

### Chapter 3. The Surgical Team

좋은 프로그래머와 나쁜 프로그래머는 그 능률이 10배 가까이 차이 난다.

적은 인원의 프로그래머끼리 작업하는 것이 소통 비용 측면에서 더 효율적이다.

적은 인원끼리 작업하는 것은 효율적이거나, 큰 시스템 구축에 있어서 많은 인원이 필요한 것은 사실이다.

Mills의 제안에 따르면 이 일은 surgical team을 구성함으로써 능률을 향상시킬 수 있다. 10명으로 이루어진 팀의 역할은 다음과 같이 나눌 수 있다.

1. Surgeon: 전체적인 디자인과 코딩, 테스트, 문서화작업을 도맡는다.
2. Copilot: surgeon의 보조를 담당한다.
3. Administrator: 프로젝트와 관련된 돈, 사람, 환경 관리를 담당한다.
4. Editor: surgeon의 디자인 초안을 비판하고 재구성한다.
5. Two Secretaries: administrator와 editor 각각의 조수
6. Program clerk: 프로젝트와 관련된 기록과 결과를 관리한다.
7. Toolsmith: 신뢰도 있고 만족스러운 기기 환경을 제공한다.
8. Tester: 프로그램에 대한 테스트 케이스들을 돌린다.
9. Language lawyer: 해당 언어를 마스터하여 유용한 조언을 제공한다.

이렇게 surgical team을 구성하면 각자 구현할 부분을 나눌 필요가 없고, 서로의 디자인이 충돌하는 문제를 막아줄 수 있다.

### Chapter 4. Aristocracy, Democracy, and System Design

성당 건축에서 건축가 각각의 개성을 드러내지 않는 일관된 디자인은 신의 영광을 더욱 드러내준다.

더 간단한 구현과 더 많은 기능들을 어느정도 희생해야 일관성 있는 프로젝트를 만들 수 있다.

일관성을 달성하기 위해서는 적은 인원이 디자인하는 것이 좋지만, 큰 시스템 개발에선 많은 사람의 손이 필요하다. 이에 대한 해결방법은, 첫째로 구조를 짜는 것과 실제 구

현을 하는 사람들을 나누는 것이다, 둘째로는 surgical team을 사용하는 것이다.

구조를 짜는 사람에게 조언자를 한 명 붙여주어 다양한 아이디어를 제시하게 하고, 최종 결정은 구조를 짜는 사람에게 맡겨라.

구조를 짜는 동안 구현을 맡은 인력들은 기다려야 하는 문제가 있을 수 있다. Blaauw에 따르면, 구조화, 구현, 실현의 단계를 나누어 병렬적으로 진행할 수 있는 것들은 동시에 진행하는 것이 좋다. 예를 들어 구현을 하는 사람들은 구조에 대한 희미한 가정을 가지고 data flow나 대략적인 흐름을 구상할 수 있다.