

SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증 모델

박정현 박영식* 정효택* 김상훈**

한국전자통신연구원 책임연구원

한국전자통신연구원 실장 *

한국전자통신연구원 PL **

I. 서론

가전제품, 무기, 자동차, AI, 비행기, 로봇, 우주 항공, 게임, 영화, 사물 인터넷, 핸드폰, 바이오, 시뮬레이션, 기상 예측, 빅데이터, 유통, 의료 등 우리의 삶의 모든 분야에서 SW의 중요성은 날로 증대하고 있다. 더욱이 항공기나 무기 그리고 자동차와 같이 인간의 생명이나 안전과 직접 관련된 제품에서의 SW 비중이나 중요성은 계속 증대하고 있다. 이렇듯 지능 정보화시대에 모든 분야에서 SW의 역할과 중요성이 날로 커지고 있는 만큼 SW의 복잡성이나 규모 또한 매우 커지고 있다.

이렇게 복잡하고 규모가 큰 SW 개발을 성공적으로 완료하기 위해서는 개발 방법론이나 개발 환경 및 툴 그리고 개발 과정 등이 모두 잘 준비되고, SW 개발에 참여하는 설계 및 코딩 개발자, 아키텍처, 시험 및 품질담당자 등 프로젝트 매니저를 포함하여 프로젝트 참여자 모두의 능력과 태도가 무엇보다도 중요하다. 실제 SW 개발 프로젝트의 성공률은 미국을 포함하여 대부분 50% 이상을 넘지 못하고 있는 실정이다. 실패 원인으로는 프로젝트 참여자들 간 의사소통의 한계에 따른 요구사항 및 기능에 대한 잘못된 이해로 발생하는 지연 등이 40% 이상을 차지하고, 그 외 PM(Project manager)이나 프로젝트에 참여하는 개발자들의 문제 해결 능력 한계와 사용자 요구사항에 대한 대응체계 미흡 등이 있다. 특히, SW 개발 R&D(Research & Development) 프로젝트 수행에서도 이와 같은 프로젝트 실패 요인은 예외가 아니며 더욱이 TRL 수준에 따라 사용자의 구체적인 요구사항 파악에 대한 어려움은 가중되고 설계 및 구현 등 프로젝트 수행 전 과정에서 발생하는 여러 가지 이슈와 리스크, 그리고 의사소통 문제로 프로젝트 성공률이 높지 않다. 또 SW 개발 R&D 프로젝트의 경우 개발 기술을 성공적으로 완료했다 해도 개발된

* 본 내용은 박정현 책임연구원 (☎ 042-860-5512, jh-park@etri.re.kr)에게 문의하시기 바랍니다.

** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.

기술이 비즈니스로 연계될 확률은 쉽지 않으며 나아가 개발 기술이 성공적인 제품으로 시장에 연착될 확률 또한 높지 않다.

더구나 프로젝트를 통해 개발하는 기술이나 제품에 포함되는 SW 비중은 더욱 커지고 SW 복잡성과 규모 또한 더욱 커지는 상황에서, SW 개발 R&D 프로젝트에서 소스 코드에 기반을 둔 개발 기술이 기업에 이전되어 제품으로 만들어진 후 고객이 사용하고 운영하는 과정에서 결함이나 장애가 발생하였을 경우 재개발에 따른 비용 증대는 물론이고 기술 이전 기업과 제품 사용 고객의 불만 증대, 기술과 제품에 대한 고객 신뢰 저하로 이어지므로, 프로젝트 시작 시점부터 개발 기술과 코드에 대한 검증이 매우 중요하다. 따라서 성공적인 SW 개발 R&D 프로젝트 수행과 개발 기술에 대한 고객 만족 및 신뢰 증대를 위해서는, 기술 개발의 시작 시점부터 완료에 이르기까지 개발 기술과 코드 등 결과물에 대한 품질 관리가 무엇보다도 중요하며, 나아가 기술 개발을 완료하는 시점에서 개발 기술의 기업 이전에 앞서 개발 기술과 소스 코드에 대한 추가적인 검증 과정이 필요하다.

이에 본 고에서는 SW 개발 R&D 프로젝트에서 개발 기술과 코드를 검증하고 관리하는 개발 기술 및 코드 검증 모델을 소개한다. 이를 위해 먼저 SW 개발 R&D 프로젝트에서 개발 기술 검증 내용과 방법을 기술하고, 개발 기술 완료 후 기업 이전에 앞서 개발 기술과 소스 코드 검증을 위한 절차와 환경을 기술한다. 그리고 SW 개발 R&D 프로젝트에서 개발 기술 및 소스 코드 검증 모델 영향에 대한 설문 결과를 정리하고, 결론으로 요약한다.

II . 개발 기술 검증

개발 기술과 코드 검증을 위해서는 요구사항 정의와 요구사항별 문서화된 테스트 케이스 준비 단계, 코딩 및 디버깅 및 테스트 단계, 소스 코드 및 오픈 소스 라이선스 검증 단계, 그리고 품질 요원에 의한 개발 기술과 코드의 2 차적 검증 단계를 포함하여 기본적으로 5 단계 과정으로 이루어진다.

1. 테스트 케이스 기반 요구사항 검증

개발자는 사용자 요구사항을 도출 및 정의하고, 사용자 요구사항을 개발할 시스템 기능으로 표현하는 시스템 요구사항[1],[2],[6],[7]을 기술한다. 준비한 요구사항에 대해 동료 검토를 통해

보완하고 정의된 요구사항을 검증할 테스트 케이스와 검증 방법(Test/Inspection/Demonstration/Analysis/Simulation/Modeling)을 정의한다. 다음은 활동에 대한 결과물 예이다.

- 요구사항 검증보고서([표 1] 참조)
- 사용자 & 시스템 요구사항 정의서/요구사항별 테스트 케이스/요구사항 추적표

[표 1] 테스트 케이스 기반 요구사항 검증 보고서

순번	시스템 요구사항 ID	테스트 케이스 ID	검증 방법	동료검토	QA 검토

2. 코딩 & 디버깅 & 테스트

개발자는 개발할 요구사항 및 기능에 대해 동작 시나리오와 데이터 흐름을 정의한 후 각각의 클래스와 메소드 정의를 통해 코딩을 수행한다. 또한, 코딩 과정에서는 기능 동작에 따른 취약점과 테스트 코드를 고려하여 코딩과 디버깅을 반복적으로 수행하고, 요구사항정의 단계에서 준비한 테스트 케이스를 이용하여 기능 검증과 동시에 결함을 제거한다. 이후 기능 구현 과정에서 이용했던 테스트 케이스를 보완하여 통합 및 시스템 시험에 활용하도록 한다. 다음은 활동과 결과물 예이다.

- 각 요구사항과 기능에 대한 클래스 및 메소드 정의
- 각 클래스 및 메소드별 코딩 및 디버깅
- 각 클래스 및 메소드에 대한 실패, 성공 테스트 데이터 준비 및 시험
- 가독성과 의존성을 고려한 코드 리팩토링
- 반복적 작업을 통해 코드 품질 및 완성도 향상
- 업데이트 테스트 케이스/결함보고서
- 사용자 & 시스템 요구사항 정의서/코드/요구사항별 테스트 케이스/요구사항 추적표

3. 정적분석

개발자는 개발한 소스 코드에 대해 코드 결함과 취약성 제거를 위해 정적분석[3],[4],[8]을 진행한다. 정적분석을 통해 발생한 코드 취약점이나 결함에 대해 추가적으로 코드 보완 작업을 진행한다. 보완된 소스 코드에 대해 정적분석을 추가로 진행하여 메모리 릭(leak)이나 실행 오류(run time error)와 같은 치명적인 코드 결함이나 보안 취약점 등의 코드 결함이 추가적으로 발

생하지 않으면 변경한 코드에 대해 회귀 시험을 진행하고 테스트 케이스를 업데이트 한다. 소스 코드에 대한 치명적 결함이 없도록 정적분석과 회귀 시험 과정을 반복적으로 수행한 후 최종적인 소스 코드에 대한 정적분석 결과서를 준비한다.

- 소스 코드 정적분석 실행 및 치명적 결함 제거
- 코드 수정(클래스, 메소드, 변수 등)에 따른 회귀 시험
- 업데이트 테스트 케이스
- 반복적 작업을 통한 코드 품질 및 완성도 향상
- 정적분석 결과서([표 2] 참조)

[표 2] Code Compliance Report

프로젝트 이름		개발자 이름		날짜		QA 검토
프로그램 언어		라인수		총 결함수		
코딩표준 규칙		보안 취약 규칙		실행오류 규칙		
결함	중요	결함	중요	결함	중요	
	마이너		마이너		마이너	

4. OSS 라이선스 검증

개발자는 개발 기술에 대한 정적분석을 완료한 후 소스 코드에 대한 OSS(Open source software)[5] 라이선스 검증 과정을 진행한다. 이는 개발한 소스 코드에서의 OSS 사용 여부와 OSS 사용에 따른 제약 사항을 사전에 파악하고 대처하기 위함이다. 또한, 개발 소스 코드에 대한 OSS 라이선스 검증에 앞서 개발 소스에 대해 파일별, 혹은 기능별, 모듈별, 클래스별, 혹은 디렉토리별로 구분하여 OSS 사용 여부와 사용 버전 그리고 사용 내용 등을 사전에 구분 처리하여 OSS 라이선스 검증을 용이하게 하도록 한다. OSS 라이선스 검증을 진행한 후 OSS 사용에 따른 제약 사항이 없는 경우 개발자는 개발 소스에 대한 OSS 라이선스 검증 보고서를 준비한다. 아래는 활동과 결과물 예이다.

- 개발 소스에 대한 OSS 사용 구분 명시
- OSS 사용에 따른 제약 사항 제거
- OSS 라이선스 검증 보고서([표 3] 참조)

[표 3] OSS 라이선스 검증 보고서

프로젝트 이름						개발자 이름		QA 검토
검증 대상 파일 수						분석 파일 수		
분석 바이트 수						공개 SW 사용 파일 수		
준법성 주의 파일 수						준법성 위반 파일 수		
Bill of Materials	준법성	컴포넌트	버전	라이선스	결함	해당파일수		
공개 SW 라이선스 종합 요약								
검증 및 분석 의견								

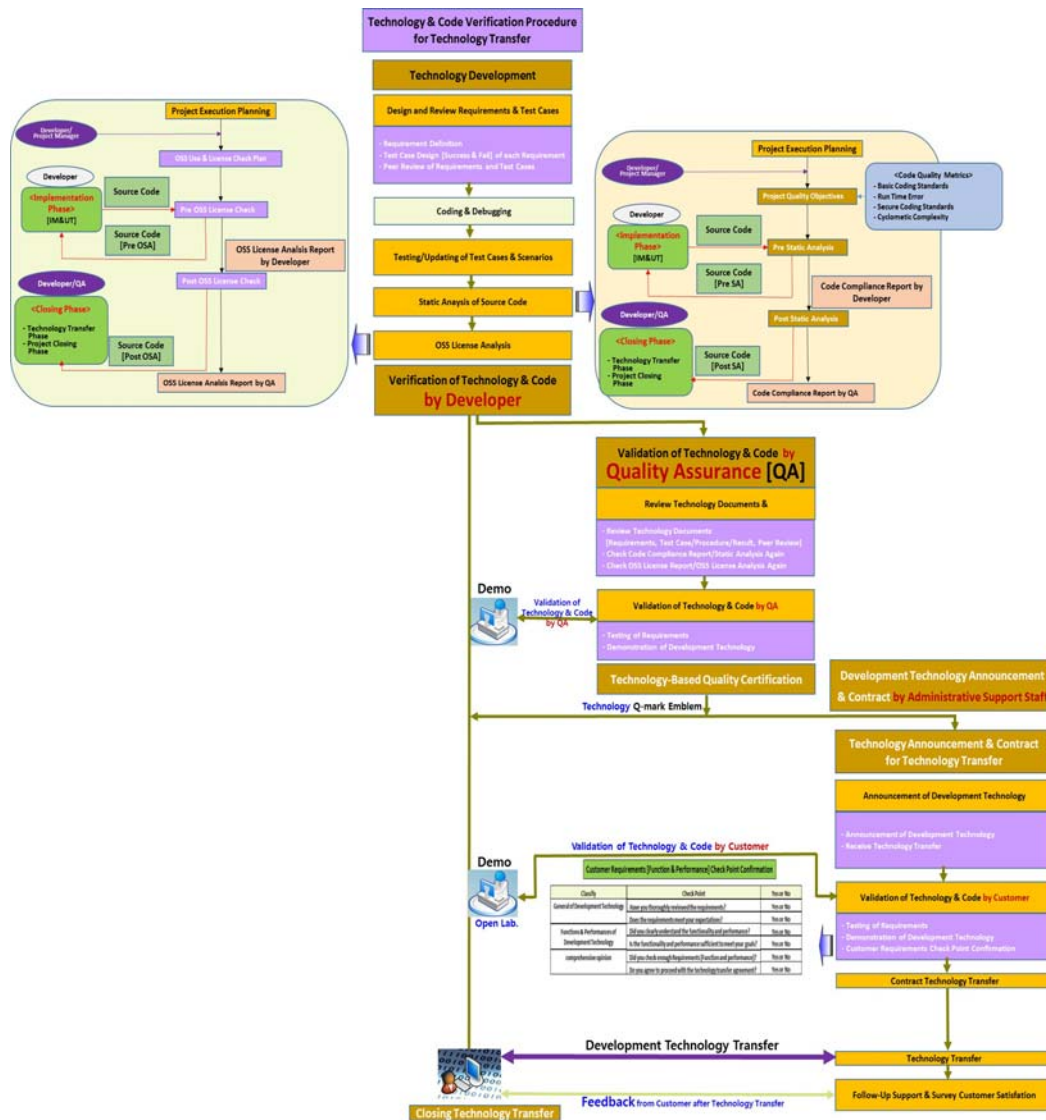
5. QA 검증

프로젝트 수행 및 기술 개발 과정에서 개발자는 개발 기술과 소스 코드에 대해 1차적으로 검증과정을 거친 후 개발 기술에 해당하는 문서와 소스 코드를 QA(Quality assurance)로 보낸다. QA는 개발자가 보낸 소스 코드와 요구사항 정의서, 시험 절차서 및 시험 결과서 등 개발 기술 관련 문서를 검토하고, 요구사항별 테스트 케이스와 시험 절차서 그리고 시험 결과서를 바탕으로 개발 기술을 검증한다. 또 QA는 개발 기술에 대한 문서적 검증과 시연을 통해 개발 기술 검증을 마친 후 개발 소스 코드에 대해 정적분석과 OSS 라이선스 검증을 2차적으로 진행한다. QA에 의한 요구사항 검증과 정적분석, 그리고 OSS 라이선스 검증을 완료하면 QA는 개발 기술에 대한 검증보고서를 준비하여 조직의 최고 책임자 승인을 득하여 개발 기술 검증과 코드에 대해 인증을 부여한다. 다음은 활동과 결과물 예이다.

- 요구사항 정의서, 시험절차서, 시험결과서
- 테스트 케이스, 동료검토서, 결함보고서
- 요구사항 추적표 검증
- 정적분석 보고서(by QA)
- OSS 라이선스 검증 보고서(by QA)
- 개발 기술 시연 & 확인
- 요구사항 검증보고서(by QA)

III . 개발 기술과 코드 검증 모델

SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증을 위해 요구사항 정의 및 요구사항별 테스트 케이스 준비, 동료 검토, 설계와 구현 단계에서 디버깅과 시험, 그리고 개발 소스에 대한 정적분석과 OSS 라이선스 검증 과정이 개발자에 의해 1 차 진행된다. 이후 프로젝트 종료



<출처> ETRI 자체 제작

[그림 1] 개발 기술과 코드 검증 절차

및 기술 개발을 완료하여 기업에 이전 하기에 앞서 개발 기술과 코드에 대해 품질담당자에 의한 추가적인 검증 과정이 진행된다. 이에 대한 절차 및 환경은 다음과 같다.

1. 개발 기술 및 코드 검증 절차

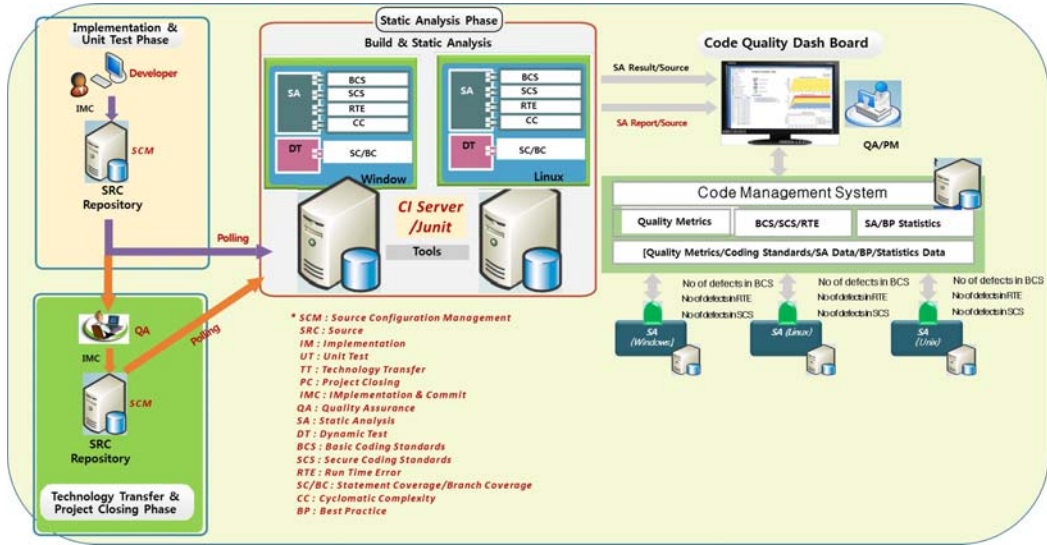
[그림 1]은 SW 개발 R&D 프로젝트에서 개발한 기술과 코드에 대한 검증 절차이다. 이 절차는 기본적으로 프로젝트에서 개발한 기술을 기업에 이전하는 경우나 TRL(Technology readiness level) 5 이상의 SW 개발 R&D 프로젝트에서 개발한 기술과 코드에 대해서 필수적으로 진행되는 검증 프로세스이다. 개발 기술과 코드 검증 모델은 먼저 개발자에 의해 요구사항 정의 및 요구사항별 테스트 케이스 준비와 이를 검증하는 부분, 테스트와 디버깅을 통해 설계 및 구현한 기술을 검증하는 부분과 개발 소스 코드에 대한 취약점과 OSS 라이선스를 검증하는 부분, 그리고 품질담당자에 의해 개발 기술과 코드를 확인하고 검증하여 릴리즈하는 부분, 마지막으로 기업 이전을 위해 오픈 랩을 통해 사용자에게 의해 개발 기술을 확인하고 검증하는 개발 기술 데모 수행 부분으로 이루어진다.

2. 개발 기술과 코드 검증 환경

SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증을 위한 정적분석 환경, OSS 라이선스 검증 환경, 그리고 프로젝트 수행 및 기술 개발 과정에서 발생한 다양한 문서와 소스, 그리고 이슈, 위험, 및 프로젝트 관리 노하우를 포함한 각종 BP(Best practices)에 대한 공유 및 관리 환경은 다음과 같다.

가. 소스 코드 검증 환경

[그림 2]는 SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증을 위한 정적분석 환경이다. 요구사항 정의 및 분석이 이루어진 후 설계 및 구현 단계에서 개발자는 코딩 기준을 고려하여 코딩과 디버깅 그리고 자체 테스트 과정을 반복 수행한다. 이 과정에서 개발자는 정적분석 툴을 이용하여 개발 소스에 대한 취약점 점검을 진행하여 개발 소스에 대한 코딩 표준 준수율, 메모리 릭, 그리고 실행 오류 등을 검증한 정적분석 결과서를 준비한다. 이후 개발 소스 코드에 대해 동적 시험을 진행하여 개발 기술 기능을 검증함과 동시에 결함을 줄이고 개발 소스 코드의 완성도를 높인다. 또 개발한 소스의 공유 및 재활용을 위해 선별 후 BP 풀로 이관한다.

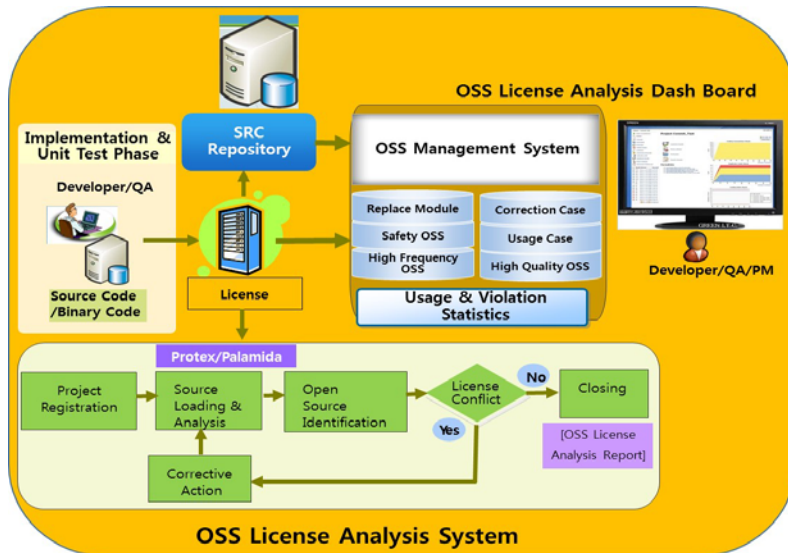


<출처> ETRI 자체 제작

[그림 2] 소스 코드 정적분석 환경

나. OSS 라이선스 검증 환경

[그림 3]은 SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증을 위한 OSS 라이선스 검증 환경이다. 개발자는 요구사항 정의 및 분석, 그리고 시스템설계 단계에서 기술 구현 방법과 오



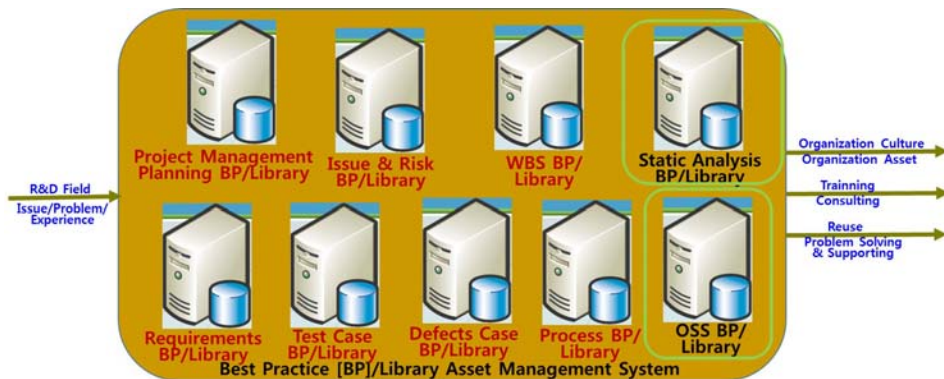
<출처> ETRI 자체 제작

[그림 3] OSS 라이선스 검증 환경

폰 소스 활용 여부를 검토한다. 또 오픈소스를 사용할 경우 오픈 소스 라이선스 문제를 검증하고 오픈 소스를 활용하여 기술과 코드 개발을 진행한다. 이후 개발한 기술과 코드에 대한 디버깅 및 테스트 과정을 거쳐 검증하고, 오픈 소스 툴을 이용하여 개발 소스 코드에 대해 OSS 라이선스 검증 작업을 진행하여 OSS 라이선스 검증보고서를 준비한다. 또 개발한 소스의 공유 및 재활용을 위해 선별 후 BP 풀로 이관한다.

다. 개발 기술과 코드의 자산화 및 공유 환경

[그림 4]는 SW 개발 R&D 프로젝트에서 개발한 기술과 코드 그리고 프로젝트 수행 및 기술 개발 과정에서 발생한 시스템 개발 자료와 프로젝트 관리 자료에 대한 자산화 및 공유 환경 체계이다. 공유 및 관리 시스템에는 프로젝트 수행에서 발생한 이슈나 위험 관리, 의사소통, WBS 등의 정보와 프로젝트 수행에서 생성한 각종 기술 개발 문서와 정적분석이나 OSS 라이선스 검증이 완료된 코드를 포함하며 이는 다른 프로젝트 수행에서 재활용, 공유 및 교육용으로 활용할 수 있도록 한다.

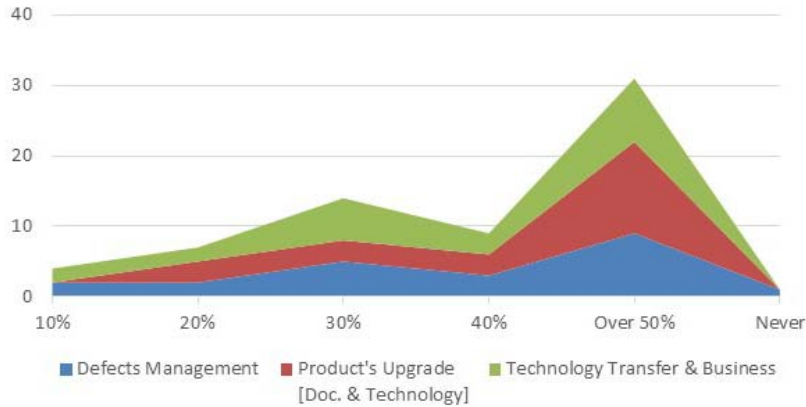


<출처> ETRI 자체 제작

[그림 4] 개발 기술과 코드 자산화 및 공유 환경

Ⅳ . 개발 기술과 코드 검증 모델이 미치는 영향

[그림 5]는 SW 개발 R&D 프로젝트에서 결함 제거, 개발 기술 및 코드 완성도 향상, 그리고 기술 이전과 사업화 측면에서 개발 기술과 코드 검증 모델의 긍정적 영향^{1),2)}과 개선 수준에 대한 설문 응답 결과에 대한 요약이다. SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증 모



<출처> ETRI 자체 제작

[그림 5] 개발 기술 및 코드 검증 모델이 미치는 영향

델이 개발 기술에 대한 결함 제거 측면에서 50% 이상 직·간접적으로 좋은 영향을 주었다고 41%가 응답하였고, 30~40% 이상 좋은 영향을 주었다고 37%가 응답하였다. 또 개발 코드를 포함하여 연구 결과물 완성도 항상 측면에서도 60% 이상 좋은 영향을 주었다고 50% 이상이 응답하였고, 대부분의 응답자는 연구 결과물 완성도 항상에 최소 30% 이상의 좋은 영향을 주었다고 답하였다. 그리고 개발 기술의 기업 이전과 사업화에도 50% 이상 좋은 영향을 주었다고 41%가 응답하였고, 30~40% 이상 좋은 영향을 주었다고 응답한 사람도 41%였다.

V. 결론

지금 이 순간에도 각 국가, 기업, 조직에서 프로젝트가 수많은 영역에서 수행되고 있다. 그 중에서도 R&D 프로젝트가 차지하는 비중은 날로 높아가고 있다. 이렇게 많은 프로젝트를 수행하고 있지만, 실제 프로젝트 성공률은 50% 이상 되지 않는다. 미국이나 유럽에서는 프로젝트 실패율을 줄이기 위해 PMP(Project Management Professional)나 PRINCE2(Projects IN Controlled Environments version 2)와 같은 프로젝트 관리 방법이나 표준 프로세스 수립 및 활용을 제시하고 있으나 이 또한 기업이나 조직의 환경과 여건에 따라 조정하여 적용 및 내재화 하기까지는 상당한 시간과 시행착오가 따른다. SW 개발 R&D 프로젝트 수행 환경에서도 이와 같은 문제는 예외가 아니며 이에 본 고에서는 개발 기술과 소스 코드 검증을 위한 방법과 절차, 그리고 필요한 환경 및 체계를 포함한 모델을 소개했다. 제시된 모델에는 개발 기술 검증을 위해 문서적

테스트 케이스 기반 요구사항 정의와 동료검토, 설계 및 구현 단계에서 개발자에 의한 시험 및 구현과 정적분석 및 오픈소스 라이선스 검증, 그리고 프로젝트 종료 및 완료 단계에서 품질담당자에 의한 확인과 소스 코드에 대한 2 차적 검증 방법과 절차 그리고 개발 기술과 소스 코드의 자산화 환경이 기술되었다. 제시된 모델은 SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증 및 릴리즈를 위해 확대 적용 가능하며 아울러 프로젝트 수행에서 생성한 기술 문서나 코드 그리고 프로젝트 수행 관리에 대한 노하우 문서를 자산화하고 공유, 재활용 및 교육용으로 사용할 수 있도록 한다. 또 SW 개발 R&D 프로젝트에서 개발 기술과 코드 검증 모델은 결함 제거, 코드 완성도, 개발 기술의 기업 이전 및 사업화에 10~30%의 좋은 영향을 주고 있음을 설문을 통해 확인할 수 있었다. 앞으로 모델 보완 및 확대 적용을 통해 개발 기술과 코드를 포함해서 SW 개발 R&D 프로젝트 수행에서 생성한 결과물에 대한 완성도 및 품질 향상에 지속적으로 기여할 것으로 기대한다.

[참고문헌]

- [1] Jeong-Hyun Park et al, "Q-Mark Model for Quality of R&D Projects," FGCN2014, SERSC Proc., Suwon, Rep. of Korea, Nov. 21-22, 2014, pp.1743-1756.
- [2] Jeong-Hyun Park et al, "Q-Mark Model Assessment Using CBAM(Cost Benefit Analysis Method)," in Int. Conf. Manag. Eng. Technol., Portland, OR, USA, Aug. 2-6, 2015, pp.1805-1808.
- [3] Jeong-Hyun Park et al, "Static Analysis for Code Quality on R&D Projects of SW Development," ETRI Magazine, 2017. 2.
- [4] NIPA, "Static Analysis using Open Source," GCS School, 2015. 5.
- [5] NIPA, "Understanding and Management of Open Source," GCS School, 2016. 5.
- [6] Quality Innovation Section, "ETRI R&D Standard Process V4.0," ETRI, 2016. 8.
- [7] Quality Innovation Section, "Q-mark 6.0," ETRI, 2017. 4.
- [8] Steve McConnell, "Code Complete," Microsoft Press, 1993. 5.

* 본 연구는 '경영품질성과 향상을 위한 프로세스 개선 및 혁신전략 연구 과제(170A1110)' & '언어 장벽 없는 국가 구현을 위한 자동 통번역 산업 경쟁력 강화 사업(17ZS1210)'수행 결과임 / 본 연구 결과는 ETRI 공식 견해와 다를 수 있음