



Iteration 기반 소프트웨어 개발 프로세스

김강모
(kangmo@nanolat.com)

Agenda

- Part 1. **소개**(Introduction)
- Part 2. **계획**(Planning)
- Part 3. **설계**(Design)
- Part 4. **구현**(Implementation)
- Part 5. **테스트자동화**(Test Automation)
- Part 6. **검증**(Verification) 및 **문서화**

Agenda - Part 1. 소개(Introduction)

- **개발 프로세스의 필요성**
 - 정해진 인력/시간투입 => 고품질의 소프트웨어
- **Iteration 기반 개발 프로세스**
 - 계획-설계-개발-테스트-검증-평가-개선을 여러 차례 반복
- **단계별 산출물 및 개발 툴**
 - 각종 문서의 형식, 키포인트
- **Redmine 소개**
 - 웹상에서 스케줄을 실시간으로 관리
- **PM의 역할**
 - 요구사항 정립, 스케줄 관리, 설계 방안 제시, 리스크 관리, 측정

Agenda - Part 2. 계획(Planning)

- PM
 - 요구사항 분석
 - 개발자 할당
 - 추정시간 산출
 - 개발자 변경
 - 스펙제외(Feature Cut)
- 개발자
 - 요구사항 이해
 - 기반기술 파악
 - 개발툴 실습
 - 프로토타이핑
 - 설계안 브레인스토밍

Agenda - Part 3. 설계(Design)

- PM
 - 아키텍트 역할(Architect Role)
 - 추정시간 재산출
 - 스펙제외(Feature Cut)
 - Spec에 빠진내용 없는지 요구사항 검증
- 개발자
 - 프로토타이핑 개선
 - 설계문서 작성, 리뷰
 - 설계보강
- 모두
 - Simple! Simple! Simple!!!
 - 더 간단하게 할 수는 없는가?
 - 그렇게 (복잡하게) 해야 하는 이유는 무엇인가?

Agenda - Part 4. 구현(Implementation)

- PM
 - 개발방안 제시(기반기술, 구현방법, 테스트 방법, 버그관리)
 - 코드리뷰
 - 요구사항 부합 확인
 - 빠진 구현 항목 없는지 확인
 - 각종 테스트 실시
- 개발자
 - 구현계획 공유
 - 구현
 - 코드리뷰 (툴 : Eclipse)
 - 버그등록 및 해결

Agenda - Part 5. 테스트자동화(Test Automation)

- PM
 - 테스트 방안 제시
 - 테스트 자동화 정도 측정
 - 개발자별 Line Coverage 측정
 - 테스트 계획 리뷰
- 개발자
 - 테스트 계획 공유
 - 테스트 자동화 (툴 : Google Test Framework)
 - 테스트 정도를 Line Coverage로 측정 및 개선 (툴 : GCOV)
 - 성능 튜닝 (툴 : GPROF)

Agenda - Part 6. 검증(Verification) 및 문서화

- PM
 - 모듈별 Line Coverage 측정
 - 정적분석 (툴 : Valgrind)
 - 매뉴얼 레이아웃 결정
- 개발자
 - 코드 리뷰
 - 설계 리뷰
 - 테스트 리뷰
 - 매뉴얼 작성