

SW 공학 소개

과목 소개

- 소프트웨어 개발을 체계적으로 수행하여 프로젝트를 성공적으로 이끌고, 품질 높은 제품을 개발할 수 있는 방법들을 학습
 - 소프트웨어 공학에 대한 전반적인 개념과 주요 이론들을 이해
 - 요구사항 분석, 설계, 구현, 테스트에 이르는 전 개발 과정에서 기존에 검증된 Best Practice들을 습득
 - 소프트웨어 개발에 있어서 Generative AI 도구들의 활용

BP(Best Practice)

AI-Augmented Software Engineering

Generative Artificial Intelligence

- Generative artificial intelligence (also generative AI or GenAI) is artificial intelligence capable of **generating** text, images, or other media, using generative models.
- Generative AI models **learn the patterns and structure** of their input training data and then generate **new data** that has similar characteristics.

챗GPT가 할 수 있는 컴퓨터 코딩 기능

인간의 언어를
이해해 직접 코딩

코딩의 설명

코딩의 디버깅
(Debugging)

코딩의 개선

다른 컴퓨터
코딩 언어로 번역

인공지능
프로그램 코딩

MVP (Minimum Viable Product)

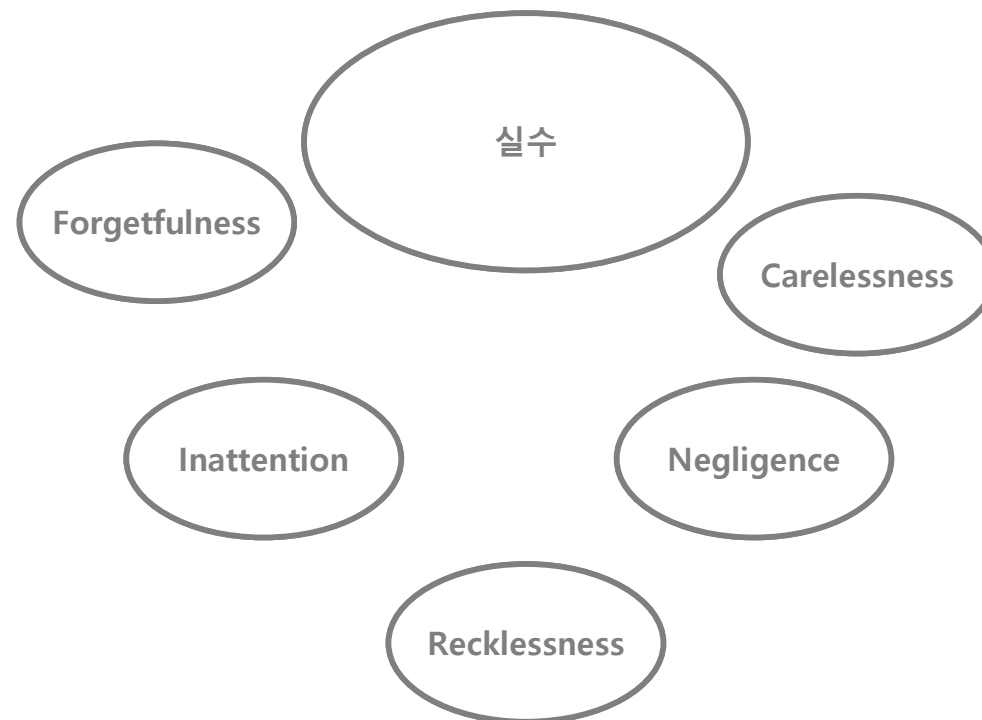
- 최소기능을 갖춘 제품 또는 서비스로 시장에서의 검증과 사용자 피드백 수집에 중점
- 실제 사용자와 시장에 진입하여 검증하는 단계에서 사용
- 실제 사용자가 제품을 경험하고 피드백을 제공
- 프로토타입은 아이디어의 구체화에 초점을 두는 반면 MVP는 실제 사용자와 시장에서 검증되는 단계

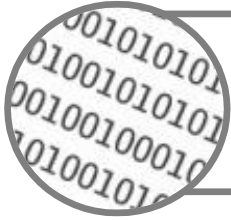
성적 평가

- 중간고사 : 20%
- 기말고사 : 30%
- 과제 4개 : 40%
- 출석: 10%



Human Intensive work (Errors are part of human nature)





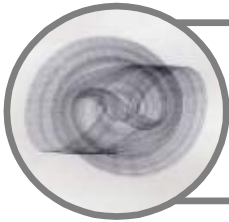
Binary

```
x = 0.0

for i in range(10):
    x = x + 0.1
    if x == 1.0:
        print (x, '= 1.0')
    else:
        print (x, 'is not 1.0')
```

가

else

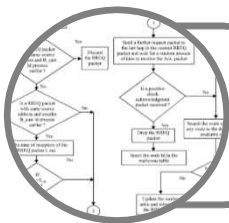


Invisibility

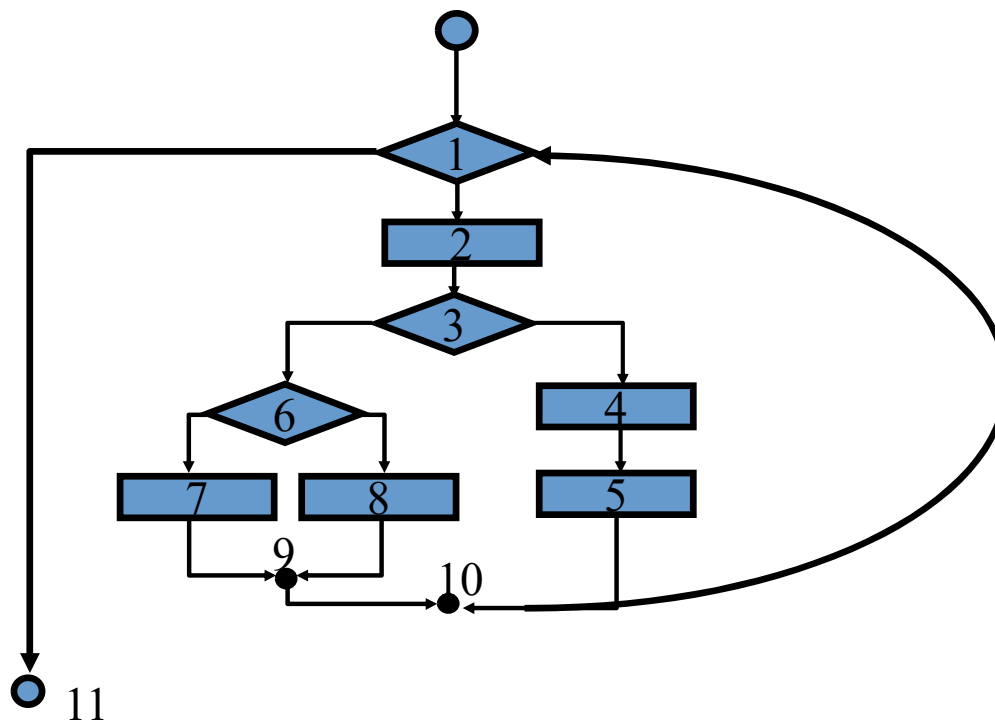
```
x = 3
for j in range(x):
    for i in range(x):
        print (i)
x = 2
```

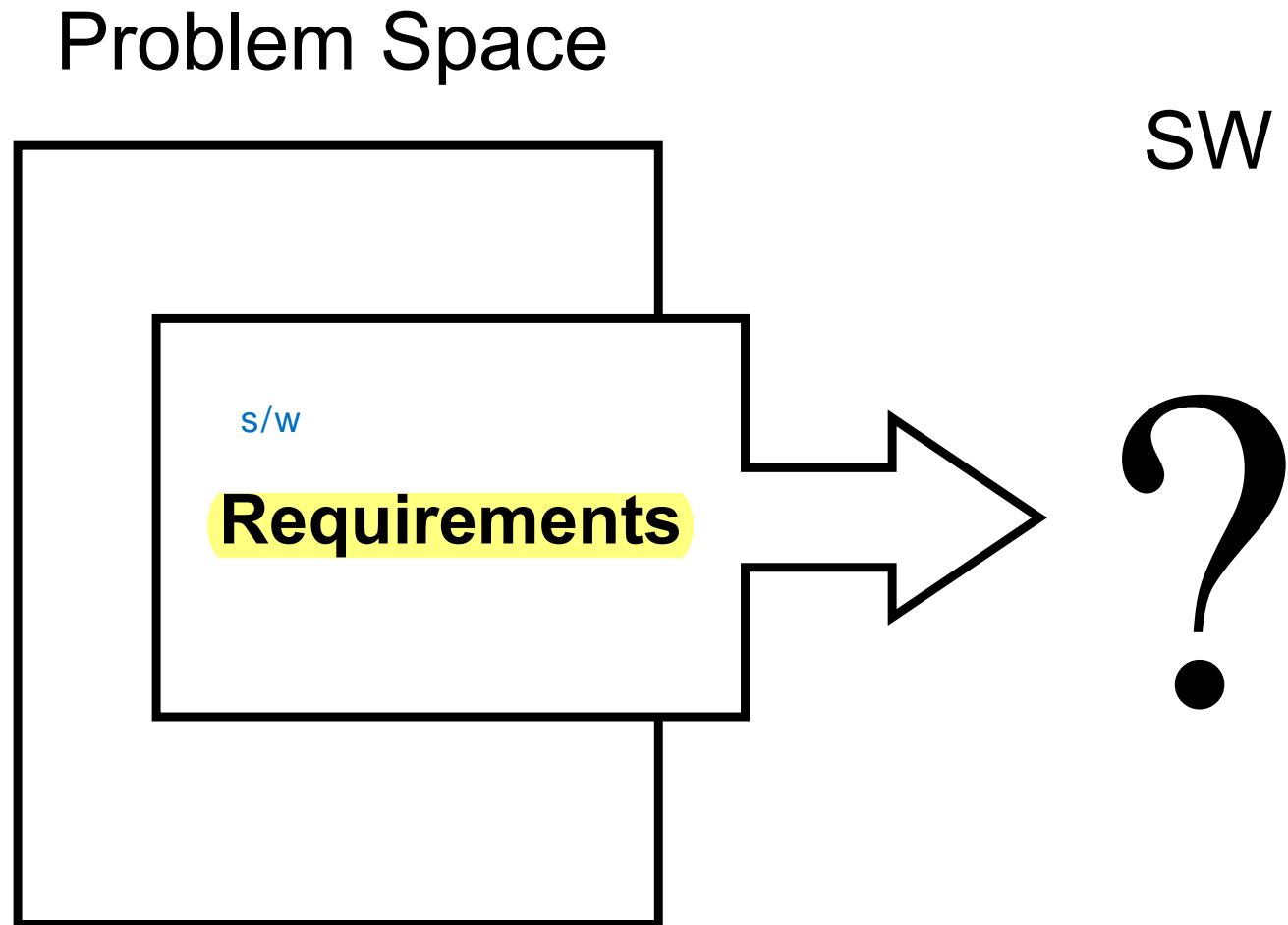
”

”””



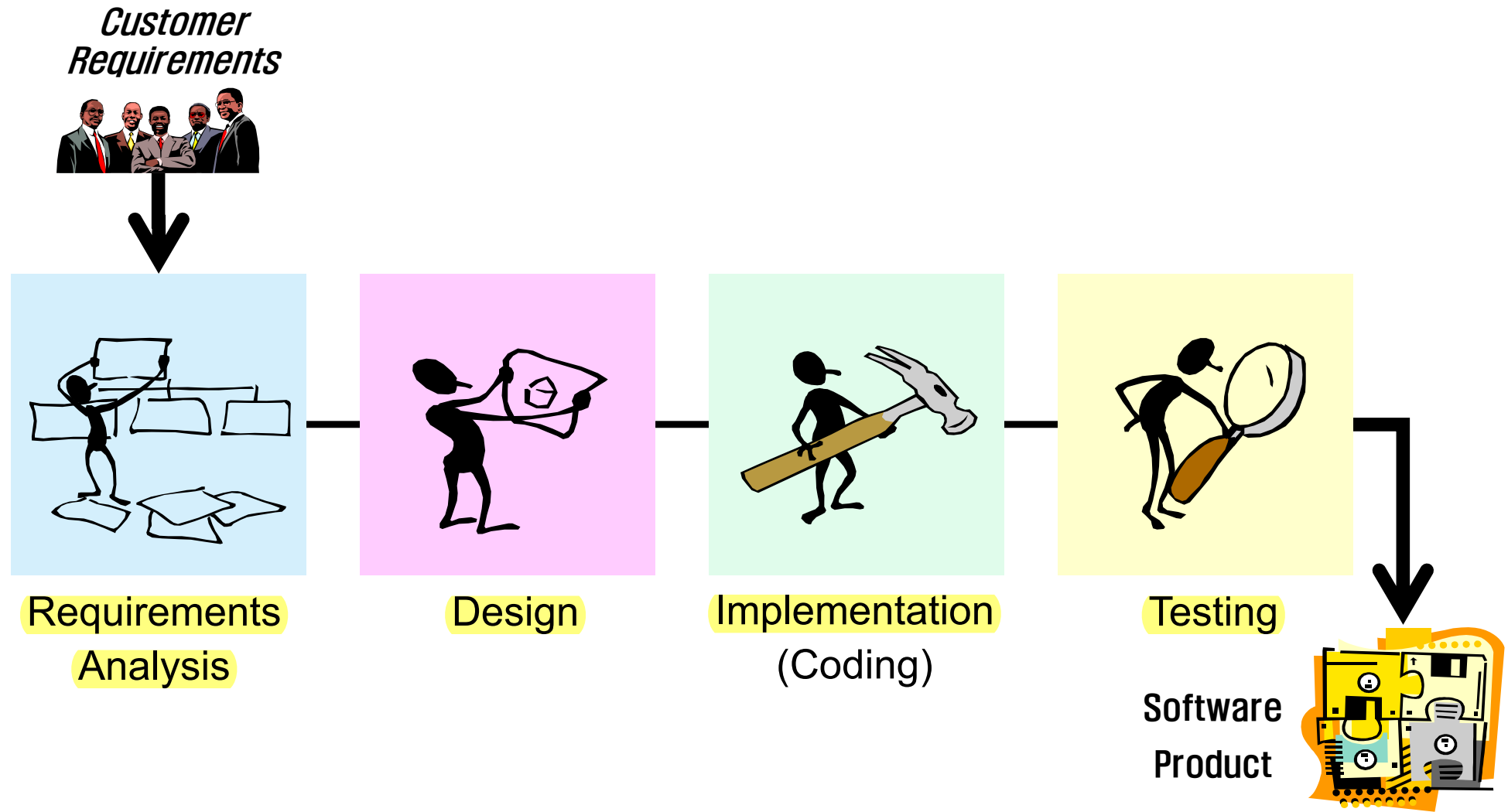
Non Linearity





SW 개발의 주요 단계 asile

maturity()



사람

구성원들의 능력



기술

필요 기술에 대한
숙련도



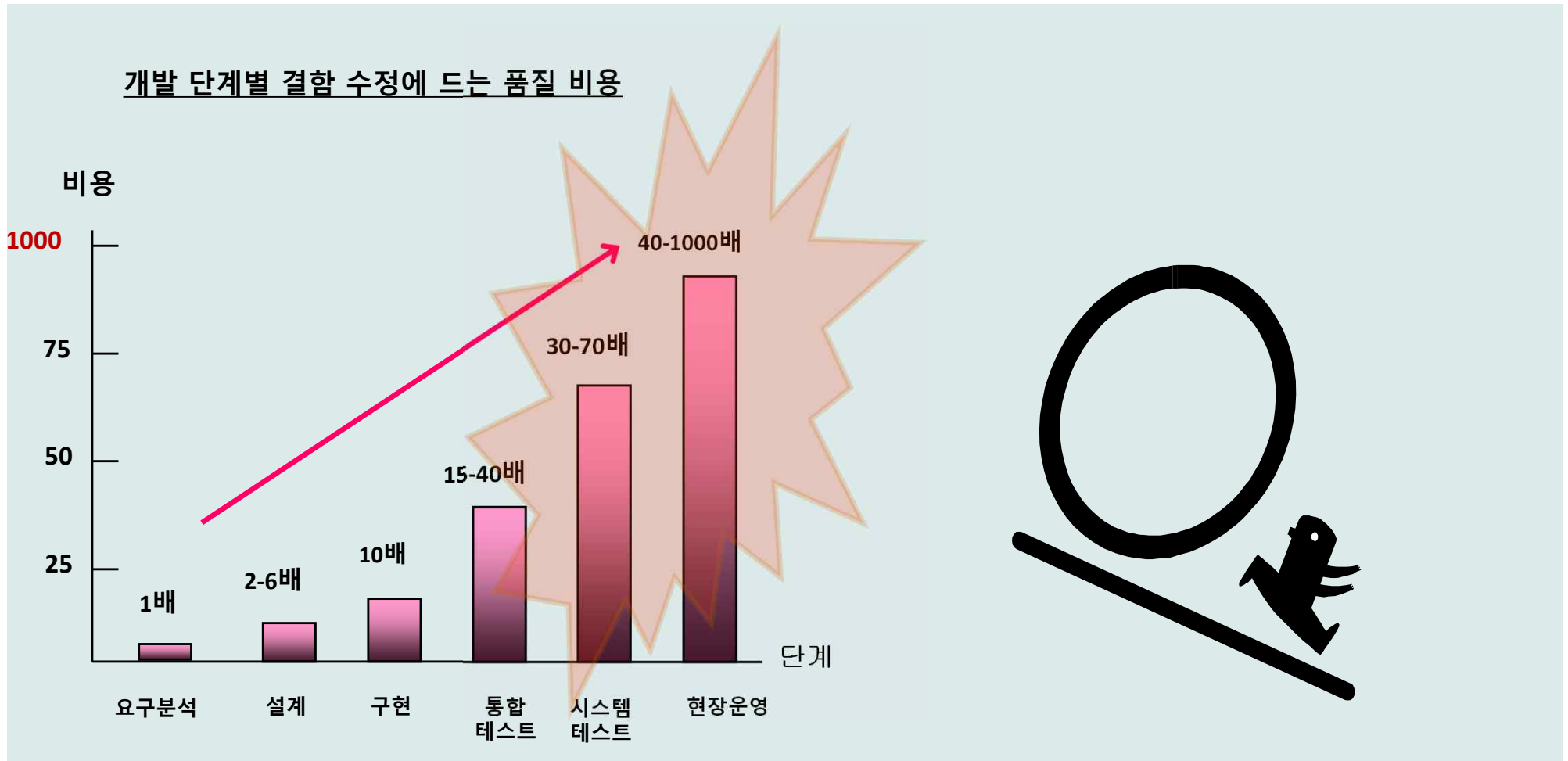
BALANCED
INTEGRATION

프로세스

일하는 방식



눈덩이 효과(Snowball Effect)



출처) Kaplan, C., R. Clark, and V. Tang, *Secrets of Software Quality: 40 Innovations from IBM*, Mc-Graw-Hill, 1995

SW 공학의 배경

- Software 위기(SW crisis)
 - Software 수요 증가에 비해 공급 및 개발의 어려움
- Software 위기의 해결
 - 다른 분야에서 사용했던 공학(Engineering) 패러다임을 이용하자는 결론
 - 1968년 NATO conference에서 Software 공학(Software Engineering) 제안됨

software crisis -> software engineering

?

+ gen.ai가 ^{sw}가 (,)

공학 (Engineering)

- 의미

- 실제적 문제(Practical Problem)를 해결하거나
- 실제적인 산출물을 생산해내기 위해
- 자원과 비용을 효과적으로 활용하면서
- 과학적 지식을 적용하는 것

- 공학과 Software 공학

- 공학

- ✓ 업무분야에서 문제 발생 시, 실무자가 적절한 해답을 찾을 수 있도록 체계적으로 정리된 기술적 지식을 제공

- Software 공학

- ✓ Software 개발 기술, 절차 및 도구의 우수한 사례(Best Practice)들을 정리하여 Software 개발 시, 누구나 당면한 문제를 해결할 수 있도록 체계적인 기술적 지식을 제공

SW 공학이란?

● 정의

- Software의 개발, 운용, 유지보수 및 폐기에 대한 체계적인 접근 방법

- IEEE

<->

- ✓ Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of SW

● 특징

- Software 개발 전 과정에 걸쳐 필요한 이론, 개념 및 기술을 다룸

- Software 개발 과정에서 생성되는 모든 산출물이 그 대상이 됨

● 목표

- Software 개발이 체계적이고 공학적인 방법으로 이루어져 추정된 비용과 기간에 고객이 원하는 품질 높은 Software를 개발하는 것

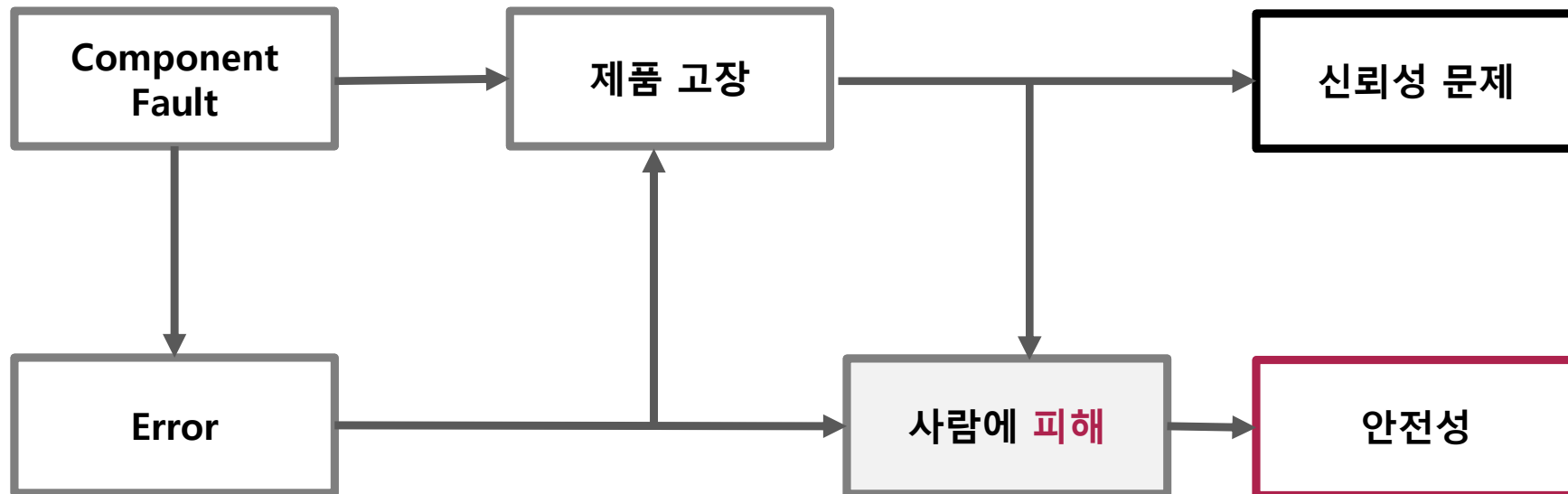
Best Practices()

안전성과 신뢰성

- **Safety**, focus
제품이 문제가 발생했을 때 사람에 대한 피해를 최대한 줄이는 (Risk Reduction) 것이 목표
- **Reliability**
‘제품 고장률(failure rate)을 낮춰서 오랫동안 쓸 수 있는 제품을 만드는 것이 목표

$$\frac{1}{1} > \frac{100}{1} \quad (\quad)$$

->



ISO 26262

- 차량을 위한 기능 안전성 표준 (2011년 11월)
(Functional Safety standard)
- 국제표준(International standard)
 - ISO/TC 22, SC03, WG16
 - ✓ TC (Technical Committee)
 - ✓ SC (Sub Committee)
 - ✓ WG(Working Group)