

애자일 스크럼 팀이 SW 품질과 테스트에 주력하게 만드는 3가지 방안

2012. 10. 22. (제29호)

목 차

- I. 서론
- II. 강력한 의사소통 라인 구축하기
- III. 유연한 테스트 타이밍
- IV. 빠르고 효과적인 테스트 방법
- V. 결론

Key Message

SW품질과 테스트는 애자일 방법론을 도입하는 된 핵심적인 이유이다. 애자일을 도입한 개발팀이 산출물의 품질향상을 위해 필요한 의사소통라인 구축, 유연한 테스트 타이밍, 빠르고 효과적인 테스트 방법을 살펴본다.

■ 서론

- 애자일 방법론을 성공적으로 도입한 뒤 개발에 초점을 두고 테스트는 신경을 덜 쓰는 경향이 있음
 - 산출물이 매 주기 발전되면서 요구사항의 변화도 잦고, 변화에 빠르게 대응해야 함
 - 개발의 속도, 유연성과 더불어 효과적이고 집중된 테스트와 균형을 이뤄야 함
 - 스크럼 팀은 최대한 효과적이고 철저하게 테스트를 하면서 품질을 향상시켜야 함
- 스크럼 팀과 의사소통 하여, 테스트를 유연하고 빠르고 효과적으로 만들어서 테스트 팀이 테스트에 집중할 수 있도록 해야 함
 - 질 높은 산출물을 만들 수 있는 스크럼과 테스트 팀의 능력을 향상시키는 세 가지 방법을 살펴볼 것임

■ 강력한 의사소통 라인 구축하기

- 개인 간 소통만이 아니라 팀의 비즈니스 의사소통도 혁신시킬 필요가 있음
- 개발하면서 결함이 발견되기 전에, 요구사항단계부터 변화가 발생하면 테스트 팀과 소통을 하는 것이 좋음
 - 예) 개발자가 모르는 워크플로우(Workflow)를 발견, 기능이 실행되지 않거나 문제 발생
 - 주기 동안에 이슈를 발견하고 고치는 것이 애자일 방식임
- 의사소통의 원활함을 위해서는 스크럼 팀 구성원을 편안하게 만드는 것이 중요함
 - 팀원 서로가 편안하게 대화할 수 있도록 만들기
 1. 팀원끼리 서로 소개시키기
 2. 스크럼 미팅에 매일 2분 정도씩 각 팀원이 좋아하는 활동, 취미 등을 공유하기

3. 매니저는 대화를 이끌어내고, 리드하고, 활성화시키기
 4. 일상의 사소한 질문들로 단조로움을 깨고 서로 편하게 말하게 하기
 5. 재미요소를 더하고 편안한 소재로 대화를 자극하기
- 팀원끼리 편안해 지면 서로 중요한 상세사항을 자연스럽게 공유하게 됨
 1. 이슈나 변화가 본인의 일에 미치는 영향뿐만 아니라 다른 이들에게 주는 영향까지 생각하게 됨
 2. 팀이 한번 의사소통 하게 되면 정보를 공유하는 것이 더욱 쉬워짐

■ 유연한 테스트 타이밍

- 테스트 팀 매니저로서 회의를 계획하고 진행하는데 시간을 좀 더 투자하는 것이 중요
 - 이메일로 요청사항을 전달하기보다 직접 대면하여 정보 얻기
 - 스크럼 팀 안에서 테스트 팀원이 하는 일과 친숙해지기
- 엄격함보다는 유연성을 가지고 결정하기
 - 각 유저스토리가 개별의 테스트 케이스를 가져야 한다는 생각 버리기
 - 통합할 수 있는 테스트 케이스들은 통합하기
 - 테스트 팀을 믿고, 언제 어떻게 테스트 케이스를 쓸지를 유연하게 결정하기
 - 팀원들을 테스트에 집중할 수 있도록 만들고 불필요한 프로젝트에 참여시키지 말 것
- 테스트를 수행하는 인력을 관찰하는데 시간을 투자하기
 - 멘토가 되어 코치하여, 주기 전반에 걸쳐 테스트와 개발이 진행되도록 현실적인 해법으로 찾아낼 것

■ 빠르고 효과적인 테스트 방법 적용하기

- 빠르고 효과적인 테스트를 수행하는 테스터는 부지불식간에 수동적(자동화 반대 개념 측면에서의)이고 자기만의 기술을 활용하여 테스트를 수행하는 경향이 있음에 따라 그들이 일을 하는 것을 관찰하여 어떻게 테스트를 수행하는지를 배워야 함

- 워크플로우의 어느 부분에 포함되는가?
- 주기 동안에 얼마나 깊게 진행하는가?
- 지금은 그냥 넘어가고 나중에 좀 더 심도 있는 테스트를 수행하는가?
- 변화하고 있는 워크플로우의 작은 부분에 먼저 집중하는 것이 좀 더 효과적임
 - 리그레이션(regression) 사이클 동안이나 다음 주기의 일부분으로써 이후에 좀 더 충분한 테스트를 더함
 - 예) 많은 테스터가 테스트를 만들 경우, 먼저 스프레드시트를 만들고 나중에 스프레드시트를 테스트 툴에 통합한 후, 모든 주기 사이클이 완성되었을 때 공식적인 테스트 케이스를 만들
 - 주기에 필요한 부분만을 테스트 하면서 계속적으로 테스트 케이스를 더함
- 밑바닥에서 건물을 설계하는 것과 같음
 - 토대에서 시작하여 토대를 수정하거나 결함을 제거해 나감
 - 토대를 만들고 구조의 다른 부분을 설계해 나감
 - 결국, 한번 각 조각들이 모두 테스트 되면 각 조각들로 전체 테스트를 만들어 결국에는 전체 구조를 체크하는 테스트를 만들게 됨

■ 결론

- 비즈니스 의사소통 향상시키기, 테스트 타이밍과 구조에 유연성 더하기, 각 주기를 진행하면서 테스트 만들기를 통해 적합하게 테스트된 산출물을 만들 수 있음
- 더불어 다른 사람과 의사소통하는 방법을 이해한 팀원을 얻게 됨
- 주기테스트를 만들고 각 테스트로 각 부분들을 테스트하고 마지막에 각 테스트 조각들을 전체 테스트로 통합하면 깊이 있고 효과적인 테스트를 얻을 수 있음

<참고 자료>

1. Agile Testing and Quality Strategies: Discipline Over Rhetoric, Ambyssoft(2012), <http://www.ambyssoft.com/essays/agileTesting.html>
2. Agile Testing: Key Points for Unlearning, ScrumAlliance(2012), <http://www.scrumalliance.org/articles/392-agile-testing-key-points-for-unlearning>
3. Top 5 Common Challenges for Agile Testing Teams, SmartBear(2012), <http://searchsoftwarequality.techtarget.com/tip/How-to-focus-an-Agile-Scrum-team-on-quality-and-testing>