

Mythical man month chapter 1 & 2 summary

20200120 차선희

carprefer

Chapter 1. The Tar Pit

큰 프로그램일수록 타르 구덩이(tar pit)에 빠져 헤어 나오지 못하는 경우가 많다. 두명의 개발자들이 창고에서 만든 프로그램은 다른 큰 회사에서 만든 프로그램보다 뛰어난 경우가 많지만, 이 프로그램을 다른 사람들이 사용하고, 발전시키기 위해서는 system화와 product화를 시키는 과정이 필요하다. 그리고 이는 단순히 프로그램 하나를 개발하는데 비해 각각 3배의 노력이 필요하다.

프로그래밍을 통해 얻는 재미들은 다음과 같다. 무언가를 만드는 데서 오는 순수한 재미, 다른 사람들에게 유용한 것을 만드는 데서 오는 재미, 복잡한 퍼즐을 해결하는 듯한 느낌에서 오는 재미, 계속 배움에서 오는 재미, 실제로 움직이고 동작하는 것을 만드는 데서 오는 재미(시와 다르게)를 얻을 수 있다.

프로그래밍을 하다 보면 안 좋은 점들도 경험하는데 이들은 다음과 같다. 프로그램은 완벽하게 동작해야 하는데서 오는 압박, 프로그램이 자신의 통제아래 있지 않은 점, 다른 사람의 잘못된 디자인에 영향을 받는 점, 짜증나는 디버깅, 뒤쳐진 프로그램이 되지 않기 위해 경쟁해야 하는 점이 있다.

Chapter 2. The Mythical Man-Month

많은 프로젝트들이 기한을 못 맞춘다. 그 이유는 예측 기술이 부족하고, 시간을 사람으로 대체할 수 없으며, 일정 모니터링이 제대로 되지 않으며, 시간이 부족할 때 manpower를 더하는 식으로 해결하려고 하기 때문이다.

Manpower를 더했을 때 시간을 줄일 수 있는 상황은 오버헤드가 없을 때의 이야기이다. 오버헤드로는 새로 들어온 사람들을 훈련시키는 시간과 늘어난 소통 시간, 일을 다시 재분배하는 시간 등이 있다. 따라서 프로젝트에 사람들을 늘림으로써 오히려 시간을 늘리는 참사가 날 수 있다.

프로젝트의 일정을 짤 때, 계획, 코딩, 테스트 단계를 거친다. 시간에 쫓겨 테스트 단계를 생략하는 것은 추후에 더 큰 문제를 낳을 수 있다. 프로젝트 일정을 적절하게 설정하지 못하는 경우가 많이 발생하며, 이를 방지하기 위해 소프트웨어 엔지니어들은 생산성과 버그 발생 빈도 등의 정보를 공유해야 한다. 프로젝트가 기한을 못 맞출 것 같다고 manpower를 추가하는 식으로는 문제를 해결할 수 없다. 프로젝트의 멤버 수는 독립적으로 나눌 수 있는 일들의 수보다 많게 해서는 안 된다.