

### 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 16. No Silver Bullet – Essence and Accident in Software Engineering

Silver bullet은 늑대인간을 죽일 수 있는 무기이다.

소프트웨어 프로젝트의 경우, silver bullet이란, 프로젝트에 들어가는 비용과 시간을 현저하게 떨어뜨려주는 것을 말하며, 제목에서 알 수 있듯이 이는 존재하지 않는다.

빠르게 발전하고 있는 하드웨어와 소프트웨어 분야는 다르다. 소프트웨어 시스템은, 다음과 같은 특성들을 지닌다.

- Complexity: 소프트웨어는 비슷한 점들을 함수로 묶어 처리하기 때문에, 반복되는

부분들이 별로 없어 복잡하다.

- Conformity: 소프트웨어는 물리학처럼 하나의 이론으로 설명되지 않는다. 인터페이스가 달라지면 그 설계 또한 달라져야 한다.
- Changeability: 소프트웨어 제품은 끊임없는 변화에 대한 압력에 시달린다.
- Invisibility: 소프트웨어는 그 구조를 시각화하기 어렵다.

과거 소프트웨어의 생산성이 크게 향상되었던 부분은 심각한 하드웨어의 제약, 어색한 프로그래밍 언어등으로 인한 장벽들을 제거하는 데서 비롯되었다.

과거 생산성 향상에 도움이 되었던 것들은 다음과 같다.

- High-level languages
- Time-sharing
- Unified programming environments
- Object-oriented programming
- Artificial intelligence
- Expert systems
- Automatic programming
- Graphical programming
- Program verification
- Environments and tools
- Workstations

유망한 무기가 될 수 있는 것들은 다음과 같다.

- Buy versus build: 소프트웨어를 만들기보다 사는 것이 더 저렴하다.
- Requirements refinement and rapid prototyping: 고객들과의 꾸준한 소통을 통해 반복적으로 개선해야한다.
- Incremental development: 소프트웨어 개발에 있어서, 한 번에 다 만들지 말고, 점진적으로 개발해나가며 기능을 더해나가야 한다.
- Great designers: 좋은 개발자를 육성해 나가야 한다.