



모든 문제에 통용되는 만능 해결책 따위는 존재하지 않는다



제 1 장. 소프트웨어 본질

March. 2018

Young-gon, Kim

ykkim@kpu.ac.kr

Department of Computer Engineering

Korea Polytechnic University



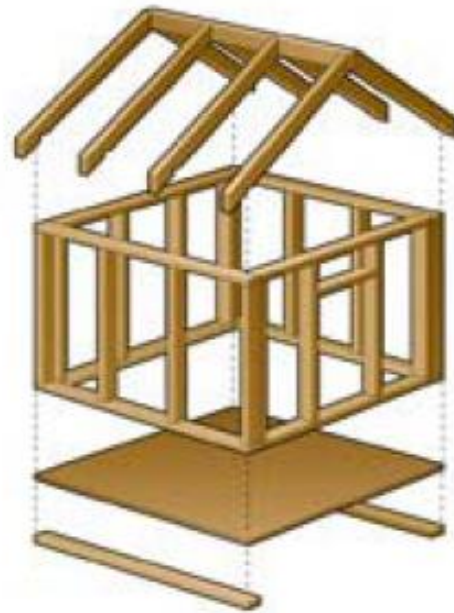
Topics covered

- ◆ 소프트웨어공학의 정의 및 특성
- ◆ 소프트웨어 응용 도메인
- ◆ Legacy Software
- ◆ Product Line Software
- ◆ 응용 소프트웨어에 대한 인식

1. 소프트웨어공학

◆ 소프트웨어 공학

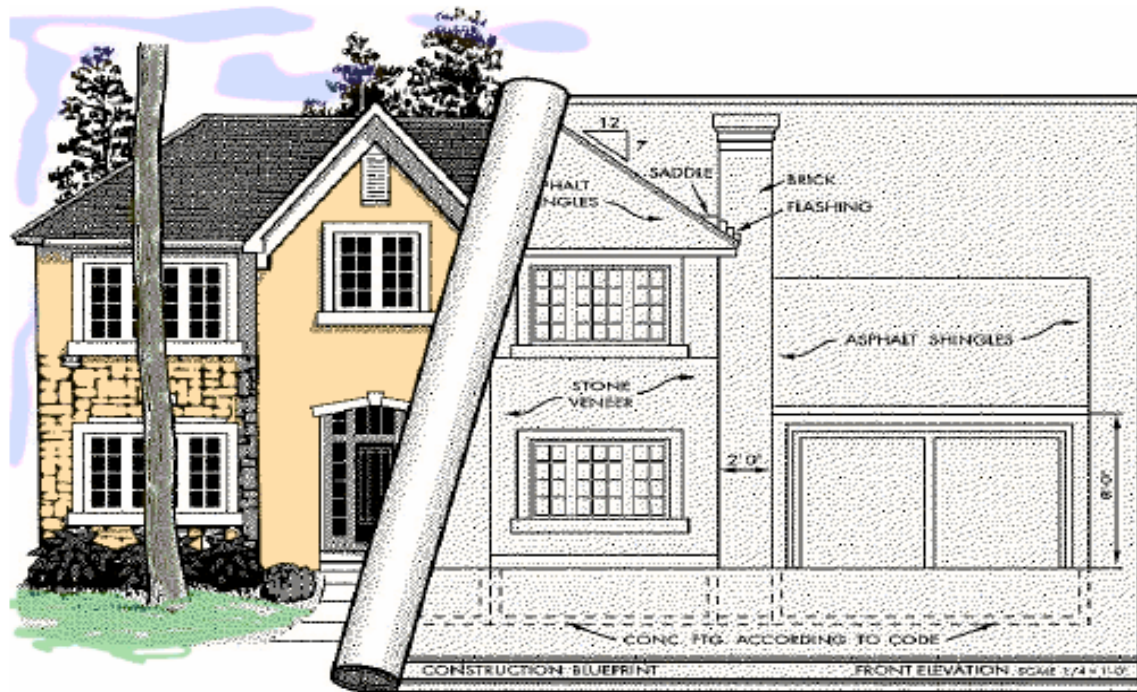
Do you want to build a dog hut?



1. 소프트웨어공학

◆ 소프트웨어 공학

Do you want to build a small house?



1. 소프트웨어공학

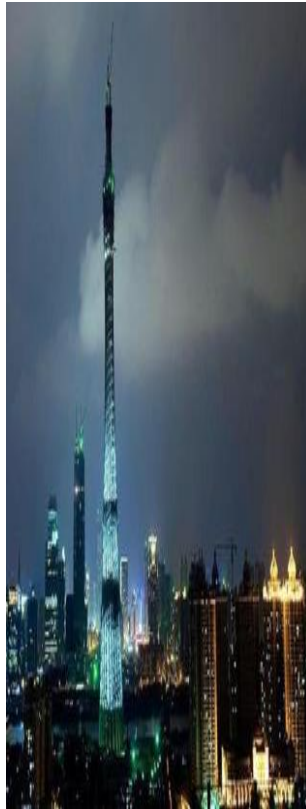
◆ Do you want to build a large-scale building?



1위 828m
버즈 칼리파



2위 634m
도쿄
스카이트리



3위 610m
광저우 TV관광
타워



4위 601m
아브라주 알
베이트 타워



5위 555m
롯데 월드 타워

1. 소프트웨어공학

◆ Can everyone build such a building?



6위 553m
CN타워



7위 509m
타이페이 101



8위 첸탑 506m
페더레이션 타워



8위 492m
상하이 월드
파이낸셜 센터



10위 484m
환구무역광장



2. 소프트웨어 정의

◆ Software 정의

- Instruction(컴퓨터 프로그램)

- 실행되면서 원하는 기능이나 함수 , 성능을 제공하는 명령어들

- Data structures

- 프로그램이 데이터를 적절하게 처리할 수 있게 해주는 자료구조

- Documentation

- 프로그램의 사용이나 운영을 나타내는 하드카피나 가상 상태의 문서

◆ Software 특성

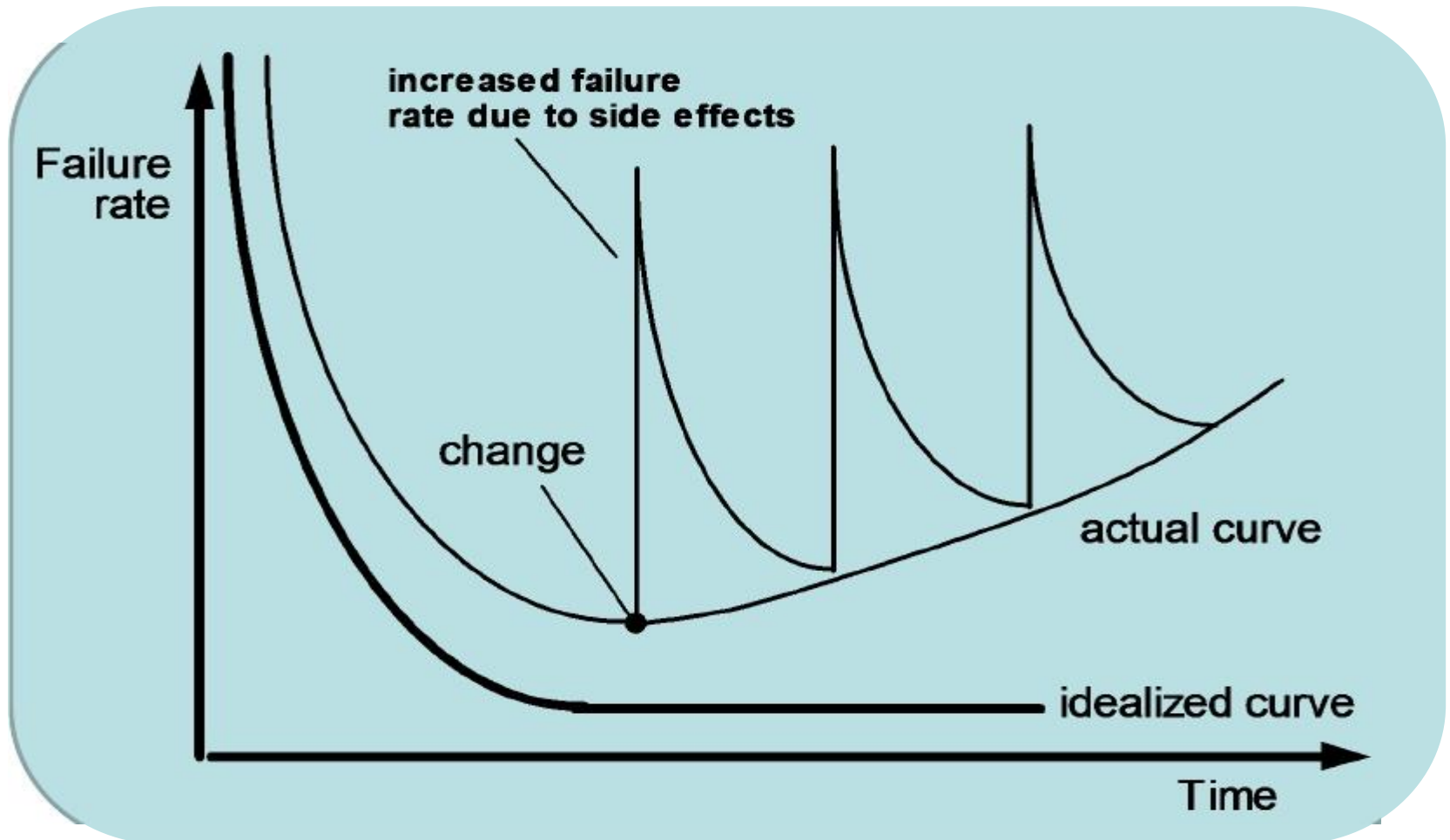
- 물리적인 요소가 아니라 논리적인 요소

- 마모 되지 않음

- 주문 제작 개념.

2. 소프트웨어 정의

◆ 마모 vs. 악화





3. Software Application Domain

◆ System Software

- 다른 프로그램에 서비스 제공 목적(컴파일러, 에디터, 파일관리, OS)

◆ Application Software

- 특수 업무상 요구해결 (비즈니스 처리, 관리/기술 측면의사결정지원)

◆ Engineering/Scientific Software

- 수치리 알고리즘 (분자생물학, 자동제어, 자동차 Test, 우주선궤도역학)

◆ Embedded Software

- 제품/시스템에 내장, 시스템 특성/기능 구현(제한적, 제어능력)

◆ Product-line Software

- 많은 고객이 사용하는 특수한 기능 (재고관리시스템)

◆ Web/mobile Application Software

- 브라우저 기반의 앱/모바일 기기에서 동작하는 소프트웨어

◆ AI Software

- 비수치적 알고리즘(로봇, 전문가시스템, 패턴인식, 인공지능망).

4. Legacy Software & Software Change

◆ Legacy

- 과거에 개발되어 **현재에도 사용중인** 낡은 하드웨어나 소프트웨어
- **중요한 자산**으로서, 새로운 아키텍처 시스템에 업데이트 필요
 - Old Method, 기술, 컴퓨터 시스템, 응용 프로그램

◆ Software Change

- 새로운 **컴퓨팅 환경이나 기술**이 필요한 경우
소프트웨어가 이를 만족시키기 위해 **적응**되어야 함
- 사업상 **새로운 요구**를 만족시키기 위해 **기능이 향상**
- 새로운 **시스템 시스템이나 데이터베이스**와
상호 운용될 수 있도록 **소프트웨어가 확장**되어야 함
- **진화하는 컴퓨팅 환경**에서 실행 가능하도록 하기 위해
소프트웨어는 재설계되어야 함.



5. Product Line Software

◆ Product line software

- 특정시장 분야 또는 업무에 필요, 공통으로 관리되는 특징 공유 ,
규정된 방식에 따라 **핵심 자산들로부터 개발된**
일련의 소프트웨어 집약 시스템
- 소프트웨어 프로덕트 라인을 개발할 경우
 - 동일한 **기본 애플리케이션 및 데이터 구조**를 사용
 - 프로덕트 라인 간에 **재사용이 가능한 소프트웨어 컴포넌트** 이용
- 소프트웨어 프로덕트 라인 **자산**
 - 요구사항 , 구조 , 디자인 패턴 , 재사용 가능한 컴포넌트.

◆ Product line software





6. 응용 소프트웨어에 대한 인식

◆ 응용 시스템 개발에 대한 의견

- 소프트웨어가 **완성**되기 위해 왜 그렇게 **많은 시간**이 필요한가 ?
- 소프트웨어 **개발 비용**이 왜 그렇게 **많이** 드는가 ?
- 소프트웨어를 **고객에게 전달하기 전**에 왜 **모든 에러**를 찾지 못하는가 ?
- 개발된 프로그램을 **유지보수**하는데 왜 그렇게 많은 **시간과 노력**이 들어야 하나 ?
- 소프트웨어를 **개발하고 유지보수**할 때 왜 **진척 정도를 측정**하기가 어려운가 ?



SE에 대해 자주 물어보는 질문(FAQ)

- ◆ 소프트웨어란 무엇인가?
- ◆ 소프트웨어 공학이란 무엇인가?
- ◆ 소프트웨어 비용이란 무엇인가?
- ◆ 소프트웨어 공학과 컴퓨터 과학의 차이는 무엇인가?
- ◆ 소프트웨어 공학과 시스템 공학의 차이는 무엇인가?
- ◆ 소프트웨어 프로세스는 무엇인가?
- ◆ 소프트웨어 프로세스 모델은 무엇인가?
- ◆ CASE(Computer-Aided Software Engineering)란 무엇인가?
- ◆ 좋은 소프트웨어의 특징은 무엇인가?
- ◆ 소프트웨어 공학이 직면한 도전 과제는 무엇인가?



Homework

◆ Chapter1. 소프트웨어 본질

1.1 소프트웨어 특성

1.2 소프트웨어 응용 도메인

1.3 레거시 시스템

1.4 소프트웨어 변경