

Fast Campus - 웹 프로그래밍 SCHOOL

Day 6. Homework / by. 장기수

'기술적 부채(Technical Debt)' 란?

기술 자본이 많지 않은 개인이나 집단이 결과물을 빠르게 얻기위해 포기하는 많은 것들이 결국에는 '부채(Debt)'로 돌아온다는 것이다. 모든 프로젝트는 어느정도 기술적 부채를 만들어낼 수 밖에 없지만 빠르게 결과물을 얻고자 하면 테스트나 QA가 소홀해 질 수 밖에 없다. 만약 이 부채를 방치하다가 제품에서 이슈가 발생하면, 그것을 보정하기 위해 들어가는 추가 공수는 일종의 '이자(Interest)'가 되어 개인이나 집단에 큰 부담을 준다.

기술적 부채를 줄여나가기 위하여 Continuous Integration(CI), Test Driven Development(TDD), Refactoring, Pair Programming, Code Review, Checkup과 같은 개발 방법을 적용할 수 있는데 주로 애자일의 XP(익스트림프로그래밍)에서 제시하는 방법들이다.

한편, 기술적 부채는 어찌보면 프로젝트를 진행함에 있어서 당연하게 발생하는 부산물이며 기술적 부채를 과도하게 줄이고자 한다면 비즈니스의 속도에 유연하게 대처하지 못할 수 있다. 때문에 경영진과 의사결정권자들은 기술부채와 개발속도간에 균형을 유지할 수 있도록 고민해야하며, 개발자 및 실무진들은 XP의 기법들을 이용하여 기술적 부채가 커지지 않도록 해야한다.

참조.

http://blog.lyuwonkyung.com/technical_debt/

<http://www.ciokorea.com/slideshow/19128?slide=5>

<http://egloos.zum.com/swprocess/v/2828873>

