KU KONKUK UNIVERSITY

HELISCOPE Project의 비행 운용 프로그램을 위한 검증 절차

건국대학교
Dependable Software Laboratory
이종훈

차례

1. 서론

1. 서론

- 소프트웨어 정확성의 필요성
  - 여러 분야에서 소프트웨어 사용 비중 증가
    - 항공: 비행 제어 소프트웨어
    - 철도: 차량제어, 신호제어 소프트웨어
    - ……
  - 소프트웨어 결함이 사고 발생으로 연결
  - 소프트웨어의 정확성 확보하기 위한 정형화된 검증 절차 필요

- 각 분야 소프트웨어 표준
  - 항공: DO-178B
  - 철도: EN 50129
  - ……

1. 서론

- HELISCOPE Project
  - 재난 대응 및 복구를 위한 소형 무인 헬리콥터 개발
  - 직접 조종 이외에 자동항법(무인 조종) 기능

- HELISCOPE Project의 OFP(Operational Flight Program)
  - HELISCOPE Project의 무인 헬리콥터에서 동작
  - 비행제어를 위한 소프트웨어
  - OFP의 결함은 비행제어의 이상으로 이어짐
  - 정확성 확보가 매우 중요

- HELISCOPE Project의 OFP를 위한 V&V 절차 구성
  - 관련 표준 참조
  - HELISCOPE Project의 OFP 특징 고려

무인 헬리콥터 시험비행 장면

2. 배경

2.1 표준

- IEEE 1012-2004: Standard for Software Verification & Validation
  - 소프트웨어 V&V를 위한 표준
  - 결함이 야기하는 위험 정도에 따라 SIL(Software Integrity Level) 정의
    - 위험성 분석을 통하여 정의
    - 높은 SIL에 대하여 엄격한 검증 활동 수행

- DO-178B: Software Considerations in Airborne Systems and Equipment Certification
  - RTCA에서 개발한 항공 소프트웨어 개발 및 인증 표준
  - 결함이 야기하는 위험 정도에 따라 소프트웨어의 등급 정의
    - 시스템 안전 평가
    - 높은 등급에 대하여 엄격한 검증 활동 수행

2.2 HELISCOPE Project의 OFP 특징

- Mission Critical 소프트웨어
  - 오류가 임무 실패 및 시스템 손실로 연결
- 실시간, 내장형 소프트웨어
  - 연산에 시간 제약 존재
  - 헬리콥터의 비행제어컴퓨터에서 동작
    - 제한적인 자원에 따른 안정성 저하 발생 가능
- 개발 완료된 소프트웨어
  - 개발 생명 주기가 이미 진행
  - 개발 생명 주기에 따른 V&V 진행 어려움
  - 요구사항, 설계 정보가 부족할 수 있음

3. HELISCOPE Project의 OFP를 위한 V&V 절차

3. HELISCOPE Project의 OFP를 위한 V&V 절차

- 표준을 참조하여 정형화된 절차 구성
  - IEEE 1012-2004 소프트웨어 V&V 표준 절차 참조
    - 개발 생명 주기에 따른 검증 활동
  - DO-178B 항공 소프트웨어 개발 표준 참조
    - 정적 코드 분석, 시험 기법
  - OFP의 SIL을 정의하고 그에 따라 검증 활동 적용

- HELISCOPE Project의 OFP 특징 고려
  - Mission Critical 소프트웨어
    - 정형 검증
  - 실시간, 내장형 소프트웨어
    - 정적 코드 분석
    - 시험
  - 개발 완료된 소프트웨어
    - 소프트웨어 역공학

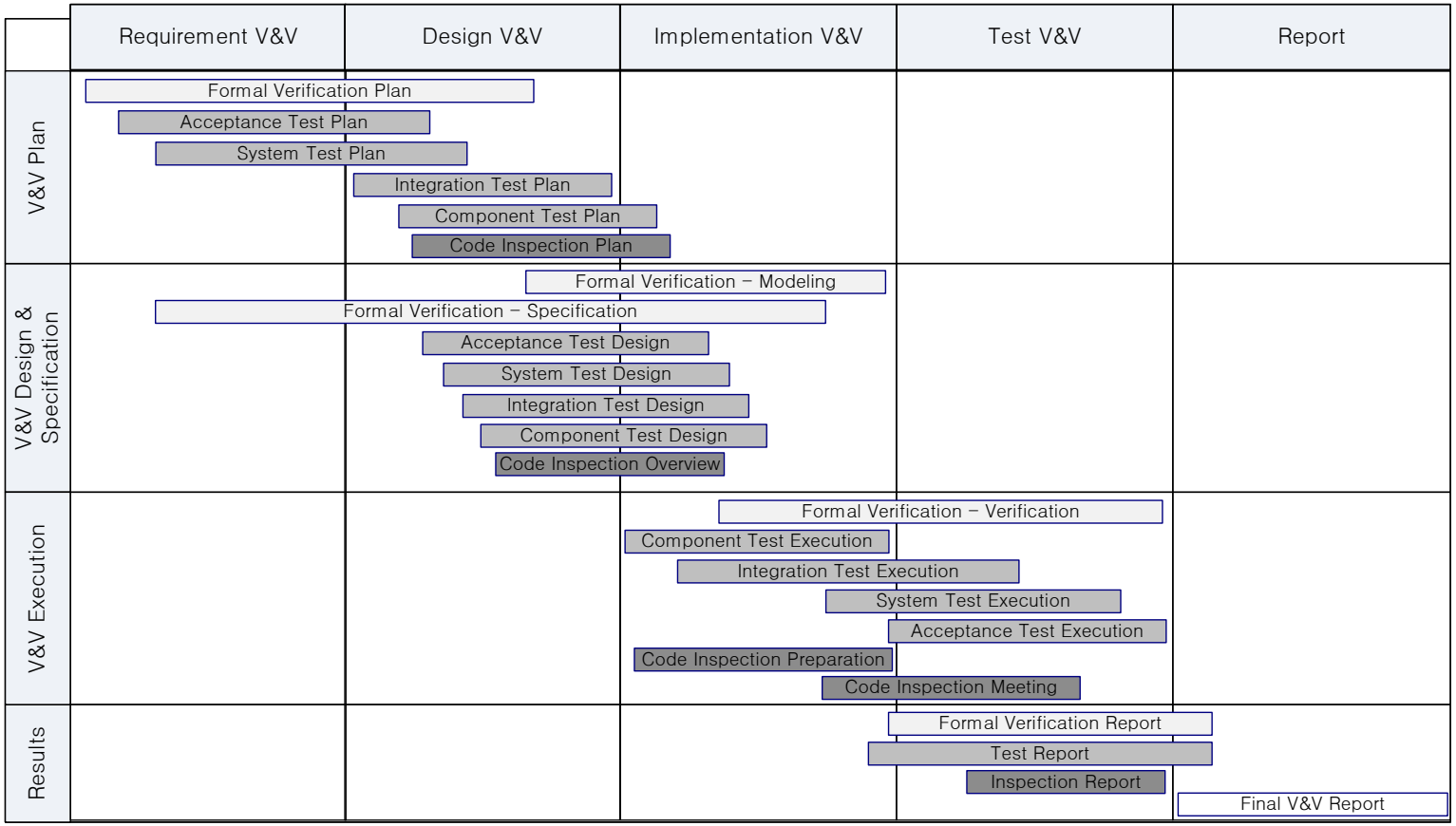
3.1 HELISCOPE Project의 OFP SIL정의

- HELISCOPE Project의 OFP 위험성 분석
  - 높은 영향력
    - 자동 항법 모드의 연산 오류, 연산 시간 초과는 치명적
    - 헬리콥터 비행 상태에 큰 영향
    - 임무의 성공/실패 여부

Consequence	정의	SIL
Catastrophic	미션 실패, 시스템 손실 등	4
Critical	미션의 부분적 실패, 시스템 일부 손실 등	3
Marginal	미션 수행 능력 저하, 시스템에 피해 등	2
Negligible	미션 수행 능력 저하, 시스템 성능 저하 등	1

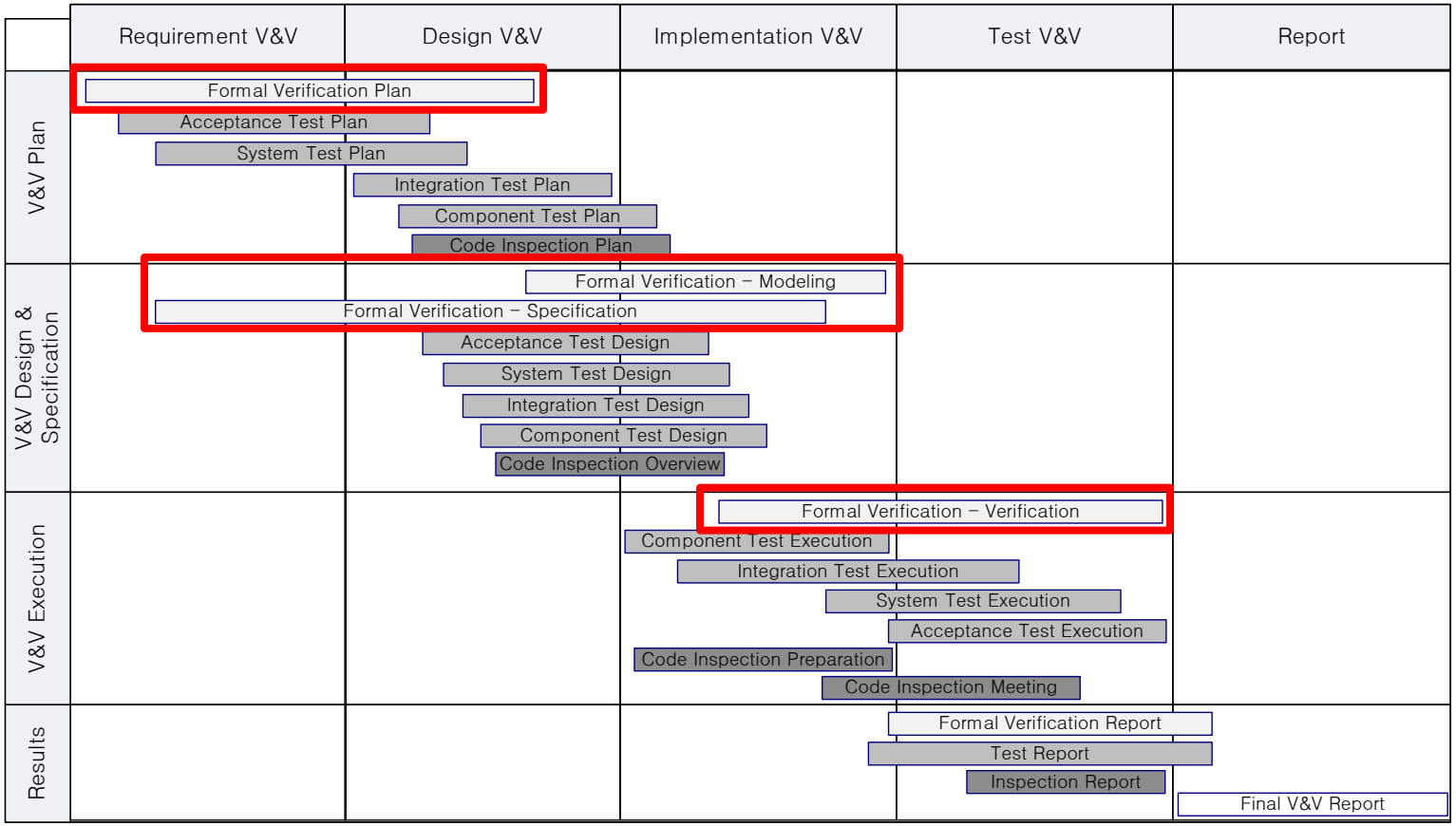
3.2 검증 기법 – 검증 활동

- 개발 생명 주기에 따른 검증 활동



3.2 검증 기법 – 정형 검증

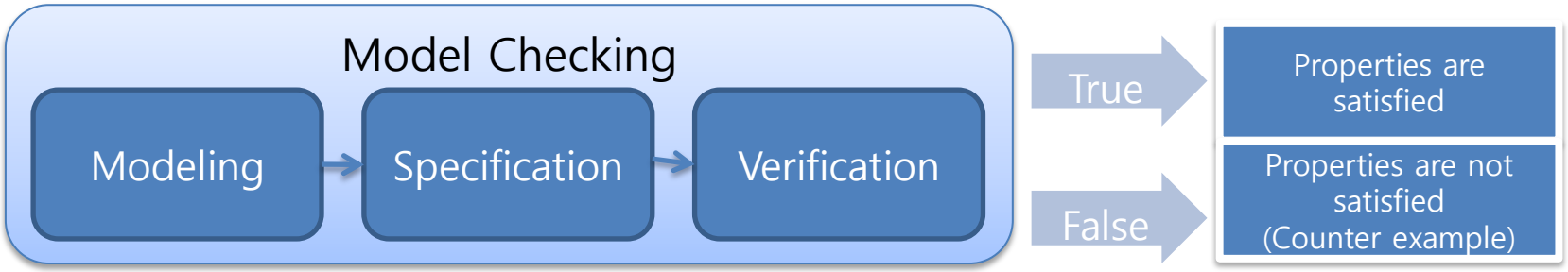
- 개발 생명 주기에 따른 검증 활동



3.2 검증 기법 – 정형 검증

- Mission Critical 소프트웨어인 HELISCOPE Project의 OFP 정확성 확보

- Model Checking
  - 모델 체킹 도구를 이용한 검증 자동화
    - 모델링
    - 속성 명세
    - 검증(자동화)
  - 반례를 통한 오류 상황 제시
    - 반례: 명세 속성을 만족하지 않는 모델의 상황



3.2 검증 기법 – 정형 검증

- 정형 검증 과정
  - 모델링
    - 시스템을 오토마타 형태로 표현
    - 오토마타를 모델 체킹 도구의 입력 언어로 변환
  - 속성 명세
    - 시스템이 만족해야 하는 속성들을 명세
    - 모델 체킹 도구에 알맞은 형태로 재명세
  - 검증
    - 모델이 속성을 만족하는지 검증
    - 모델 체킹 도구에 모델과 속성을 입력하여 자동화 검증
    - 속성을 만족하지 않을 시, 반례 제시
    - 반례에 대한 분석

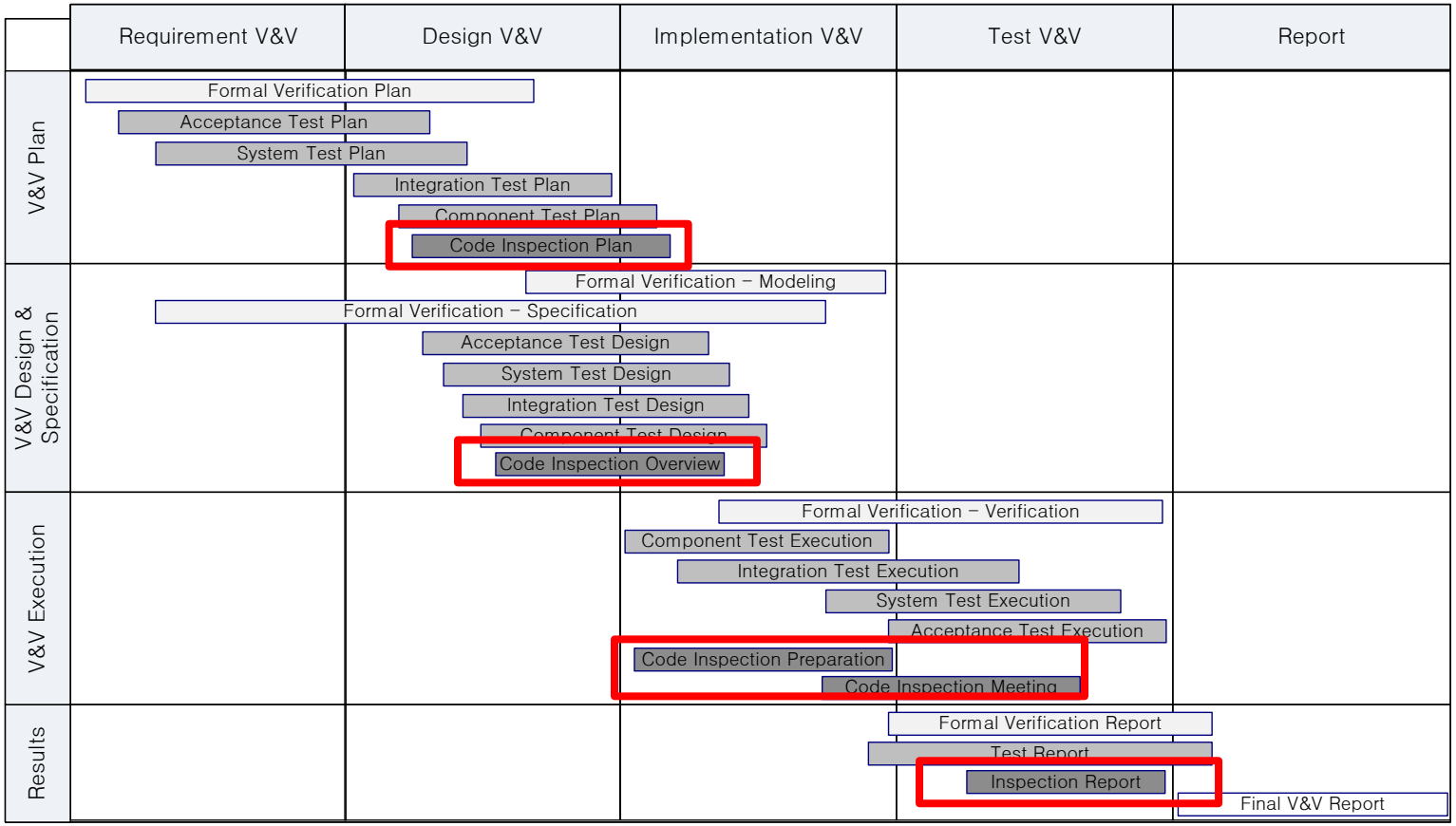
3.2 검증 기법 – 정형 검증

- 모델 체킹 도구

도구	특징	입력 언어	속성 명세
SPIN	분산 소프트웨어의 검증에 적합	PROMELA	LTL
UPPAAL	실시간 시스템의 모델링 및 증명과 검증 수행에 적합	Timed Automata	Query (CTL과 유사)
VIS	유한 상태 시스템의 시뮬레이션 및 정형 검증, 설계	Verilog	CTL
NuSMV	동기적, 비동기적 유한 상태 시스템의 표현 가능	NuSMV input Language	CTL, LTL
BLAST	C 프로그램을 위한 모델 체킹 도구	Program source	Temporal safety Property

3.2 검증 기법 – 정적 코드 분석

- 개발 생명 주기에 따른 검증 활동



검증 기법 – 정적 코드 분석

- 내장형 소프트웨어인 HELISCOPE Project의 OFP 분석
  - 소프트웨어를 실행하지 않고 코드를 분석
  - 내장형 소프트웨어에 치명적인 오류 탐지
    - 메모리 할당/해제, 범위를 벗어난 데이터, 논리의 오류 및 비효율성 등
  - 시험에서 발견하지 못하는 문제들을 발견

- 검사(Inspection)
  - 검사를 이용하여 코드 분석
  - HELISCOPE Project의 OFP 특징을 고려한 Checklist 작성
    - Checklist: 검사할 항목에 대한 목록

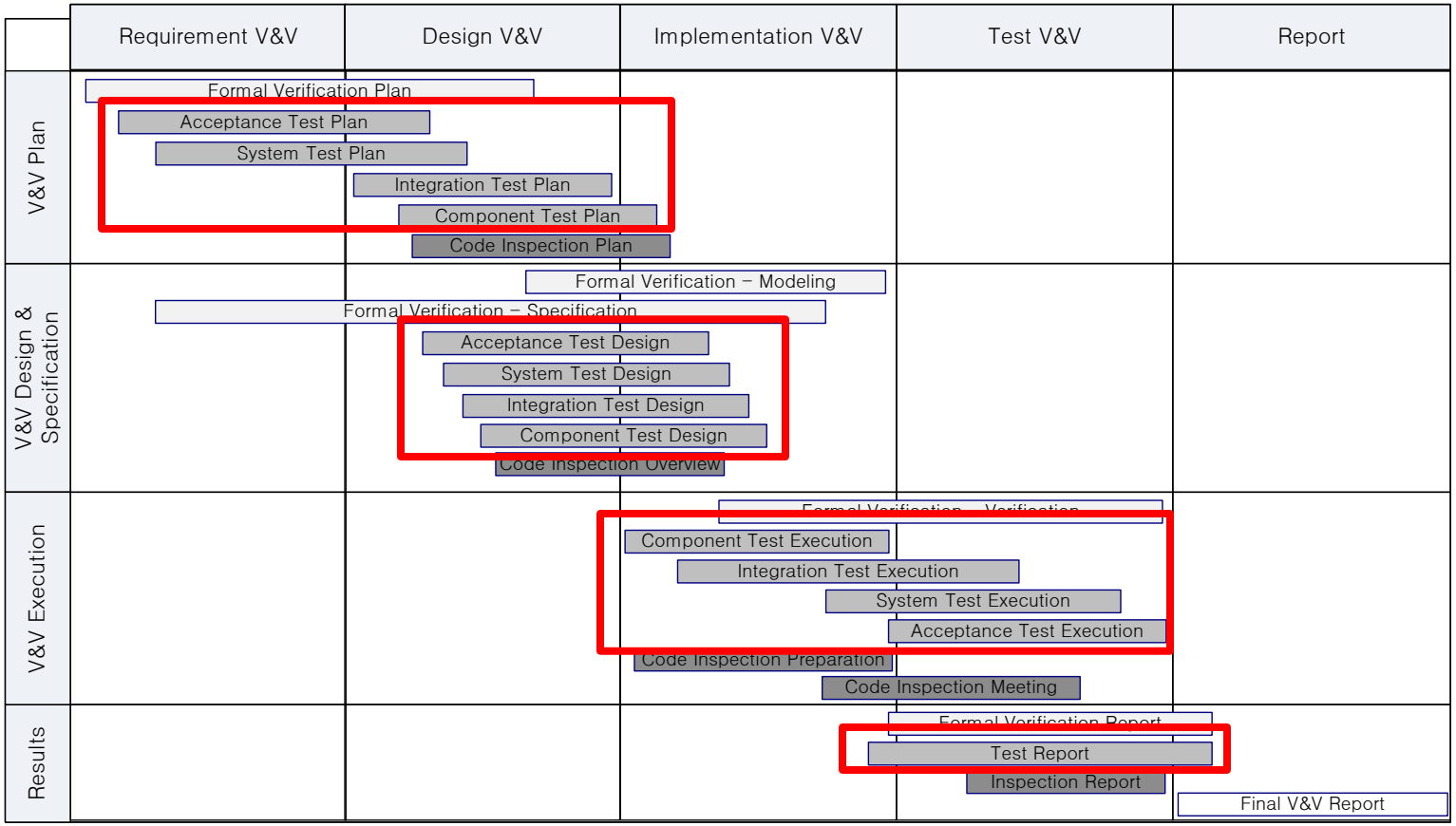
검증 기법 – 정적 코드 분석

- 분석 자동화
  - 도구를 사용하여 분석 자동화
  - 도구에 따라 Checklist 지정 가능

도구	언어	상용
Coverity	C, C++, JAVA	Commercial
LDRA Testbed	C, C++, Ada83, Ada95, Assembler	Commercial
Cppcheck	C, C++	Open source
RATS	C, C++, Python, Perl, PHP	Open source
Flawfinder	C, C++	Open source

3.2 검증 기법 – 시험

- 개발 생명 주기에 따른 검증 활동



3.2 검증 기법 – 시험

- HELISCOPE Project의 OFP를 실행하여 동작의 정확성 확보
  - 4 Level 시험 구성

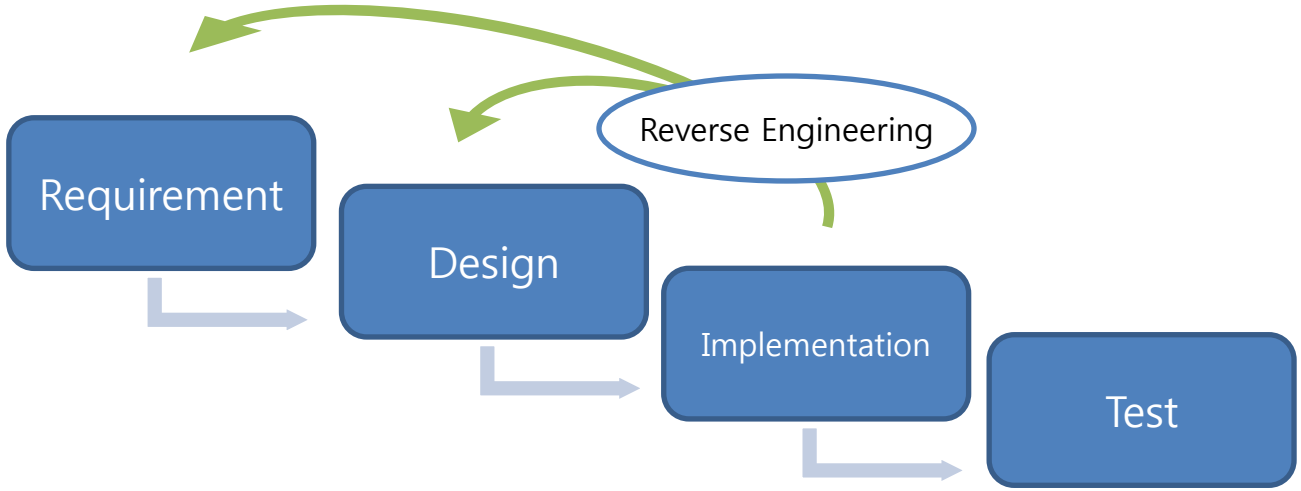
시험 Level	대상	시험 환경	특징
Component Test	OFP 모듈	개발 환경	-
Integration Test	OFP 모듈간 통합 테스트	개발 환경	모듈간 통합 후의 결점 탐지
System Test	전체 시스템	테스트 환경	HILS Simulator 환경 구성
Acceptance Test	전체 시스템(SW/HW간 동작)	실제 환경	내장형 소프트웨어 테스트 기법 고려

  - SIL에 따른 커버리지 적용

Structural Coverage	SIL 1	SIL 2	SIL 3	SIL 4
Statement Coverage	X	O	O	O
Decision/Condition Coverage	X	X	O	O
MC/DC Coverage	X	X	X	O

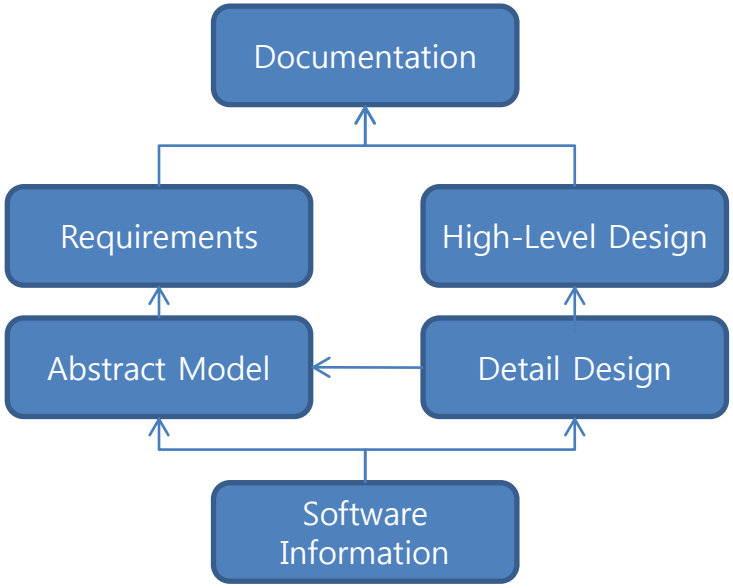
3.3 소프트웨어 역공학

- 개발이 완료된 소프트웨어
  - 개발 생명주기의 요구사항, 설계, 구현 단계가 진행
  - 요구사항, 설계 정보 누락 가능성
  - 검증을 수행하기 위해 역공학을 통하여 요구사항, 설계 정보 보완



3.3 소프트웨어 역공학

- 소프트웨어 역공학 과정
  - 설계 복구
    - 소스 코드를 분석하여 구조 분석
    - 각 부분에 대한 기능을 분석
    - 고수준 설계 구조 생성
  - 요구사항 분석
    - 시스템 구조를 분석하여 추상화 모델 작성
    - 비기능/기능적 요구사항 분석
  - 문서화
    - 요구사항, 설계 정보를 검토(Review)
    - 문서화
      - SRS(Software Requirements Specification), SDD(Software Design Description) 등



4. 결론 및 향후 연구

4. 결론 및 향후 연구

- HELISCOPE Project의 OFP를 위한 V&V 절차
  - 표준을 참조하여 V&V 절차 구성
    - 소프트웨어의 위험성을 분석하여 SIL정의
    - SIL에 따라 검증 기법을 다르게 적용
  - HELISCOPE Project의 OFP 특징 고려
    - 높은 SIL에 대응하기 위한 정형 검증 기법 구성
    - 코드 분석, 테스트 기법으로 소프트웨어 정확성 확보
    - 역공학 과정을 통하여 요구사항, 설계 정보 보완

- 향후 연구
  - HELISCOPE Project의 OFP를 대상으로 절차에 따라 V&V 수행