

# 과 제 제 출

일 자 : 2016. 5. 2.

제 출 자 : 황 원 주

## ‘기술적 부채(Technical Debt)’에 대해 알아보기

중학교때 재미있는 선생님 한분이 개그우먼 사진 한장과 미모의 여배우사진 한장을 학생들에게 보여주면서 하신 말씀이 있다. “공부에는 때가 있다. 지금 시간의 노력이 미래의 배우자 얼굴을 좌우한다.” 지금의 노력이 나중에 결과로 나타난다는 말을 은유적으로 표현 하시기 위한 얘기였을 것이다.

기술적 부채에 관해 조사하면 조사할 수록 머릿속을 떠나지 않는 개념이 있었다. 바로 모럴 해저드”Moral Hazard”다. 모럴해저드는 금융에서 나온 개념으로 정보를 가진 자와 정보를 가지지 못 하자가 서로 게임을 할 때 정보를 가진 자가 정보를 가지지 못한 자가 모르는 정보를 이용하여 본래의 업무에 최선을 다 하지 않음으로써 발행하는 부작용 내지는 문제점 때문에 정보를 가지지 못한 자에게 피해를 입게 되는 현상을 말한다.

프로그래머 혹은 개발자가 시간 또는 경제적 제약 혹은 도덕적 해이등의 이유로 프로그램을 지저분하게 만드는 행위는 일종의 기술적 빚을 안는 것과 같다는 개념이다. 이 기술 부채는 은행빚과 마찬가지로 이자비용을 발생시킨다. 이는 빠르지만 지저분한 설계와 코드 때문에 나중에 더 많은 추가 작업을 해야 한다는 것을 의미한다. 프로그램 개발이 진행되고 커질수록 부채의 크기 또한 커지는 건 당연한 일 일 것이다.

만들어진 프로그램을 수정하는 일은 새로운 프로그램을 만드는 일 보다 어려울 수 있다. 특히 만드는 사람과 수정하는 사람이 다르다면 이견 두배 이상의 수고가 드는 일이다. 그런데 이런 기술적 부채는 빚을 진사람과 이를 해결하는 사람이 다른 경우가 아주 많다. 예를들면 빚을 진 사람은 최초 개발자인데 갚고있는 사람은 유지보수팀일 수 도 있다.

이런 부채가 만들어 지는 이유를 크게 네가지로 나눈 방법을 ‘기술부채 사분면’에서 설명하고 있다. 그 네가지 경우는 부채가 무분별한가? 신중한가? 에 따라 나누고, 의도되었나? 혹은 우연한가? 에 따라 나눴다.

첫번째의 경우 무분별한 부채가 의도된 경우이다. 예를들어 개발팀이 좋은 설계방법을 알고있고, 능력 또한 있으나 코드를 지저분하게 만든 경우이다. 이유는 물리적 시간 부족 등으로 들 수 있다. 이러한 경우가 생기는 원인은 대표적으로 “아웃소싱”위주의 기업시장 문화에서 볼 수 있다. 보통 관공서나 대기업들은 경제적 또는 관리적(혹은 책임회피)등의 문제로 프로그램 관련 문제를 공개 입찰등의 방법으로 아웃소싱 한다. 이 과정에서 경제 논리가 들어오고 SI개발자들은 본인의 능력과 상관없이 물리적 시간이 부족해진다. 그에 따라 의도하였던 의도하지 않았던 지저분한 코드를 사용하게 되고 이에 따른 부채가 생기게 된다. 원래 해결 방법이라 함은 프로그램이 진행 되는 도중 설계변경등의 과정을 거쳐 빠르게 부채를 청산 하여 원래의 기능을 다 할 수 있게 해야 한다. 하지만 우리나라의 경우 비정상 적인 계약 행태 등(이른바 ‘갑’질)으로 현실상 어려운 경우도 많이 있다 하겠다.

두번째의 경우는 신중하게 결정된 의도된 부채이다. 개발팀이 그 부채를 안고 간다는 사실을 인지하고 있고, 부채를 지는 것보다 일정을 앞당기는 것이 더 효과적이라고 판단했기 때문이다. 예를들면 KTX가 터널내 지하수 수위 조절 설비에 관한 프로그램을 개발한다고 생각 하자. 프로그램을 개발하는 도중 지금은 프로그램이 정상 작동 하지만 몇년 후 수위조절용 모터등 설비 교체시 호환이 어려운 프로세스인걸 감지했다. 이걸 해결하고 싶지만 개통예정일이 얼마 남지 않았고 철도가 멈

춤으로써 발생하는 비용을 고려했을 때 어느것이 효과적인지 판단이 필요할 것이다. 이런 경우 일단 부채를 가지고 프로그램을 실행하고 다음 부채가 발생 하기 전 까지 부채를 해결하는 노력이 필요할 것이다.

세번째의 경우는 최악의 경우인 설계를 할 줄 모르는 개발팀의 무분별한 부채 양산이다. 답이 없는 경우고, 심지어 부채의 액수가 얼마인지도 모른다. 기존 프로그램을 다 버리고 새로 만드는 것이 현명하다 하겠다.

네번째의 경우는 우연하게 발생되었지만 신중한 부채이다. 개발팀이 부채의 내용을 알고 있지만 개발 환경의 변화 등의 이유로 발생한 경우이다. 예를 들면 1년짜리 프로그램 개발을 거의 마치고 결과물이 나올때 쯤 새로운 시스템 혹은 기술이 나오게 된 경우가 있을 수 있다. 이 경우 상황에 따른 판단이 중요하며 부채라기 보다는 기술적 발전에 관한 관심이나 연구에 관한 중요성에 관한 이야기라고 할 수 있다.