SURESOFT

Intelligence for Safe World

# CT

## 코드 동적 검증 도구

SURESOFT

# TABLE OF CONTENTS

## 차별점

### Coverage View

함수, 테스트별 커버리지 정보 및 MC/DC 가이드 제공

### Rapid Coverage Achieve

시스템 커버리지 측정(COVER 제품) 후 커버되지 않은 부분만 CT로 테스트

### Scalability

크고 복잡한 SW 및 대용량 테스트데이터(1억 개) 시험 지원

### Team Testing

테스트 협업 환경에 최적화된 기능 제공

## 강점

**국제인증**

- Cert. from TÜV
- Tool Qualification 제공

**타깃 테스팅**

- 간편한 타깃 환경 설정
- 버튼 한 번에 타깃에서 테스트 실행

**유연성**

- 도메인 요구 사항에 빠르게 대응

**적용사례**

- 100개 이상의 고객사에 적용
- 자동차, 국방, 철도, 원자력, 의료, 로봇

## 기대 효과

### CT

**01 품질 향상**
SW 품질 향상 및 고객의 신뢰 확보

**02 비용 절감**
테스팅 및 유지보수 비용 절감

# Mission Critical 도메인의 안전성 및 신뢰성 인증 필요

- 테스트를 통해 구조적인 커버리지 목표를 달성해야 함
- 실제 타깃 환경에서 테스트가 필요함


Table 12 — Structural coverage metrics at the software unit level

| | | ASIL | | | |
|---|---|---|---|---|---|
| | | A | B | C | D |
| 1a | Statement coverage | ++ | ++ | + | + |
| 1b | Branch coverage | + | ++ | ++ | ++ |
| 1c | MC/DC (Modified Condition/Decision Coverage) | + | + | + | ++ |

01

Table 15 — Structural coverage metrics at the software architectural level

| | | ASIL | | | |
|---|---|---|---|---|---|
| | | A | B | C | D |
| 1a | Function coverage [a] | + | + | ++ | ++ |
| 1b | Call coverage [b] | + | + | ++ | ++ |

[a] Method 1a refers to the percentage of executed software functions. This evidence can be achieved by an appropriate software integration strategy.

[b] Method 1b refers to the percentage of executed software function calls.

02

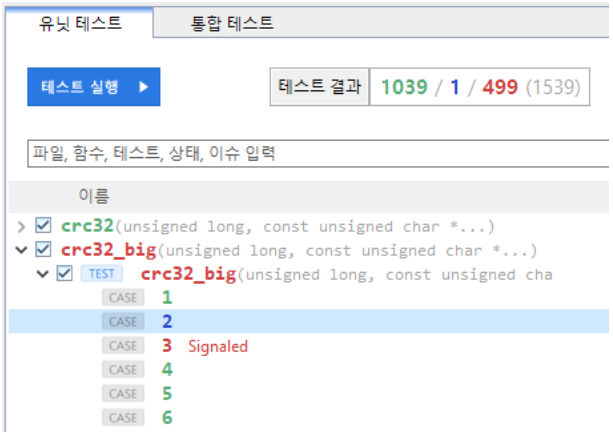
**ISO 26262의 SW 파트**

# 단위/통합 테스트 수행 및 커버리지 측정

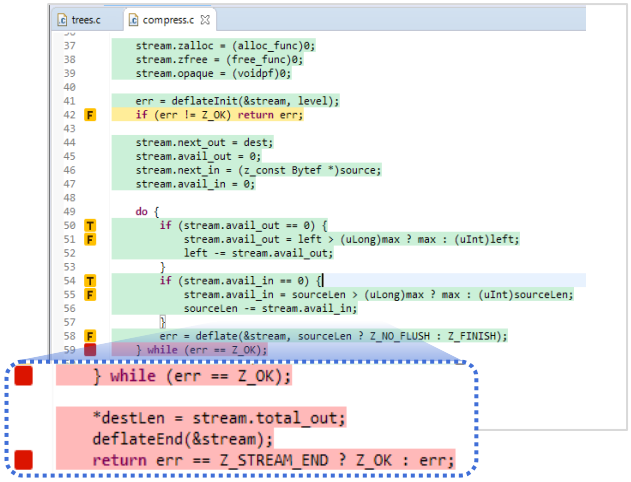
- 테스트 자동 생성 및 커스터마이징 가능
- 문장, 분기, 함수, 함수 호출 커버리지 및 MC/DC 측정 가능

## 테스트 결과

성공
실패
오류



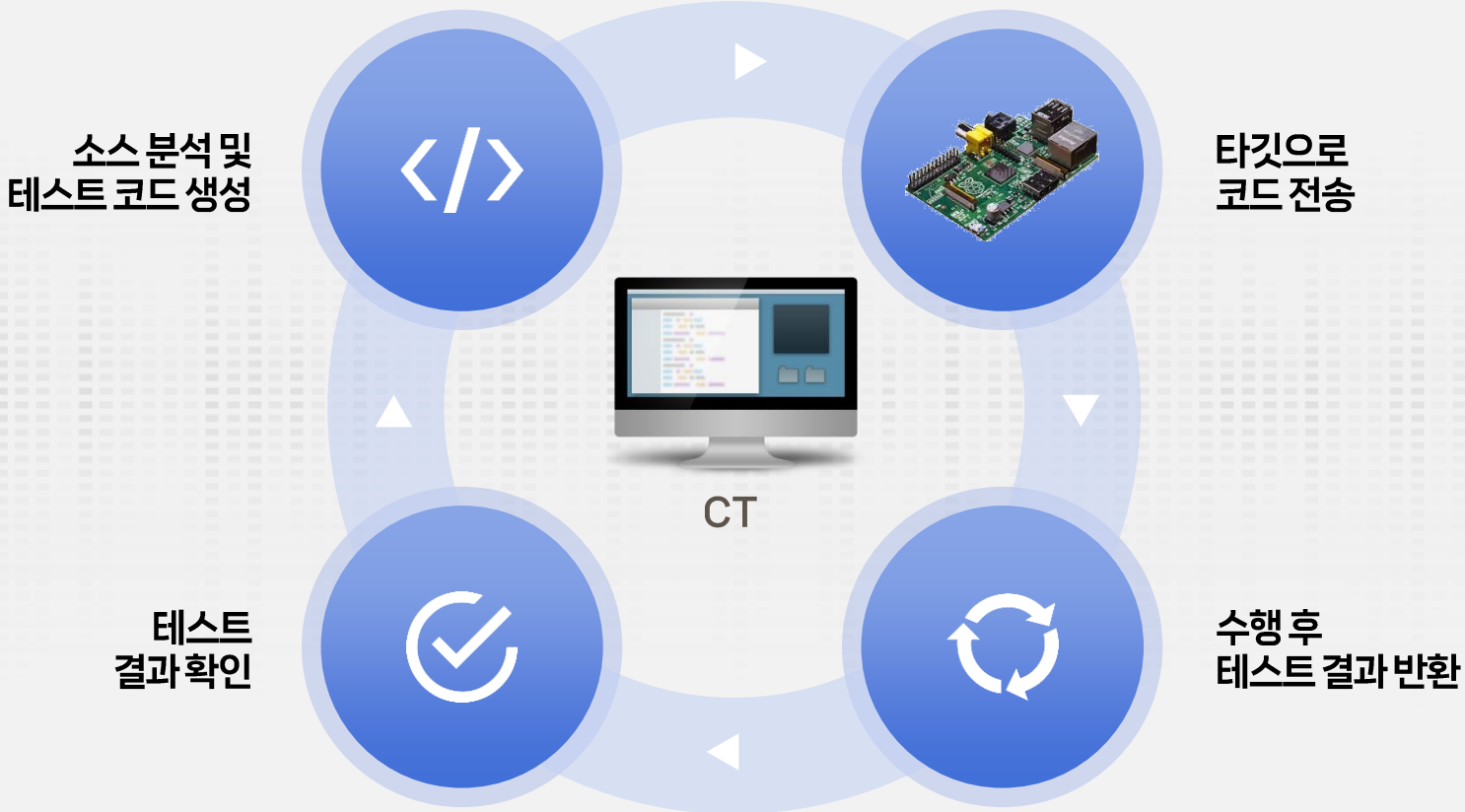

## 커버되지 않은 소스 코드 확인




## 전체 커버리지 및 함수별 커버리지

# 실제 타깃 환경 테스트

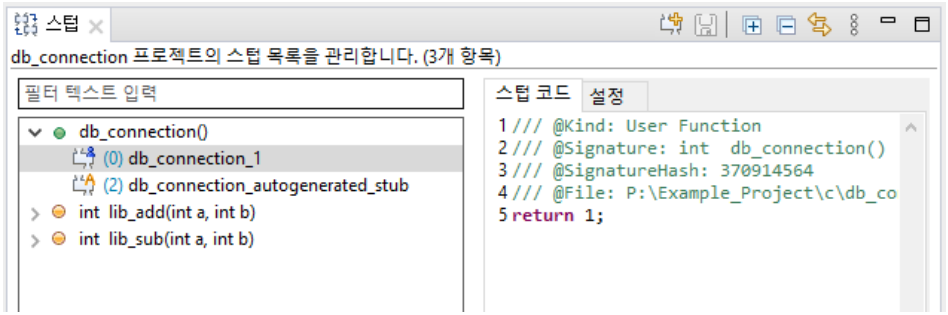
- 버튼 한 번에 실제 타깃에서 테스트 실행 및 결과 확인 가능(디버거 연동)
- 다양한 환경 구성 및 통신 방식(Serial/Ethernet/JTAG) 지원



# 스텁 기능

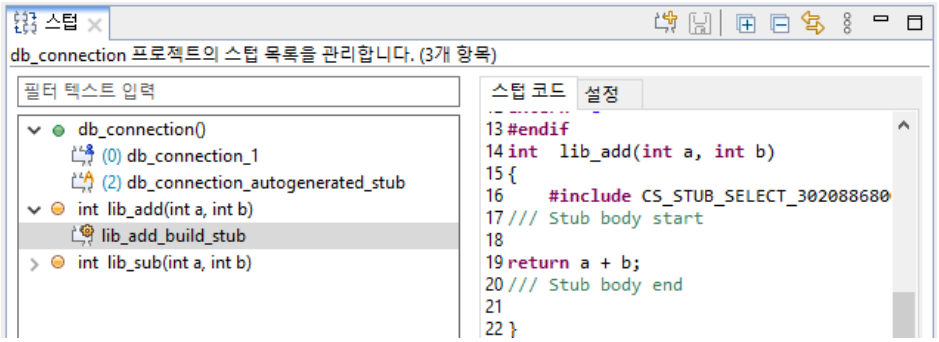
- 제어하기 어려운 원본 함수를 스텁(대역 함수)으로 대체하는 기능

## 원본 함수 대신 스텁으로 원하는 값을 반환하도록 테스트




Ex) db_connection 함수에서 1을 반환하려면 드라이버 설치 후 DB 연결 필요
▶ 스텁 기능을 사용하여 쉽게 1을 반환하도록 설정

## 정의가 없는 함수(라이브러리 등)의 스텁 자동 생성




Ex) lib_add 함수는 소스 코드에 정의가 없고 .lib 파일 링크 필요
▶ .lib 파일이 없을 때 스텁이 자동 생성되어 원하는 값을 반환하도록 할 수 있음

함수 타입

| 타입 | 설명 |
|---|---|
| 함수 | 정의가 없는 함수 |
| 함수 | 정의가 있는 함수 |

스텁 타입

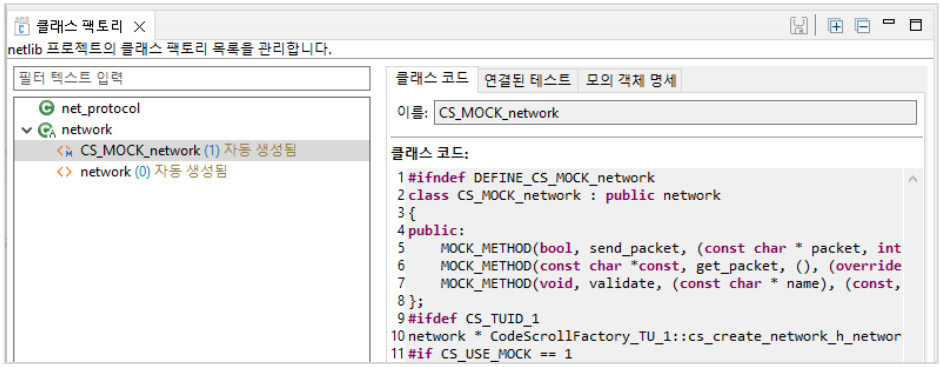
| 타입 | 설명 |
|---|---|
| 사용자 스텁 | 사용자 정의 스텁 |
| 빌드 스텁 | 자동 생성된 스텁 |

```
1
2   int lib_add(int a, int b);
3   int lib_sub(int a, int b);
4
5   int db_connection() {
6       // db connection
7
8
9
10
11  }
12
13  int func(int c1, int c2, int c3) {
14      if(db_connection()) {
15          if(c1 && c2 || c3) {
16              c1 = lib_add(c1, c2);
17              c2 = lib_sub(c2, c3);
18          }else {
19              return 0;
20          }
21      }
22      return c1 + c2;
23  }
24
```

# 모의 객체(Mock) 기능

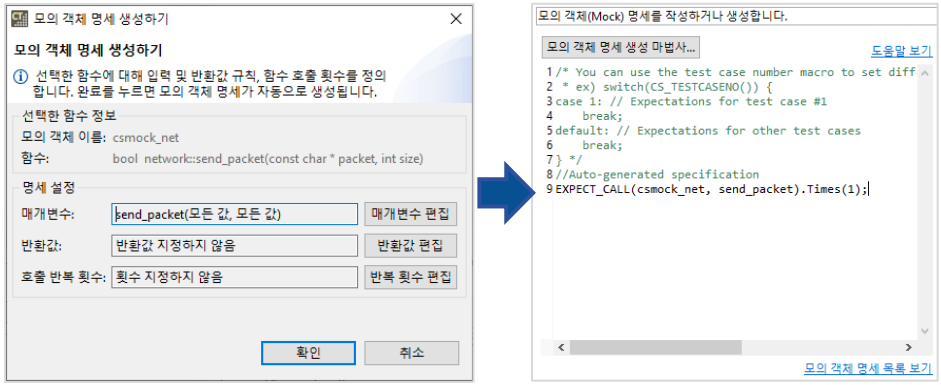
- C++ 테스트를 할 때 객체 종속성을 빠르고 안정적으로 대체할 수 있는 기능

## 테스트 대상의 객체 의존성 자동 식별 및 코드 생성




Ex) 추상 클래스에 대해 자동으로 생성된 Mock 객체 생성 코드

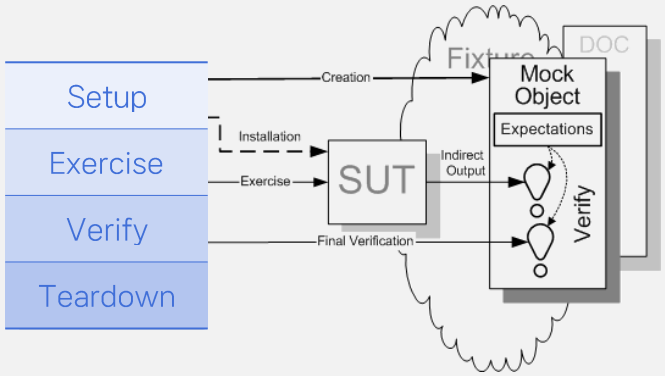
## GUI를 통한 모의 객체(Mock) 명세 입력




Ex) Google Mock 매크로 형태로 객체에 대한 기대(Expectation) 입력 가능

## 행동 검증(Behavior Verification)

테스트 대상의 간접 출력(Indirect Output)에 대한 검증




```
Mock function called more times than expected - returning
default value.
    Function call: send_packet(0000022FF9240004 pointing to
" ", 1)
          Returns: false
        Expected: to be never called
          Actual: called once - over-saturated and active
```

Ex) 0번 호출되도록 명세를 작성한 send_packet 함수가 한 번 호출되어 실패

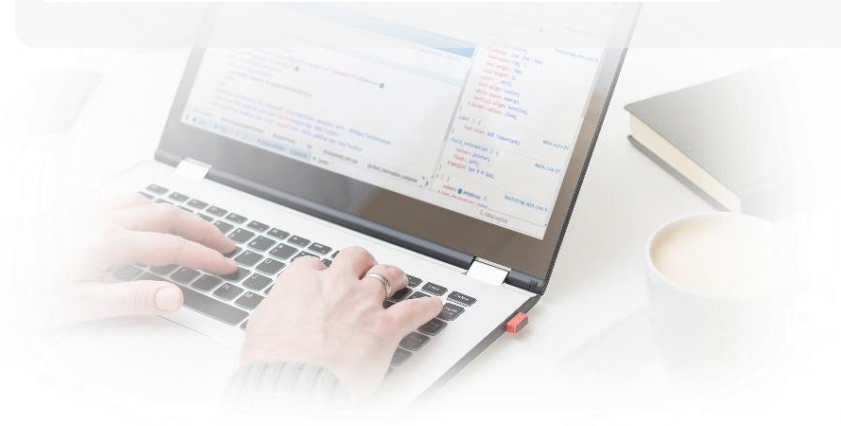
# 제어 흐름 그래프(Control Flow Graph)

- 소스 코드 이해를 도와주는 제어 흐름 그래프 제공
- 소스 코드와 연동하여 커버리지 결과 확인

## 소스 코드

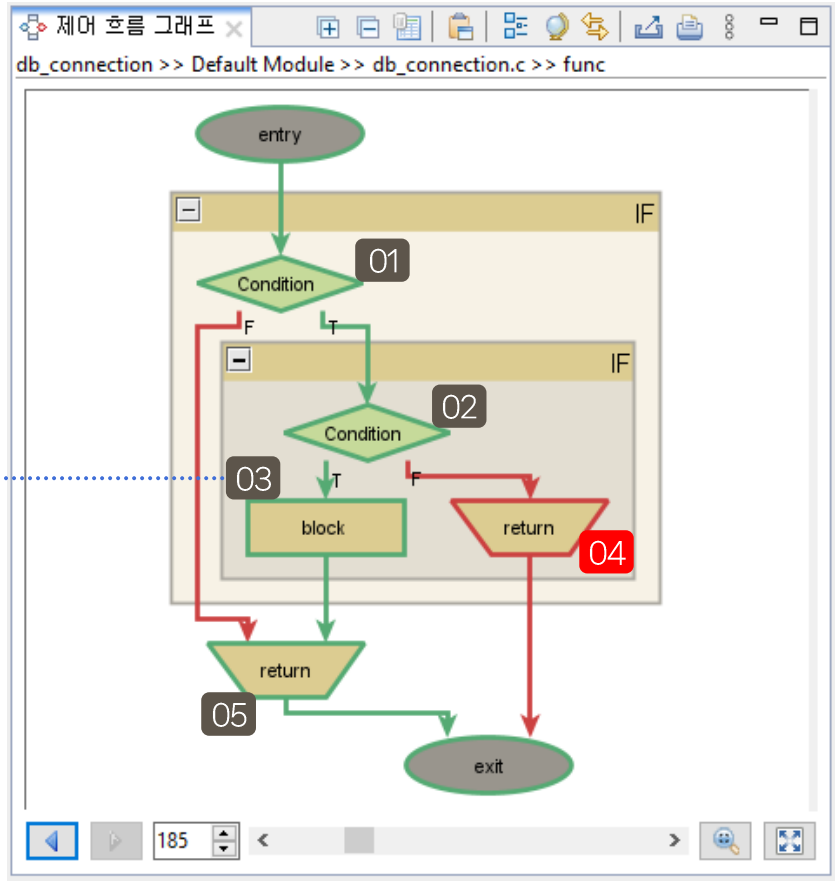
```
int func(int c1, int c2, int c3) {
    if(db_connection()) {
        if(c1 && c2 || c3) {
            c1 = lib_add(c1, c2);
            c2 = lib_sub(c2, c3);
        }else {
            return 0;
        }
    }
    return c1 + c2;
}
```

01 02 03 04 05



## 제어 흐름 그래프

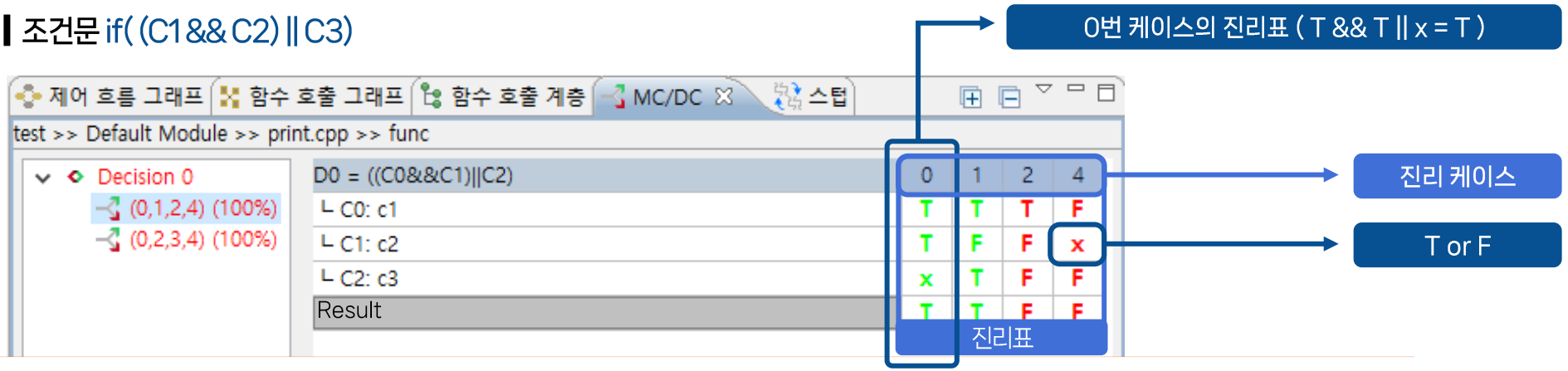
커버됨 커버되지 않음



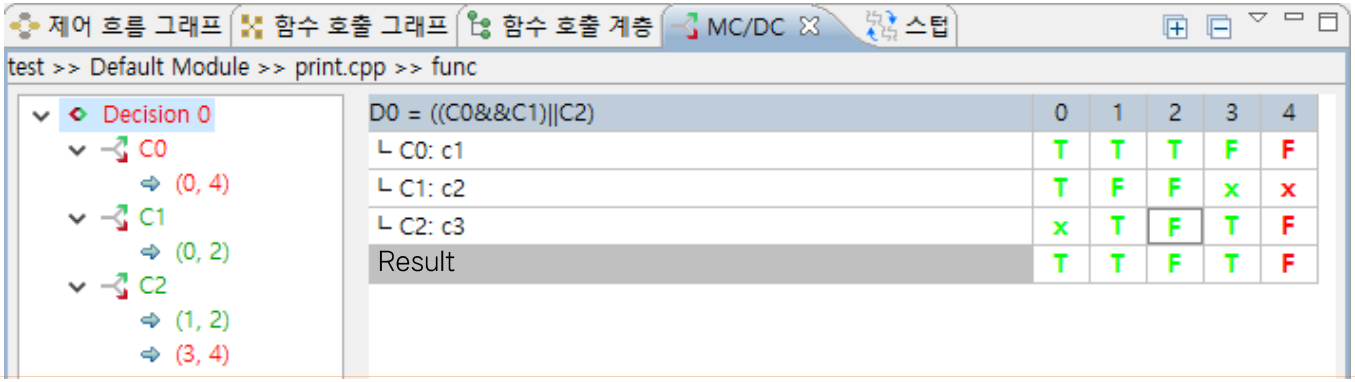
# MC/DC

- 도메인에서 요구하는 MC/DC 목표 달성을 위한 가이드 제공

## 조건문 if( (C1 && C2) || C3)




MC/DC n%를 만족하는 조합을 알려주는 기능(n값은 옵션에서 설정)
(Ex. 진리 케이스 0, 1, 2, 4만 커버하면 조건 C0, C1, C2를 100% 만족함)




각 조건(C0, C1, C2)을 만족하는 pair와 그 커버 여부를 알려주는 기능
(Ex. 4번 진리 케이스가 커버되지 않았기 때문에 C0는 커버되지 않음/0, 2번 진리케이스가 커버되었기 때문에 C1은 커버됨)

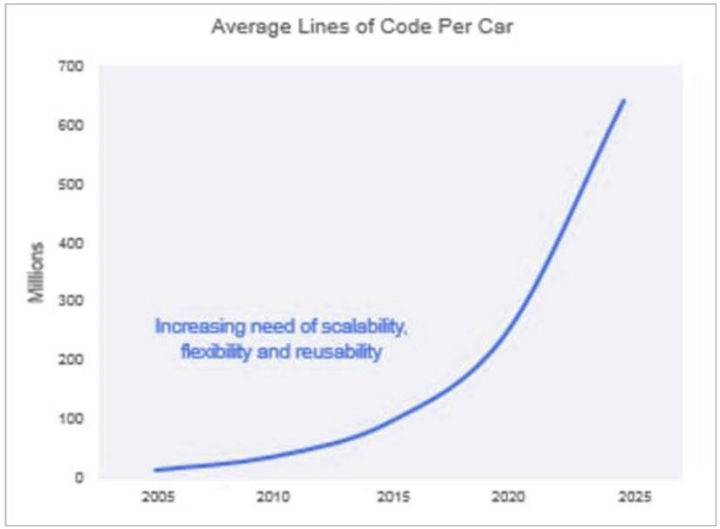
# 확장성(Scalability)

- 크고 복잡한 SW 및 대용량 테스트데이터 시험 가능

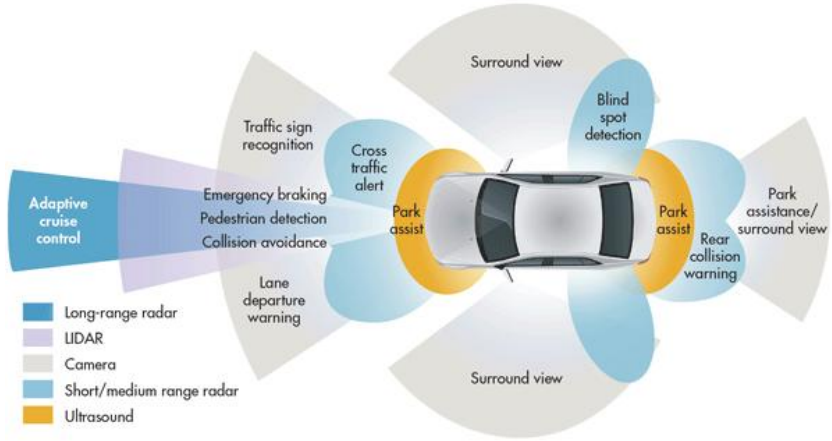
## 시험 대상 SW의 크기는 계속 증가

Tomorrow's Vehicle

6X more lines of code




## 대용량 테스트데이터 시험 필요(예: ADAS SW)




## CT 지원 가능 범위

01 하나의 프로젝트에 5,000개 이상의 함수 지원

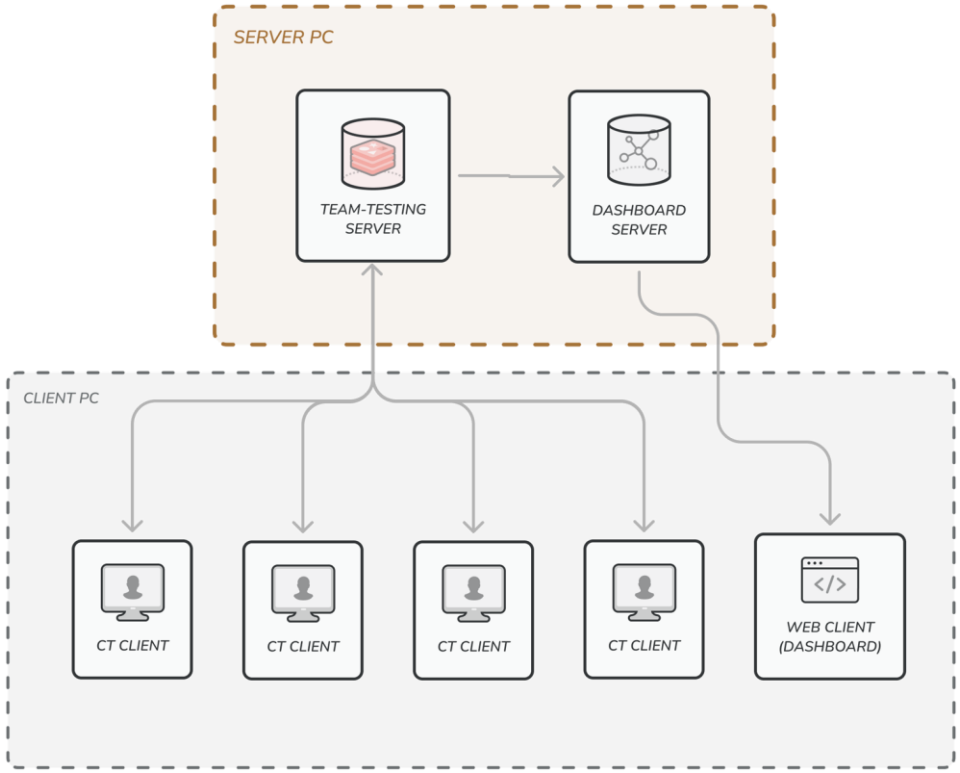
02 하나의 함수(테스트)에 10,000개 이상의 입/출력 변수 지원

03 하나의 함수(테스트)에 1,000,000개 이상의 테스트 케이스 지원

# 팀 테스팅 기능

- 테스트 협업 환경에 최적화된 기능을 제공하여 생산성 향상

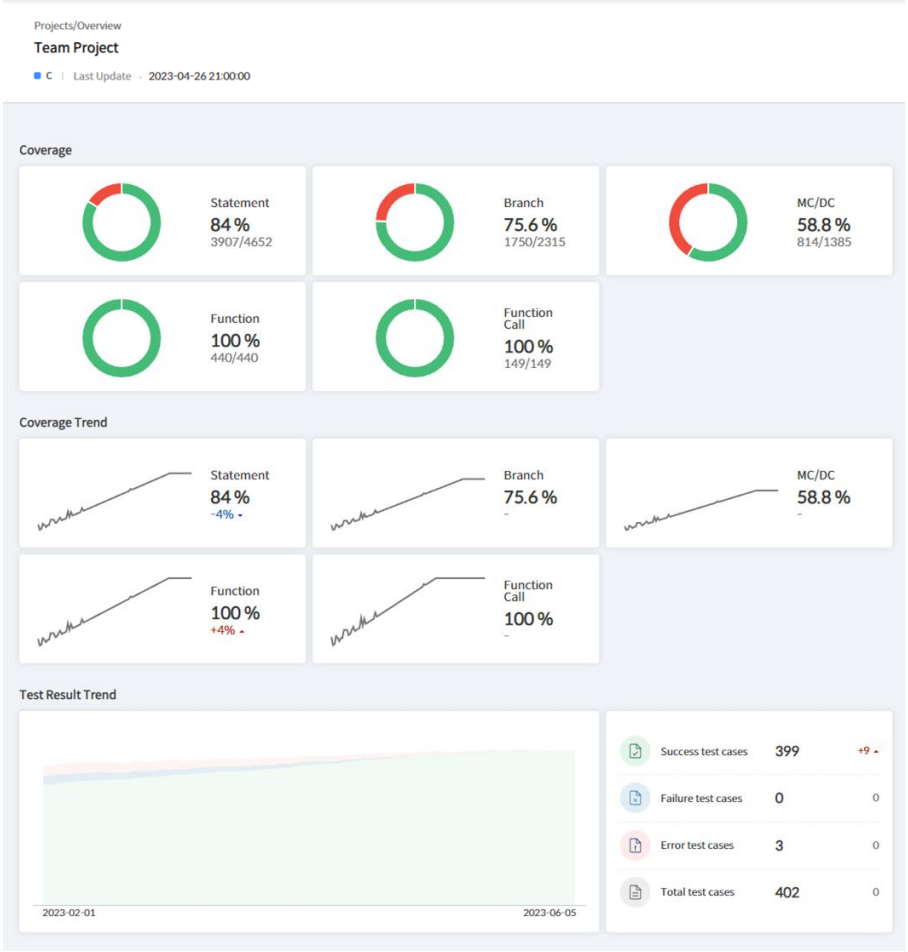
## 팀 테스팅 서버를 통해 테스트 환경 원클릭 공유




**테스트 진행 중 변경 사항 자동 공유**

테스트 대상 소스 코드 또는 구성이 변경된 경우에 다른 클라이언트에게 변경 사항이 자동으로 공유됨

## 대시보드에서 전체 진행 상황 실시간 확인(자동 취합)

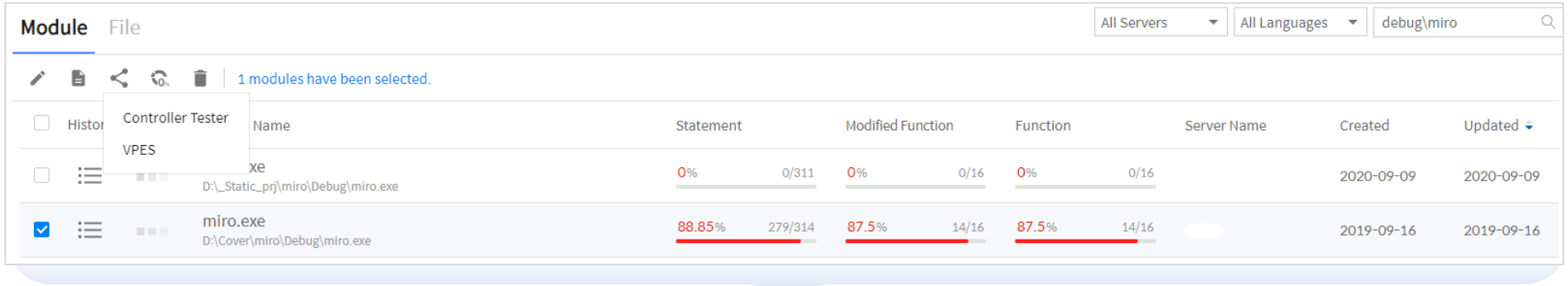


# COVER 제품과 커버리지 공유

- Top-Down 방식으로 빠르게 커버리지 목표 달성

## COVER로 시스템 테스트를 수행하며 커버리지 측정

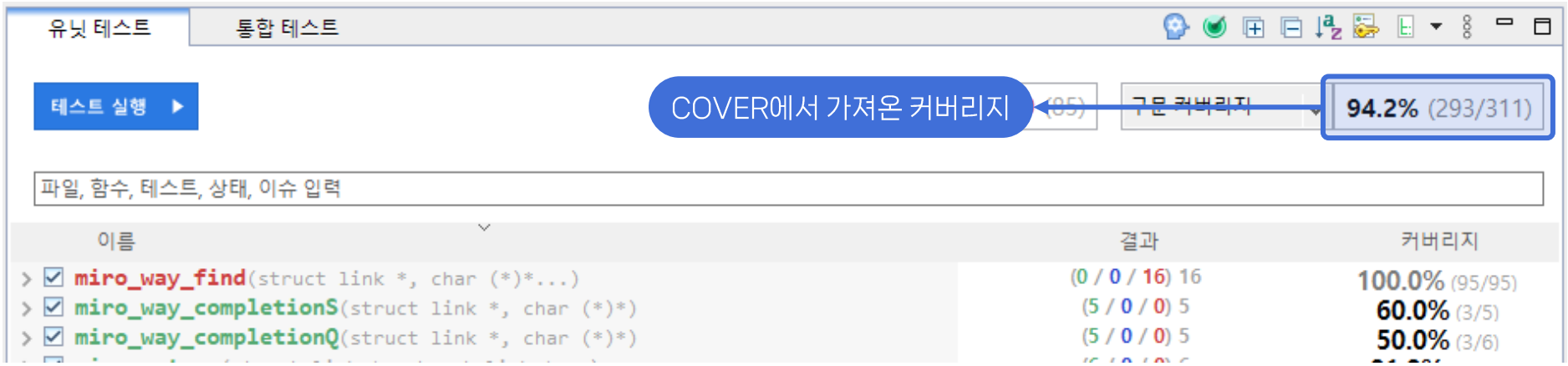
COVER




**시스템 테스트로 달성한 커버리지 결과를 가져오기**

## COVER에서 미달성된 부분만 CT 에서 테스트

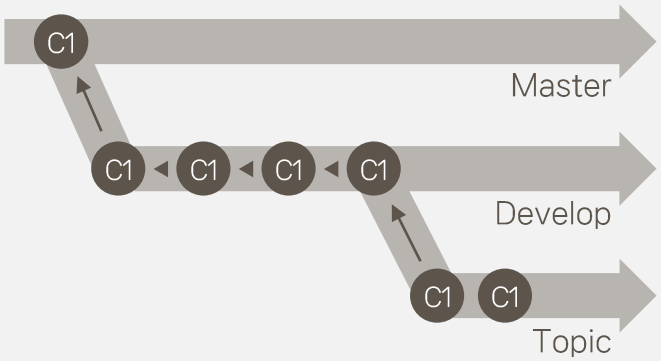
CT

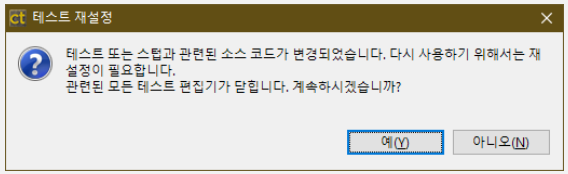


# 테스트 재사용

- 소스 코드가 변경된 이후에 테스트를 재사용하기 위한 편의 기능 제공

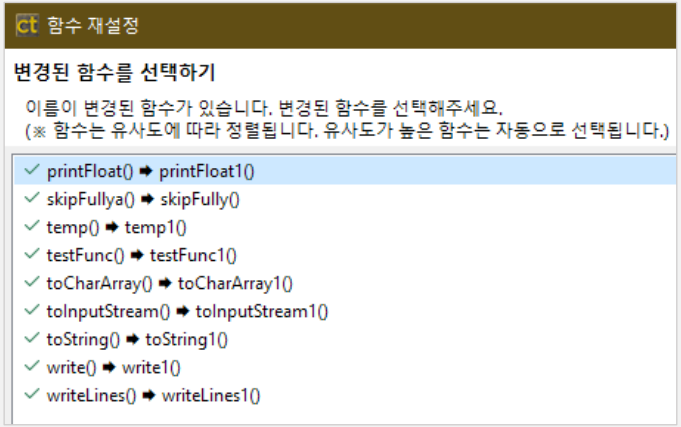
## 소스 코드의 변경을 자동으로 감지 (무결성 검사)



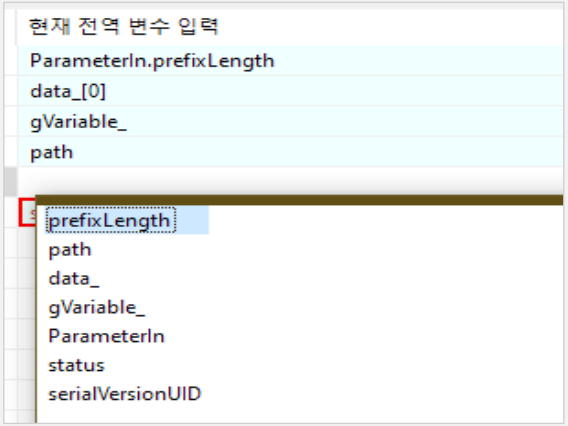




테스트와 관련된 변경 사항 자동 감지

## 변경 대상에 대한 유사도 기반 추천/자동 맵핑 기능 제공 (일괄 수정)




맵핑 대상 함수 유사도 기반 자동 맵핑




맵핑 대상 변수 유사도 기반 추천 및 유효성 검사

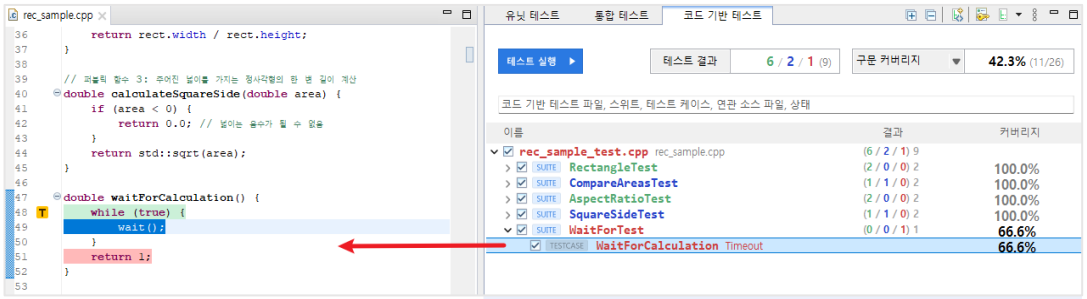
# 코드 기반 테스트 (Code-based Testing)

- 개발자에게 친숙한 Google Test 형식으로 테스트 작성
- 코드 기반 테스트로 CT의 모든 기능을 동일하게 사용

## CTEST 매크로 기반
## 테스트 작성 및 커버리지 결과 확인

**주요 특징**
Key Features

① Google Test 매크로 호환 지원
② 기존 Google Test 코드 재사용 가능
③ 테스트 케이스 구조 자동 파싱




## 모든 CT 기능 및 표준 인증 지원

| CT의 모든 분석/검증 기능 100% 포함 | 커버리지 측정 지원 | 요구사항 추적성 연결 | 모든 정적/동적 분석 리포트 생성 기능 |
|---|---|---|---|
| | Statement , Branch, MC/DC | Requirements Traceability | |

## 구조화 테스트 vs 코드 기반 테스트

| 구분 | 구조화 테스트 | 코드기반 테스트 | ★ 상호 보완적으로 활용 시 더 높은 커버리지 달성 가능 |
|---|---|---|---|
| 테스트 형식 | CT 자체 포맷 | Google Test 형식 | |
| 작성 방식 | GUI로 설계 | GTest 매크로 작성 | |
| 강점 | AI없이 자동생성 | ★ 복잡한 C++객체 | |

## CTEST 매크로 작성 예시 Code Example

**① CTEST : 단순 테스트 케이스**

```
CTEST(Math, Addition) {
  CTEST_ASSERT_EQ(5, Add(2, 3));
}
```

**② CTEST_F : 공통 Fixture 사용**

```
class CalcFix : public ::ctest::Fixture { ... };
CTEST_F(CalcFix, InitVal) {
  calc.setup(init_val);
  CTEST_ASSERT_EQ(105, calc.add(5));
}
```

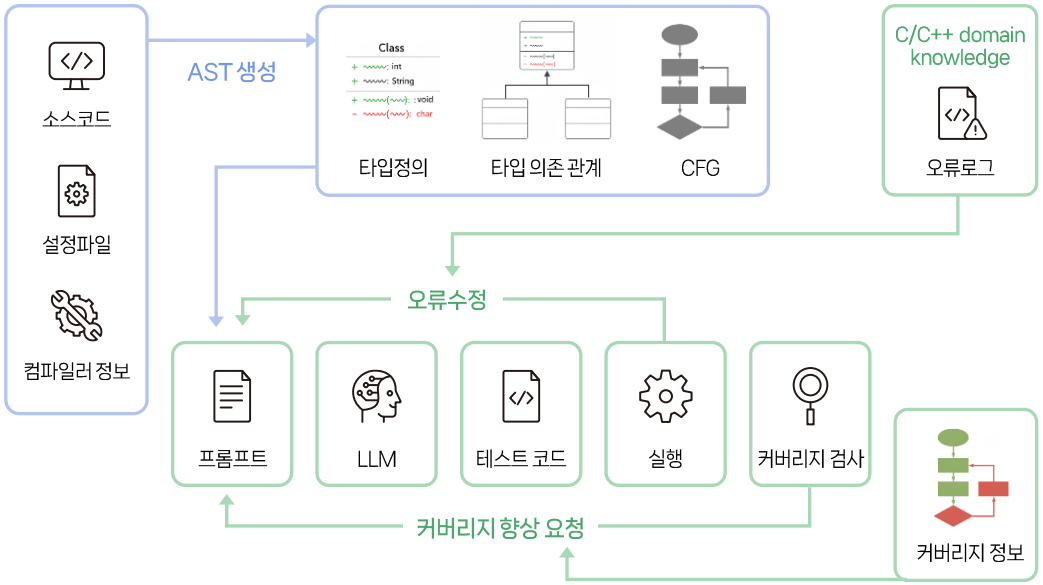
**③ CTEST_P : 파라미터화 테스트**

```
CTEST_P(ParamTest, IsEven) {
  CTEST_EXPECT(GetParam() % 2 == 0);
}
INSTANTIATE_CTEST_SUITE_P(Evens, ...);
```

# AI 기반 테스트 자동화

- 정확한 분석 정보 기반 AI 테스트 자동 생성 및 수정
- Closed-Loop 방식으로 지속 개선하여 일반 생성 대비 높은 품질 달성

## Closed-Loop Process & Analysis Flow




### 분석 정보 제공

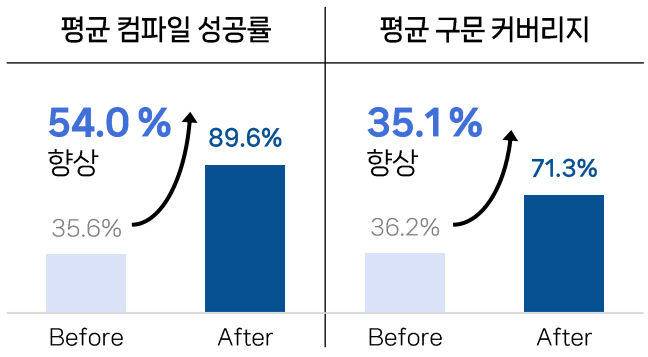
① **구문 트리 (AST)** : 함수 시그니처 및 타입 파악

② **제어 흐름 (CFG)** : 분기 조건 및 경로 분석

③ **빌드 환경** : 헤더, 매크로, 컴파일러 옵션 식별

## 분석 정보 적용 및 병합 효과

● Before : 단순 생성
● After : 분석 기반




**구조화 + AI 병합 시**
(최종 커버리지)

**최종 평균 76.8% 달성**

상호 보완 효과
**5.5%p**
**추가 상승**

## Self-Healing 자동 수정

88가지 변경 케이스 컴파일 성공 비율
(Ninja 프로젝트 기준)

**73.8%**

# AI 도우미

- 복잡한 C++ 스텁 작성 가이드 및 런타임 오류 자동 분석
- AI 챗봇으로 CT 사용법 문의, 문제 해결, 기능 실행(MCP) 지원

## AI 가이드 주요 기능

### 런타임 오류 정밀 분석

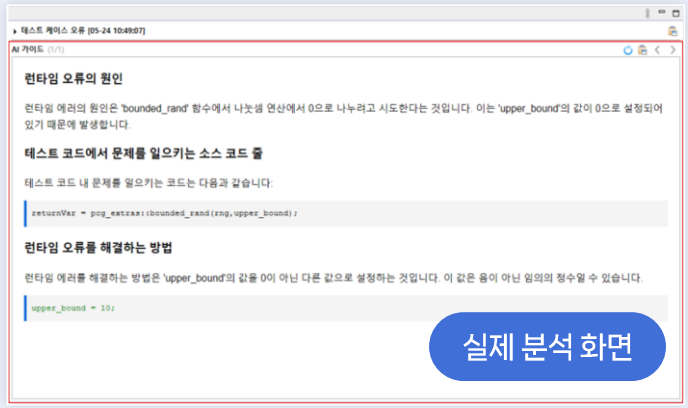
테스트 실행 오류 문제 원인 자동 진단
및 해결책 제시

### 스텁(Stub) 코드 자동생성 가이드

C++ 의존성 분석 기반 컴파일 가능한
스텁 코드 제안

## 런타임 오류 분석 가이드

- 오류 원인 **자동 식별**
- 해결 코드 **즉시 제공**




## AI 챗봇 & MCP 연동

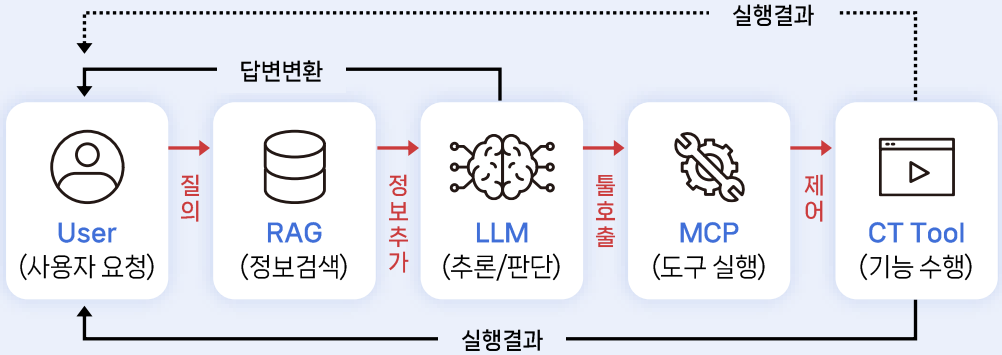
### 자연어 문의 및 트러블슈팅 (RAG)

"커버리지 측정 제외 메뉴 찾아줘"와 같은
자연어 질의 시 답변

### CT 기능 직접 실행 (MCP)

단순 답변을 넘어 테스트 생성, 설정 변경 등
도구 기능을 직접 호출하여 수행

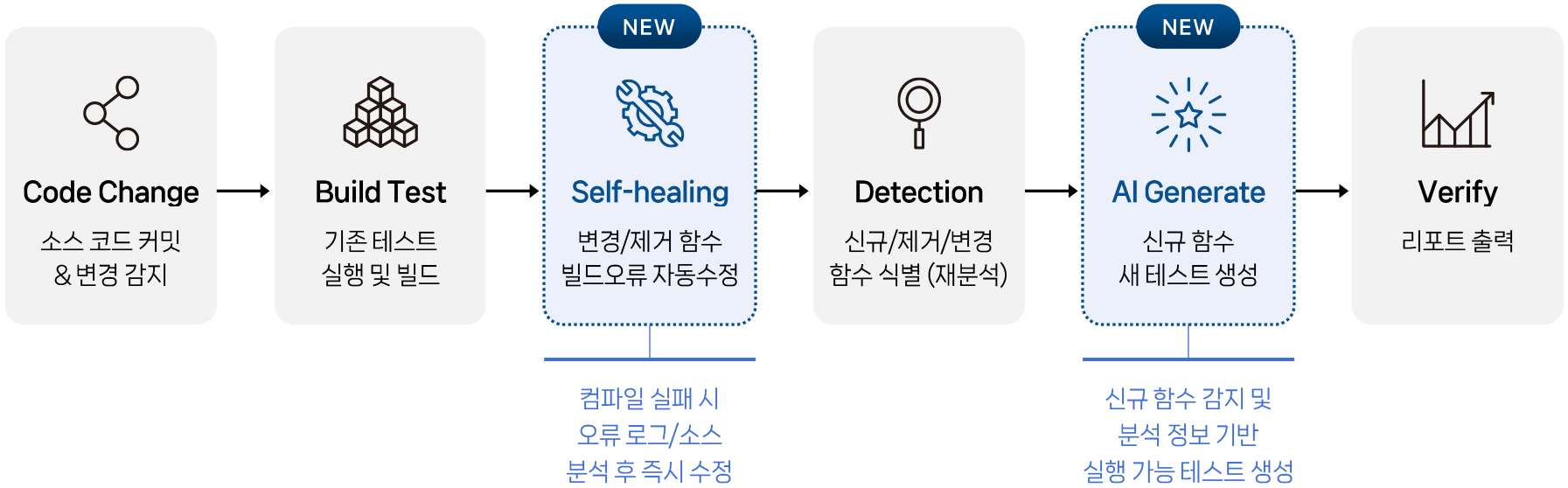
## RAG 기반 질의 및 MCP 실행 흐름



# CI/CD 환경 연동 (Pipeline Integration)

- CI/CD 파이프라인 통합 지원 (Jenkins Plugin, CLI, Docker)
- 파이프라인에서 AI 테스트 자동 생성 및 Self-healing까지 수행 가능

## AI-Driven Test Automation Pipeline




**1** **Jenkins Plugin & CLI**

프로젝트 생성부터
보고서 생성까지
CLI로 제공

**2** **End-to-End Automation**

소스 커밋부터
검증까지 파이프라인 내
완전 자동화

**3** **AI-Powered Maintenance**

자동 생성 (Generation)과
자동 수정 (Self-healing)
결합

## 상세 환경

| 구분 | | 세부사항 |
|---|---|---|
| 지원 언어 | | - C, C++ |
| 개발 환경 | 운영체제 | - Windows 7 이상 (64bit)<br>- Linux (Debian, Red Hat 계열) |
| | 다국어 지원 | - 한국어, 영어 지원 |
| | 컴파일러 지원 | - 최신 컴파일러 지원 (C++ 23) (GCC 15, Clang 21, VS 2026)<br>- ARM 계열<br>- Freescale 계열<br>- GreenHills 계열<br>- Keil 계열<br>- Renesas 계열<br>- 기타 100개 이상 지원 |
| | 타깃 인터페이스 | - Ethernet, Serial (UART), JTAG<br>- Debugger 연동 (TRACE32, iSYSTEM, Code Composer Studio, IAR C-SPY, PLS UDE 등) |

## 하드웨어 권장 사양

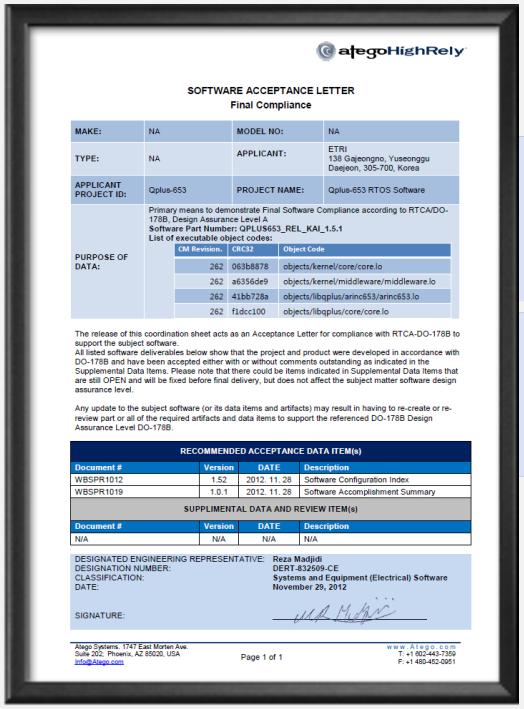
| CPU | RAM | 저장공간 |
|---|---|---|
| 3GHz, 8코어 | 16 GB 이상 | 10 GB 이상 |

# 무인항공기 DO-178B Level A 획득 사례(ETRI, KAI)

**검증대상** Qplus-653 무인 항공기용 RTOS

ETRI 한국전자통신연구원 www.etri.re.kr  KAI 한국항공우주산업㈜

: 미국 항공 인증기관 아티고(Atego)로부터 항공 인증 최고 수준 'DO-178B Level A' 획득



atego HighRely

SOFTWARE ACCEPTANCE LETTER
Final Compliance

| | | | |
|---|---|---|---|
| MAKE: | NA | MODEL NO: | NA |
| TYPE: | NA | APPLICANT: | ETRI<br>138 Gajeongno, Yuseonggu<br>Daejeon, 305-700, Korea |
| APPLICANT PROJECT ID: | Qplus-653 | PROJECT NAME: | Qplus-653 RTOS Software |
| PURPOSE OF DATA: | Primary means to demonstrate Final Software Compliance according to RTCA/DO-178B, Design Assurance Level A<br>**Software Part Number: QPLUS653_REL_KAI_1.5.1**<br>**List of executable object codes:** | | |

| CM Revision | CRC32 | Object Code |
|---|---|---|
| 262 | 063b8878 | objects/kernel/core/core.lo |
| 262 | a6356de9 | objects/kernel/middleware/middleware.lo |
| 262 | 41bb728a | objects/libqplus/arinc653/arinc653.lo |
| 262 | f1dcc100 | objects/libqplus/core/core.lo |

The release of this coordination sheet acts as an Acceptance Letter for compliance with RTCA-DO-178B to support the subject software.
All listed software deliverables below show that the project and product were developed in accordance with DO-178B and have been accepted either with or without comments outstanding as indicated in the Supplemental Data Items. Please note that there could be items indicated in Supplemental Data Items that are still OPEN and will be fixed before final delivery, but does not affect the subject matter software design assurance level.

Any update to the subject software (or its data items and artifacts) may result in having to re-create or re-review part or all of the required artifacts and data items to support the referenced DO-178B Design Assurance Level DO-178B.

**RECOMMENDED ACCEPTANCE DATA ITEM(s)**

| Document # | Version | DATE | Description |
|---|---|---|---|
| WBSPR1012 | 1.52 | 2012. 11. 28 | Software Configuration Index |
| WBSPR1019 | 1.0.1 | 2012. 11. 28 | Software Accomplishment Summary |

**SUPPLIMENTAL DATA AND REVIEW ITEM(s)**

| Document # | Version | DATE | Description |
|---|---|---|---|
| N/A | N/A | N/A | N/A |

DESIGNATED ENGINEERING REPRESENTATIVE: **Reza Madjidi**
DESIGNATION NUMBER: **DERT-832509-CE**
CLASSIFICATION: **Systems and Equipment (Electrical) Software**
DATE: **November 29, 2012**
SIGNATURE:

Atego Systems, 1747 East Morten Ave, Suite 202, Phoenix, AZ 85020, USA
info@Atego.com
Page 1 of 1
www.Atego.com
T: +1 602-443-7359
F: +1 480-452-0951

DO-178B Level A 인증서

## High-level Requirements Test

- SW가 타깃에 대한 high-level 요구사항을 충족하는지 확인
- Use ARINC 653 Conformance Test Suite

## Low-level Requirements Test

- SW가 low-level 요구사항 및 MC/DC 커버리지를 만족하는지 확인
- 도구 : CT

## 커버리지 분석

- SW의 요구사항 커버리지와 구조적 커버리지를 분석
- 도구 : COVER & CT

## 정적 분석

- SW가 코딩 표준을 준수하였는지 확인
- 도구 : STATIC

## 소스-바이너리 일치성 검사

- 소스 코드와 컴파일러가 생성하는 목적 코드의 일치성을 확인

(출처: http://www.dt.co.kr/contents.html?article_no=2012121302010257731002)

SURESOFT

Intelligence for Safe World

# Thank You