

2013. 6. 17. [제56호]

# 애자일을 새롭게 적용하는 개발팀을 위한 세 가지 기본원칙

소프트웨어공학센터 경영지원TF팀

## C o n t e n t s

- ▶ 서론
- ▶ 1. 애자일 회고
- ▶ 2. 전사적 테스트
- ▶ 3. 자체편성-자기주도적 팀
- ▶ 4. 결론 : 애자일 개발 원칙의 중요성

**Key Message**

SW개발팀이 애자일 적용을 통해 처음 개발을 시도할 때, 어디서부터 시작해야 할지 몰라 핵심 포인트를 잃어버리는 경우가 있음. 애자일 방법론 도입초기에 개발팀이 출발점으로 잡아야하는 세 가지 기본원칙인 애자일 회고(retrospectives), 전사적 테스트(whole-team testing), 자체편성-자기주도적 팀(self-organized, self-directed teams) 등을 소개함.

▶ **서론**

- 팀이 애자일 적용을 통해 개발을 시작할 때, 출발점을 찾는 것은 벅찬 임무일 수 있음
  - 애자일 숙련자들이 애자일 기술을 완성하는 것에 대한 적절한 지점에 대해 논의할 때, 초보자들은 기본적인 애자일 개발의 원칙들조차 놓칠 수 있음
  - 애자일 적용 초보자들은 "만약 애자일 테스트를 적용하기를 원한다면 어디서부터 시작해야 할까?, 어떤 애자일 기술을 먼저 시작해야 하나?"등의 질문을 할 수 있음
- 지난해 스웨덴 말모의 Øredev 개발자 컨퍼런스가 열리기 전, 참가자들이 5분 단위로 주제에 대해 논의하고 투표하는 모임 형식인 "린 커피(Lean Coffee)"로 불리는 것을 개최했을 때, 이러한 질문들이 제기됨
  - 여기서 애자일 회고, 전사적 테스트, 자체편성-자기주도적 팀이라는 세 가지 애자일 개발 원칙들을 실제로 약 60초 만에 동의하게 되었으며, 이러한 세 가지 애자일 출발점들을 각각 소개하고자 함

▶ **1. 애자일 회고**

- 애자일 회고(retrospective)란 프로젝트 프로세스를 면밀히 살펴보고 배우며, 해결책을 모색하거나 새로운 방법을 시도하기 위한 계획을 수립하면서 일의 진행사항을 검토하는 것임
  - 때때로 "사후 분석(post-mortem)"또는 "사후 강평(AAR: After Action Review)"이라고 불리는 애자일 회고는 대단히 효과적임
  - 그것은 실제로 발생한 것에 기반을 두는 실증적인 것으로 새로운 과정으로써가 아니라 실험적인 것을 실행함으로써 시도하는 것이 필요함

- 애자일 회고의 가장 단순한 형태는 프로젝트에 대해 지금까지 무엇이 맞고 틀린지 그리고 다음에는 무엇을 바꿀지에 대해 모든 팀원이 확인하도록 하는 것임
  - 이러한 방식으로 틀이 잡히면, 무책임한 비난과 음울한 과거에서 벗어나 개선에 초점을 맞춘 미래지향적 논의를 지속하는 것이 과제가 됨
  - 예를 들어 이러한 방식을 사용한 애자일 회고는 팀을 개선하기 위한 아이디어들의 목록을 작성하고, 팀원들에게 그들이 가장 선호하는 것 세 가지를 선택하게 한 뒤 리스트를 추려서, 목록에서 순서가 높은 몇 가지를 실제로 진행하는 것에 중점을 둠

## ▶ 2. 전사적 테스트

- 다른 애자일 개발 원칙은 전사적 테스트(Whole-team testing)으로 이것은 전체 프로젝트팀이 테스트 전략을 결정하고 결과에 대한 책임을 분배하는 조직적 계획임
  - 이것은 사무실 한편에서 개발자 혼자 테스트하는 대신 테스트 대상, 테스트 방법, 자동화 대상, 검토 대상을 결정하기 위해 테스터가 분석가, 프로그래머, 관리자 등과 함께 일하는 것을 의미함
- 기존의 개발자가 사무실 구석에서 벗어나서 작업하고 스스로의 애자일 회고를 실행하며 "품질 보장"에 책임을 지는 "테스트 팀"인 애자일 조직에 들어갔을 때 불편할 수도 있음
  - 이러한 방식으로 성과를 내는 것은 가능하지만, 팀이 책임을 나누지 않는다면 빠른 프로그래밍이 항상 제품의 빠른 출시를 의미하지 않고 어떤 사람들은 야근을 하고 어떤 사람들은 업무를 마쳤기 때문에 4시에 퇴근하는 이상한 최적화가 될 수도 있을 것임
- 전사적 테스트를 수용하는 것은 단지 하나의 업무만을 완수하는 것이 아니라 소프트웨어를 실행시키는 것으로, 팀의 각 부분의 목표를 변화시키는 것을 의미함
  - 즉, 프로그래머들의 작업은 소프트웨어가 코드화 되었을 때가 아니라 소프트웨어가 실행될 때 완성되는 것임
  - 리스크 프로파일(risk profile)에 대한 전혀 다른 면들을 팀원들에게 제공함으로써, 전사적 테스트는 책임전가를 방지할 수 있고, 테스터들이 다른 팀원들의 생각을 이해하도록 함

### ▶ 3. 자체편성-자기주도적 팀

- 애자일 회고 이후 변경 사항이라는 힘든 부분이 발생함
  - 팀은 테스터를 분석가로 또는 분석가를 테스터로 다시 선임하길 원할 수도 있고, 테스터는 빌드 시스템(build system) 운영을 시작하기를 원할 수 있으며, 사실을 기반으로 하는 품질관리(QA)를 기다리는 대신 프로그래머가 스토리를 완성하자마자 "완성(done)"을 입증하는 자동 점검 실행을 프로그래머가 결정할 수도 있음
  - 이것이 자체편성-자기주도적 팀(Self-organized, self-directed work teams)이 작업하는 방법으로, 그들은 변경 사항을 실행하고, 변경 사항이 만들어 낸 결과물에 의해 다음 애자일 회고에서 변화를 판단함
- 계획 지향적인 팀은 승인을 필요로 함
  - 그들은 변경 사항에 대한 문서를 필요로 하거나 이번 주에 휴가인 사람으로부터 승인을 요구할 것이고, 조직의 경우는 외부조직에게 지분매입을 요구할 것임
  - 무슨 이유든지 간에 계획 지향적 팀들은 변경 사항에 대해 스스로 결정할 능력을 가지고 있지 않으며, 이러한 결단력의 부족은 애자일 적용에 있어 효율성을 저해함

### ▶ 4. 결론: 애자일 개발 원칙의 중요성

- 계속적으로 새로운 기술들이 등장하기 때문에, 1년 이상 지속되는 기술에 관한 기사를 작성하는 것은 어려움
- 애자일 회고, 전사적 테스트, 자기주도적 팀은 실제로 발생하는 그날의 문제들에 기반을 두어 팀원들 스스로 전략을 명확히 하도록 하기 때문에 중요함
- 애자일 회고는 실행으로 바꾸는 것에서부터 시작하고, 전사적 테스트는 팀이 전체 팀의 문제로서 품질에 대해 생각하도록 하며, 자기주도적 팀은 이러한 변화들을 만드는 역할을 부여받음
  - 이러한 세 가지 애자일 개발 원칙들이 애자일의 모두는 아닐 수 있지만, 그것들로부터 시작한다면 소프트웨어 품질팀은 적합한 애자일 전략들을 얻을 수 있을 것임
- 애자일 테스트의 형식보다 내용에 대해 만족할 만한 리트머스 테스트를 원한다면, 애자일 회고, 전사적 테스트, 자체편성-자기주도적 팀이 적합함

### 참고 자료

1. <http://net.tutsplus.com/articles/general/the-principles-of-agile-development/>
2. [http://community.dynamics.com/b/executive\\_insight/archive/2013/03/25/innovate-to-stay-relevant-the-power-of-agile-development.aspx](http://community.dynamics.com/b/executive_insight/archive/2013/03/25/innovate-to-stay-relevant-the-power-of-agile-development.aspx)
3. <http://www.inquid.com/how-%E2%80%9Cagile%E2%80%9D-is-prince2-as-a-project-management-methodology-and-framework.html>