

1장 소프트웨어 공학의 정의와 역사

1. 공학이란 과학의 실용적인 응용 분야
2. 소프트웨어 공학 정의의 재구성
3. 다시, '소프트웨어 공학'
4. 전진, 앞으로
5. 소프트웨어 공학의 탄생
6. 패러다임의 전환

공학이란 과학의 실용적인 응용 분야

학습을 위한 접근과방식의 과학이므로 과학의 기술과 전략을 채택할 필요가있다.

"학습"



소프트웨어는 작게 나눠 개발한 다음에 실험하고 거기에서 얻은 결과를 바탕으로 다시 가설을 세우고 실험하고 거기서 얻은 결과를 다음 사이클에 반영하는 행위

소프트웨어 공학 정의의 재구성

소프트웨어 개발을 주도하기 위한 도구로서 실용적인도구

- 테스트 가능성
- 배포 가능성
- 속도
- 변수 통제
- 지속적인 배포

다시, '소프트웨어 공학'

공학은 제대로 임무를 완수할 기회를 높이기 위해 적용하는 프로세스와 관례다.

→ 이게 공학이라는 의미라니 단순히 동작하게하는것 뿐만아니라 프로세스와 관례를 설정하는것, 다시 한번 개발에대한 의미를 느꼈다.

전진, 앞으로

어떻게 전진하고 소프트웨어를 만들어 낼 수 있을까?

지배적인 권위자의 결정과 같은 스타일에서 벗어나



독단적인 신조에 이의를 제기하고 반박하며, 출처가 무엇이든 유행이나 상투적인 나쁜 생각과 훌륭한 생각을 구분하기 위해 우리에게 지적 자유를 허락하는 접근법이다.

소프트웨어 공학의 탄생

1960년대 저장된 프로그램이라는 아이디어를 기반으로 만들어졌다.

“소프트웨어 위기”

하드웨어와 소프트웨어와의 반전 속도차이

프레드 브룩스가 말했다.

기술이든 관리 기법이든 그 자체만으로 생산성과 안정성과 단순성 측면에서 10년
내에 획기적인 발전을 약속하는 단일 개발 방법론은 존재하지 않는다

소프트웨어 개발에서 개선 속도에 대한 브룩스의 관찰은 진실로 남아있다.

→ 소프트웨어가 뒤쳐지는 것처럼 보이지만, 그건 비현실적으로 빠른 하드웨어 발전때문이었구나

패러다임의 전환

패러다임 전환이란

아이디어는 우리가 이런 변화를 만들 때 그과정의 일부로서 이제는 더 이상 올바르지 않다고 알게 된 몇 가지 아이디어를 버리게 될 것 이라는 사상을 암묵적으로 포함한다.

소프트웨어 개발을 과학적인 바업론과 과학적인 합리주의에 근본을 두는 진정 한 공학 분야로 취급