Chapter 2, 3

`Chapter 2`

낙관주의

- 일정대로 잘될 것이라고 낙관적으로 전망하지 마라
    - 버그는 필연적이고 일정은 다른 일정에 꼬리에 꼬리를 물고 있다.
    - 따라서 낙관적인 계획 수립은 실패의 원인이다.

맨먼스

- 소프트웨어 개발에서 인력과 기간은 상호 동등하지 않다.
    - 소프트웨어 개발에서 분업을 하면 커뮤니케이션에 따른 노력이 증가한다.
    - 분업을 할수록 노동자는 늘어나고 커뮤니케이션에 필요한 비용, 노력이 증가하게 되는 것이다.
        - n ⇒ 2 * (0.5n + a)
        - n(n-1)/2 : 상호간 커뮤니케이션일 경우
    - 인력이 추가된다고 일정이 단축되기보단 오히려 연장될 수 있는 것이다.

시스템 테스트

- 디버깅에 절반을 할애하고 계획 수립, 코딩 순으로 시간을 할애하라
    - 테스트에 충분한 시간을 쏟지 않으면 결과가 좋지 않다.
        - 심리적으로도 좋지 않으며 재정적으로도 악영향을 끼친다.

비겁한 견적

- 기간 내에 할 수 있는 견적을 짜라

되풀이되는 일정 참사

- 지연된 프로젝트가 발생 시
    - 낙관적으로 생각하지 말라

    - 맨먼스로 해결할 수 없다. : 새로 투입한 인원 훈련 기간에 시간이 걸린다.
        - 차라리 원래 인원으로 작업 진행하는 것이 낫다.
    - 일정 자체를 가능하도록 수정해야한다.
    - 불필요한 임무를 제거해야한다.
- 인력은 최소로 일정은 최대로라는 원칙으로 계획을 잡아야 한다.

`Chapter 3`

문제

- 뛰어난 프로그래머와 평범한 프로그래머의 능력 차이는 10배 정도라 한다.
- 그러나 대규모 프로젝트에서 아무리 소수 정예라 할지라도 대 인원의 속도를 따라잡을 수 없다면 그들이 만든 프로젝트는 구닥다리가 되어버린다는 문제가 있다.

밀스의 제안, 운영방식

- 대형 프로젝트에서 외과 수술 팀같은 커뮤니케이션 방식을 제안하였다.
- 프로그래밍과 행정, 기록 등의 업무를 나누어 업무의 전문성을 높이고 커뮤니케이션을 단순화 할 수 있다.
- 프로그래밍을 개인적인 업무가 아닌 팀 단위, 즉 사회적 행위로 변화시켰다.