2010년 8월 28일

Visual Studio Camp #1

소프트웨어 품질 향상을 위한 다양한 테스트 기법

엄준일
Microsoft Visual Studio ALM MVP

NCsoft

Agenda

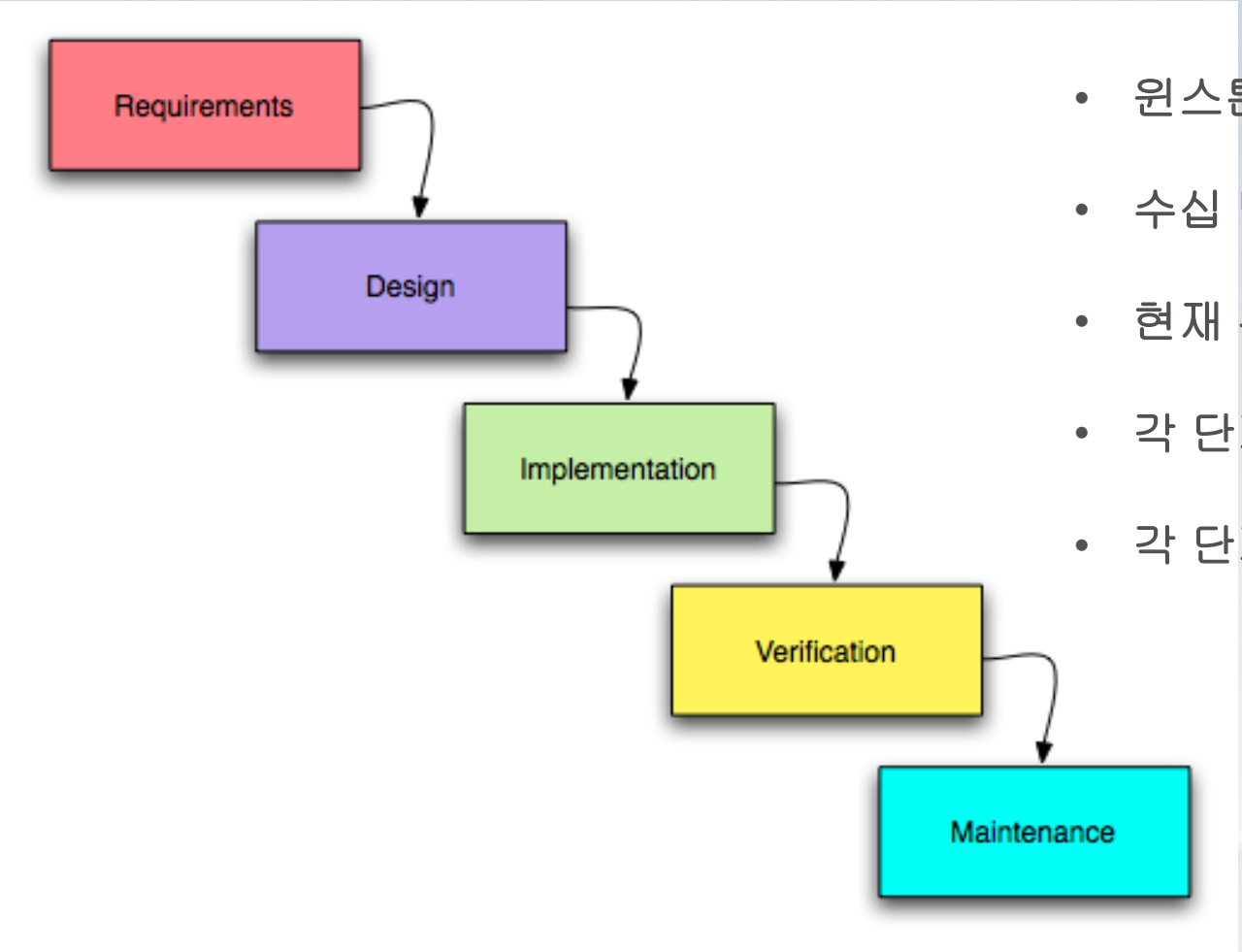
목표

소프트웨어 품질 향상을 위한 여러 가지 개념과 테스팅 기법을 알아봅니다.

- 왜 테스팅을 해야 하나?
- 테스팅 기법
- 여러 가지 테스팅 데모
  - 단위 테스트
  - 메뉴얼 테스트
  - 성능 프로파일
  - 부하 테스트

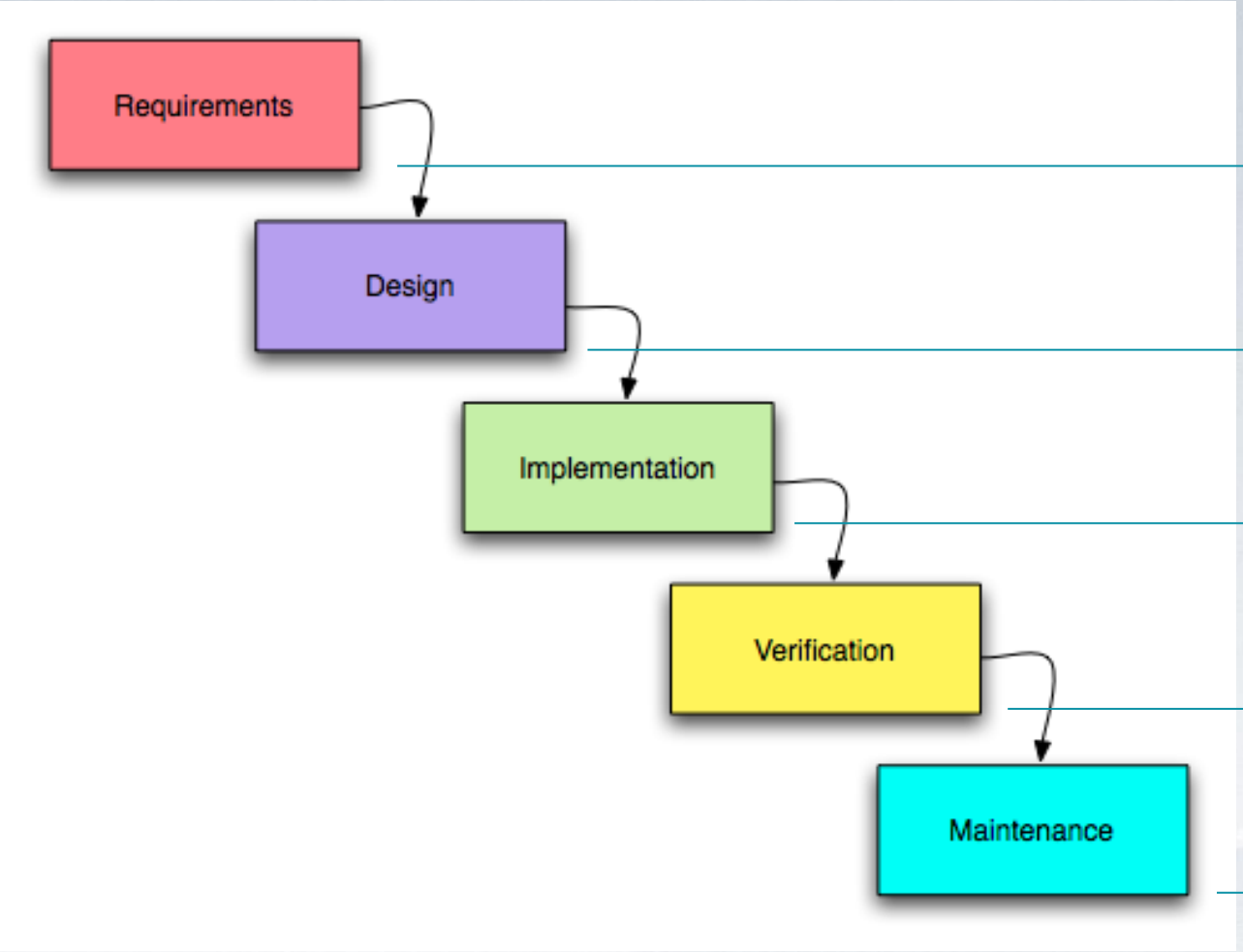
왜 테스트를 해야 하나?

폭포수 개발 프로세스(Waterfall Development Process) 란?



- 윈스톤 로이스(**Winston Royce**) 에 의해 만들어졌으며,
- 수십 년 동안 사용되어 지고,
- 현재 우리나라에서 흔히 볼 수 있는 프로세스로,
- 각 단계는 순차적으로 진행된다.
- 각 단계가 완료되지 않으면 다음 단계가 시작되지 않는다...

폭포수 개발 프로세스(Waterfall Development Process)



- 요구사항 수집
- 분석

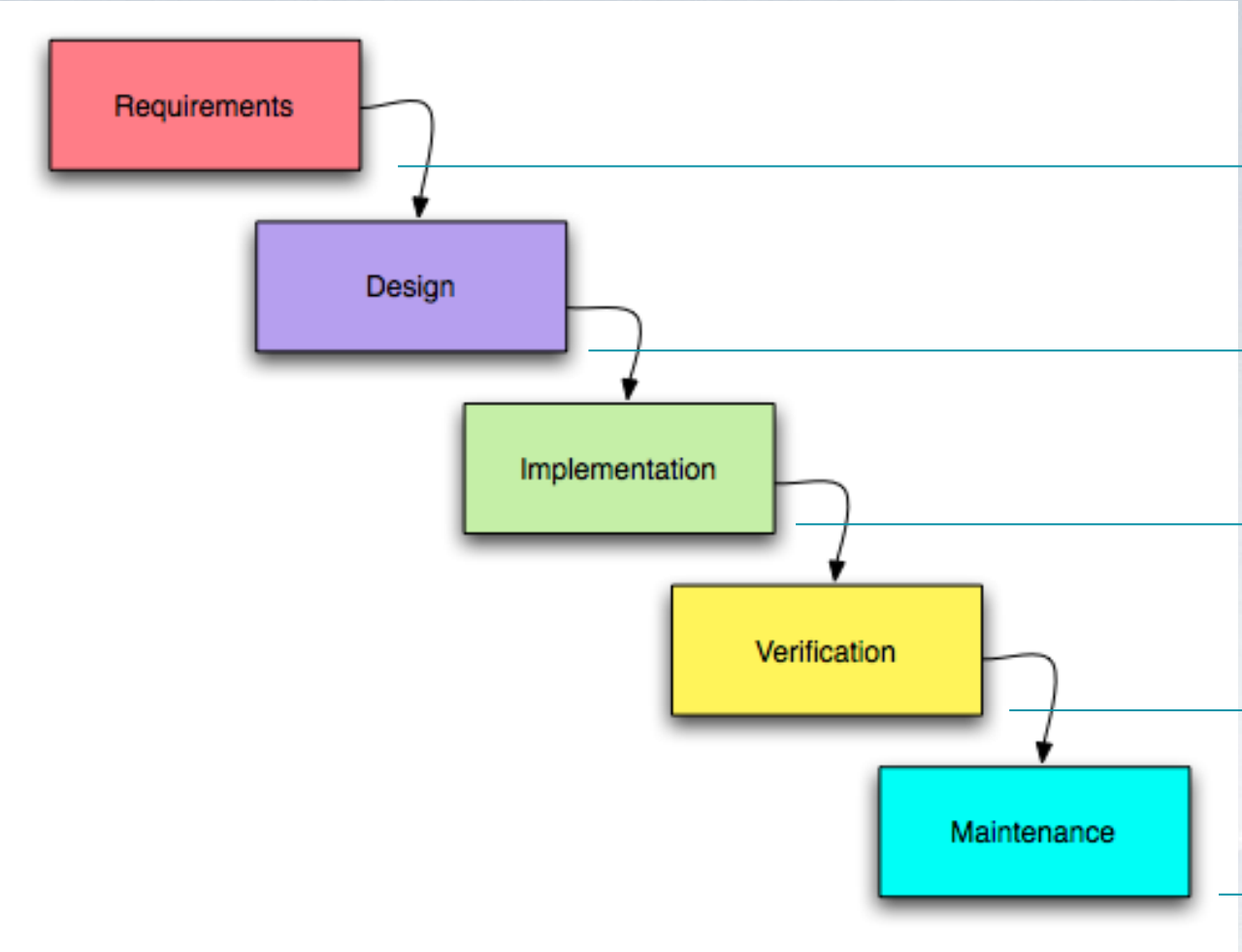
- 설계 및 디자인
- 계획 수립

- 구현

- 단위 테스트 / 통합 테스트
- 문서화

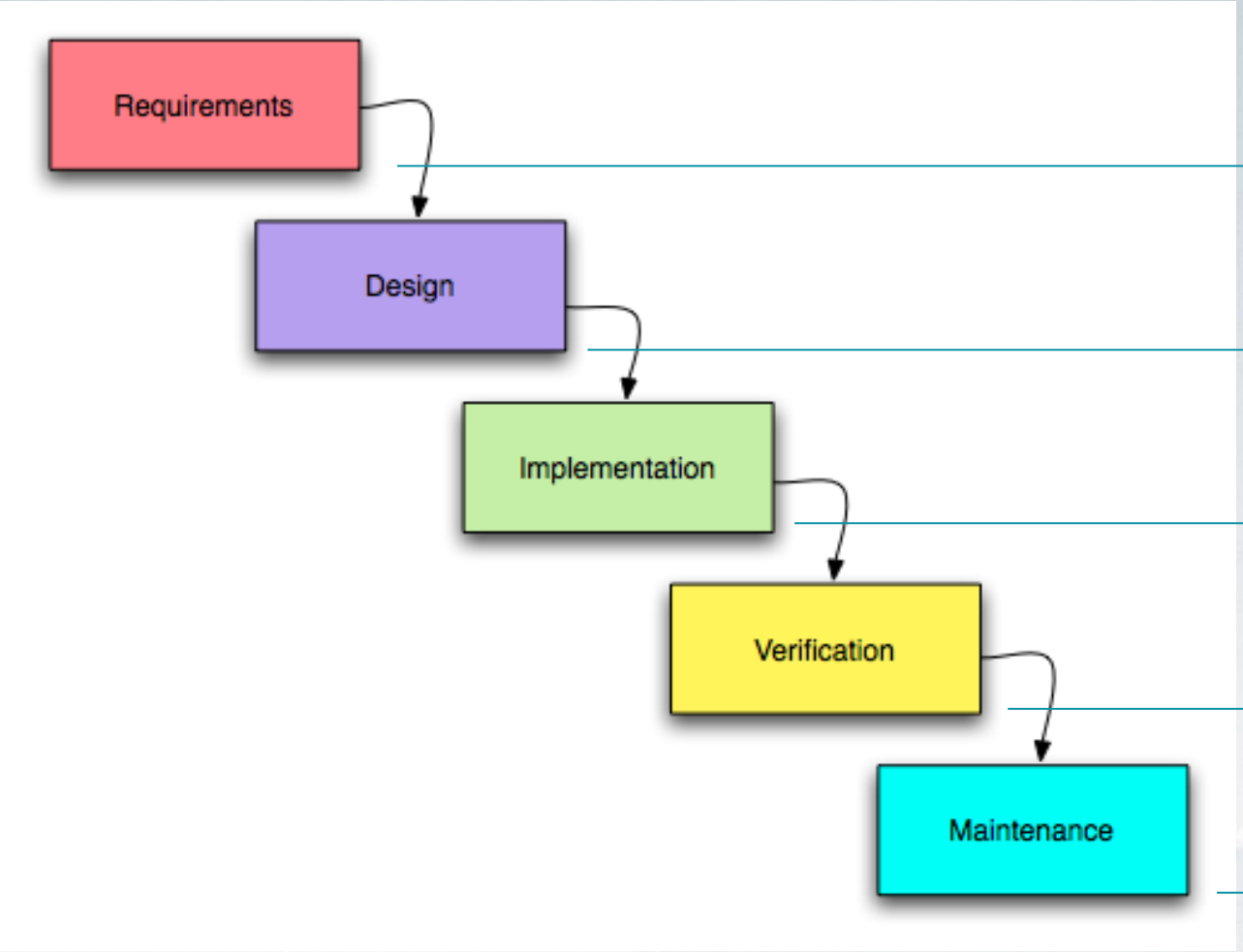
- 배포 및 인수인계
- 유지/보수

폭포수 개발 프로세스(Waterfall Development Process) 장점



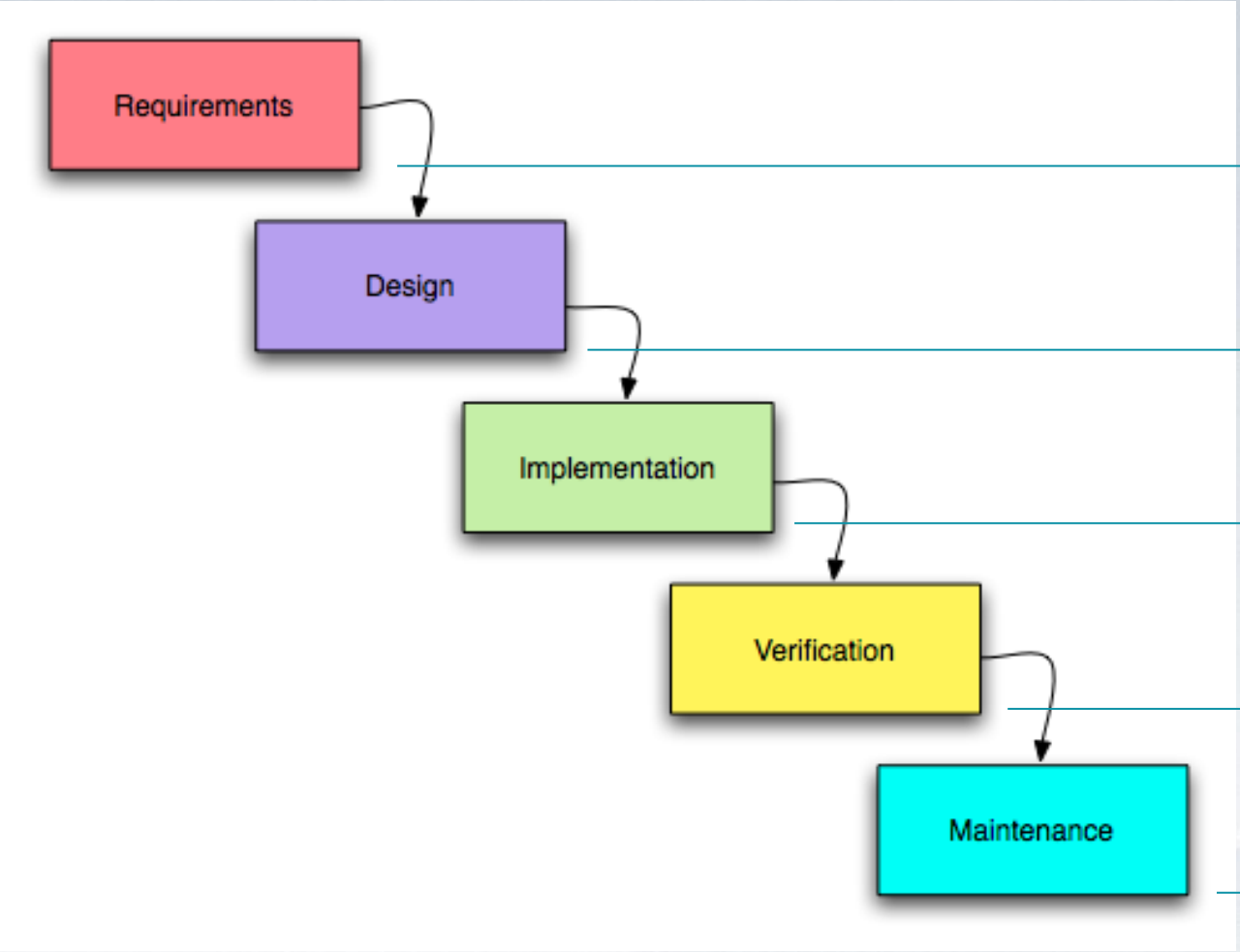
- 초기 요구사항을 기반으로 Needs 를 파악
- 요구사항에 꽉 맞는 소프트웨어 디자인/계획 수립
- 기능 명세로 요구사항/Needs 명확한 구현
- 구현된 기능의 테스트로 결함/버그 제거
- 잘 만들어진 소프트웨어 운영 ^^

폭포수 개발 프로세스(Waterfall Development Process) 단점



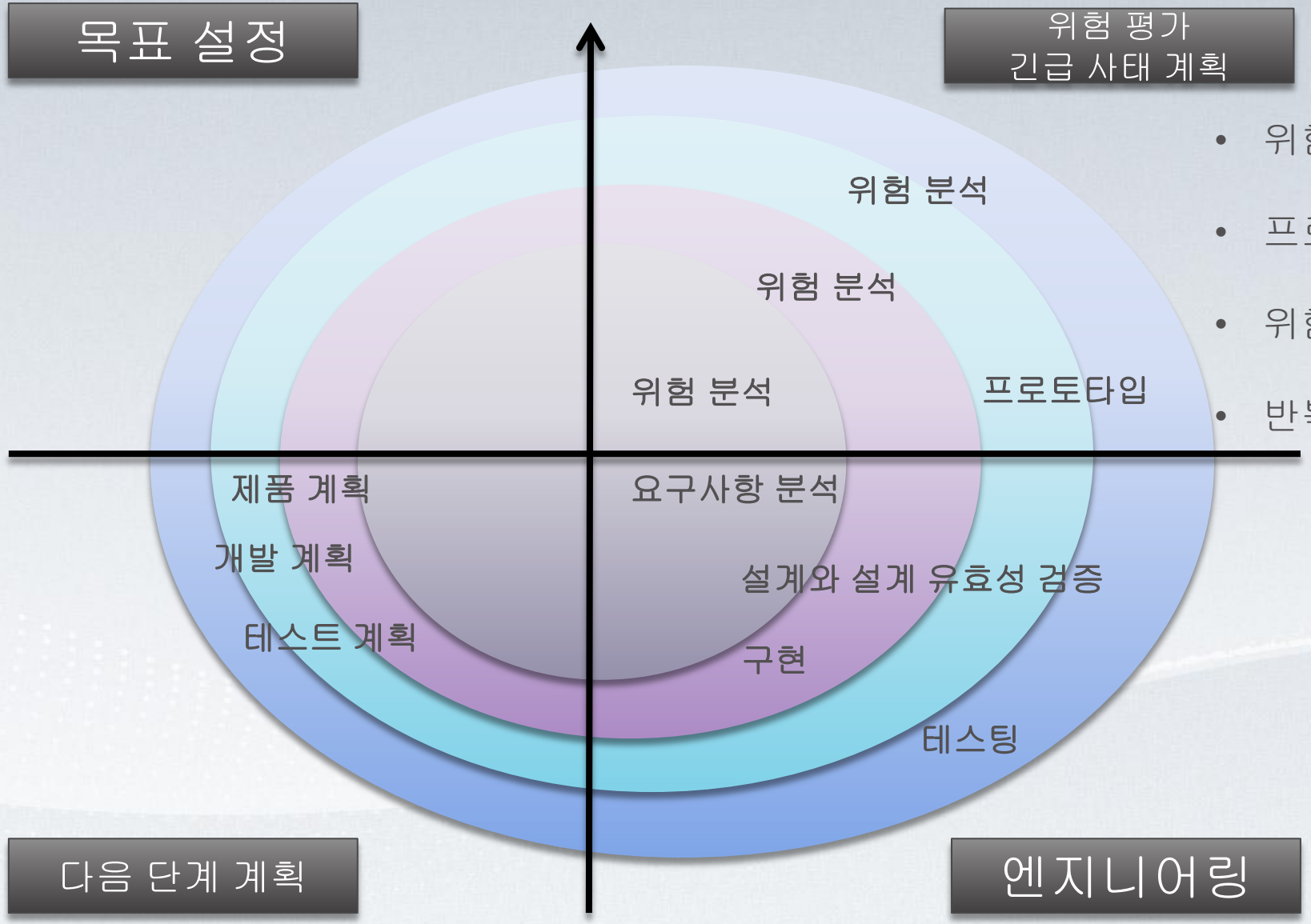
- 초기 요구사항은 완벽할 수 없다
- 요구사항이 완벽하지 않으면 디자인 작업을 할 수 없다
- 구현 중에 요구사항/디자인이 변하는 것은 개발 프로세스 위배
- 잘못된 디자인과 구현으로 올바른 검증이 불가능
- 이제부터가 진정한 개발 시작!!

폭포수 개발 프로세스(Waterfall Development Process) 현실



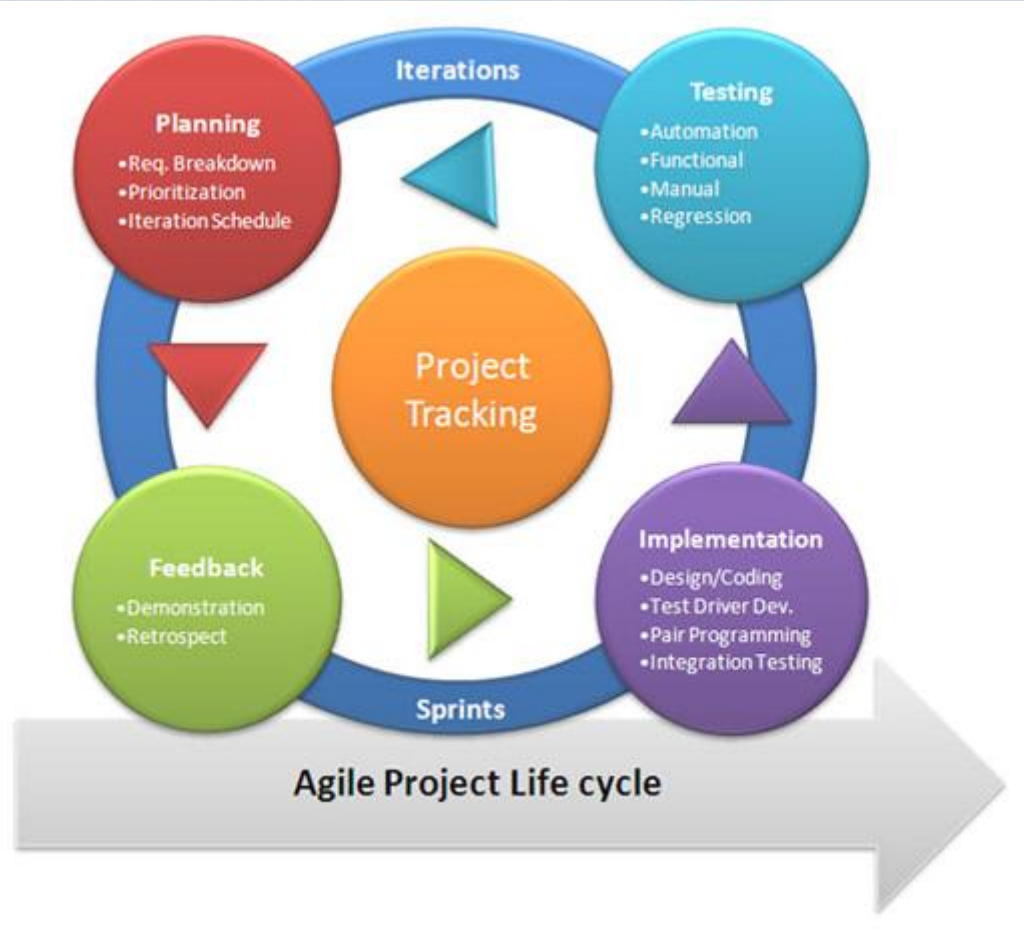
- 고객/사용자도 모르는 자신의 요구사항
- 그나마 분석/설계된 어정쩡한 디자인
- 어정쩡한 디자인으로 구현하지만, 드디어 본 모습을 들어내는 고객의 다이나믹한 Needs
- 변화하고 진화하는 고객 요구사항 반영 오류만 안나면 일단 테스트 성공^^;
- 불쾌한 소프트웨어로 고객의 피드백 수렴 버그 수정, 안정화 작업

베리 뷤(Barry Boehm)의 나선형 모델



- 위험을 줄이고 최소화 하기 위해,
- 프로토타입을 만들어보고,
- 위험 요소를 분석/평가 하여
- 반복하는 개발 프로세스

애자일 개발 프로세스


Iterations
Planning
•Req. Breakdown
•Prioritization
•Iteration Schedule
Testing
•Automation
•Functional
•Manual
•Regression
Project Tracking
Feedback
•Demonstration
•Retrospect
Implementation
•Design/Coding
•Test Driver Dev.
•Pair Programming
•Integration Testing
Sprints
Agile Project Life cycle

즉, 모든 마지막 단계는 바로 테스팅이다.

검증되지 않으면 불량품일 수도 있다.

그러나...

살충제 페러독스

테스팅 기법

다양한 테스팅 기법

기능 테스트
동등 클래스 분할
경계 값 분석
조합 분석
구조적 테스트
블록 테스팅
결정 테스팅
조건 테스팅
기본 경로 테스팅
리스크 분석 테스트
사이클로매틱 복잡도 분석
모델 기반 테스트
모델 설계
베이시안 도해 모델
페트리 넷
비기능 테스트
부하 테스트
성능 테스트
스트레스 테스트
호환성 테스트
접근성 테스트
사용성 테스트
보안성 테스트
SaaS/S+S 테스트
인터넷 기반
서비스/패키지 기반
에뮬레이션

블랙박스, 화이트박스, 그레이박스 테스팅

Input

Output

코드레벨 이해 없이 접근하여 테스트
- 동등 클래스 분할
- 경계 값 분석
- 조합 분석

소스 코드 기반 테스트
- 블록 테스팅
- 결정 테스팅
- 조건 테스팅
- 기본 경로 테스팅

- 테스트 설계 시 사용자 관점 -> 블랙박스 접근법
- 테스트 효율성을 고려한 -> 화이트박스 접근법

테스트 방법과 대상 선정

데이터 중심 단위 테스트

- 데이터의 트랜잭션을 보장 향상
- 데이터의 통일성 보장 향상
- 데이터의 품질 향상

- 데이터베이스의 부하 증가
- 더미(Dummy) 데이터의 증가
- 테스트 실행 성능 저하

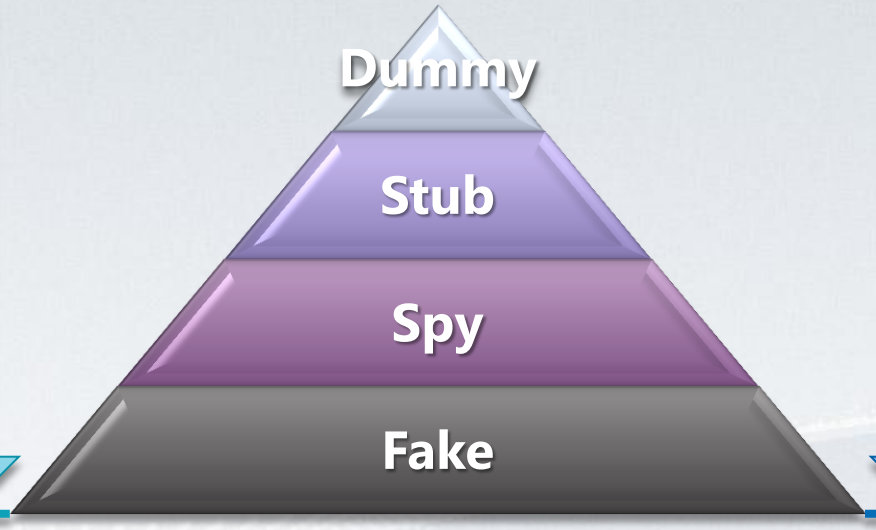
사용자 중심 단위 테스트

- 최소한의 런타임 동작을 보장
- 관련 있는 컴포넌트간의 연관된 오류 최소화
- 컴포넌트간에 자연스럽게 결합도 감소

- 사용자 동작 시 오류 최소화
- 런타임 오류 최소화

- 테스트를 위하여 추상화된 패턴을 강제화
- 패턴을 통한 리팩토링 양의 증가로 생산성 저하

- 테스트의 정확도 증가
- 소프트웨어 품질 향상



- 테스트 복잡도 증가
- 테스트 코드 양의 증가

테스트의 복잡성을 단순화

BDD + Mock

테스트 교육 및
테스트 전문가/팀 필요

여러 가지 테스팅 데모

Thank you.

한국 Visual Studio 공식 팀
http://vsts2010.net
@vsts2010