



# RECOMMENDED SKILLS & KNOWLEDGE FOR SOFTWARE ENGINEERS

-Team 3-

김계성

김무진

류규현

이정현



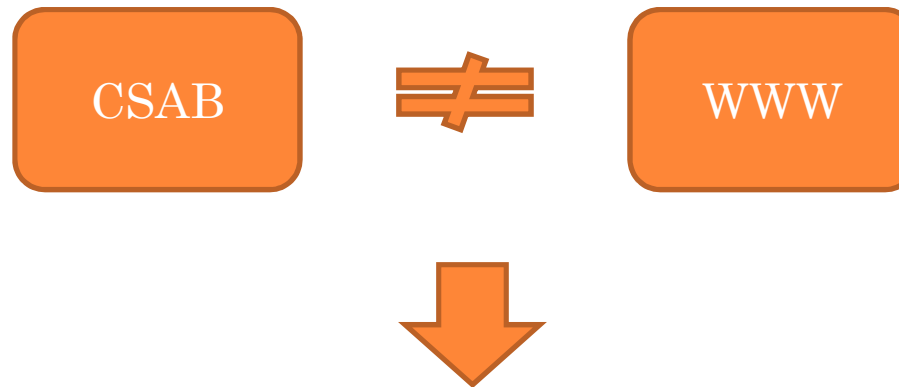
# 목차

1. Introduction
2. Computer science versus software engineering
3. Recommended software engineering skills and knowledge
  - 3.1 Computing Theory
  - 3.2 Software Practice
  - 3.3 Engineering Economy
  - 3.4 Customer and Business Environment
4. Practical implications
5. Summary



# 1. INTRODUCTION

Computer Science 에서 사용하는  
Skill & Knowledge



소프트웨어 공학의 진짜 가치를 평가하는데 어려움이 있다.



## 2. COMPUTER SCIENCE VERSUS SOFTWARE ENGINEERING

- Science(과학)의 정의

연구의 결과물로서 체계화된 지식의 한 부분.

과학적 방법과 실험을 통하여 얻게 된 일반적 법칙, 원리, 지식

- Engineerinf(공학)의 정의

실습, 실험, 연구에 의해 얻어진 자연 과학과 수학적 지식

그리고 인류의 이익을 위해 자연의 힘과 물질을

경제적으로 활용하는 방법을 적용하는 직업



## 2. COMPUTER SCIENCE VERSUS SOFTWARE ENGINEERING

- Science(과학) Branch

원리에 관하여 이론적이고 핵심적인 지식의 확장과 관련

- 공학(Engineer)

이론적 지식의 경제적 적용과 활용에 관련이 있다.



## 2. COMPUTER SCIENCE VERSUS SOFTWARE ENGINEERING

- Equation 1

Engineering = Scientific theory + Parctice +  
(Engineering) Economy

- Equation 2

Software Engineering = Computing theory + Practice +  
(Engineering) Economy



### 3. RECOMMENDED SOFTWARE ENGINEERING SKILLS AND KNOWLEDGE

- Recommended skills and knowledge

능숙한 설계, 디자인, 저비용의 효과적인 컴퓨팅 시스템 유지

가능소프트엔지니어들을 위해 적절하게 직업적으로 활용할 수

있도록 함



### 3. RECOMMENDED SOFTWARE ENGINEERING SKILLS AND KNOWLEDGE

- Skill - 능력 또는 소질에 의해 개발되어 능숙하게 다루는 힘
- Knowledge - 연구, 조사, 관찰 또는 경험에 의해 얻게 된

생각이나 사실

➔ 이러한 기술과 지식을 소유한 사람과 그렇지 못한 사람들을 고려하여 더 능력있는 소프트웨어 조직을 고려해야 한다.





### 3. RECOMMENDED SOFTWARE ENGINEERING SKILLS AND KNOWLEDGE

개개인적으로 최소한의 Skills과 knowledge를 가져야 하지만

모든 소프트웨어 엔지니어들이

여기에 제시된 Skills과 knowledge에 능숙하는 것 보다는

각각의 관련이 있는 적합한 기술 또는 핵심 지식에 능숙하게 만드는

최소한 한 명의 소프트웨어 엔지니어가 소프트웨어 프로젝트에 요구됨



# 3. RECOMMENDED SOFTWARE ENGINEERING SKILLS AND KNOWLEDGE

누구에게 필요한가?

졸업생 또는 졸업하지 않은 이들에게

이러한 skills과 knowledge가 요구됩니다.

어디서?

일반대학 또는 단과대학

왜?

그들은 가장 적절하고 적합한 자격이 있기 때문입니다.



## 3.1 COMPUTING THEORY

아주 많은 수, 다양한 계획을 실행 그리고 그것을 가능하게 하는 것  
더 일찍 몇몇 알려진 이론을 위반했기 때문에 동작하지 않을 수 있는  
계획을 버리는 것

이론적으로 실행 가능한 계획 저비용적 고려 -> 경제성 평가하는 것

Computer science는 computing의 이론과 관련성을 제공한다.



## 3.2 SOFTWARE PRACTICE

### Recommended computing theory skills and knowledge

- Programming language concepts
- Data structure concepts
- Database system concepts
- Relational algebra
- Operating system concepts
- Software architectures
- Computer architectures
- Automata theory and Petri nets
- Computability theory and Turing machine theory
- Complexity Theory
- Linguistics and parsing theory
- Computer graphics
- Set theory
- Predicate logic
- Formal proofs
- Induction



## 3.2 SOFTWARE PRACTICE

Table 1. Recommended computing theory skills and knowledge

- Programming language concepts
- Data structure concepts
- Database system concepts
- Relational algebra
- Operating system concepts
- Software architectures
- Computer architectures
- Automata theory and Petri nets
- Computability theory and Turing machine theory
- Complexity Theory
- Linguistics and parsing theory
- Computer graphics
- Set theory
- Predicate logic
- Formal proofs
- Induction



## 3.2 SOFTWARE PRACTICE

Table 2. Recommended software product engineering skills and knowledge

- Requirements, analysis, and requirements engineering
- Software design
- Code optimization and semantics preserving transformations
- Human-computer interaction, and usability engineering
- Specific programming language
- Debugging techniques
- Software-software and software-hardware integration
- Product family engineering techniques and reuse techniques
- CASE/CASE tools



## 3.2 SOFTWARE PRACTICE

**Table 3. Recommended software quality assurance skills and knowledge**

- Task kick-offs, previews, and readiness reviews
- Peer reviews, inspections, and walk-throughs
- Software project audits
- Requirements tracing/Quality Function Deployment(QFD)
- Software testing techniques
- Proofs of correctness
- Process definition and process improvement techniques
- Statistical process control
- Technology innovation



## 3.3 ENGINEERING ECONOMY

경제란 : 자원을 절약과 효율적인 사용

공학의 궁극적 목적 : 최소의 비용으로 최대의 효과를 창출 (이익의 극대화)

- 최소의 비용으로 사람들이 원하는 바를 만족 시킬 수 있는가?

### 소프트웨어 경제학(Economics)

경제학은 프로그래밍 프로젝트 진행에 드는 비용을 선택 해야 할 때, 우선적으로 과학적인 방법을 통하여 결정하여야 한다. 즉 올바른 결정을 내리기 위하여 여러 방법과 모델을 제시한다.





## 3.3 ENGINEERING ECONOMY

공학 경제에서 요구되는 기술과 지식

Time value of money

Economic equivalence

Inflation

Depreciation

Income taxes

Decision making among alternatives

Decision making under risk and uncertainty

Evaluating replacement alternatives

Evaluating public activities

Breakeven

Optimization



## 3.4 CUSTOMER AND BUSINESS ENVIRONMENT

소프트웨어 기술자는 해당 소프트웨어 제품과 서비스가 고객, 그리고 고객이 하는 일에 어떠한 영향을 주는지를 알 지 못한다면 그 제품과 서비스는 고객에게 제공 할 수 없다.

이러한 이유로 소프트웨어 기술자는 다음 사항을 숙지하고 있어야 한다.

- 고객이 누구이고 어떠한 일을 하는가.
- 우리의 제품과 서비스 중 그들이 이용하는 것이 있는가.
- 언제 어디서 그리고 왜 우리의 제품과 서비스를 이용하는가.
- 우리의 제품과 서비스가 기존의 의도와는 다르게 사용되고 있지는 않은가.  
(만약 그렇다면 그 이유는?)
- 고객이 하는 일에 우리의 제품과 서비스가 어느 정도의 영향을 미치는가.
- 제품과 서비스가 고객에게 전달되는 과정에서 어떠한 외부 제약과 규제가 존재하는가.

