있다.

(2) 분석된 정보를 데이터베이스에 저장

소스 네비게이터에서 추출된 소스코드의 정보가 담긴 SNDB파일들을 읽어 들인다. 바이너리 코드로 작성되어 있기 때문에 Script를 제작할 때 사용이 용이하도록 데이터베이스를 재구성하여 저장한다. 먼저 Extract.jar파일을 실행하여 소스 네비게이터의 설치 경로와 덤파일이 있는 경로를 입력한다. 입력이 완료되면 각 SNDB 파일들이 생성 완료되고 ExtractInfo.jar 파일을 실행한 경로에 recoveryDB.db 파일이 생성된다. recoveryDB.db 파일은 SNDB의 정보들을 SQLite를 이용하여 저장해놓

은 데이터베이스이다. 소스 네비게이터에서 분석을 실패 하였거나 데이터베이스 생성이 실패하였다면 생성된 파일의 크기가 0byte일 것이다. 파일이 생성되었다고 무조건 성공으로 판단할 수 없다.

(3) 데이터베이스의 정보로 DotScript 제작

```

digraph xx {
  node[shape=record, style=filled, fillcolor=lightyellow]:
  VisualGraph [label=<<{<FONT FACE="boldfontname">
  "VisualGraph</FONT>}>|+nodeData: ArrayList<String[]>
  <BR/>+edgeData: ArrayList<String><BR/>+wordStack:
  Stack<String><BR/>+countStack: Stack<HashMap<String,
  Integer>><BR/>+bodyStack: Stack<HashMap<Integer,
  Boolean>><BR/>+statementCount: int<BR/>+statementCheck:
  boolean|+checkStmnt(Statement: Statement, int: int):
  String[]<BR/>+makeHashMap():HashMap<String, Integer>
  r<<BR/>>]
  ...
  "BadSmellXML"->"BadSmellDB" [style=dotted,
  arrowhead=vee];
  "BadSmellXML"->"DrawGraph" [style=dotted,
  arrowhead=vee];
  "Main"->"ClassDiagramXML" [style=dotted,
  arrowhead=vee];
  "TestMain"->"GetDeclaration" [style=dotted,
  arrowhead=vee];}
  dotting
  C:\SWV\toolchain\badsmell\ClassDiagram_2018_11_29
  -23_54_31.svg
  done

```

다음은 생성된 DotScript의 예제이다. 데이터베이스에 저장된 소스코드의 분석된 정보는 클래스 다이어그램과 결합도 다이어그램을 생성할 수 있다.

클래스 다이어그램은 각 클래스들 간의 연관관계를 나타낸 다이어그램이다. 클래스 이름, 인스턴스 변수, 메서드의 정보로 이루어진 테이블의 다이어그램이다. 각 테이블들은 Association, Dependency, Generalization으로 연결되어 있다. Association은 한 모듈에서 다른 모듈의 인스턴스 변수를 호출하는 경우이다. 짝 찬 화살표에 점선으로 노드를 연결한다. Dependency는 한 모듈에서 다른 모듈의 메서드를 호출하는 경우이다. 짝 찬 화살표에 실선으로 노드를 연결한다. Generalization은 모듈 간의 상속 관계를 나타낸다. 빈 화살표에 실선으로 노드를 연결한다.

결합도 다이어그램은 자료 결합도부터 내용 결합도까지의 점수를 그래프에 표시한다. 각 노드들은 클래스의 이름이며 클래스 간의 결합도를 그래프에서 확인할 수 있다. 각 결합도는 가중치가 적용되어 있어 하나의 결합도일지라도 종류에 따라 값의 차이가 커진다. 1개의 내용 결합도가 있다면 10.2개의 자료 결합도와 같은 값으

로 판단하고 중요도를 증가시킨다. 다음은 각 결합도 가중치의 예이다.

- Stamp(SP) = 2 \* Data(DT)
- Control(CR) = 3 \* Data
- External(ET) = 4.8 \* Data
- Common(CM) = 7.5 \* Data
- Content(CT) = 10.2 \* Data

3.3 설계 문서 자동화 기술

3.3.1 PAL(Process Asset Library)

PAL(Process Asset Library)은 프로젝트 수행 도중에 업무 프로세스에서 나오는 각종 문서들을 유지보수 및 관리하는 역할을 수행하여 문서 관리를 좀 더 쉽고 편리하게 할 수 있도록 지원한다[20].

PAL에는 프로세스, 프로시저 서술내용, 정책, 표준, 지침, 문서 템플릿, 체크리스트, 트레이닝 코스, 그리고 툴 등의 자산이 포함되며, 자산들은 소프트웨어를 개발하고 테스트하거나 또는 유지보수, 소프트웨어 산출물과 프로세스를 보증, 소프트웨어 프로젝트를 관리하는 모든 사람들에 의해 쉽게 접근될 수 있다. 자산은 끊임없이 추가되고 수정될 수 있다. PAL의 자산은 다음과 같이 4가지 범주로 분류된다. ① 프로젝트 관리(Project Management)는 프로젝트를 수행함에 있어 스테이크 홀더들이 필요한 정보나 자원을 위해 프로젝트 활동으로부터 발생하는 지식, 기술, 노하우, 그리고 기술에 대한 애플리케이션이 프로젝트 관리 범주로 분류된다. ② 프로덕트 개발(Product Development)은 고객의 요구사항을 제품화(소프트웨어, 하드웨어, 서비스 등)하는 과정에서 산출되는 자산들이 여기에 해당된다. ③ 구조적 지원(Organizational Support)은 프로젝트 관리, 프로덕트 개발, 그리고 획득의 생명주기에서 사용되는 프로세스 자산이 해당된다. ④ 획득(Acquisition)은 공급자로부터 소프트웨어 시스템을 획득하는 것에 관한 모든 자산이 해당되며, 이러한 자산은 입찰요청서와 요구사항 템플릿 등을 포함한다. 그림 3은 PAL의 범주이다.

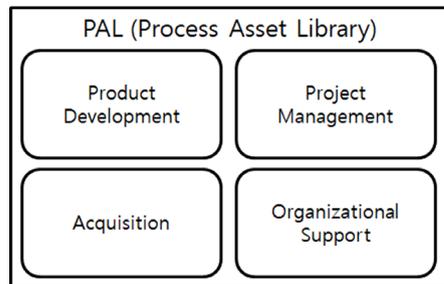


그림 3 PAL의 범주 Fig. 3 PAL Category

3.3.2 PAL 기반의 문서 생성 자동화 방법

그림 4는 PAL 기반의 문서 생성 자동화 방법이다[21]. Data Preprocessor는 데이터베이스에 저장된 프로젝트 데이터 리스트에서 특정 정보를 추출하여 XML 파일로 만든다. PAL에 XML 언어를 HTML 언어로 자동 변환하기 위한 XSLT 기반의 변환틀을 저장한다. XML 파일을 PAL에 입력하여 HTML 파일로 자동 변환한다. 결과적으로 모든 정보는 HTML 기반의 문서로 자동 생성될 수 있다.

그림 5는 XSLT를 이용하여 XML 데이터를 HTML 데이터로 자동 변환된 결과이다[22]. 프로젝트 관리 데

이터를 XML 파일로 저장한다. XML 파일을 HTML로 변환하는 룰을 작성하여 XSLT 파일로 저장한다. XML 파일의 헤더정보에 XSLT 파일의 경로를 저장한다. 웹 브라우저(인터넷 익스플로러, 크롬 등)로 XML 파일을 조회하면 XML 파일의 데이터가 HTML 형태로 자동 변환되어 출력된다.

3.4 육군 정보체계 소프트웨어 프로세스 가시화

육군 정보체계 소프트웨어 품질 가시화 환경은 웹 기반의 서비스를 제공하고, Tools, Server, Dev, Toolchain, Webapps, src의 6개의 분류로 나뉜다[23]. 각 분류에 해당되는 도구들은 언제나 추가, 수정, 삭제가 될 수 있

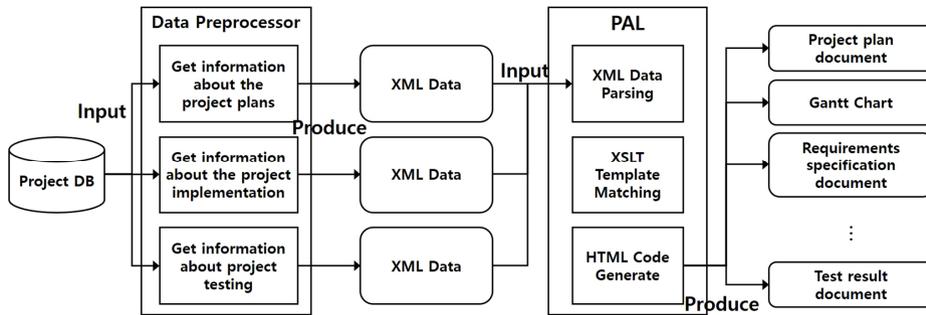


그림 4 PAL 기반의 설계 문서 생성 자동화 방법

Fig. 4 The Method for Automated Generation of the PAL based Design Document

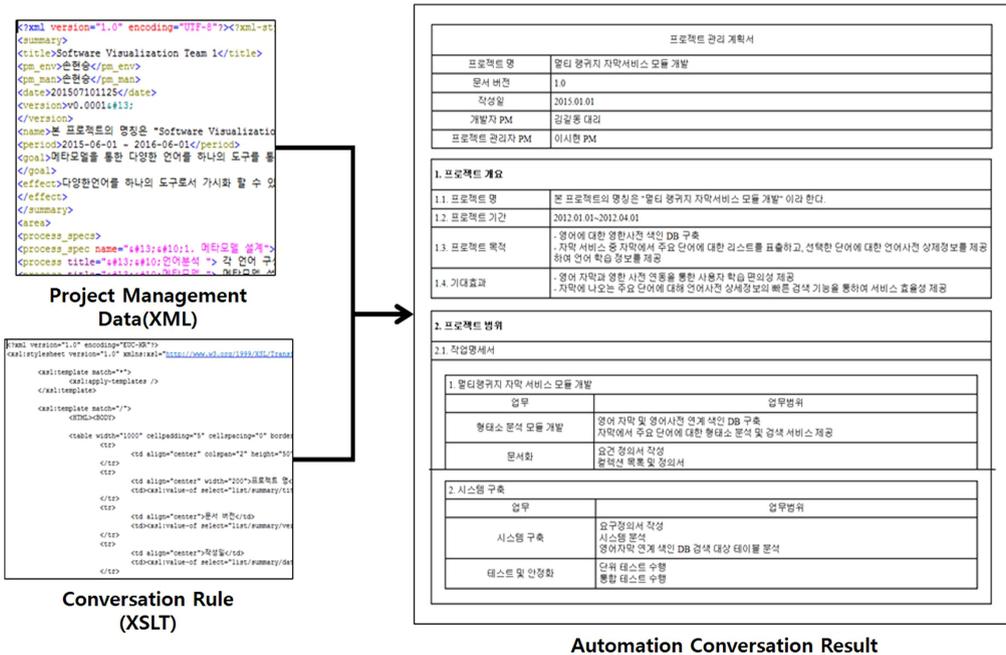


그림 5 자동 변환 결과

Fig. 5 Automation Conversion Result

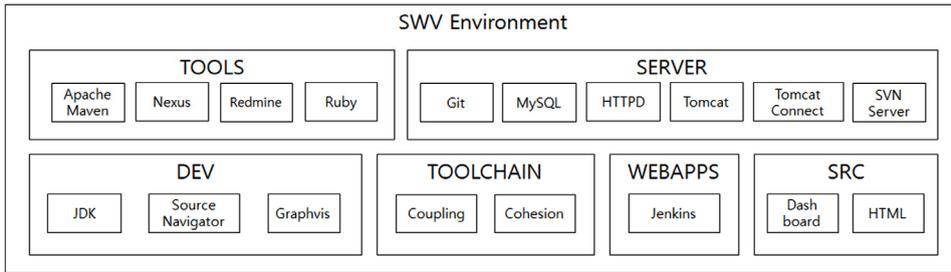


그림 6 소프트웨어 가시화 환경 프레임워크  
Fig. 6 SW Visualization Environment Framework

는 Plug & Play 형태의 특징을 가진다. 가시화 환경 설치자는 자신의 업무 환경에 맞는 다양한 도구를 분류에 맞게 삽입하거나 삭제할 수 있다. 그림 6은 소프트웨어 품질 가시화 환경 프레임워크이다.

Tools는 프로젝트 관리 도구나 지속적 통합 도구들이 포함된다. 도구 실행을 위한 필수 유틸리티 또한 포함된다. Server는 웹 서버 구동 및 프레임워크 전체 환경 구성에 필수적인 도구가 포함된다. Dev는 소프트웨어 품질 가시화 환경에서 유용하게 사용되는 도구들이 포함된다. Toolchain은 소프트웨어 품질 측정을 위한 틀체인들이 포함된다. Webapps는 Tomcat 관련 도구가 포함된다. src는 품질 점수 표시를 위한 대시보드, 그리고 기타 웹페이지(HTML, JSP)가 포함된다.

육군 정보체계 소프트웨어 품질 가시화 환경 프레임워크의 구축을 위해 도구들을 설치한다면, 웹 환경 구축, 프로젝트 관리도구 설치, 유틸리티 설치, 품질 측정 도구 순으로 설치가 되어야 정상적인 설치 및 운용이 가능하다.

웹 환경 구축에서 HTTPD는 Apache에서 제공하는 웹 서버로, 웹 브라우저를 통해 웹서버에서 제공되는 HTML 등의 파일에 접근할 수 있다. JDK는 Java Development Kit으로써, 자바 프로그램을 구동하고 개발하기 위한 환경을 지원한다. 또한 JDK는 Tomcat 웹서버를 구축하기 위해 반드시 필요하다. Tomcat은 자바 기반의 웹서버이며, 웹 브라우저를 통해 웹서버에서 제공되는 JSP, HTML 등의 파일에 접근할 수 있다. Tomcat Connect는 Tomcat 서버와 HTTPD 서버를 연동하기 위해 필요하다.

프로젝트 관리 도구 설치에서, Jenkins는 지속적인 통합 도구(Continuous Integration)로써, 공개 소프트웨어로 제공되며, Tomcat 환경에서 실행된다. MySQL은 데이터베이스 관리 시스템으로, 데이터베이스를 관리할 수 있다. Redmine은 공개 소프트웨어로 제공되는 프로젝트 관리 도구로써, 기본 데이터베이스로 MySQL을 사용한다. Ruby는 Redmine 실행을 위해 필요하다.

유틸리티 설치 과정에서, SVN Server는 프로젝트의

소스코드 관리 기능을 제공하는 Subversion 서버이다. Source Navigator는 가시화 틀체인 내에서 소스코드의 구조를 추출하는데 사용되는 공개 소프트웨어 도구로, C/C++에 최적화되어 있다. Graphvis는 가시화 틀체인 내에서 데이터베이스에 저장된 소스코드 구조를 그래프화하는데 사용되는 공개 소프트웨어 도구이다. Git은 프로젝트 소스코드 관리 기능을 제공하는 Git 서버이다. Apache Maven은 자바용 공개 소프트웨어 프로젝트 관리 도구이며, Nexus는 프로젝트 소스코드의 저장소 기능을 제공한다.

품질 측정 과정에서, Coupling은 소프트웨어 소스코드의 결합도 품질 점수를 측정할 수 있는 틀체인이다. Cohesion은 소스코드의 응집도 품질 점수를 측정할 수 있는 틀체인이다. 마지막으로 Dashboard는 측정된 품질지표 점수를 표시하고, 품질지표 그래프를 보여준다. 그림 7은 육군 정보체계 소프트웨어 품질 가시화 환경 구축 과정이다.

#### 4. 사례연구

육군 정보체계 소프트웨어 개발 관리 프로세스 개선 사례 적용 결과는 보안상의 이유로 외부 공개가 제한되어있다. 본 논문에서는 이와 유사한 형식의 그림으로 대체한다.

##### 4.1 개발환경

개발 플랫폼은 전자정부 표준 프레임워크 3.2를 사용한다. 전자정부 표준 프레임워크는 디자인 패턴 및 소스코드의 재사용이 원활한 ‘Spring’기반의 표준 프레임워크이다[24]. 정보시스템 개발을 위해 필요한 기능 및 아키텍처를 미리 만들어 제공함으로써 효율적인 어플리케이션 구축을 지원한다. 또한 공공사업에 적용되는 개발 프레임워크의 표준 정립으로 응용 소프트웨어 표준화, 품질 및 재사용성 향상을 목표로 한다.

개발언어는 Java, JSP, HTML, CSS, Javascript를 사용한다. Java는 전자정부 표준 프레임워크 내에서 사용되는 객체지향 언어로, 서버 단에서 처리하는 기능을 구현한다. 컴퓨터 네트워크 접근 기능 탑재와 같은 프로그램

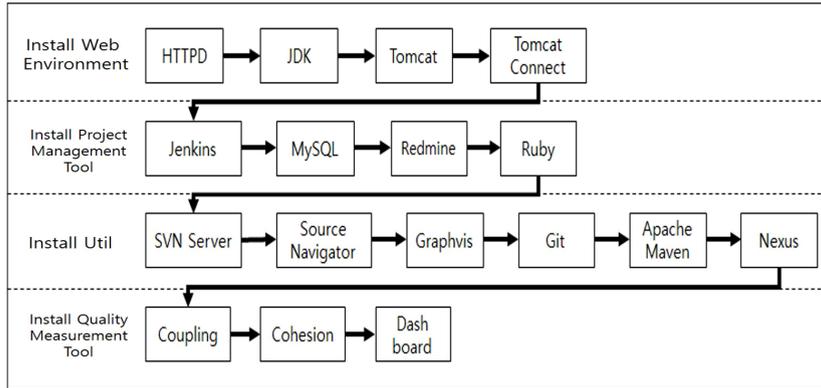


그림 7 육군 정보체계 소프트웨어 가시화 환경 구축 과정  
Fig. 7 Army Information System SW Visualization Environment

이 여러 운영체제에서 실행될 수 있다. JSP는 Java를 이 용한 서버 사이드 스크립트 언어로 안정적이고, 유지보수 가 쉽다. HTML, CSS, Javascript는 하나의 웹페이지를 구현하기 위해 사용되는 언어이다. 전자정부 표준 프레임 워크 내에서 사용되는 스크립트 언어로, 웹 홈페이지를 구축하여 사용자에게 시각적으로 보여줄 수 있도록 한다.

개발도구는 이클립스, Maven, MySQL, 레드마인, 젠 킨스, Git을 사용한다. 이클립스는 공개 소프트웨어 개발 툴로 운영체제의 제한 없이 사용 가능하다. Maven은 자 바용 프로젝트 관리 도구이며, 아파치 라이선스로 배포 되는 오픈 소스 소프트웨어이다. MySQL은 세계에서 가 장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템으로, 다중 스레드, 다중 사용자 형식의 구조질의 어 형식의 데이터베이스 관리 시스템이다. Git은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시 스템이다. 레드마인은 오픈소스 프로그램으로 웹 기반의 프로젝트 관리와 버그 추적 기능을 제공하는 도구이다. 레드마인에 등록되어 있는 개발업무 정보를 관리할 수 있도록 한다. 젠킨스는 소프트웨어 개발 시 지속적 통합 서비스를 제공하는 툴이다. 다수의 개발자들이 하나의 프로그램을 개발할 때 버전 충돌을 방지하기 위해 각자 작업한 내용을 공유 영역에 있는 저장소에 빈번히 업로 드 함으로써 지속적 통합이 가능하도록 해준다.

4.2 오픈소스 기반 버전 개선

기존의 윈도우 기반 시스템은 레드마인과의 연결이 가능하였다. 하지만 CentOS는 레드마인과의 연결에 문 제가 있어, 레드마인의 기능을 오픈소스 버전으로 개선 하였다. 또한 오픈소스로의 확장을 위해, 오라클/큐브리 드/MySQL 데이터베이스를 레드마인에 연결할 수 있도 록 개선하였다.

4.2.1 레드마인 기능 개선

레드마인의 ‘일감’은 검색조건에 입력된 상태와 유형 을 데이터베이스에 저장하여 사용자가 검색한 조건을 계속 유지한다[25]. 그러나 검색한 조건의 유형을 요구 사항이라고 했을 때, ‘새일감만들기’를 하면 유형이 ‘요 구사항’으로 유지되는 것이 아니라 항상 첫 번째 유형 값으로 생성된다. 이 기능을 사용하기 위해서는 다음과 같은 절차로 수정한다.

1) redmine/app/views/issues/index.html.erb 수정

데이터베이스에 저장된 tracker\_id가 null이면 일반 링크를 출력하고 tracker\_id 값이 있을 때는 이 tracker\_id 를 인자로 넘기는 링크를 출력하도록 해 일감의 유형을 계속 유지할 수 있도록 한다.

```

<% unless @query.filters['tracker_id'].nil? or @quer
y.filters['tracker_id'][:values].nil? %>
  <%= link_to I(:label_issue_new), _new_project_is
sue_path(@project, :issue => { :tracker_id => @quer
y.filters['tracker_id'][:values].first } ), :class => 'icon
icon-add new-issue' %>
<% else %>
  <%= link_to I(:label_issue_new), _new_project_is
sue_path(@project), :class => 'icon icon-add new-is
sue' %>
<% end %>
  
```

2) 레드마인 확인

정상적으로 수정되었는지 확인하기 위해 검색조건을 ‘요구사항’으로 선택한다. 새 일감만들기 시 유형이 자동 으로 요구사항이 선택된다. 그림 8은 레드마인의 기능 개선결과이다.

4.2.2 레드마인과 오라클 연동

레드마인은 공식적으로 오라클 데이터베이스를 지원 하지 않는다. 그렇기 때문에 단지 Ruby-oci8 드라이버



그림 8 레드마인의 기능 개선 결과  
 Fig. 8 The Result of Redmine's Functional Improvement

와 oracle\_enhanced-adapter를 설치해도 오라클 데이터베이스는 동작하지 않는다. 그것은 오라클 데이터베이스가 NULL, 테이블 길이, 간트차트 등의 문제가 있기 때문이다. 본 논문에서는 레드마인에 오라클을 동작하게 하기 위해서 직접 레드마인 코드를 변경하였다.

1) NULL 문제 해결

NULL 문제는 오라클에서 “”와 같이 문자열 값이 없는 경우를 NULL로 인식하기 때문에 발생한다. 데이터베이스 쿼리에서 NULL 값이 허용되지 않은 상태에서 NULL로 인식하기 때문에 SQL 에러가 발생한다. 해결 방법은 아래와 같이 redmine/db/migrate/ 폴더 내에 모든 파일을 null을 허용할 수 있도록 변경한다.

```
수정전
:default => "", :null => false
or
:default => ""
```



```
수정후
:default => "", :null => true
```

2) 테이블 길이 문제

오라클의 테이블 길이는 30으로 제한되었다. 그러나 레드마인에서 생성하는 몇 개의 테이블들이 30을 초과하기 때문에 이 테이블 명들을 아래와 같이 30이하의 값으로 변경한다.

```
수정전
open_id_authentication_associations
```



```
수정후
open_id_auth_associations
```

3) 간트차트 문제

간트차트 문제는 단순히 일감을 생성하였을 때는 문제가 없다가 특정한 일감에 종료 날짜를 입력하면 페이지가 로드되지 않는 현상이 발생한다. 이 문제는 오라클이 사용하는 날짜표시 방법이 레드마인과 달라서 발생한다. 이를 해결하기 위해서는 다음과 같이 redmine/config/initializers/oracle.rb 파일을 추가한다. self.emulate\_

dates\_by\_column\_name의 값이 true로 설정되어 있어야 한다.

```
ENV['TZ'] = 'UTC'
ActiveSupport.on_load(:active_record) do
  ActiveRecord::ConnectionAdapters::OracleEnhanced
  Adapter.class_eval do
    self.emulate_dates_by_column_name = true
  end
end
```

레드마인과 오라클 연동 시 발생하는 문제를 해결한 결과, 아래의 명령어를 통해 오라클 데이터베이스에 마이그레이션 명령어로 데이터를 입력할 수 있다. 그림 9는 정상적으로 실행되는 결과이다.

```
rake db:migrate
rake redmine:load_default_data
```

```
[root@localhost redmine-3.4.6]# RAILS_ENV=development bundle exec rake redmine:load_default_data
/usr/local/lib/ruby/gems/1.9.1/gems/activerecord-4.2.8/lib/active_record/connection_adapters/abstract_adapter.rb:1694: warning: already initialized constant HABTM_Roles

Select language: ar, az, bg, bs, ca, cs, da, de, el, en, en-GB, es, es-PA, et, e
u, fa, fi, fr, gl, he, hr, hu, id, it, ja, ko, lt, lv, mk, mn, nl, no, pl, pt, p
t-BR, ro, ru, sk, sl, sq, sr, sr-YU, sv, th, tr, uk, vi, zh, zh-TW [en] ko
=====
Default configuration data loaded.
[root@localhost redmine-3.4.6]# ./run.sh
=> Booting WEBrick
=> Rails 4.2.8 application starting in development on http://localhost:3000
=> Run 'rails server -h' for more startup options
=> Ctrl-C to shutdown server
=> Debugger enabled
[2018-07-17 07:07:29] INFO WEBrick 1.3.1
[2018-07-17 07:07:29] INFO ruby 1.9.3 (2014-11-13) [x86_64-linux]
[2018-07-17 07:07:29] INFO WEBrick::HTTPServer#start: pid=4226 port=3000
```

그림 9 오라클 마이그레이션의 출력 결과  
 Fig. 9 The Output Result of Oracle Migration

4.3 코드 가시화

코드 가시화는 진행 중인 개발업무의 가시화 관련 정보를 확인할 수 있다. 각 개발업무 리스트에 프로젝트 별로 Class, Method, Member 등이 적용되어 있으며, 가시화 링크 클릭 시 해당 항목에 맞는 상세정보를 확인할 수 있다.

코드 가시화 리스트에서 '가시화' 버튼을 클릭하면, 해당 프로젝트의 소스코드에 관한 가시화 내용을 확인할 수 있다. 프로젝트 디렉토리 구조와 화면 내 결합도, 클

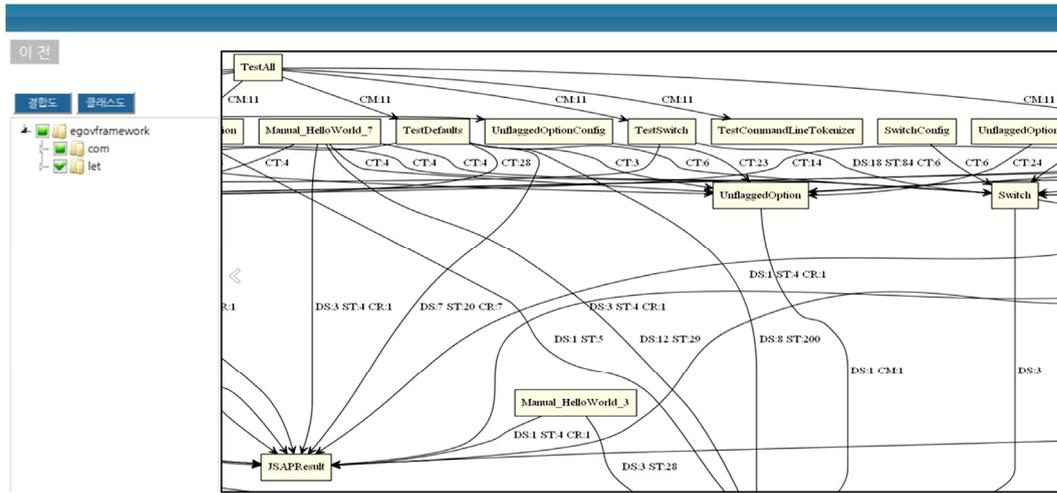


그림 10 코드 가시화 출력 화면  
Fig. 10 Code Visualization Output Screen

래스 버튼을 클릭하여, 해당 프로젝트 내 결합도와, 각 클래스 간의 관계 등을 확인할 수 있다. 그림 10은 코드 가시화 출력화면이다.

표 3 XML 소스코드의 예  
Table 3 XML Example

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="requirement_spec.xsl"?>
<list doc_name="요구사항 정의서" period="2018.01.01~2018.11.29." project_name="Project1" version="1.0">
  <modification_history>
    <history date="-" version="-" writer="-"/>
  </modification_history>
  <outlines>
    <outline title="# 주제">객체지향 코드의 요구사항 검증을 위한 유스케이스 추출 방안</outline>
    <outline title="# 품질특성">요구사항 추적성 : 개발공정별로 요구사항의 일관성 유지여부를 확인
      코딩 표준 준수율 : SW개발에 적용하는 코딩 표준에 대한 준수율 여부를 확인</outline>
  </outlines>
  <requirements>
    <requirement>
      <id>REQ-F-001</id>
      <name>실행 옵션 설정</name>
      <detail>본 도구는 아래의 세 가지 옵션을 선택하여 실행할 수 있어야 한다.
        1. ALL : 소스 분석에서 가시화까지 모든 단계를 수행
        2. DOT : 파싱 혹은 분석 과정을 수행하지 않고, 이미 존재하는 DB 파일로부터 그래프를 추출
        3. DUMP : SNDB 파일을 덤프한 결과를 파일로 추출</detail>
      <priority>2</priority>
      <sign/>
    </requirement>
    ...
  </requirements>
</list>
```

#### 4.4 설계 문서 자동화

본 절은 설계 문서 자동화 개발 결과로 요구사항 정의서의 예를 제시한다. 요구사항 정의서를 자동 생성하기 위해 XML 파일과 XSLT 파일을 만들어야 한다. 표 3은 요구사항 정의서 정보가 저장된 XML 파일의 소스코드이다.

이를 기반으로 요구사항 정의서 변환률이 저장된 XSLT 파일을 생성한다. 그림 11은 웹 브라우저를 통해 XML 파일을 실행한 결과이다.

프로젝트명	Project1	프로젝트기간	2018.01.01~2018.11.29
문서명	요구사항 정의서	버전	1.0

### Project Overview

일지	변경	변경내역	작성지
-	-	-	-

#### 1. Overview

- # 주제  
객체지향 코드의 요구사항 검증을 위한 유스케이스 추출 방안
- # 품질특성  
요구사항 추적성 : 개발공정별로 요구사항의 일관성 유지여부를 확인  
코딩 표준 준수율 : SW개발에 적용하는 코딩 표준에 대한 준수율 여부를 확인

#### 2. Functional Requirements

요구사항 ID	요구사항 명	상세 요구사항	우선순위	확인
REQ-F-001	실행 옵션 설정	본 도구는 아래의 세 가지 옵션을 선택하여 실행할 수 있어야 한다. 1.ALL : 소스 분석에서 가시화까지 모든 단계를 수행 2.DOT : 파싱 혹은 분석 과정을 수행하지 않고, 이미 존재하는 DB 파일로부터 그래프를 추출 3.DUMP : SNDB 파일을 덤프한 결과를 파일로 추출	2	
REQ-F-002	결합도 검증관리 관리	결합도 검증을 위한 데이터베이스 유지는 객체지향 관리한다. 이는 추후 관리 확장을 용이하기 위함이다.	2	
REQ-F-003	코드의 도구에 대한 특질성	각각의 오픈 소스 도구들은 추후 다른 도구를 추가할 사용하거나 대체할 가능성이 존재한다. 그러므로 구현 단계에서 OCP를 준수하도록 코드를 작성해야한다.	3	
REQ-F-004	공통 그래프 모델	본 도구는 결합도 그래프 및 클래스 다이어그램 등의 각종 그래프를 출력한다. 추후 출력된 그래프의 종류와 형태가 변경될 수 있으며, 이때 코드의 수정을 최소한으로 줄이기 위한 추상적인 공통 그래프 모델을 정의하고자 한다. 이 그래프 모델은 노드(Node), 엣지(Edge)의 기본적인 요소로 이루어져 있으며, 본 도우에서 출력하는 모든 그래프는 이 모델을 확장하여 사용한다.	3	

그림 11 요구사항 정의서 예시  
Fig. 11 Example of Requirement Description

4.5 육군 정보체계 소프트웨어 프로세스 가시화

육군 정보체계 소프트웨어 프로세스 가시화 개발 결과는 다음과 같다. 소프트웨어개발관리시스템은 소프트웨어 가시화 체계에서 대시보드 역할을 수행한다. 기존의 대시보드에 육군의 업무를 수행할 수 있도록 개발관리 기능을 추가하였다. 소프트웨어개발관리시스템은 개발진행관리, 개발진행현황, 시스템관리 등으로 구성된다.

개발진행관리는 레드마인에 등록되어 있는 프로젝트의 진행내용을 관리할 수 있다. 개발진행관리는 개발업무현황, 개발상세정보관리, 아키텍처 가시화 등으로 구성된다.

개발업무현황은 레드마인에서 생성된 프로젝트를 가져와 개발업무명으로 표시한다. 또한 프로젝트의 시작과 종료날짜를 출력해 사용자에게 보여준다. 사용자가 리스트 중 하나의 프로젝트를 클릭하면, 레드마인 팝업창에 결

SW Development Status

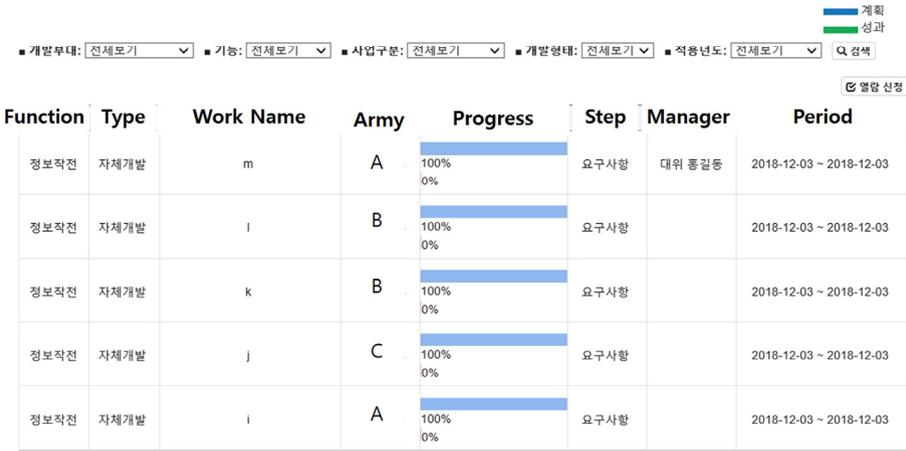


그림 12 소프트웨어 개발현황 화면  
Fig. 12 SW Development Status UI

1) Project Requirement Information

2) Project Gantt Chart

3) Achievement Degree of Requirements

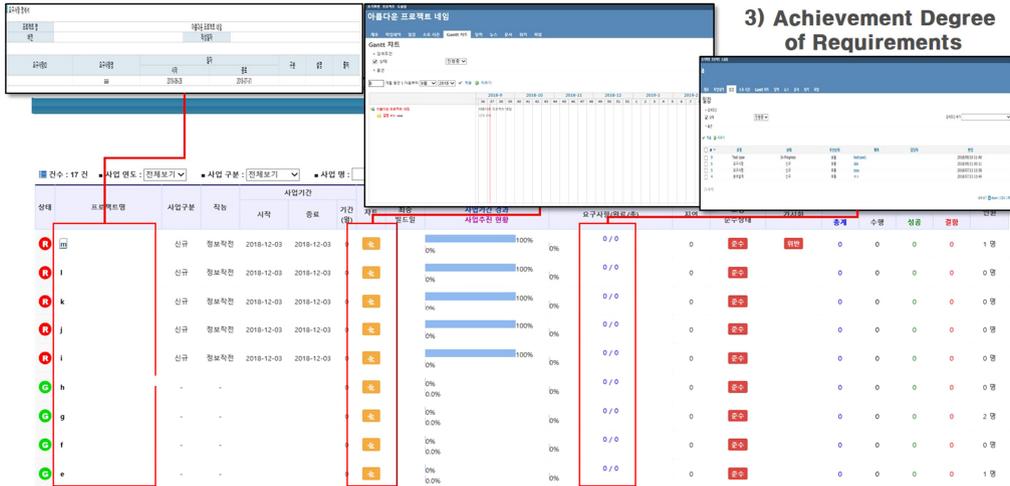


그림 13 소프트웨어개발진행현황 화면  
Fig. 13 SW Development Progress UI

과가 출력되어 해당 프로젝트의 정보를 확인할 수 있다.

소프트웨어개발현황은 현재까지 진행된 소프트웨어개발 진도와 현 단계를 표시하며, 사용자는 열람 신청을 통해 소프트웨어개발현황을 확인할 수 있다. 소프트웨어개발현황은 그림 12는 소프트웨어개발현황의 예이다.

개발진행현황은 현재 진행 중인 개발업무정보를 리스트 형식으로 보여주며, 각 개발업무에 대한 요구사항명세서, 간트차트, 요구사항 달성도, 코딩준수상태, 가시화위반 등 상세정보를 확인할 수 있다. 소프트웨어개발현황은 데이터베이스에 레드마인 프로젝트 리스트를 요청하면 요청받은 리스트 항목마다 요구사항 정보를 View 테이블을 참조하여 반환받는다. 이후에 프로젝트 리스트의 정보를 사용자에게 보여준다. 마지막으로 시스템관리자는 소프트웨어개발관리시스템 내의 설정 역할을 한다. 그림 13은 소프트웨어개발진행현황의 예이다.

## 5. 결론

소프트웨어 가시화는 인력과 비용을 최소화하면서 소프트웨어의 품질 관리를 수행한다. 소프트웨어 가시화는 일반 사용자에게 친숙하며, 응용 프로그램과의 호환성 등 많은 장점이 있기 때문에 대다수 윈도우 기반으로 설치되고 관리된다. 이러한 이유로 육군 정보체계 소프트웨어에 도입되었다. 하지만 윈도우는 구입비용, 잦은 버전 업데이트 등의 문제로 적극적인 도입 및 확산이 어렵다. 이러한 문제를 해결하기 위해, 본 논문에서는 육군 정보체계 소프트웨어의 품질 개선을 위한 개방형 OS기반의 소프트웨어 가시화 방법을 제안하였다. 독립적인 소프트웨어 의존도를 줄이기 위해 리눅스 기반으로 독립적이고 확장에 유연한 소프트웨어 가시화 체계를 구축하였다. 코드 가시화는 기존에 발견하지 못한 문제점을 찾아내어 잠재된 오류 문제를 해결할 수 있었다. 실제 문서 자동화 기술은 PAL을 기반으로 개발했으며, 문서 관리를 좀 더 쉽고 편리하게 할 수 있다. 마지막으로 소프트웨어 가시화 자동화 기술은 개발 전 과정의 상태를 확인하고, 유기적 연계를 통해 현재 개발 상태의 원인을 쉽게 파악할 수 있다. 향후 연구로 본 연구의 전체 과정을 확장하여 다양한 분야의 기업 환경에 적용하고자 한다.

## References

- [1] S. E. Lee, S. H. Baek, S. J. Kang, W. M. Song, S. E. Lee, W. S. Jang, D. G. Jung, J. M. Jo, *SW Development Quality Management Manual*, NIPA SW Engineering Center, Seoul, 2013.
- [2] B. K. Park, W. S. Jang, S. Y. Moon, R. Y. C. Kim, and Y. S. Kim, "Advanced Practices on Software Visualization for Improving Code Quality of the Military Defense Software," *Special Interest Group on Defense Software Workshop*, 2019. (in Korean)
- [3] B. K. Park, W. S. Jang, K. D. Kim, and R. Y. C. Kim, "A Simplified Test Maturity Model (sTMM) for Small and Midsize Test Organization," *Journal of KIISE*, Vol. 45, No. 6, pp. 522-532, Jun. 2018. (in Korean)
- [4] B. K. Park, B. K. Jeon, and R. Y. C. Kim, "Improvement Practices in the Performance of a CPS Multiple-Joint Robotics Simulator," *Applied Sciences*, Vol. 10, No. 1, pp. 185-198, Nov. 2019.
- [5] Architecture Explorer Work Item Extension. (2012, Jun 13). [Online]. Available: <https://marketplace.visualstudio.com/items?itemName=vs-publisher-71049.ArchitectureExplorerWorkItemExtension> (download 2020, May. 11)
- [6] Class Visualizer. (2011, Nov 29). [Online]. Available: <http://class-visualizer.net/> (download 2020, May. 12)
- [7] Code\_Swarm. (2014, Jan 13). [Online]. Available: [https://github.com/rictic/code\\_swarm](https://github.com/rictic/code_swarm) (download 2020, May. 12)
- [8] JIVE. (2007, Oct 21). [Online]. Available: <https://www.cse.buffalo.edu/~jive/> (download 2020, May. 11)
- [9] X-Ray. (2008, Jul, 5). [Online]. Available: <http://xray.inf.usi.ch/xray.php#intro> (download 2020, Jun. 2)
- [10] M. Lanza, S. Ducasse, "CodeCrawler - An Extensible and Language Independent 2D and 3D Software Visualization Tool," *Tools for Software Maintenance and Reengineering. RCOST/Software Technology Series*, Franco Angeli, pp. 74-94, 2005.
- [11] NOONBESOFT. (2015, Jun 01). [Online]. Available: <http://www.noonbesoft.com/>
- [12] McCabe T.J., "A Complexity Measure, Software Engineering," *IEEE Transactions on Software Engineering*, Vol. SE-2, No. 4, pp. 308-320, Dec. 1976.
- [13] J. H. Lee, J. H. Park, E. Y. Byun, H. S. Son, C. Y. Seo, and R. Y. C. Kim, "Plug-in Diverse Parsers Within Code Visualization System with Redefining the Coupling and Coheision in the Object-Oriented Paradigm," *KIPS Tr. Software and Data Eng.*, Vol. 6, No. 5, pp. 229-234, May 2017. (in Korean)
- [14] B. K. Park, H. E. Kwon, H. S. Son, Y. S. Kim, S. E. Lee, R. Y. C. Kim, "A Case Study on Improving SW Quality through Software Visualization," *Journal of KIISE*, Vol. 41, No. 11, pp. 935-942, Nov. 2014. (in Korean)
- [15] B. K. Park, S. Y. Moon, C. Y. Seo, and R. Y. C. Kim, "A Constructive Practice of Software Visualization Mechanism for Code Quality Improvement on the Army's Software Resource Management," *Proc. of the KIISE Korea Computer Congress 2017*, Vol. 44, No. 2, pp. 161-163, Dec. 2017.
- [16] Source Navigator. (2014, Apr 5). [Online]. Available: <http://sourcnav.sourceforge.net/> (download Jun 20)
- [17] SQLite/ (2020, Jun 18). [Online]. Available: <https://www.sqlite.org/> (download, Jun 20)

[18] Graphviz, (2016, Dec. 25). [Online] Available: <https://www.graphviz.org/> (download, Jun 22)

[19] JavaParser. [Online] Available: <https://javaparser.org/> (download, Jun 23)

[20] W. S. Jang, J. S. Hwang, D. H. Kim, C. Y. Seo, R. Y. C. Kim, B. H. Park, S. E. Lee, and Y. S. Kim, "Constructing with XML Data and XSLT for Process Asset Library(PAL)," *The 2015 Fall Conference of the KIPS*, Vol. 22, No. 2, pp. 956-958, 2015.

[21] G. H. Kang, H. S. Son, G. S. Yi, R. Y. C. Kim, and S. E. Lee, "Automatic Document Generation for Effective Project Management Plan Specification," *The 2015 Fall Conference of the KIPS*, Vol. 22, No. 2, pp. 959-961, 2015.

[22] W. S. Jang, H. S. Son, R. Y. C. Kim, and J. H. Lee, "A Practice of Software Development Process Visualization for Army Information System Management," *Journal of KIISE*, Vol. 45, No. 9, pp. 904-910, Sep. 2018.

[23] W. S. Jang, H. S. Son, and R. Young Chul Kim, "A Constructive Practice of the Visualized Development Process of Military Human Resource Management for High Quality Software," *Proc. of the KIISE Korea Computer Congress 2017*, Vol. 44, No. 2, pp. 140-142, 2017.

[24] eGovFrame. [Online]. Available: <https://www.egovframe.go.kr/>

[25] Redmine: Project Management Tool, [Online]. Available: <http://www.redmine.org> (download, Jun 25)



박 보 경

2008년 홍익대학교 컴퓨터정보통신(학사). 2012년 홍익대학교 일반대학원 소프트웨어공학(석사). 2012년~현재 홍익대학교 일반대학원 소프트웨어공학 박사과정. 관심분야는 소프트웨어공학, 요구공학, 역공학, 소프트웨어 가시화, 저전력 및 성능 가시화, AI 기반 소프트웨어 개발 방법론



서 채 연

2014년 홍익대학교 소프트웨어공학(학사). 2009년~2014년 유한대학교 경영정보학과 겸임조교수. 2016년~현재 선문대학교 IT교육학부 계약제교수. 2019년~현재 홍익대학교 소프트웨어융합학과 강사. 관심분야는 소프트웨어 개발 방법론, 메타 모델, 비즈니스 프로세스 모델, 디자인 씽킹, 컴퓨테이셔널 씽킹, 교육 프로세스, 코딩 교육, 소프트웨어 교육, 소프트웨어 자동화 도구 개발, 소프트웨어 프로세스 및 가시화, 메타모델 설계 및 모델 변환



김 기 두

2003년 홍익대학교 컴퓨터정보통신(학사). 2005년 홍익대학교 소프트웨어공학(석사). 2014년 홍익대학교 소프트웨어공학(박사). 2015년~현재 한국정보통신기술협회 디지털 인프라팀 팀장. 관심분야는 소프트웨어 테스트, 테스트 프로세스, 테스트 성숙도 모델, GS(Good Software)인증, 고성능 컴퓨팅(HPC), 3D Printing, 항공용 위성항법시스템(GPS) 검증



이 중 훈

1994년 Univ. of Missouri-Columbia 전산공학 박사. 2016년~2018년 육군본부 정보체계관리단장. 2019년~현재 홍익대학교 SW융합학과교수. 관심분야는 소프트웨어공학, 디지털트윈, 모델링&시뮬레이션, 정보보호, 소프트웨어 가시화, 시큐어 코딩, NCW, 빅데이터



김 영 철

2000년 Illinois Institute of Technology (IIT)(공학박사). 2000년~2001년 LG산전 중앙연구소 Embedded system 부장. 2001년~현재 홍익대학교 소프트웨어융합학과 교수. 관심분야는 테스트 성숙도 모델(TMM), 모델 기반 테스트, 소프트웨어 가시화(소프트웨어 프로세스, 문서화, 코드 가시화), 저전력, 성능, 정적분석기 개발, 신재생에너지 통합관리 시스템