

# manmonth

## Chapter 1 : The Tar Pit

큰 규모의 프로그래밍 프로젝트는 소규모 프로젝트와 종류가 다른 관리 문제를 겪는다. Tar Pit(타르 구덩이)에 갇힌 동물이 탈출할 수 없는 것처럼, 아무리 강한 동물이 오랜 시간 노력을 해도 결국 타르 구덩이가 이기는 것과 같이 큰 프로젝트들도 tar pit에 잡히는 경향이 있다.

우린 두 명의 기업가가 큰 규모의 팀보다 더 좋은 결과물을 내는 것을 알게 되었다. 그렇다면, 왜 큰 규모의 팀들이 진작에 두 명 팀으로 대체되지 않은걸까?

개발자에게 그의 시스템에서 그가 가진 데이터를 이용해서 돌아가는 프로그램을 만드는 것은 그의 것에서만 동작하는 특화된 알고리즘만 있어도 되고 별다른 유지 보수나 실험을 위한 문서가 필요하지 않다. 하지만, product를 만드는 것은 위에서 언급했던 것들을 모두 고려하는 최소 3배 이상의 작업이 요구된다.

단일 프로그램 대신에 같이 동작하는 프로그램의 system을 만들고자 한다면, 다른 프로그램이 입력으로 사용할 수 있는 출력을 생성하고 다른 프로그램과 가능한 모든 조합으로 철저히 테스트하는 과정에서 최소 3배 이상의 작업이 요구된다.

즉 위의 두가지 특징을 고려하면 product의 형태로 팔리는 system은 단일 프로그램에 비해 최소 9배 이상 비용이 필요하다.

## Chapter 2 : The Mythical Man Month

많은 소프트웨어 프로젝트가 다른 모든 이유를 합친 것보다 달력 시간 부족으로 인해 실패하였다.

- 추정 기술들은 모든 것이 잘될 것이라는 가정 하에 덜 발달되어 있다.
- 사람과 시간은 교체할 수 없다.
- 소프트웨어 관리자들은 경영진에 대해 완강히 반대하지 않는다.
- 일정 진행 상황이 제대로 모니터링되지 않고, 다른 분야에서 입증된 기술은 소프트웨어에서 채택되지 않는다.
- 만약 프로젝트가 연기되었다면, 더 많은 사람들이 추가되고 이는 불에 휘발유를 뿌리는 것과 같은 결과를 얻게 된다.

프로젝트에서 더 많은 일들을 처리할 수록 실패할 가능성이 커지고 전체적인 프로젝트가 연기 될 가능성이 커진다. 즉, 큰 프로젝트는 반드시 늦어지게 된다.

Head count를 doubling하는 것은 오버헤드가 없다면 시간을 절반으로 줄여준다. 이때 오버헤드는 다음의 두 가지 경우가 존재한다.

- 기술, 목표 및 팀의 운영 방식에 대한 교육으로 인한 오버헤드.

이는 나뉘질 수 없으며 headcount에 선형적으로 증가.

- 소통에 의한 오버헤드로 2명, 혹은 그 이상의 사람이 모일수록 악화됨.

이 모든 것 때문에, 사람을 추가하는 것은 이상적인  $1/N$ 보다 낮은 비율로 일정을 단축시키고, 실제로 한 지점을 넘어 길어지게 한다.

따라서, 소프트웨어 업무의 스케줄을 짜려면, Planning, Coding, Debugging으로 나누어서 시간을 할당해야 한다.

추정하기 쉬운 작업 부분은 결국 시간이 덜 걸리고, 스케줄로 인한 압박 때문에 테스트에 소요하는 시간을 줄이는 것은 결국 계획했던 것 이상으로 미뤄지는 더 나쁜 상황으로 이어지게 된다.

경영진이 원하는 날짜를 맞추기 위해 무리한 일정을 짜는 것은 다른 공학 분야보다 소프트웨어 분야에서 더 흔하다. 왜냐하면 우리는 하드 데이터가 없고 직감만 있기 때문이다.

소프트웨어 개발자들은 생산성, 버그 발생률, 추정 규칙과 같은 것에 대한 데이터를 더 많이 공유할 필요가 있다.

경영자들은 그들의 척추를 다지고 직감을 방어해야 한다. 왜냐하면 그들은 경영진의 희망적인 생각보다는 낮기 때문이다.

만약 프로젝트가 연기되었다면, 일을 다듬고 시간표를 지키려고 노력해라. 늦은 프로젝트에 인력을 더하는 것은 프로젝트를 연기시킬 뿐이다.

## Reference - Summary

<https://kartick.substack.com/p/summary-of-the-mythical-man-month>