

VPES

프로세스 리포팅 도구

SURESOFT

TABLE OF CONTENTS

Why VPES	3p
Needs	5p
Solution	6p
Features	7p
Specification	23p
References	24p
ROI	27p

VPES

VPES(브이피에스)는 프로세스 지원 및 품질 관리를 수행하는 시험 협업 도구로, 무기체계 SW 프로세스를 중심으로 산출물 관리 및 시험프로세스 관리를 제공합니다.

핵심경쟁력
(차별점/강점)

신뢰성 시험 프로세스 지원



문서 검증 관리



시험 통합 관리



AI 문서 자동생성



AS-IS

- 1 시험 절차 및 예외사항 관리 어려움
- 2 기술 문서 검사 및 추적성 검토 어려움
- 3 시험 도구상세 결과 직접 작성 어려움
- 4 변화하는 소스코드의 시험 관리 어려움
- 5 업무 효율 증가 필요



To-BE

- 1 WEB 기반 간편한 입력을 통한 데이터관리
- 2 문서 추적성 검증으로 편리한 검토
- 3 문서 자동 생성 기능으로 보고서 자동 생성
- 4 형상 관리 기능으로 누락 없는 시험 진행
- 5 AI 기능으로 SDD 완전 자동 생성

차별점

Unrivaled

국내 유일 무기체계 SW 개발
프로세스 관리 도구

Compatibility

코드 기반 검증 솔루션과 연동

Quick Setup

쉽고 빠른 서버 기반 설치 환경 제공

Process

무기체계 SW 개발 프로세스 지원

강점

독보적

- 국내 유일한 무기체계 SW 프로세스 관리 도구
- 기술 문서 검증 기능 지원
- 한글/엑셀/워드 산출물 자동 생성 및 완벽 호환

도구 연동

- STATIC, CT, COVER 도구 연동
- 플러그인 방식의 쉬운 결과 업로드

유연성

- 20년 업력의 개발 및 공급 경험
- 도메인 요구 사항에 대한 빠른 대처

프로세스

- 기술 문서 검증 프로세스 지원
- 신뢰성 시험 프로세스 지원

기대 효과

[무기체계 SW 신뢰성 및 품질 향상의 기반 마련]

무기체계 SW 개발 프로세스 지원

- 프로세스 전 단계 지원
- 프로세스 단계별 가이드 지원
- 가이드 영상으로 편의 제공

기술 문서 검증 및 관리

- 기술 문서방사청 기준 점검
- 문서 정합성 검증
- 문서간 일치성 검증

신뢰성 시험 및 결과 모니터링

- 간편한 시험 설정 및 수행 관리
- 신뢰성 시험의 목표 및 진척 관리
- 웹에서의 예외 처리 관리

산출물 작성 비용 절감

- 한글/엑셀/워드 리포트 자동 생성
- 한글/엑셀/워드 리포트 완벽 호환
- 전체 산출물의 96.7% 자동 생성

01

자체 문서 검증 기능으로
기술 문서 검증 및 관리

02

테스트 자동화 도구와의
간편한 연동으로 테스트
데이터를 통합 관리

03

무기체계 SW 개발
프로세스가 익숙하지 않은
고객도 쉽게 따라할 수 있는
고접근성 시스템 확보

04

신뢰성 시험 결과를
지속 누적, 관리하여
시각화함으로써
품질 이력을 확인

05

신뢰성 시험 결과
리포트 자동 생성으로
산출물 작성 시간 단축



- 01 기술 문서 검토 불편 → 기술 문서 검사 및 추적성 관계 검토의 번거로움
- 02 신뢰성 시험 미흡 → 시험 절차 및 예외 사항 관리의 어려움
- 03 문서 산출물 작성 부담 → 시험 도구의 상세 결과를 보고 사람이 직접 작성
- 04 시험 산출물 검토 불편 → 시험 산출물 시험 도구의 상세 결과와 직접 비교해야 하는 어려움



신뢰성 시험의 불편 사항에 대한 해결책이 필요

기술 문서 검토 불편

- 기술 문서 규격 검사의 어려움
- 문서 간 추적성 검토 번거로움

신뢰성 시험 미흡

- 신뢰성 시험 절차 수립 어려움
- 예외 사항의 처리 미숙

문서 산출물 작성 부담

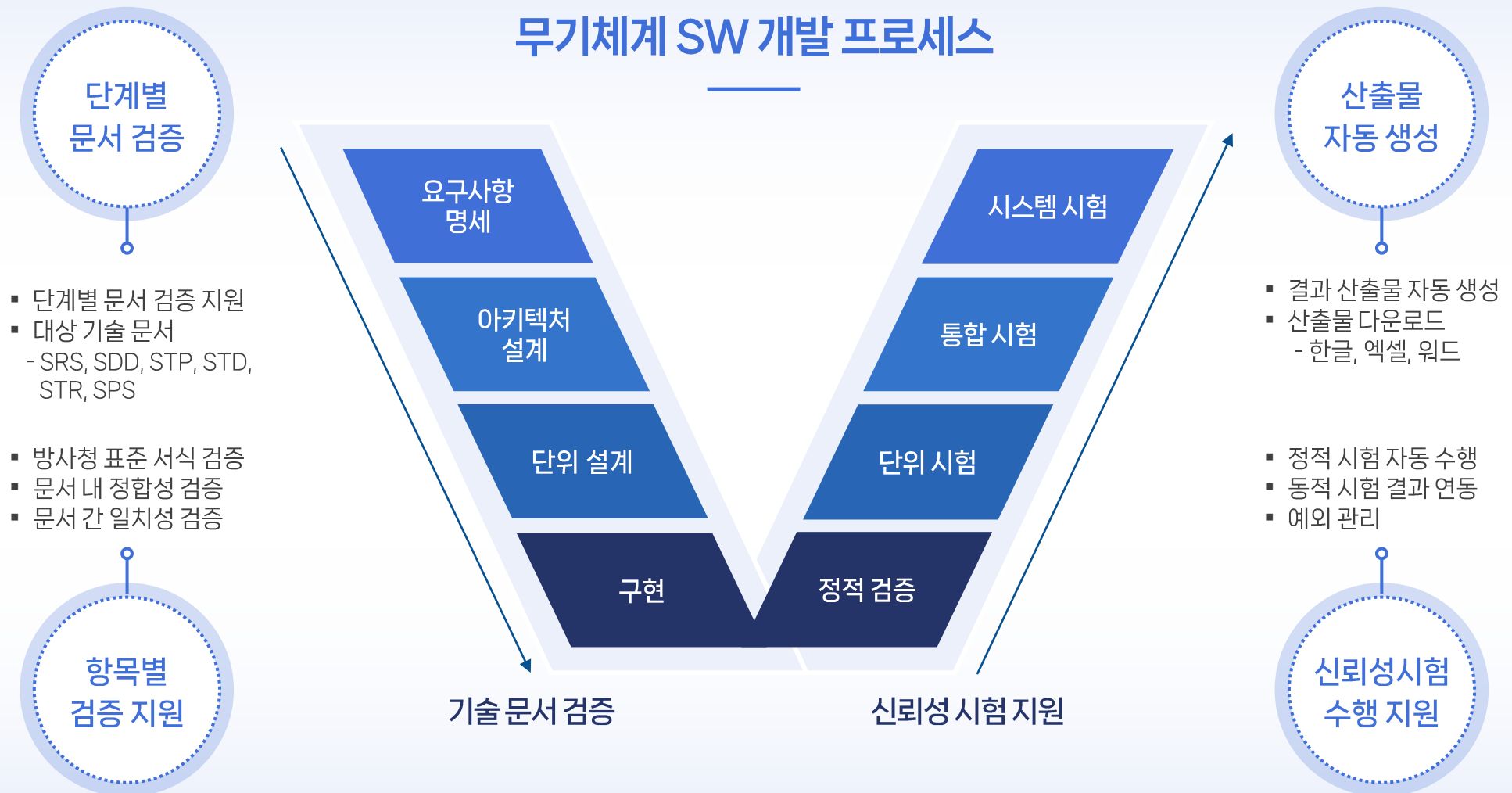
- 11종의 시험 산출물 작성에 드는 시간 비용 발생

시험 산출물 검토 불편

- 시험 산출물의 내용과 시험 도구 상세 결과 비교의 번거로움

신뢰성 시험 관리

- 무기체계 SW 개발 프로세스를 지원하여 문서 검증, 이력 관리와 자동 산출물을 제공



기술 문서 검증 및 관리 (방사청 가이드 기준)

- 무기체계 SW 기술 문서 작성 가이드 기준

일반 검사

- SW 기술 문서 작성 서식 기준 검사
- 필수 항목 검사
- 규격 검사, 여백 검사, 머리말/꼬리말 검사 지원

위배: 194건

Example

TEST_SW-SRS.hwp

내용	상세 내용	구역	쪽
해당 제목의 내용이 존재하지 않습니다.	위배 제목: 3.2.5 정보관리 CSC 기능	2	17
제목 형식 설정이 문서 작성 기준과 다릅니다.	기준: 160 / 사용한 설정: 180 / 위배위치 제목: 6. 참고사항	2	31
제목 이후 본문 첫 문장은 제목 시작 만큼 들여쓰기 되어야 합니다.	위배 제목: 3.4.4 소프트웨어 협상할록 내부 자료 요구사항	2	24

문서 정합성 검사

- 요구 사항, 구성 요소, 시험 항목 목록에 대한 상세 내용 검사
- 조건 미충족 목록과 시험 생략 목록에 대한 검사
- 미충족 목록과 세부 시험 결과 목록에 대한 검사

위배: 2건

Example

TEST_SW-STD.hwp

시험 식별자	시험 항목	시험 상세
T-DEMO-SFR-000		X
1. 시험상세 목록에 식별자가 존재하지 않습니다		
T-DEMO-SFR-001		X

문서 간 일치성 검사

- 기술 문서 추적표에 작성된 요구 사항 간의 일치성 검사
- 시험 절차서와 결과서 간 일치성 검사
- 문서 간 일치 관계표(추적표) 제공

SRS - SDD 위배: 4건

Example

TEST_SW-SRS 다운로드 TEST_SW-SDD 다운로드

식별자	내용
D-DEMO-SFR-001	SDD에만 존재하고, SRS에는 존재하지 않습니다. - KEY: D-DEMO-SFR-001
D-DEMO-SFR-111	SDD에만 존재하고, SRS에는 존재하지 않습니다. - KEY: D-DEMO-SFR-111

문서 간 일치 관계표

SRS / SDD	D-DEMO-SFR-001	D-DEMO-SFR-004	D-DEMO-SFR-002	D-DEMO-SFR-003	D-DEMO-SFR-005	D-DEMO-SFR-006	D-DEMO-SFR-007	D-DEMO-SFR-008	D-DEMO-SFR-009	D-DEMO-SFR-010	D-DEMO-SFR-011	D-DEMO-SFR-012
R-DEMO-SFR-001	+	✓										
R-DEMO-SFR-002			✓									
R-DEMO-SFR-003				✓								
R-DEMO-SFR-004		✓										
R-DEMO-SFR-005					✓							

기술 문서 검증 및 관리 (일반 검사)

- 2024-6호 기준 44개 항목 검사

요구사항

설계

구현

시험

평가

검증 항목 설정

☒ 일반 검증
 ☒ 정합성 검증
 ☒ 일치성 검증

☐ 일반 검증 항목 전체 선택

☒ 용지 규격 설정
 ☒ 용지 여백 설정
 ☒ 머리말 스타일 속성
 ☒ 꼬리말 스타일 속성
 ☒ 그림 제목 유/무

☒ 그림 제목 정렬
 ☒ 그림 제목 서체
 ☐ 그림 제목 글자 크기
 ☒ 그림 제목 위치
 ☒ 그림 제목 행간

☒ 그림 제목 장평
 ☐ 그림 제목 자간
 ☒ 도표 제목 유/무
 ☒ 도표 제목 정렬
 ☒ 도표 제목 서체

☐ 도표 제목 글자 크기
 ☐ 도표 제목 위치
 ☒ 도표 제목 행간
 ☒ 도표 제목 장평
 ☒ 도표 제목 자간

☐ 그림 글자처럼 취급 옵션 유/무
 ☒ 도표 글자처럼 취급 옵션 유/무
 ☒ 그림 상단 하단 1줄 공백
 ☐ 도표 상단 하단 1줄 공백
 ☒ 제목 상단 1줄 공백

☒ 2레벨 제목 상단 2줄 공백
 ☒ 본문 내용 제목 들여쓰기
 ☒ 1레벨 제목 세 페이지 시작
 ☒ 제목 정렬 확인
 ☒ 제목 서체 확인

☒ 제목 글자 크기
 ☒ 제목 행간
 ☒ 제목 장평
 ☒ 제목 자간
 ☒ 본문 정렬

☒ 본문 서체
 ☒ 본문 글자
 ☒ 본문 행간
 ☒ 본문 장평
 ☒ 본문 자간

☒ 제목 내용 유/무
 ☒ 목차와 제목 일치성
 ☒ 방사형 서식과 목록 일치성

저장

닫기

상 다운로드 : 2_SDD_예제_위배.hwp

필터

검색어를 입력하세요.

10

상세 내용

위배 꼬리말 제목 설정 한쪽

기준 20.0mm 문서 10.0mm

위배 머리말 제목 설정 한쪽

기준 20.0mm 문서 10.0mm

기준 160 사용한 설정 310

위배 장절 3.4.2 유지보수를 위한 고려사항

본문 소프트웨어의 이식성

기준 1행 공백

위배 장절 4.2.3.1.10.1 활동 다이어그램

기준 1행 공백

위배 장절 4.2.3.1.10.1 활동 다이어그램

위배 제목 4.2.3.1.10.2 시퀀스 다이어그램

기준 1행 공백

위배 장절 4.2.3.2.10.1 활동 다이어그램

2	38	그림 하단 공백이 기준과 다릅니다.
2	38	그림 상단 공백이 기준과 다릅니다.
2	39	해당 제목의 내용이 존재하지 않습니다.
2	49	그림 상단 공백이 기준과 다릅니다.

기술 문서 검증 및 관리 (문서내 정합성 검사)

- 문서내 목록과 상세 내용 간 충족 미충족 검사

요구사항 설계 구현 시험 평가

· 위배 : 1건			· 대상 다운로드 : 1. SRS_예제.hwp		
필터 Q 검색어를 입력하세요. 10 ≡ 위배 정렬 ↓					
요구사항 식별자	요구사항	요구사항 상세			
▼ R-DEMO-SFR-031	O	X			
1. 요구사항 상세목록에 식별키가 존재하지 않습니다					
R-DEMO-SFR-001	O	O			
R-DEMO-SFR-002	O	O			
R-DEMO-SFR-003	O	O			
R-DEMO-SFR-004	O	O			

기술 문서 검증 및 관리 (문서간 일치성 검사)

- 기술문서 추적표에 작성된 요구사항 간의 일치성
- 절차서와 결과서 간 일치성
- 일치 관계표

요구사항

설계

구현

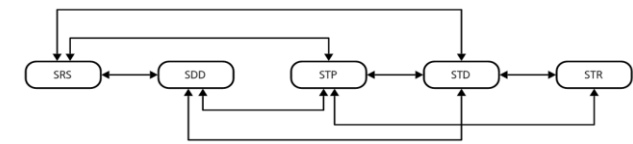
시험

평가

문서 간 일치 관계표

SRS / SDD	D-DEMO-SFR-001	D-DEMO-SFR-004	D-DEMO-SFR-002	D-DEMO-SFR-003	D-DEMO-SFR-005	D-DEMO-SFR-006	D-DEMO-SFR-007	D-DEMO-SFR-008	D-DEMO-SFR-009	D-DEMO-SFR-010	D-DEMO-SFR-011	D-DEMO-SFR-012	D-DEMO-SFR-013	D-DEMO-SFR-014	D-DEMO-SFR-015	D-DEMO-SFR-016	D-DEMO-SFR-017	D-DEMO-SFR-018	D-DEMO-SFR-019
R-DEMO-SFR-001	❌	✅																	
R-DEMO-SFR-002			✅																
R-DEMO-SFR-003				✅															
R-DEMO-SFR-004		✅																	
R-DEMO-SFR-005					✅														
R-DEMO-SFR-006						✅													
R-DEMO-SFR-007							✅												
R-DEMO-SFR-008								✅											
R-DEMO-SFR-009									✅										
R-DEMO-SFR-010										✅									
R-DEMO-SFR-011											✅								
R-DEMO-SFR-012												✅							
R-DEMO-SFR-013													✅						
R-DEMO-SFR-014														✅					
R-DEMO-SFR-015															✅				
R-DEMO-SFR-016																✅			
R-DEMO-SFR-017																	✅		
R-DEMO-SFR-018																		✅	
R-DEMO-SFR-019																			✅
R-DEMO-SFR-020																			
R-DEMO-SFR-021																			
R-DEMO-SFR-022																			
R-DEMO-SFR-023																			
R-DEMO-SFR-024																			
R-DEMO-SFR-025																			
R-DEMO-SFR-026																			
R-DEMO-SFR-027																			
R-DEMO-SFR-028																			
R-DEMO-SFR-029																			
R-DEMO-SFR-030																			
R-DEMO-SFR-031																			
R-DEMO-SFR-0010002	❌																		

기술문서 추적 관계



문서 간 일치성 검증

SRS	SDD	STP	STD
SDD 2건	STP 35건	STD 36건	STP 35건
	STD 36건	STD 1건	STR 37건
			STR 36건

문서 간 일치성 검증 결과

SRS - SDD 위배: 2건		SRS 다운로드	SDD 다운로드
필터	검색어를 입력하세요.	10	다운로드
식별자	내용		
D-DEMO-SFR-001	SDD에만 존재하고, SRS에는 존재하지 않습니다. - KEY: D-DEMO-SFR-001		
R-DEMO-SFR-001	SRS에만 존재하고, SDD에는 존재하지 않습니다. - KEY: R-DEMO-SFR-001		

문서 간 일치 관계표

닫기

신뢰성 시험 자동화

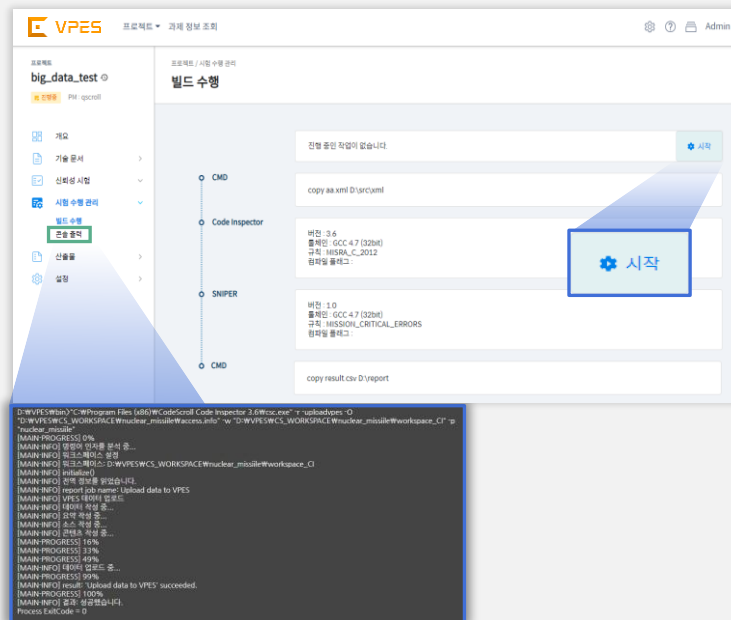
요구 사항

설계

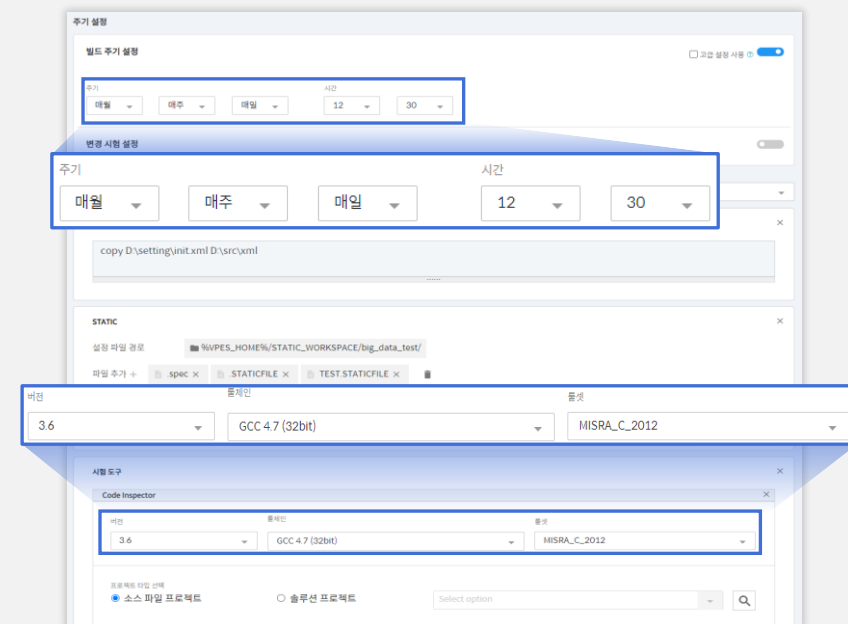
구현

시험

평가



- 클릭 한 번으로 코딩 규칙/소스코드 메트릭 검증
- 클릭 한 번으로 취약점 검증
- 형상 관리 연동 시 최신 소스코드로 검증

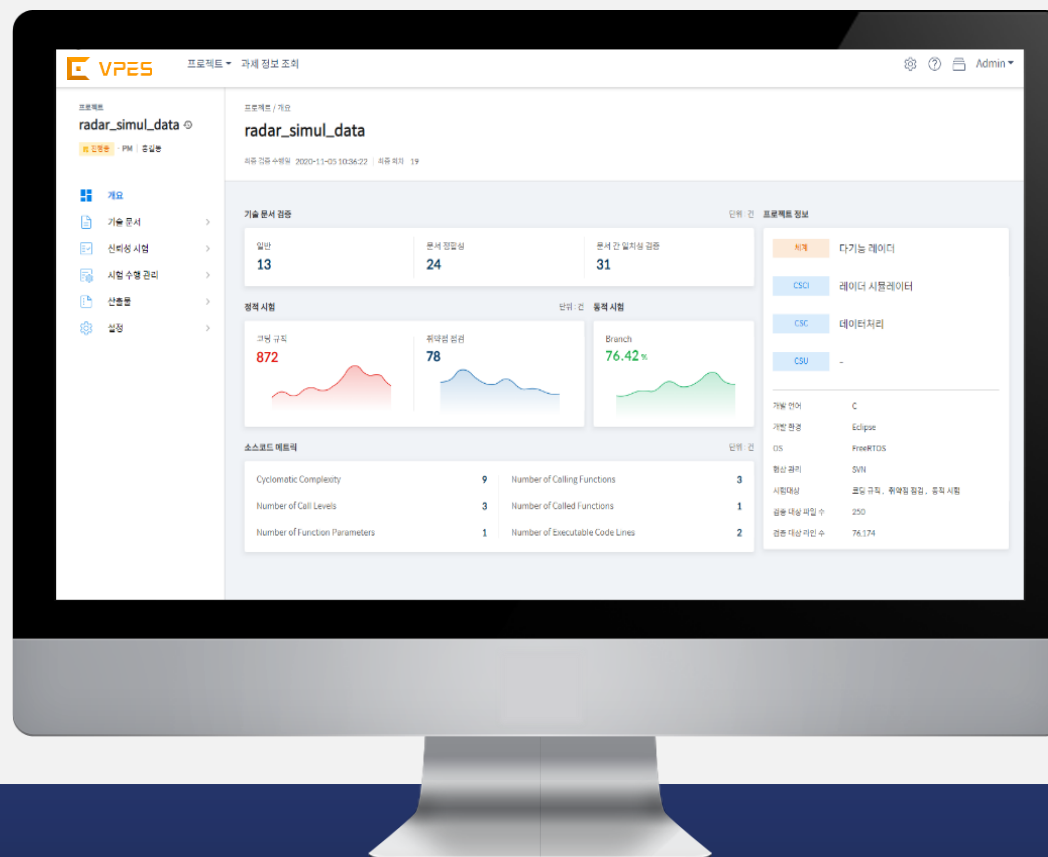


- 툴체인, 규칙, 개발 환경 정보 등의 손쉬운 설정
- 신뢰성 시험 주기를 간편하게 설정
- 형상이 변경되었을 때 신뢰성 시험 자동 수행

시험 결과 모니터링

- 회차 기준으로 시험 결과 이력 관리

- 01 회차별 정적 시험, 동적 시험, 소스코드 메트릭의 **이력** 표시
- 02 코딩 규칙 결과와 취약점 검증 결과의 요약 정보와 상세 정보 표시
- 03 파일별 코드 실행률과 함수별 코드 실행률 표시
- 04 소스코드 메트릭의 점검 기준과 기준을 초과한 **결함 수**를 보여주며 **메트릭의 상세 결과** 표시
- 05 정적 시험과 동적 시험의 예외 현황을 표시하고, 검증 도구와 VPES의 예외 현황을 합산하여 제공



- 06 프로젝트 요약 정보를 표시하고, 파일 체크섬 정보를 제공하여 테스트 대상 소스가 일치하는지 여부 확인 가능

체계 관리를 위한 계층 그룹 대시보드

- 전체 체계의 계층 구조를 통한 프로젝트 통합 관리

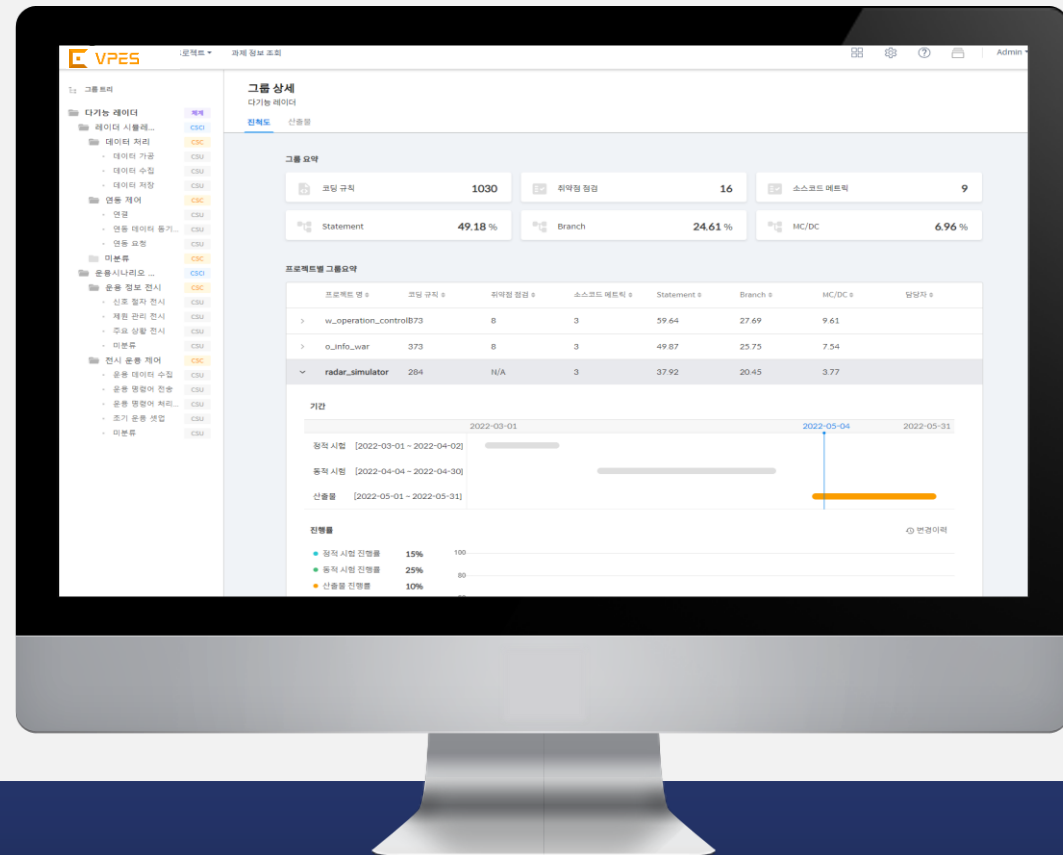
01 전체 계층 그룹 구조 표시

02 선택된 계층 그룹에 따른 시험 결과 요약 정보 표시

03 계층 그룹에 속해 있는 프로젝트별 그룹 요약 정보 표시

04 계층 그룹에 속해 있는 프로젝트별 기간 및 진행률 표시

05 여러 프로젝트들로 이루어진 상위 체계의 11종 산출물에 대한 자동 통합 생성이 가능



체계 관리기반 계층 그룹 산출물

- 계층 그룹 기준에 따른 산출물 생성

요구사항

설계

구현

시험

평가

계층 그룹

다기능 레이더 [체계]

체계기준 산출물 범위

시뮬레이터 [CSCI]

CSCI 기준 산출물 범위

데이터 처리 [CSC]

계산 모듈 [CSU]

인터페이스 모듈 [CSU]

연동 처리 [CSC]

인터페이스 모듈 [CSU]

제어 콘솔 [CSCI]

CSCI 기준 산출물 범위

데이터 처리 [CSC]

계산 모듈 [CSU]

인터페이스 모듈 [CSU]

체계 기준 산출물

[별지1-별첨5] 동적시험 자체 또는 제3자검증 수행결과 요약표

번호	과일명	기준치	측정치	시험도구	비고
1	printPar 설치	Branch	100%	Branch	12.3 ControllerTester2.2
2	valloc.h	Branch	100%	Branch	83.33 ControllerTester2.2
3	mkdtemp c	Branch	100%	Branch	50.0 ControllerTester2.2
4	writepipe ch	Branch	100%	Branch	50.0 ControllerTester2.2
5	regen_inte rnal.h	Branch	100%	Branch	50.0 ControllerTester2.2
6	GLFWWin _create.h	Branch	100%	Branch	50.0 ControllerTester2.2
7	state.h	Branch	100%	Branch	50.0 ControllerTester2.2
8	reduce.c	Branch	100%	Branch	50.0 ControllerTester2.2
9	size.h	Branch	100%	Branch	50.0 ControllerTester2.2

[별지1-별첨5] 소스코드 메트릭 측정 결과

번호	항목	기준치	측정 결과	시험도구	비고
1	Cyclomatic Complexity	20	부적격 (위해 간수:54)	Codinspector 3.4	
2	Number of Call levels	6	부적격 (위해 간수:15)	Codinspector 3.4	
3	Number of Function parameters	8	부적격 (위해 간수:10)	Codinspector 3.4	
4	Number of Calling Functions	8	부적격 (위해 간수:10)	Codinspector 3.4	
5	Number of Called Functions	10	부적격 (위해 간수:41)	Codinspector 3.4	
6	Number of Executable Code Lines	200	부적격 (위해 간수:8)	Codinspector 3.4	

제표리소트 : 예외 부분 및 사유를 첨부하였는가?

예 □ 아니오 □

[별지2-별첨5]

번호	과일명	기준치	측정치	시험도구	비고
1	printPar 설치	Branch	100%	Branch	12.3 ControllerTester2.2
2	valloc.h	Branch	100%	Branch	83.33 ControllerTester2.2
3	mkdtemp c	Branch	100%	Branch	50.0 ControllerTester2.2
4	writepipe ch	Branch	100%	Branch	50.0 ControllerTester2.2
5	regen_inte rnal.h	Branch	100%	Branch	50.0 ControllerTester2.2
6	GLFWWin _create.h	Branch	100%	Branch	50.0 ControllerTester2.2
7	state.h	Branch	100%	Branch	50.0 ControllerTester2.2
8	reduce.c	Branch	100%	Branch	50.0 ControllerTester2.2
9	size.h	Branch	100%	Branch	50.0 ControllerTester2.2

CSCI 기준 산출물

[별지1-별첨5] 동적시험 자체 또는 제3자검증 수행결과 요약표

번호	과일명	기준치	측정치	시험도구	비고
1	printPar 설치	Branch	100%	Branch	12.3 ControllerTester2.2
2	valloc.h	Branch	100%	Branch	83.33 ControllerTester2.2
3	mkdtemp c	Branch	100%	Branch	50.0 ControllerTester2.2
4	writepipe ch	Branch	100%	Branch	50.0 ControllerTester2.2
5	regen_inte rnal.h	Branch	100%	Branch	50.0 ControllerTester2.2
6	GLFWWin _create.h	Branch	100%	Branch	50.0 ControllerTester2.2
7	state.h	Branch	100%	Branch	50.0 ControllerTester2.2
8	reduce.c	Branch	100%	Branch	50.0 ControllerTester2.2
9	size.h	Branch	100%	Branch	50.0 ControllerTester2.2

[별지1-별첨5] 소스코드 메트릭 측정 결과

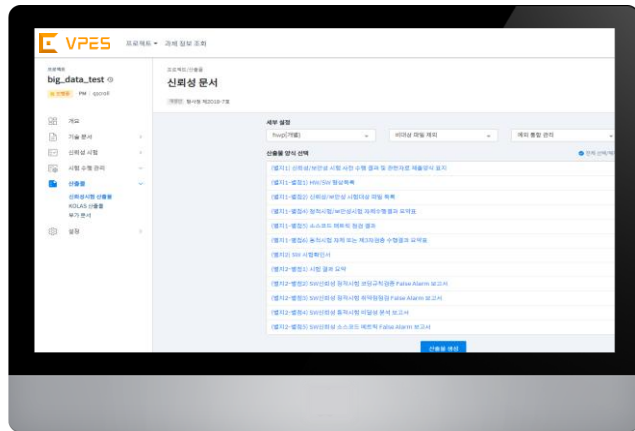
번호	항목	기준치	측정 결과	시험도구	비고
1	Cyclomatic Complexity	20	부적격 (위해 간수:54)	Codinspector 3.4	
2	Number of Call levels	6	부적격 (위해 간수:15)	Codinspector 3.4	
3	Number of Function parameters	8	부적격 (위해 간수:10)	Codinspector 3.4	
4	Number of Calling Functions	8	부적격 (위해 간수:10)	Codinspector 3.4	
5	Number of Called Functions	10	부적격 (위해 간수:41)	Codinspector 3.4	
6	Number of Executable Code Lines	200	부적격 (위해 간수:8)	Codinspector 3.4	

제표리소트 : 예외 부분 및 사유를 첨부하였는가?

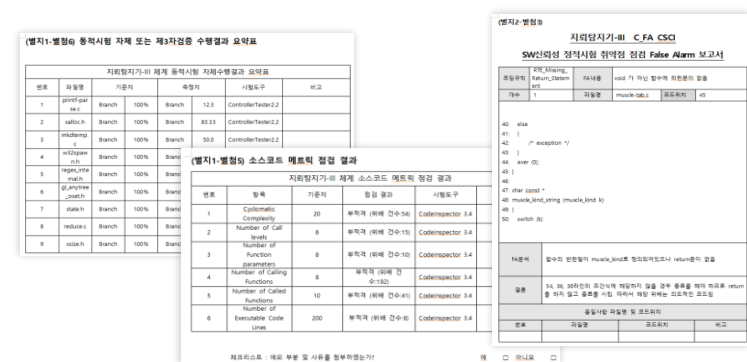
예 □ 아니오 □

산출물 자동 생성

- 신뢰성 시험 리포트 11종 산출물 다운로드 가능
- 시험 도구로부터 업로드된 데이터 기반의 산출물 한글/엑셀/워드 문서 자동 생성
: 방위사업청 매뉴얼 제2018-7호, 2020-1호, 2022-6호 「무기체계 SW 개발 및 관리 매뉴얼」의 신뢰성 시험 리포트 11종 지원
- 산출물 추가 기능 제공
: 소프트웨어 산출물 명세서 3장 제공(SPS), 소프트웨어 설계 명세서(SDD) 5장 제공, False Alarm 보고서 예외 개별 및 통합 관리 기능
- 전체 산출물 다운로드
: 한글, MS 워드, MS 엑셀 대응



신뢰성 시험 리포트 생성



신뢰성 시험 리포트 다운로드

산출물 생성 AI

- AI를 활용한 소프트웨어 설계 명세서(SDD) 자동완성
- SDD 5장 변수, 함수 내역표 자동완성
- Chat GPT, Gemini, Suresoft ALiRA, Local AI 내장 (* 별도 설치 필요)

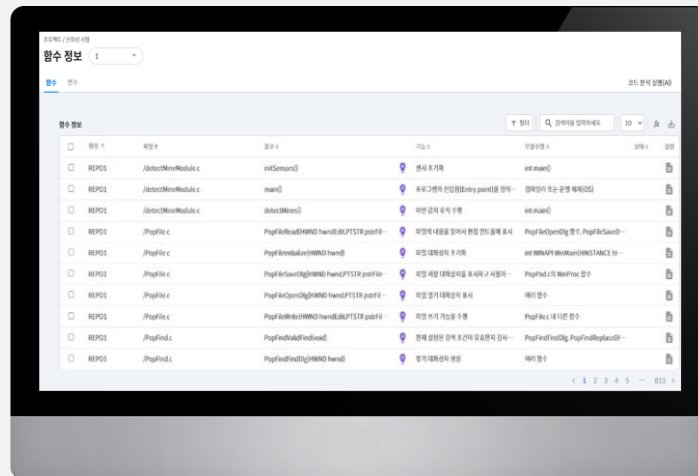
요구사항

설계

구현

시험

평가



페이지 추가

ChatGPT

Gemini

Suresoft ALiRA * 별도 설치 필요

표 4 detectMines() 함수 내역

기능	마인 감지 로직 수행		
모함수명	int main()	소스파일명	detectMineModule.c
입력	0	출력	bool
처 리			
평균 방식 감지: 함수는 먼저 모든 센서에서 얻은 값들의 합을 구하고, 그 합을 센서의 개수(‘NUM_SENSORS’)로 나누어 평균 값을 구합니다. 이 평균 값이 ‘THRESHOLD_HIGH’를 초과하는지 확인한 후, 초과한다면 ‘true’를 반환하여 지뢰 탐지를 알립니다. **개별 방식 감지:** 평균 값이 상한 임계값을 초과하지 않는 경우, 각 센서의 개별적인 값이 하한 임계값(‘THRESHOLD_LOW’)을 초과하는지 확인합니다. 이 중 하나라도 초과하는 경우, 역시 ‘true’를 반환하여 지뢰 탐지를 알립니다.			



한글



엑셀



워드

신뢰성 시험 상세 13종 보고서 다운로드 기능

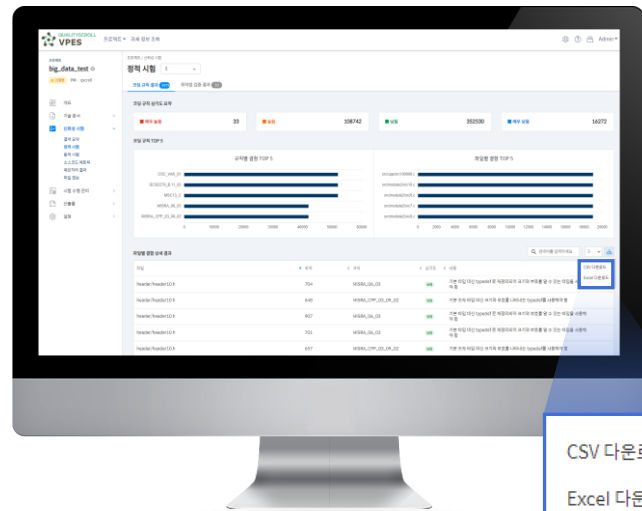
- 시험 도구로부터 업로드된 데이터 기반의 상세 시험 보고서 CSV/엑셀 문서 자동 생성

코딩 규칙 파일별 결함 상세 결과
취약점 검증 파일별 결함 상세 결과
동적 시험 파일별 코드 실행률
동적 시험 함수별 코드 실행률

정적 시험 예외 현황
동적 시험 예외 현황
소스코드 메트릭 예외 현황
소스코드 메트릭 상세 결과

프로젝트 파일 정보
문서 이력
일반 검증 결과
문서 정합성 검증 결과

문서 간 일치성 검증 결과



상세 시험 보고서 생성



파일명	함수	CC	NEL	NFP	NEF	NDF	NEL
adler32.c	adler32_combined(unsigned long, unsigned long, signed long)	2	1	2	0	0	12
adler32.c	adler32(unsigned long, const unsigned char *, unsigned int)	14	7	3	15	0	12
compress.c	compress(char *, unsigned int, unsigned char *)	1	0	0	0	0	0
compress.c	compress2(char *, unsigned int, unsigned char *, int)	1	0	0	0	0	0
blat.c	find_state(char *, struct link *)	miro_fun.c	93.75	62.5	25	-	X
blat.c	main()	Q_S.c	100	-	-	-	X
blat.c	mag_out(char *)	miro_fun.c	100	100	100	-	X
blat.c	construct()	miro_fun.c	41.34	62.5	50	-	X
blat.c	map_m_find(struct link *)	miro_fun.c	71.42	83.33	28.57	-	X
blat.c	miro_check(char *, struct link *, signed int, signed int)	miro_fun.c	100	100	100	-	X
blat.c	miro_map_gpy(char *, char *)	miro_fun.c	81.81	50	0	-	X
blat.c	miro_return(struct link *, struct link *, char *)	miro_fun.c	0	0	0	-	X
blat.c	miro_way_completion2(struct link *, struct link *, char *)	miro_fun.c	0	0	0	-	X
infoback.c	ad_int(int *)	miro_way_find(struct link *, char *)	0	0	0	-	X
infoback.c	infoback2()	Q_make_add(struct link *, char *)	55.4	47.43	7	-	X
infoback.c	infoback3()	miro_fun.c	53.57	68.75	50	-	X
infoback.c	infoback4()	Q_S.c	100	-	-	-	X
infoback.c	infoback5()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback6()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback7()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback8()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback9()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback10()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback11()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback12()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback13()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback14()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback15()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback16()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback17()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback18()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback19()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback20()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback21()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback22()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback23()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback24()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback25()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback26()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback27()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback28()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback29()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback30()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback31()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback32()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback33()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback34()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback35()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback36()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback37()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback38()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback39()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback40()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback41()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback42()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback43()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback44()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback45()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback46()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback47()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback48()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback49()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback50()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback51()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback52()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback53()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback54()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback55()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback56()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback57()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback58()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback59()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback60()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback61()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback62()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback63()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback64()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback65()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback66()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback67()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback68()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback69()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback70()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback71()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback72()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback73()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback74()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback75()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback76()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback77()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback78()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback79()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback80()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback81()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback82()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback83()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback84()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback85()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback86()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback87()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback88()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback89()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback90()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback91()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback92()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback93()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback94()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback95()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback96()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback97()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback98()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback99()	Q_S.c	62.16	46.15	35.71	-	X
infoback.c	infoback100()	Q_S.c	62.16	46.15	35.71	-	X

상세 시험 보고서 다운로드

요구사항

설계

구현

시험

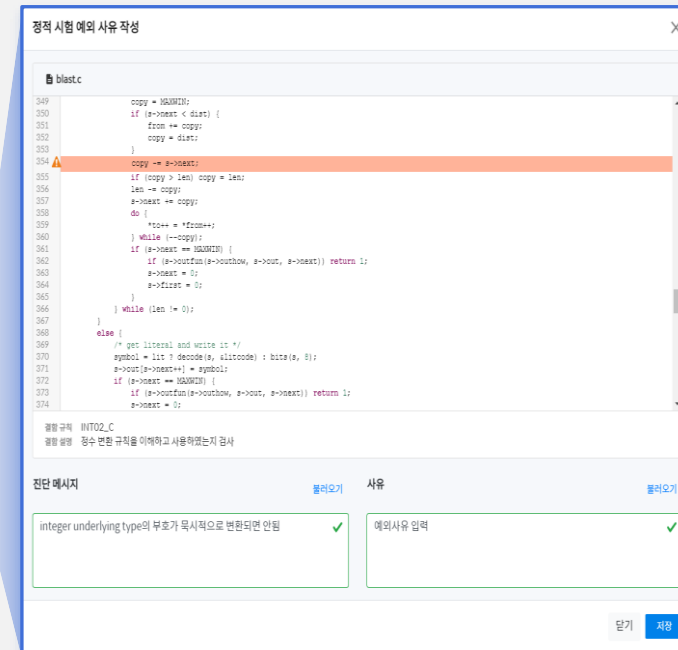
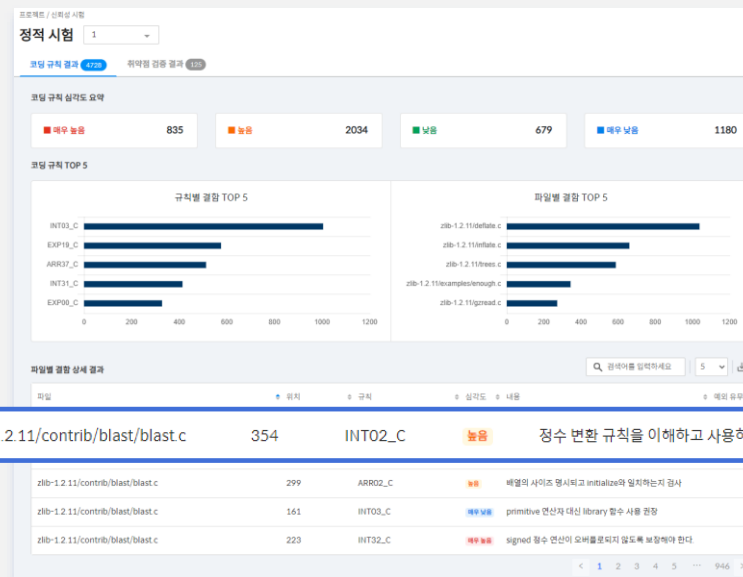
평가

시험 상세 결과 및 예외 처리 제공

- 시험 도구에서 제공하는 수준의 상세 결과를 제공하여 조직 간 결함 공유가 가능
- 개별적인 시험 도구를 실행하지 않고도 필요한 시험 결과를 확인할 수 있도록 유도
- 동일 예외에 대해서 불러오기 기능을 이용하여 간편하게 예외 작성 가능

소스코드 레벨의 결함 설명과 진단 메시지 제공 (동적 시험은 커버리지 정보 제공)

* 결함 사유를 입력하도록 하여 False Alarm 및 동적 시험 미달성 분석 보고서를 자동으로 생성



간편하고 유용한 코드 추적기능 제공

- VPES에서 관리되는 소스코드의 함수, 전역변수, 구조체 등의 코드 추적 지원
- 코드 간 호출 관계를 쉽게 추적하고, 시험 결과를 쉽게 파악하도록 유도
- 변수, 함수명 하이라이팅 지원으로 코드 구조 파악 용이

소스코드 변수 및 함수 추적 제공

정적 시험 예외 사유 작성

miro_fun.c

```

185     if(end == 2){printf("-----실패 찾기예 성공하였습니다.-----\n");break;}
186   }
187   else if(end==2||(1==(end=miro_check(moving,node,x+1,y-1))&&x>0&&y>0&&y<my&&x<mx))
188   {
189     {
190     else if(end==2||(1==(end=miro_check(moving,node,x+1,y))&&x>0&&y>0&&y<my&&x<mx))
191     {
192       node= node->prev;
193       x = x+1;
194       if(end == 2){printf("-----실패 찾기예 성공하였습니다.-----\n");break;}
195     }
196     else if(end == 2||(1==(end=miro_check(moving,node,x+1,y+1))&&x>0&&y>0&&y<my&&x<mx))
197     {
198       node= node->prev;
199       x = x+1;
200       if(end == 2){printf("-----실패 찾기예 성공하였습니다.-----\n");break;}
201     }
202     else if(end == 2||(1==(end=miro_check(moving,node,x+1,y-1))&&x>0&&y>0&&y<my&&x<mx))
203     {
204       node= node->prev;
205       x = x+1;
206       y = y-1;
207       if(end == 2){printf("-----실패 찾기예 성공하였습니다.-----\n");break;}
208     }
209   }
  
```

변수/함수명	타입	범위	종류
BAEVC_Model_125\Sources\Additional_Components\GSD\SeaWindXX11\extensions\miro_fun.c	50	FUNCTION	

결함 규칙 MISRA_C_2012_18_04
 결함 설명 +, -, *, / 연산자는 포인터 타입의 표현식에 적용 금지

전단 메시지 블러오기 사유 블러오기

포인터 타입의 수식에 연산자 +가 사용되었음	예외사유 작성
--------------------------	---------

[닫기] [저장]

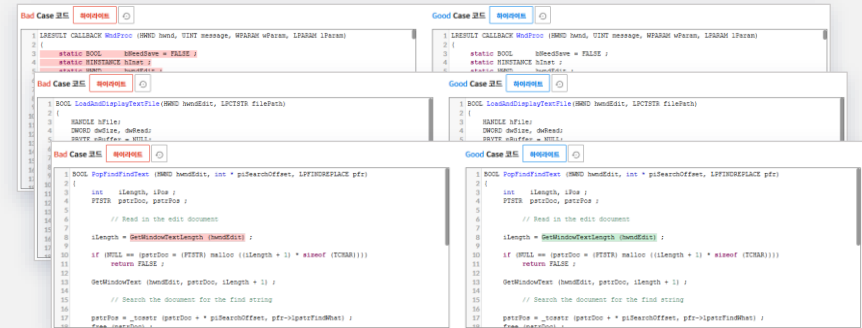
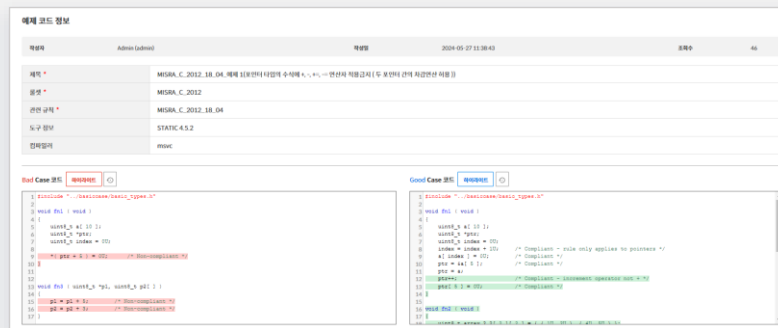
```

HOME > RTE>> 1) BAEVC_Model_125\Sources\Additional_Components\GSD\SeaWindXX11\extensions\miro_fun.c
45  }
46  }
47  //-----
48  //-----
49  //-----
50  //-----
51  //-----
52  //-----
53  //-----
54  //-----
55  //-----
56  //-----
57  //-----
58  //-----
59  //-----
60  //-----
61  //-----
62  //-----
63  //-----
64  //-----
65  //-----
66  //-----
67  //-----
68  //-----
69  //-----
70  //-----
71  //-----
72  //-----
73  //-----
74  //-----
75  //-----
76  //-----
77  //-----
78  //-----
79  //-----
80  //-----
81  //-----
82  //-----
83  //-----
84  //-----
85  //-----
  
```

개선 사례 관리 기능

- 시험 활동 개선 사례 관리 할 수 있는 기능 내재
- 개선이 필요한 사례를 관리 되는 개선 사례와 자동 매핑
- 가장 적합한 개선 사례 AI 추천

개선 사례 관리 및 매핑



형상 ▾	파일 📄	위치 📍	규칙 ⚙️	심각도 📊	내용 📝	예외 사유 📝	위배 무시 ⚙️
REPO1	mine_detector.c	3	MISRA_C_2012_21_10	높음	time.h는 include가 금지된 파일임		
REPO1	mine_detector.c		MISRA_C_2012_21_10	높음	time는 사용이 금지된 함수임		
REPO1	mine_detector.c		MISRA_C_2012_21_06	높음	printf는 사용이 금지된 함수임		
REPO1	mine_detector.c		MISRA_C_2012_21_06	높음	printf는 사용이 금지된 함수임		
REPO1	mine_detector.c		MISRA_C_2012_21_06	높음	printf는 사용이 금지된 함수임		

AI 추천
해결사례 3
Knowledge HUB

해결사례 1
해결사례 2
해결사례 3
해결사례 4
해결사례 5
해결사례 6
해결사례 7

요구사항

설계

구현

시험

평가

다양한 시험 도구와 연동하여 결과 수집

- 4종의 자사 도구와 편리한 연동
- 플러그인 방식의 쉬운 검증 결과 업로드



※ STATIC의 구 버전(CodeInspector3.7, Sniper1.4 이상) 지원 포함

오프라인 환경에서도 시험 도구 결과 수집 가능

- 데이터 내보내기를 통해 VPES에서 검증 결과 업로드 가능

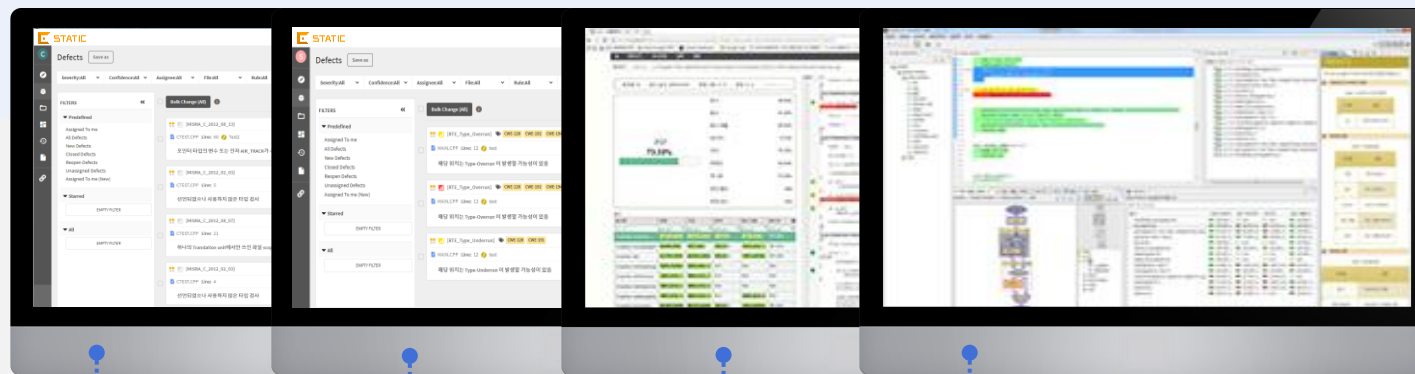
요구 사항

설계

구현

시험

평가



코딩규칙
STATIC

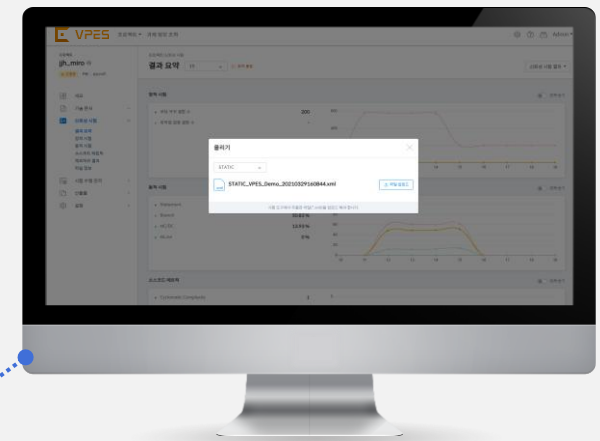
취약점 점검
STATIC

커버리지
COVER

단위/시스템 시험
Controller Tester

데이터 내보내기 (.xml)

VPES



VPES 검증 결과 업로드

요구사항

설계

구현

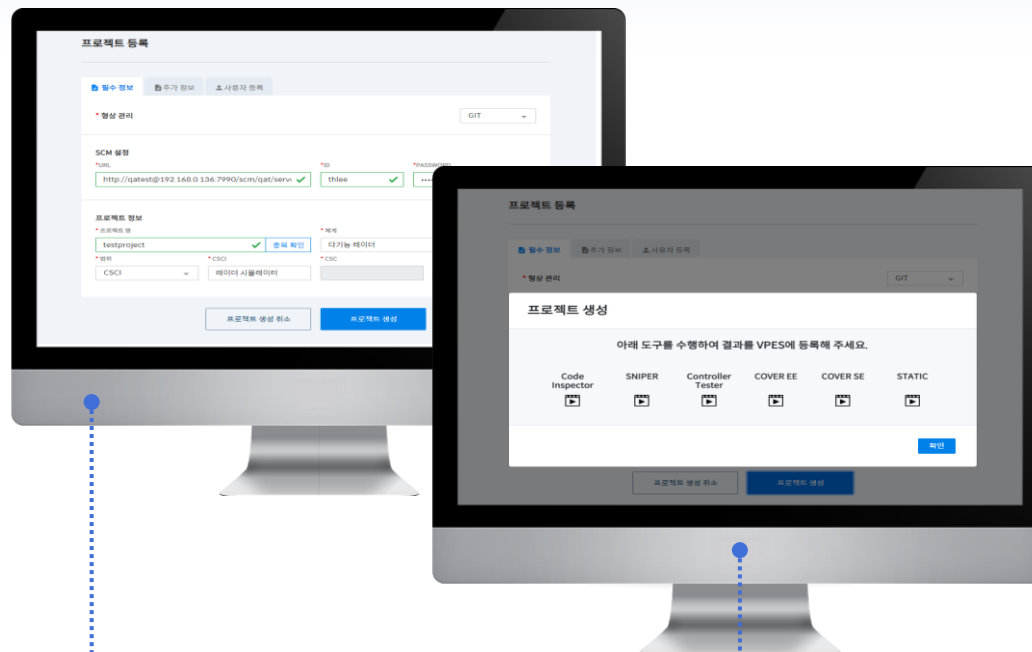
시험

평가

시험 도구 연동 및 소스코드 메트릭 가이드 제공

- 프로젝트 생성 후 다음 시험 활동 단계 동영상 매뉴얼 제공
- 시험 설정 가이드, 기술 문서 검사 가이드 동영상 매뉴얼 제공
- 소스코드 메트릭 수정을 위한 코드 리팩토링 기법 제공

프로세스 지원



프로젝트 생성 시
다음 시험 활동 안내

수정 가이드 지원



소스코드 메트릭
리팩토링 기법 제공

지원 환경

구분		세부 사항
지원 언어		C, C++, C#, Java
지원 환경	OS	Windows 7(64bit) 이상, CentOS 7 이상, Ubuntu 16 이상
	브라우저	Edge, Chrome, Internet Explorer11 이상
	언어	한국어, 영어
	도구	- STATIC(구 Code Inspector, SNIPER 포함) - CT - COVER

하드웨어 권장 사양

구분	CPU	RAM	HDD	비 고
Server (Local AI 활용 최소사양)	Intel Core i7-14700K 3.7GHz이상	32GB이상	30GB 이상	사용LLM : Qwen3-128k-30BA3B- NEO-MAX-PLUS-IQ4_XS (12Gb)
Server	Quad Core 2GHz 이상	8GB 이상	30GB 이상 여유공간 (소스 용량에 따라 증가)	



H사 적용 사례 1

01

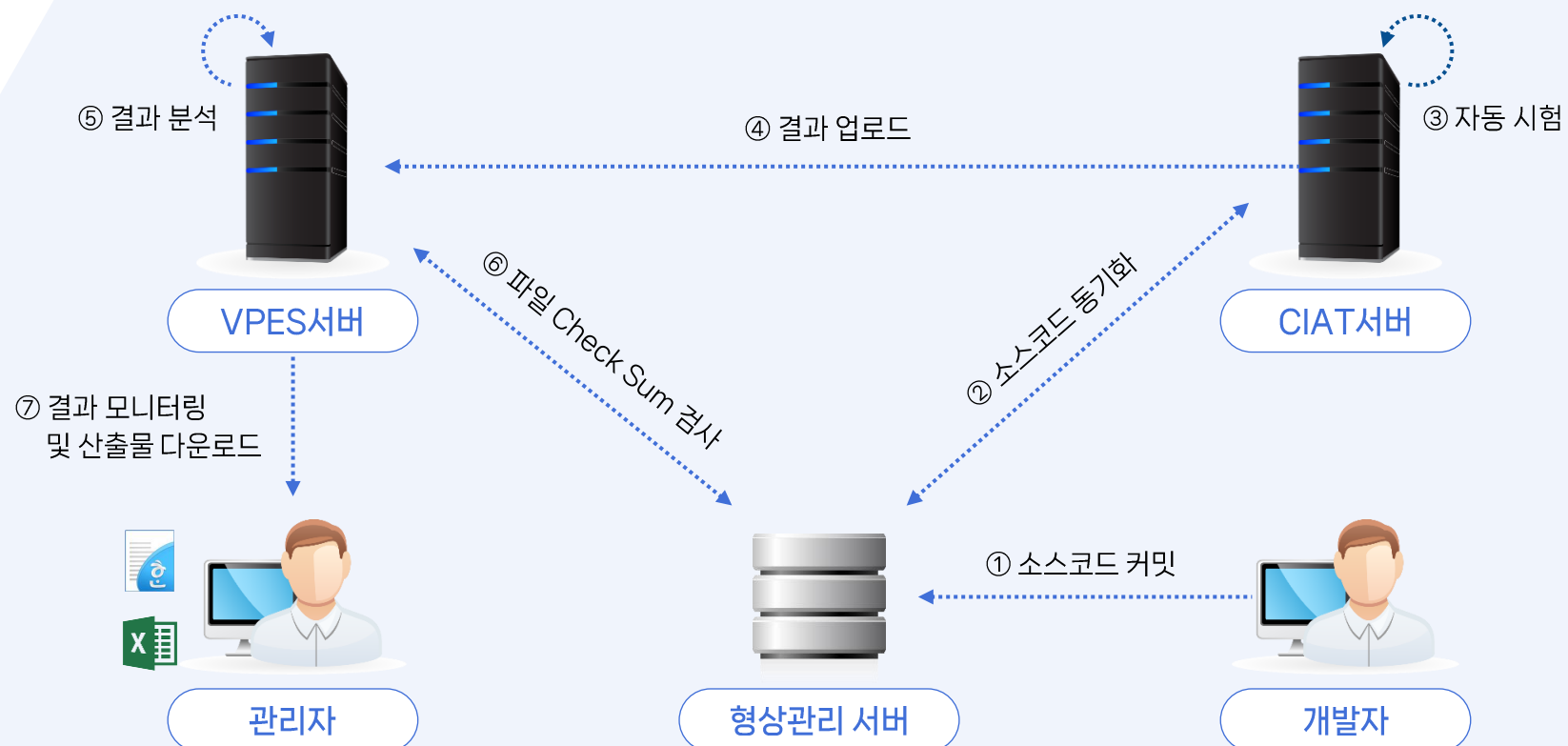
CIAT과 연동하여 자동으로 신뢰성 시험 수행 및 시험 결과 모니터링

02

소스코드 커밋으로 자동 수행 (코딩 규칙 검증, 취약점 검증, 소스코드 메트릭 검증, 단위 시험)

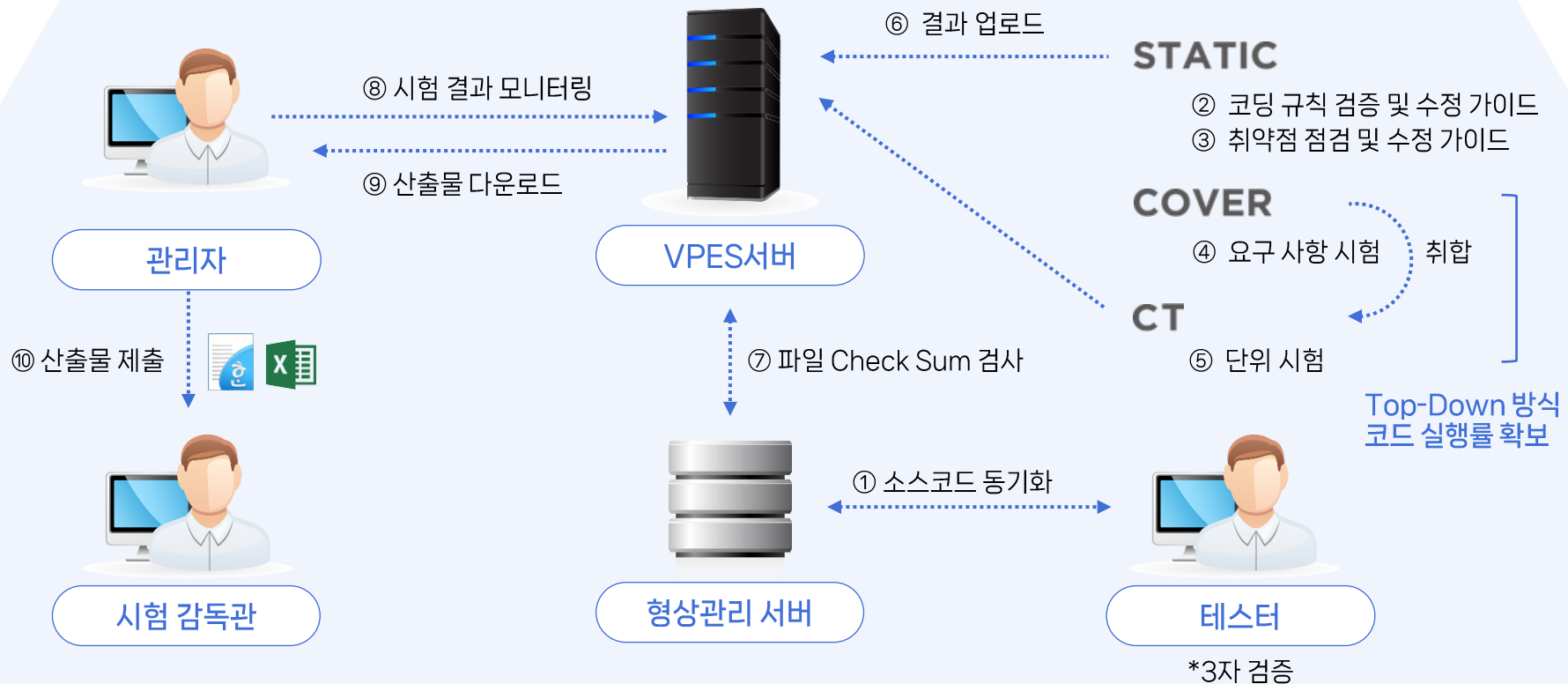
03

VPES에서 자동으로 생성한 산출물을 다운로드



H사 적용 사례 2

- 01 3자 검증을 통한 신뢰성 시험 수행
- 02 시험 도구 자체 수행 및 결과 자동 업로드
- 03 VPES에서 제공하는 시험 결과와 산출물을 통한 실사 대응

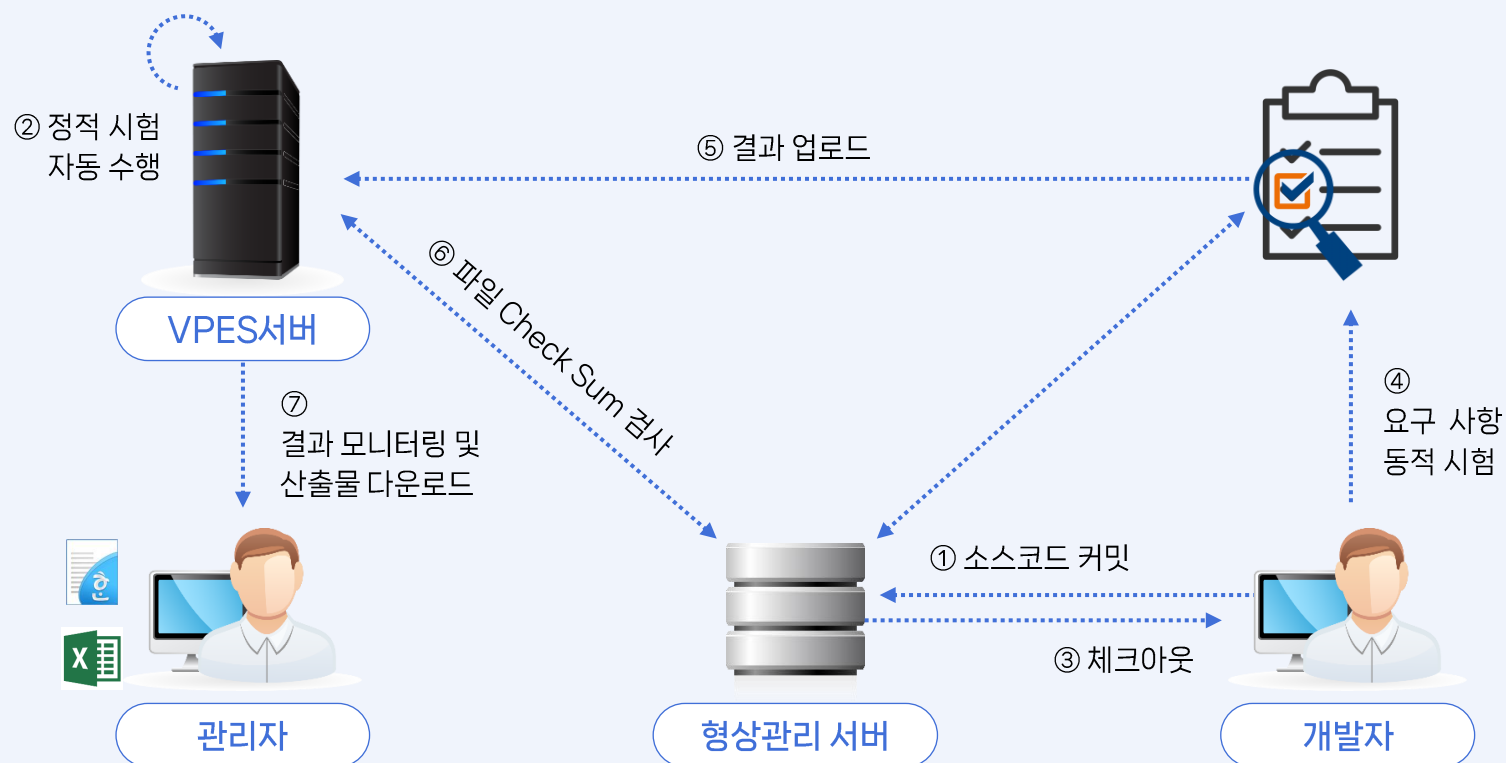


A 연구소 적용 사례

01 VPES 자체 자동 시험 기능으로 정적 검증 신뢰성 시험 자동화 및 시험 결과 모니터링

02 동적 시험 자체 수행 및 결과 수동 업로드

03 VPES에서 자동으로 생성한 산출물을 다운로드



VPES 도입으로 인한 공수 절감효과

ROI 계산하러 가기



수작업으로 신뢰성 시험 준비 시

$$\begin{aligned} & \text{시험 대상 수} \times \{ (\text{산출물 작성 소요 기간} \times \text{산출물 수정 횟수}) \\ & + \text{기술 문서 준비 기간} + (\text{LOC}/10000/24 \times \text{시험 도구 수행 횟수}) \\ & + (\text{기술 문서 최초 수정 기간} \times 2) \} \times \text{연평균 실사 횟수} \end{aligned}$$

약
125.7MM
소요

약 86% 작업량 감소

VPES 도입 시

$$\begin{aligned} & \text{시험 대상 수} \times \{ (\text{산출물 작성 소요 기간}/4 \times \text{산출물 수정 횟수}) \\ & + (\text{기술 문서 준비 기간}/30) + (\text{LOC}/10000/24 \times \text{시험 도구 수행 횟수} \times 0.33) \\ & + (\text{기술 문서 최초 수정 기간}/10) \} \times \text{연평균 실사 횟수} \end{aligned}$$

약
17.6MM
소요

약 17.6MM 소요

※ 예시 기준

✓ 시험 대상	30	✓ 산출물 작성 소요기간	7 MD
✓ 기술 문서 준비	15 MD	✓ 기술문서 최초 수정기간	5 MD
✓ 시험 도구 수행	40회	✓ 소스코드 규모 (LOC)	30,000 라인
✓ MM 단가	1,500만원		



Thank You