

# [WhitePaper]

## Target 기반 검증 여부가 SW 테스트 도구의 차이

2024. 7

(주)소프트4소프트

**Soft**  
**4**  
**Soft**

**RESORT Code Analysis**



**DNV**

DNV GL BUSINESS ASSURANCE

**CWE**  
**COMPATIBLE**

[www.soft4soft.com](http://www.soft4soft.com)

02-553-9464

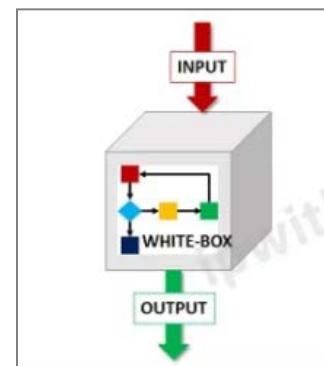
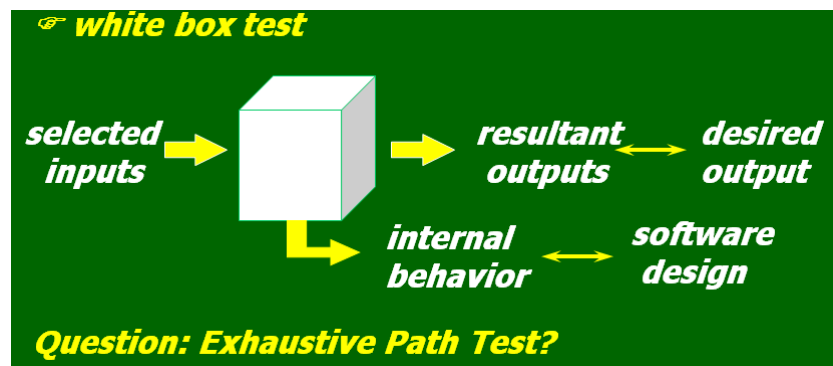


- 1. Software Testing Classification**
  - 1.1 White-box Testing**
  - 1.2 Black-box Testing**
  - 1.3 White-box Testing vs. Black-box Testing**
- 2. Automated Test Generation Method**
  - 2.1 Concrete Execution**
  - 2.2 Symbolic Execution**
  - 2.3 Concrete Execution vs. Symbolic Execution**
- 3. SW Testing Tool Comparison**

## 1. Software Testing Classification

### ◆ White-box Testing

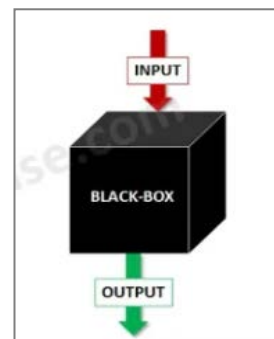
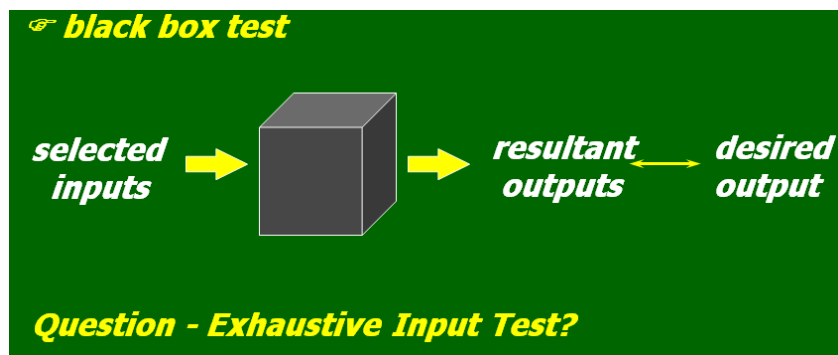
- **White-box Testing**은 프로그램의 내부 구조에 초점을 둔 how의 관점으로 코드의 논리적 구조를 테스트한다.
- **Test Case Generation**은 구조적 테스트로 구현된 프로그램 내부 구조를 보고 테스트 데이터를 선정한다. 따라서 프로그래밍 지식이 필요하다.
- **Test Level**은 unit, integration test와 같은 low level testing에 적합하다.
- **Testing Type**은 structural testing, code-based testing, path testing, fault injection testing 등이 있다.
- **White-box Testing Technique**은 Code Coverage Analysis이다: Statement Coverage, Branch Coverage, MC/DC Coverage, Path Coverage 등
- 개발자, 테스터에 의해 테스트가 수행한다.



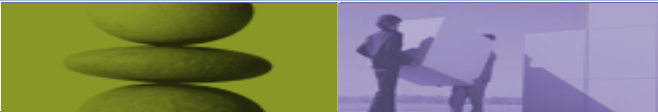
## 1. Software Testing Classification

### ◆ Black-box Testing

- **Black-box Testing**은 소프트웨어 동작에 초점을 맞춘 what의 관점으로 기능을 테스트한다.
- **Test Case Generation**은 기능적 테스트로 프로그램의 요구사항 명세로부터 테스트 데이터를 선정한다. 따라서 프로그래밍 지식이 필요하지 않다.
- **Test Level**은 unit, integration, system, acceptance test와 같은 high level testing에 적합하다.
- **Testing Type**은 functional testing, non-functional testing, regression testing, fault injection testing 등이 있다.
- **Black-box Testing에 사용되는 기법**은 Equivalence Partitioning, Boundary Value Analysis 등이 있다.
- 최종 사용자, 개발자, 테스터에 의해 테스트가 수행한다.

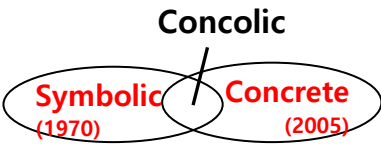


## 1. Software Testing Classification



### ◆ White-box Path Testing vs. Black-box Testing

- **White-box Path Testing**은 프로그램의 내부 구조에 초점을 둔 how의 관점으로 코드의 실행 경로(Feasible Path)를 테스트한다.
- **Black-box Testing**은 소프트웨어 동작에 초점을 맞춘 what의 관점으로 기능을 테스트한다.



Testing Technology	White-box Path Testing (how 관점, 코드의 논리적 구조인 경로를 테스트)	Black-box Testing (what 관점, 함수(모듈)의 기능을 테스트)
Test Case Generation	테스트 자동생성 (실행경로의 테스트 도메인/입력 값) - # Test Case per Function <= n (조건문 수)	테스트 수동 입력 (기능의 입력 값/기대 값) - # Test Case per Function = (min) 1
	테스트 케이스의 자동 생성으로 테스트의 노력, 시간, 비용에 크게 절감 (시간 10~100배)	전통적인 테스트 케이스의 입력 접근 방식으로 숙련된 테스트 전문가의 노력, 시간, 비용 소요 <span>Test Cost</span>
Test Execution	테스트 실행 환경 설정인 Automation Test Script <sup>1)</sup> 실행 - Test Script 작성 없음	테스트 실행 환경 설정인 Manual Test Script 작성/실행 - 테스트 케이스에 Manual Test Script 추가 <span>Extra Cost</span> - Manual Test Script에 테스트 데이터/실행 변수 추가
Test Results Verification	확장된 CFG의 MBD(Model-Based Design) 기반 테스트 실행 전/후의 경로 비교로 "Program Logic"에 대한 코드 버그 식별	기능 테스트 결과에서 미실행 코드 영역의 검증은 코드 버그 인지는 수작업의 노력 필요
Test Type	structural/requirement/code-based/path testing	Functional/non-functional testing
	Path-based Fault Injection Testing은 각 실행경로의 Path Domain기반 Fault 값 설정으로 통합 지원	Function-based Fault Injection Testing은 추가적인 테스트 환경 설정 또는 독립된 도구에서 지원

주1) Test Script: 테스트 케이스를 실행하기 위한 절차를 작성한 프로그램

### ◆ Concrete Execution

#### ➤ Key Idea

- ✓ Execution of programs using concrete data
- ✓ Execute target program on automatically generated test inputs
- ✓ Program takes the feasible path only

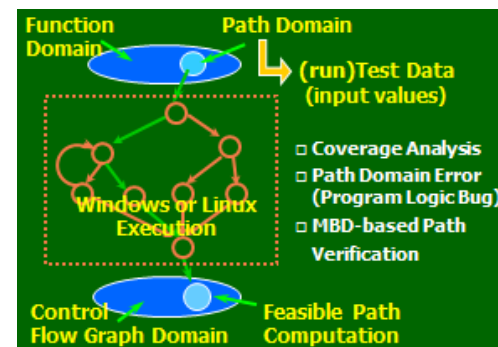
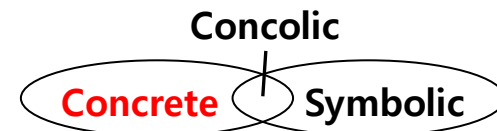
#### ➤ Limitation

- ✓ Program takes only one path determined by input values

#### ➤ Usage

- ✓ White-box Testing
  - Structural Testing, Code-based Testing, Path Testing, Fault Injection Testing, etc.
- ✓ Low Level Testing: Unit, Integration Test

#### ➤ 관련 도구: RESORT White-box Path Testing (소프트4소프트)

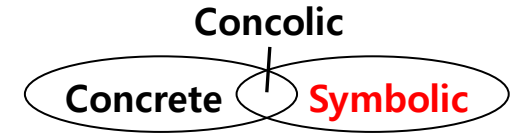




### ◆ Symbolic Execution

#### ➤ Key Idea

- ✓ Execution of programs using symbolic input values instead of concrete data
- ✓ A symbolic state represent a set of concrete states
- ✓ Program takes the basic path



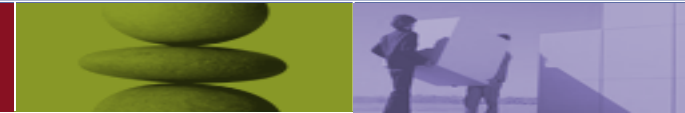
#### ➤ Limitation

- ✓ Limited by the power of constraint solver
- ✓ Scalability issues when faced with large (exponential) number of paths – path explosion problem!
  - Suppose the loop went up to 20 and the condition clauses are five?
  - $5^{20}$  possible paths!  $\Rightarrow$  3,170 years =  $5^{20}$  paths / millisecond per one test

#### ➤ Usage

- ✓ Black-box Testing
  - Functional Testing, Non-functional Testing, Regression Testing, etc.
- ✓ High Level Testing: Unit, Integration, System, Acceptance test

#### ➤ 관련 도구: 국내외 테스트 도구



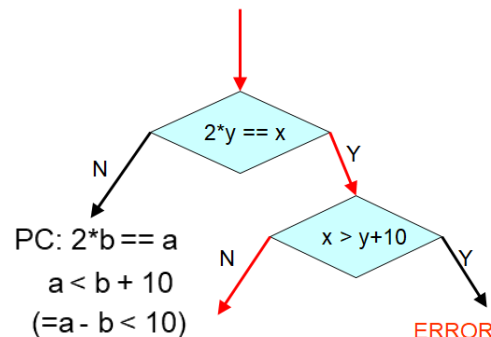
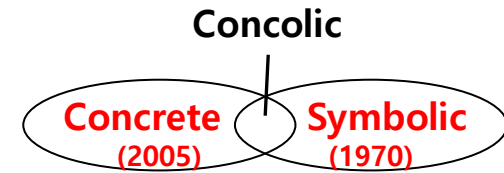
### ◆ Concrete Execution vs. Symbolic Execution

#### ➤ Test Method History

- ✓ Symbolic Execution(1970)
- ✓ Concrete Execution (2005)

#### ➤ Path condition (PC)

- ✓ Condition on the inputs to reach the location
- ✓ Logical formula over symbolic inputs
- ✓ Accumulates constraints that inputs have to satisfy for the particular path to be executed
- ✓ If a path is feasible, its PC is satisfiable



```
void testme (int x, int y) {
  z = 2* y;
  if (z == x) {
    if (x > y+10) {
      ERROR;
    }
  }
}
```

Concrete Execution	Symbolic Execution	
concrete state	symbolic state	path condition
x = 2, y = 1, z = 2	x = a, y = b, z = 2*b	2*b == a a < b + 10
Solve: 2*b == a Solution: a = 2, b = 1		

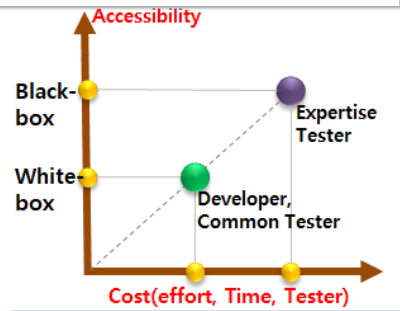
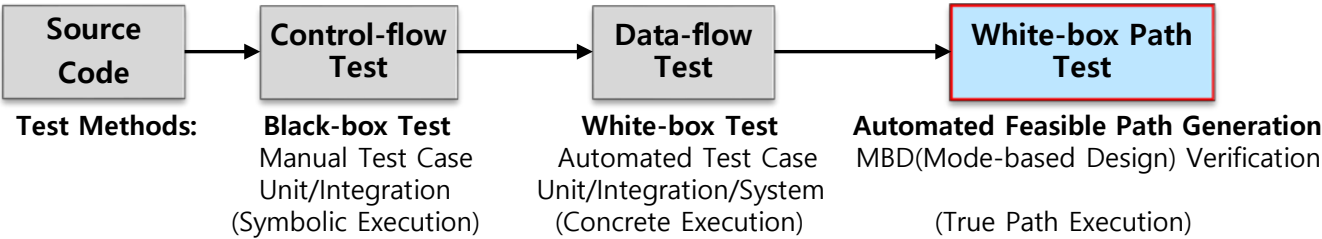


3. SW Testing Tool Comparison



- ◆ (고급 도구의 Target 기반 테스트) SW 안전성 검증의 노력/시간 절감
  - White-box Path Testing: 함수 내부 구조(소스 코드)의 실행 경로 기반 **테스트 케이스를 자동 추출**
  - Black-box Testing: 함수의 입.출력 기능(동작) 기반 **테스트 케이스를 수작업으로 작성**

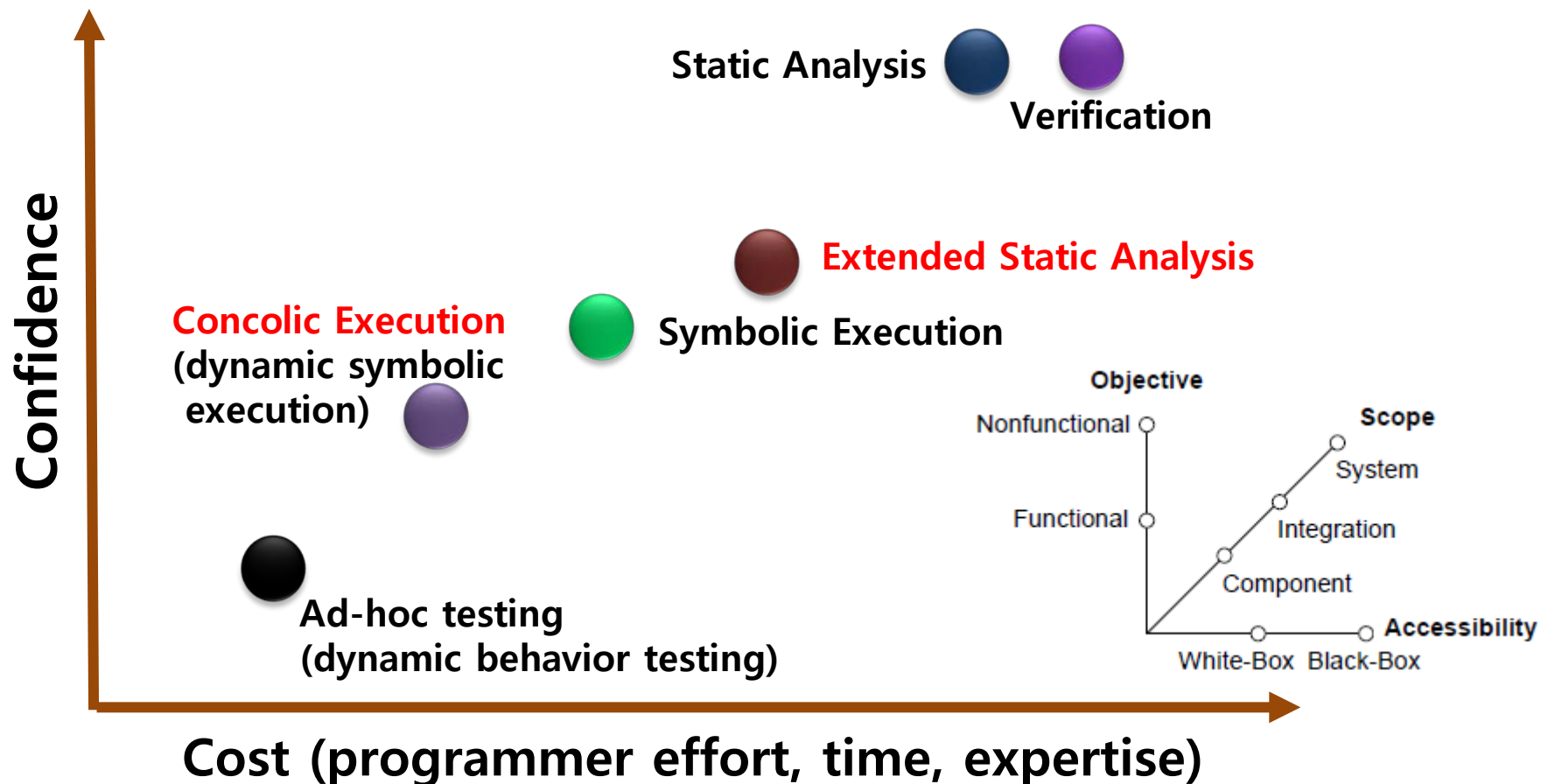
도구	테스트 유형 및 기능	White-box Path Test (RESORT)	Black-box Test (상용 도구)	기능/성능 설명
Test Level	단위 테스트	O	O	Statement, Branch, MC/DC Coverage
		O	X	(HKMC) Traceability, Consistency Coverage
	통합 테스트	O	O	Function, Call Coverage
	GUI 시스템 테스트	O (GUI SW)	X	Code Coverage
	펌웨어 시스템 테스트	SW(exe)+HW(MCU)+통신(UART) on Target	SW(code)+HW가상환경(시뮬레이션) on Host	도구 벤더의 테스트 환경 의존성: (Target) Uninterrupted Workflow vs. (Host) Interrupted Workflow
Test Cost (time, effort)	결함 주입 테스트	O	O	(White) 통합 지원(도메인 분석), (Black) 독립 도구
	Design/Exe/Cov	W/W/W	B/B/W	(White) Logic/Algorithm, (Black) Behavior
	Test Data Generation (Test Case)	테스트 데이터 (도메인, 입력 값) 자동 생성	입.출력 수작업	Test Case: 자동 vs. 수동 (Test Harness 필요) Automated SW Testing: 10~100x Faster
Verification	Test Case Execution	System(cross function)	Function	Test Case Execution Unit
	MBD 테스트 검증	자동	X	Path Verification: Program Logic Error
	Coverage Mark	O	X	테스트된 코드의 커버리지 유형 약어 표시



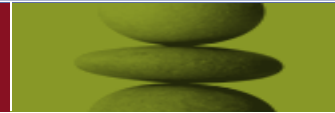
## 3. SW Testing Tool Comparison

### ◆ RESORT Advanced Validation Technologies

- (Static Analysis) Static Analysis -> **Extended Static Analysis (Inter-procedural Analysis)**
- (Dynamic Analysis) Symbolic Execution -> **Concolic Execution (White-box Path Testing)**



*If you cannot MEASURE it, you cannot IMPROVE it*



## (주)소프트4소프트

본사: 서울 서초구 서운로1길 34 한국산업기술보호협회 3층  
R&D: 대전광역시 유성구 가정북로 96 대전일자리경제진흥원 308호

Tel : 02-553-9464

기술 및 일반 문의 : [info@soft4soft.com](mailto:info@soft4soft.com)

[www.soft4soft.com](http://www.soft4soft.com)