

소프트웨어공학



강의노트

소프트웨어와 공학 일반

❖ 학습안내

이번 시간의 학습내용과 학습목표를 확인해보세요.

■ 학습내용

- 소프트웨어 공학의 정의
- 중대한 시스템
- 가용성.신뢰성.안정성.보안성

□ 학습목표

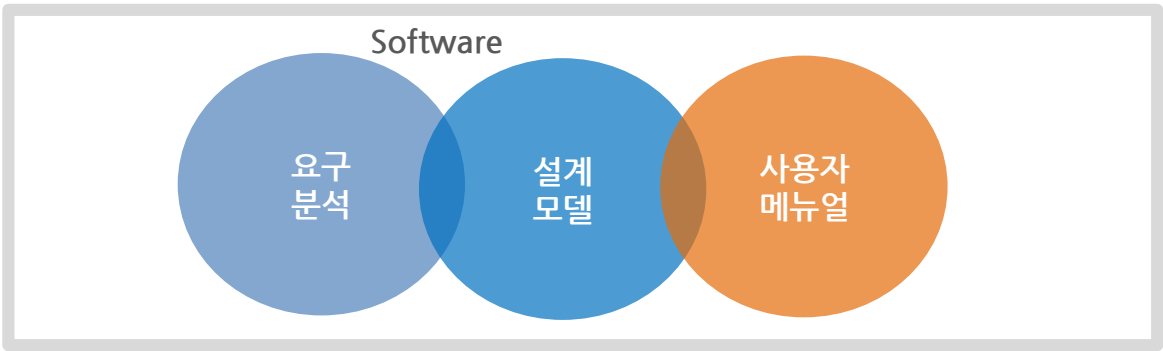
- 소프트웨어 공학에 대하여 정의할 수 있다.
- 중대한 시스템에서 보장되어야 하는 특징을 설명할 수 있다.
- 가용성,신뢰성,안정성,보안성의 필요성과 특징을 설명할 수 있다.

❖ 학습내용

[1] 소프트웨어 공학의 정의

1. 소프트웨어 공학의 정의

- ◆ 소프트웨어
 - 포괄적 개념



- 응용프로그램, 운영체제 뿐만 아니라 이에 파생되는 문서를 모두 포함하는 개념
- ◆ 소프트웨어 공학
 - 기업 정보시스템을 구성하고 있는 소프트웨어에 대하여 이를 잘 구축하고 운영함
 - 기업의 변화요구에 맞춰 수정하고 폐기하는 방법을 배우는 것

1. 1. 소프트웨어 공학의 다양한 정의

- ◆ Fritz Bauer
 - 컴퓨터 하드웨어에서 신뢰성 있게 운영되는 소프트웨어를 경제성 있게 개발하기 위해 공학적 원리를 응용하고 확립시킨 이론
- ◆ Fritz Bauer, ANSI/IEEE
 - 기계에서 안정적이고 효율적으로 작동하는 소프트웨어를 얻기 위한 올바른 공학 원칙을 수립하고 사용하는 것 (Fritz Bauer)
 - 소프트웨어 개발, 운영, 유지보수 및 폐기 과정에 적용되는 체계적인 접근 방식 및 일련의 기술(ANSI/IEEE)

❖ 학습내용

[1] 소프트웨어의 개요

1. 1. 소프트웨어 공학의 다양한 정의 (계속)

- ◆ Berry Boehm
 - 컴퓨터 프로그램을 설계, 개발, 운영, 유지보수에 관련된 문서를 작성하는 데 필요한 과학적인 지식의 실용화
- ◆ Richard R. Fairley
 - 전산학, 경제학, 경영과학 및 의사소통기술과 문제해결을 위한 공학적인 접근방안을 토대로 소프트웨어 개발에 임하는 신기술 체계
- ◆ 교수님의 쉬운 견해
 - 소프트웨어를 잘 만들고, 돌리고, 고치고, 버리는 방법을 배움

1. 2. 소프트웨어 공학에서 자주 물어보는 질문들

- ◆ 소프트웨어란 무엇인가?
 - 컴퓨터 프로그램과 그와 관계된 요구분석, 설계모델, 사용자 메뉴얼과 같은 문서를 포함
 - 소프트웨어 제품은 특정 고객을 위해 개발될 수도 있고, 일반적인 시장을 위해 개발될 수도 있음
 - 소프트웨어 제품은

일반적	- 다양한 계층의 소비자를 위해 개발 (예) 엑셀 / 워드와 같은 PC 소프트웨어
주문형	- 특정 고객의 사양에 맞게 개발 - 새로운 소프트웨어는 새로 개발하거나, 기존의 소프트웨어를 재사용하거나, 일반적 소프트웨어를 재구성하여 만들어짐

❖ 학습내용

[1] 소프트웨어의 개요

1. 2. 소프트웨어 공학에서 자주 물어보는 질문들(계속)

- ◆ 소프트웨어 공학이란 무엇인가?
 - 소프트웨어 생산과 연관된 모든 사항을 고려한 공학적인 규칙
 - 소프트웨어 엔지니어는 체계적이고 조직적인 접근 방법을 적용하여야 함
 - 직면한 문제, 개발에 따르는 제약과 자원을 활용하기 위해서 적절한 도구와 기술을 사용하여야 함
- ◆ 소프트웨어 공학과 컴퓨터과학의 차이는 무엇인가?

컴퓨터과학 이론	이론과 기초를 다룸
소프트웨어 공학	유용한 소프트웨어를 개발하기 위한 실용적인 면을 다룸

 - 컴퓨터과학 이론은 소프트웨어 공학의 완전한 지지대 역할을 하기에 충분하지 못함
- ◆ 소프트웨어 공학과 시스템공학의 차이는 무엇인가?

시스템공학	개발 하드웨어, 소프트웨어, 공정공학을 포함하여 컴퓨터 기반 시스템의 모든 측면
소프트웨어 공학	시스템 내에서 데이터베이스, 응용, 소프트웨어 인프라, 제어의 개발에 관련된 프로세스의 일부분

 - 시스템 엔지니어는 시스템의 응용, 요구 및 구조설계, 통합, 배치 등에 관여함
- ◆ 소프트웨어 프로세스란 무엇인가?
 - 소프트웨어의 개발 혹은 개선을 목적으로 하는 일련의 활동
 - 모든 소프트웨어 프로세스에서 사용되는 일반적인 활동

요구명세	시스템이 해야 할 것과 개발에 따르는 제약을 명시
개발	소프트웨어 시스템의 생산
검증	소프트웨어가 고객이 원하는 것인지를 검사
개선	변화요구를 수용하기 위해서 소프트웨어를 변경

❖ 학습내용

[1] 소프트웨어의 개요

1. 2. 소프트웨어 공학에서 자주 물어보는 질문들(계속)

- ◆ 좋은 소프트웨어란 어떠한 속성을 가지는가?
 - 소프트웨어는 사용자가 **필요한 기능과 성능을 제공**하여야 하며 **유지보수**가 가능하고, **믿을 수** 있으며 **수용**이 가능하여야 함

유지보수 가능성	소프트웨어 변경요구 를 수용 할 수 있어야 함
신뢰성	소프트웨어는 믿을 수 있어야 함
효율성	소프트웨어는 시스템 자원 을 낭비 하지 말아야 함
수용성	소프트웨어는 설계되어진 그대로 사용자가 받아들일 수 있어야 함

- 소프트웨어는 **이해**할 수 있고, **사용가능**하며, 다른 시스템과 **호환**이 되어야 함

- ◆ 소프트웨어 공학이 직면하고 있는 어려움은 무엇인가?

- 이질성, 납품시기, 신뢰성

이질성	이질성의 플랫폼, 실행환경 을 극복할 수 있는 기술의 개발이 필요
납품시기	빠른 시간 내에 소프트웨어를 납품 할 수 있는 기술의 개발
신뢰성	사용자가 신뢰 할 수 있는 소프트웨어를 개발할 수 있는 기술의 개발

❖ 학습내용





[1] 소프트웨어 공학의 정의

2. 소프트웨어 위기

◆ 소프트웨어 위기의 도래

- 하드웨어 기술의 급속한 발전, 범용컴퓨터의 광범위한 보급, 소프트웨어 엔지니어의 위기
 - 하드웨어 기술은 소프트웨어개발 능력의 발전속도보다 훨씬 빠름
 - 새로운 소프트웨어를 요구하는 시장의 수요를 감당할 수 없음
 - 기존 정보기술로 개발된 소프트웨어의 유지보수가 어려워짐
- 인건비 상승, 우수한 소프트웨어의 부족, 생산성에 대한 위기의식
- 소프트웨어 개발 프로젝트 진행의 어려움
 - 개발 예산 초과
 - 품질 저하
 - 개발 일정 지연
 - 생산성 저하

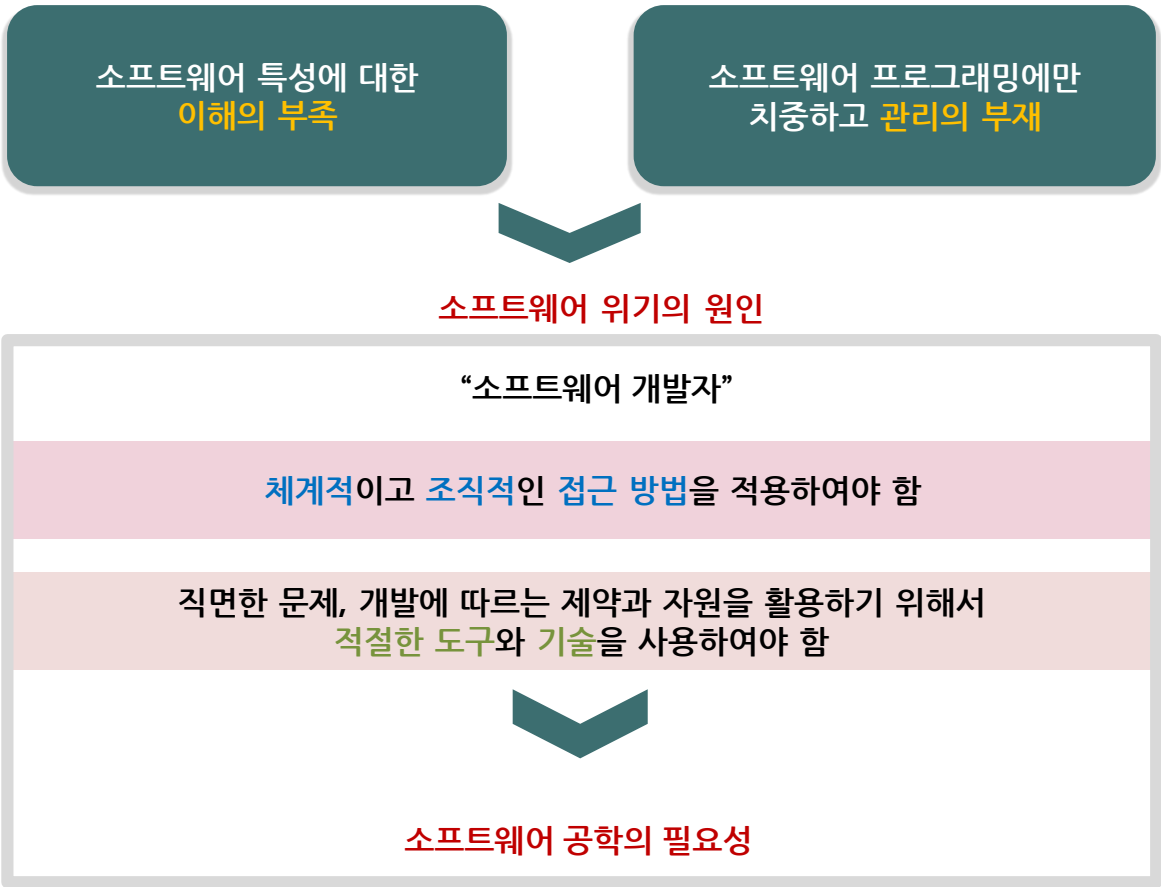
2. 1. 소프트웨어 위기의 원인

-  소프트웨어 특성에 대한 이해의 부족
-  소프트웨어 프로그래밍에만 치중하고 관리의 부재
-  소프트웨어 엔지니어는 체계적이고 조직적인 접근 방법을 적용하여야 함
-  직면한 문제, 개발에 따르는 제약과 자원을 활용하기 위해서 적절한 도구와 기술을 사용하여 함

❖ 학습내용

[1] 소프트웨어의 개요

2. 1. 소프트웨어 위기의 원인(계속)

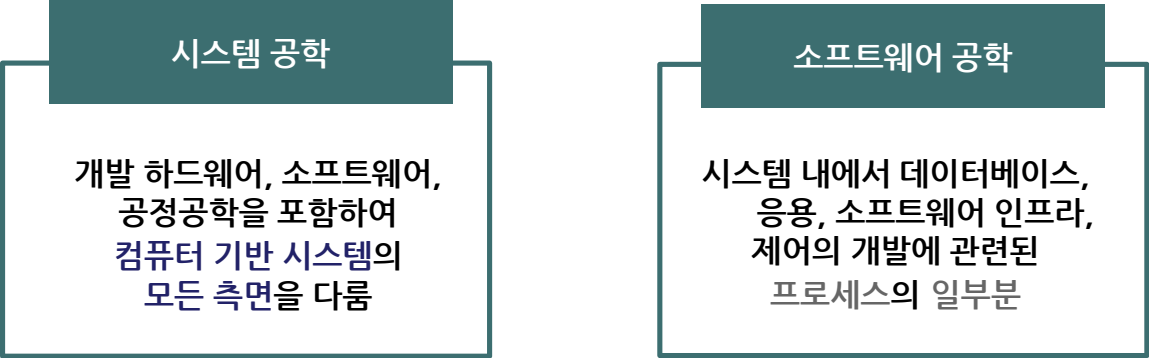


❖ 학습내용

[1] 소프트웨어의 개요

3. 시스템 공학

- ◆ 시스템 공학의 정의
 - 시스템을 **분석, 설계, 구현, 검증, 설치** 그리고 **유지보수** 하는 것
 - 시스템이 사용되는 **방법, 운영**, 건설에 적용되는 **제한**과 시스템에 의해서 제공되는 **서비스**와 관련된 것
- ◆ 시스템 공학과 소프트웨어 공학



요즘은 “소프트웨어 공학”을 보다 넓은 시각으로 해석하여 소프트웨어 공학과 시스템 공학을 거의 유사한 개념으로 바라보며, 기업의 정보시스템을 다루는 관점에서는 더욱 동일한 부분으로 봄

3. 1. 시스템 공학의 부분요소

서브 시스템 개발	시스템 통합	시스템 설치	시스템 개선 (유지보수)	시스템 해체
서브 시스템 개발	시스템 통합	시스템 설치	시스템 개선 (유지보수)	시스템 해체

- 일반적으로 **하드웨어, 소프트웨어, 통신** 개발이 동시에 이루어짐
- 각각의 서브 시스템(단위 시스템)을 구현하고 시스템을 통합하여 전체 시스템이 만들어짐
- 어떤 시스템은 상업용 구매를 통해 구입할 수도 있음
- 구현 팀 사이에 의사소통이 중요한 이슈임

❖ 학습내용

[1] 소프트웨어의 개요

3. 1. 시스템 공학의 부분요소(계속)

서브
시스템 개발

시스템 통합

시스템 설치

시스템 개선
(유지보수)

시스템 해체

- 하드웨어, 소프트웨어, 사람을 **한 곳에 통합**하여 시스템을 구성하는 과정
- 점진적으로 해결해서 서브 시스템이 한번에 하나씩 통합
- 서브 시스템 간의 인터페이스 문제는 이 단계에서 발견됨
- 시스템 컴포넌트 간에 조정되지 않은 납품 때문에 문제가 될 수도 있음

서브
시스템 개발

시스템 통합

시스템 설치

시스템 개선
(유지보수)

시스템 해체

- 시스템 완성 후, 시스템은 고객의 환경에 맞도록 설치되어야 함
 - ☒ 다른 환경
 - ☒ 동시에 두 개의 시스템이 공존
 - ☒ 운영자 훈련 시간 필요
 - ☒ 새 시스템에 대한 저항
 - ☒ 물리적 설치 문제

(예) 케이블 문제, 공간문제

서브
시스템 개발

시스템 통합

시스템 설치

시스템 개선
(유지보수)

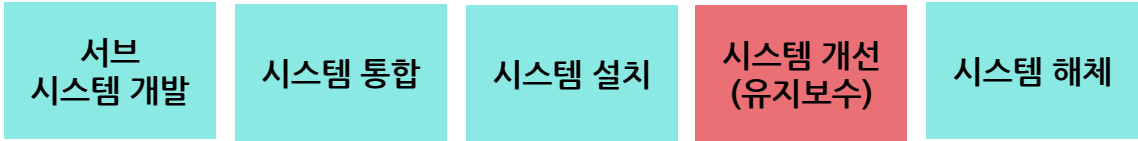
시스템 해체

- 큰 시스템은 생명주기가 길기 때문에 변경요구를 수용할 수 있어야 함

❖ 학습내용

[1] 소프트웨어의 개요

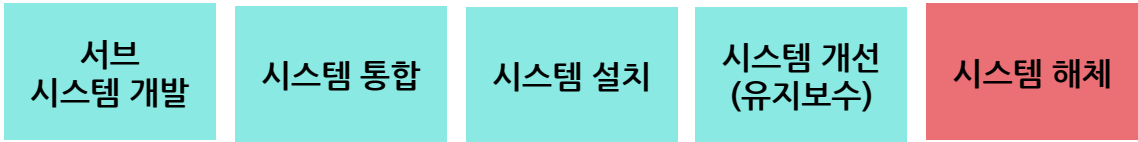
3. 1. 시스템 공학의 부분요소(계속)



- 큰 시스템은 생명주기가 길기 때문에 변경요구를 수용할 수 있어야 함

- 개선을 하기 위하여 본질적으로 비용이 듦
 - ☑ 기술적 관점, 사업의 관점에서 변경이 용이한지 분석함
 - ☑ 각 서브 시스템이 변경에 대하여 정확하게 유지, 운영되는지 분석함
 - ☑ 기존 시스템이 가진 시스템 성격을 변경하기 어려움(완전 재개발)
 - ☑ 변경이 이루어지면 시스템 구조가 훼손됨

- 레거시 시스템(Legacy Systems)을 고려하여야 함
반드시 유지되어야 하는 기존의 시스템이 존재함



- 유용하게 사용 후 시스템 사용 중지
- 환경을 오염시키는 유해물질을 제거
- 다른 시스템에서 사용할 수 있도록 데이터를 재구성하거나 변환될 수도 있음

❖ 학습내용

[2] 중대한 시스템

1. 중대한 시스템의 정의



- 기업의 정보시스템 중 특히 **기업활동의 영향**이 큰 시스템
- 중대한 시스템은 모든 시스템 관련 활동에 있어서 **1순위**로 고려하여야 함

- 01 안전이 중대한 시스템(Safety-Critical Systems)
- 02 임무가 중대한 시스템(Mission-Critical Systems)
- 03 업무가 중대한 시스템(Business-Critical Systems)
- 04 보안이 중대한 시스템(Security -Critical Systems)

2. 중대한 시스템의 특성

◆ 중대한 시스템의 특징



❖ 학습내용

[2] 중대한 시스템

2. 중대한 시스템의 특성(계속)

- ◆ 중대한 시스템의 특징(계속)
- ◆ Dependability(신뢰성, 확실성)
 - 중대한 시스템의 **가장 중요한 특징**
 - 신뢰성과 확실성이 떨어지는 시스템으로 중대한 시스템을 운영한다면 고장(Fail)의 발생확률이 높음
- ◆ 시스템의 고장으로 인해 **수리하는 비용**이 높음
 - 대부분의 중대한 시스템은 **사람이 감시**하고 **제어**하는 시스템
 - 중대한 시스템은 예상치 못한 상황을 복구할 수 있는 **관리자**가 필요
- ◆ 고장 나는 경우
 - 하드웨어 실패** 설계 상의 실수, 부속품의 오류, 하드웨어의 수명
 - 소프트웨어 실패** 분석, 설계 혹은 구현시의 오류 가능성
 - 관리자의 실수** 운영 제도적인 보완책 강구

3. 중대한 시스템의 유형

- ◆ 안전이 중대한 시스템(Safety-Critical Systems)
 - 시스템의 장애(문제)시 **안전**에 대한 위협이 높은 시스템
 - 장애 시 **생명**이나 **환경**에 큰 영향을 줌

❖ 학습내용

[2] 중대한 시스템

3. 안전이 중대한 시스템 사례 - 화학공장 방재 시스템



화학공장 방재 시스템의
소프트웨어에 이상이 생겨 방재공장이
오동작

화학물질이 대기 중에 배출

소프트웨어 개발자 또는 운영자의 단순실수로 **인명피해** 발생

인슐린 공급
프로그램

혈액 내 혈당을 측정하는
센서 값에 따라
인슐린을 체내에 공급



프로그램 오류로
제시간에 공급하지 못함

환자 위독

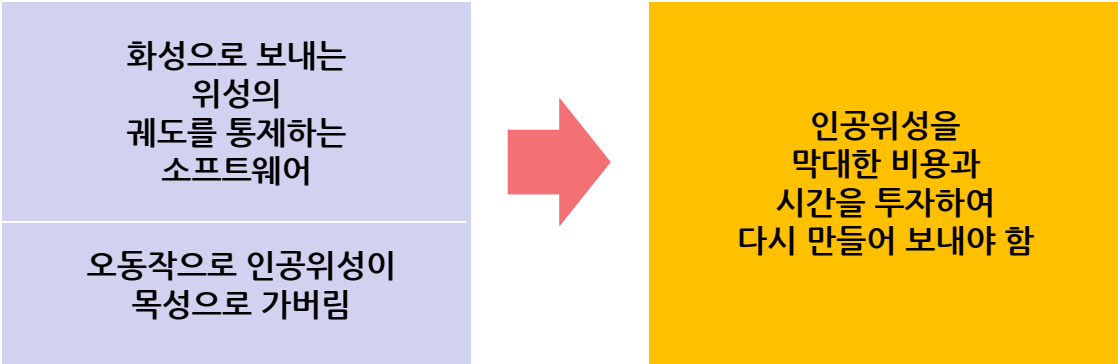
3. 중대한 시스템의 유형(계속)

- ◆ 임무가 중대한 시스템(Mission-Critical Systems)
 - 시스템의 장애(문제)시 **임무에 대한 위협**이 높은 시스템
 - 장애 시 **중요한 목적**을 달성하지 못하는 시스템

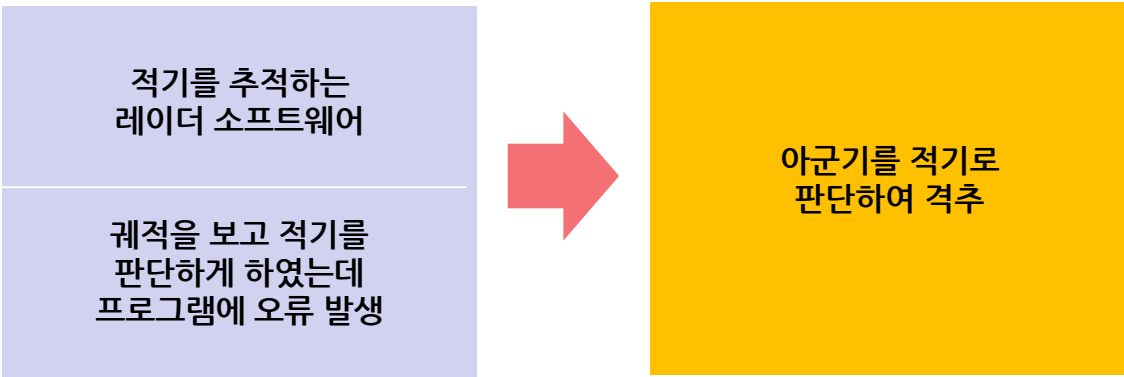
❖ 학습내용

[2] 중대한 시스템

3. 임무가 중대한 시스템 사례 - 우주 통제 시스템



3. 임무가 중대한 시스템 사례 - 적기 추적 레이더 시스템



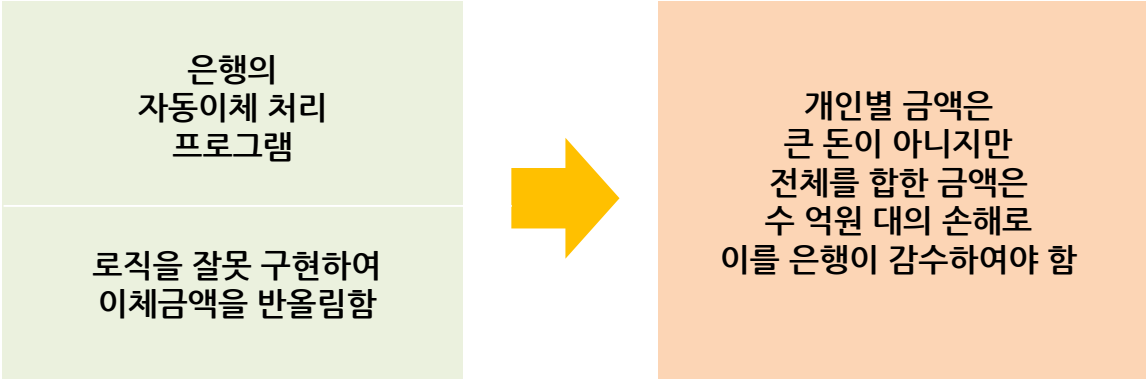
3. 중대한 시스템의 유형(계속)

- ◆ 업무가 중대한 시스템(Business-Critical Systems)
 - 시스템의 장애(문제)시 (기업 등의) **업무에 대한 위협**이 높은 시스템
 - 장애 시 많은 **금전적 손실**을 입히는 시스템

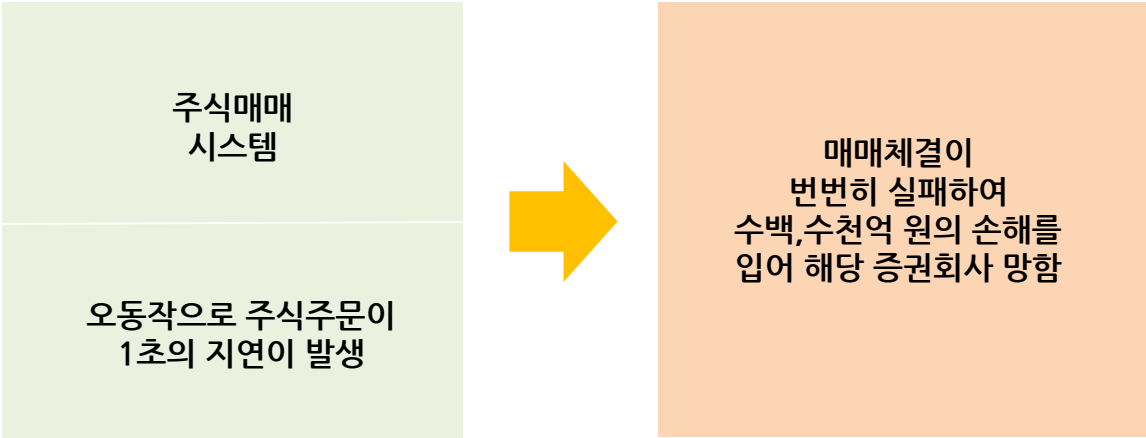
❖ 학습내용

[2] 중대한 시스템

3. 업무가 중대한 시스템 사례 - 은행의 예금 처리 시스템



3. 업무가 중대한 시스템 사례 - 증권 매매 시스템



3. 중대한 시스템의 유형(계속)

- ◆ 보안이 중대한 시스템(Security-Critical Systems)
 - 시스템에 대한 **보안(정보보호)**이 중대한 시스템
 - 보안침해사고는 **안전, 임무, 업무**에 모두 중대한 위협요소임
 - 보안요소는 모든 사항에 걸쳐있는 **절단관심사(Cross-Concern)**

❖ 학습내용

[2] 중대한 시스템

3. 중대한 시스템의 유형(계속)






◆ 보안이 중대한 시스템(Security-Critical Systems)

- | | | |
|---|--------------------|--------------------|
| 1 | 화학공장 방재시스템 Case | 인슐린 공급 프로그램 Case |
| 2 | 우주통제 시스템 Case | 적기 추적 레이더 시스템 Case |
| 3 | 은행의 예금 처리 시스템 Case | 증권 매매 시스템 Case |

앞에서 예시를 든 모든 시스템의 보안 침해사고 발생으로 소프트웨어가 오작동한다면 **중대한 문제를 야기함** 보안(정보보호)요소는 약방의 감초와 같은 요소

[3] 가용성 신뢰성 안정성 보안성

1. 시스템 확실성

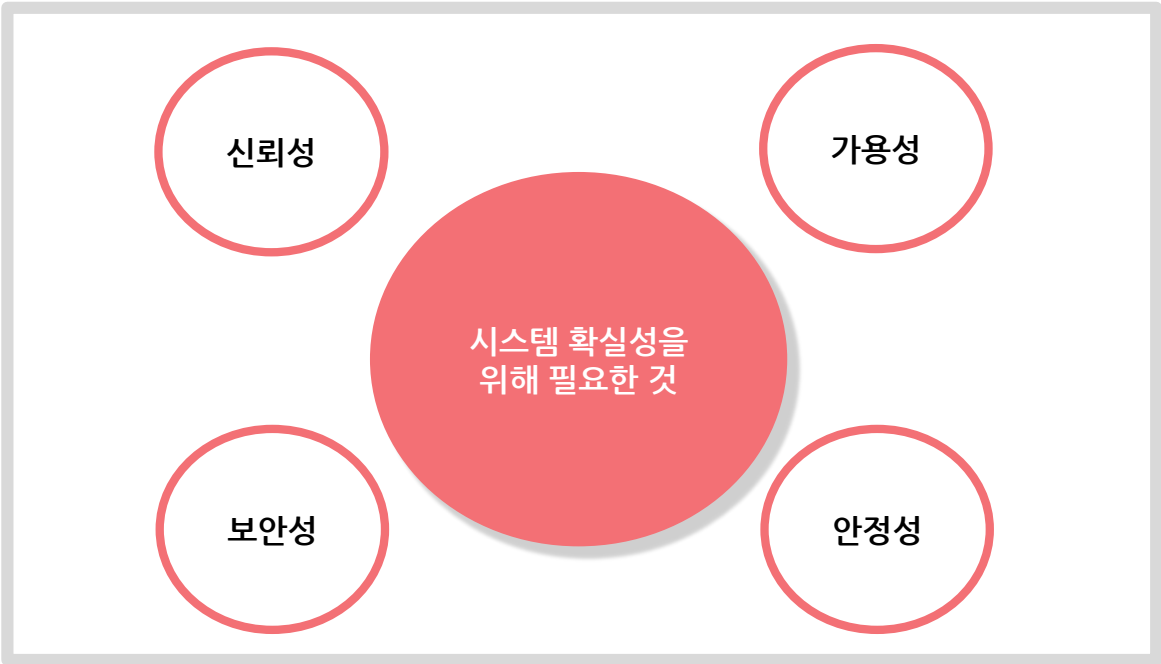
-  중대한 시스템에서 가장 중요한 시스템 특성
-  사용자가 시스템에 대하여 신뢰할 수 있는 정도
-  사용자가 생각한 대로 작동
-  정상적인 경우에 고장이 나지 않음
-  사용성과 신뢰도는 다름

※ 사용할 수 있지만 믿을 수 없는 시스템도 있음

❖ 학습내용

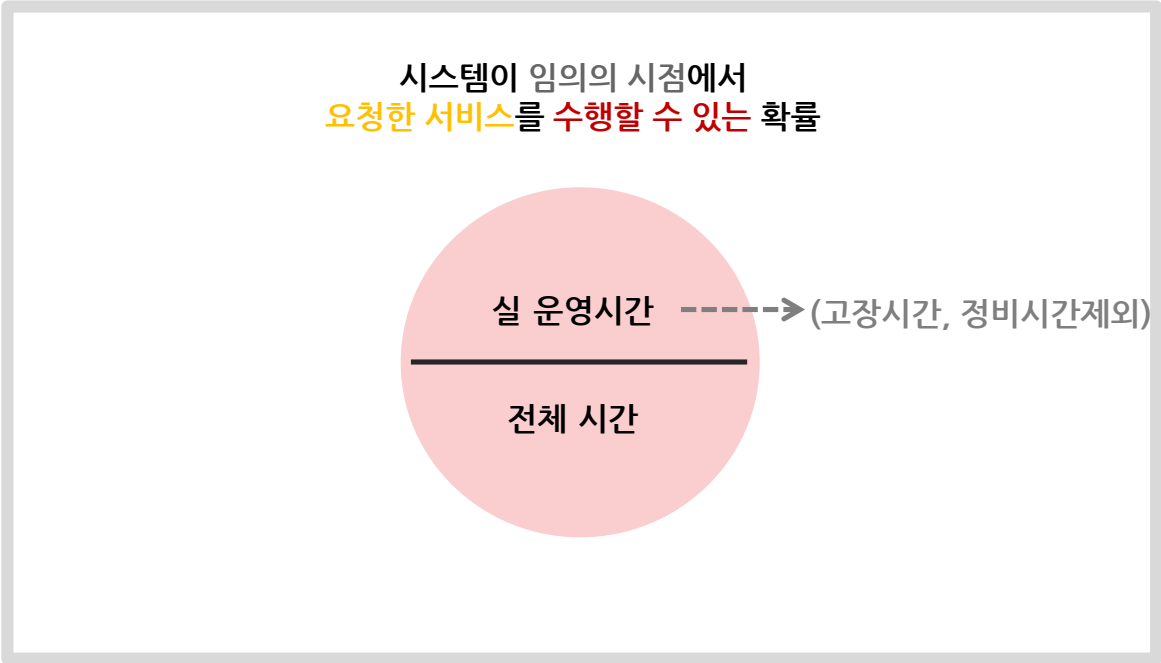
[3] 가용성 신뢰성 안정성 보안성

1. 1. 거래처리시스템(TPS: Transaction Processing System)(계속)



2. 가용성, 신뢰성

◆ 가용성(Availability)

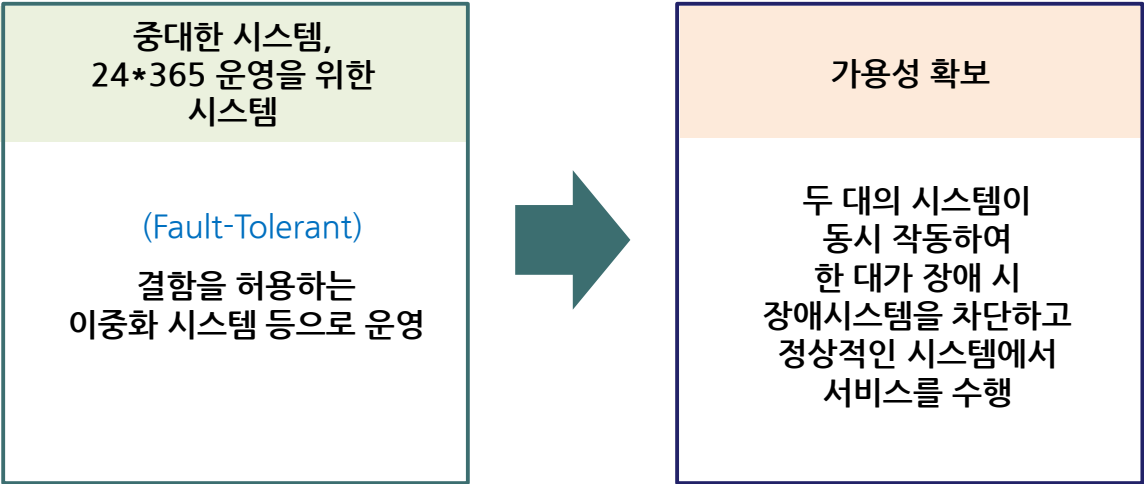


❖ 학습내용

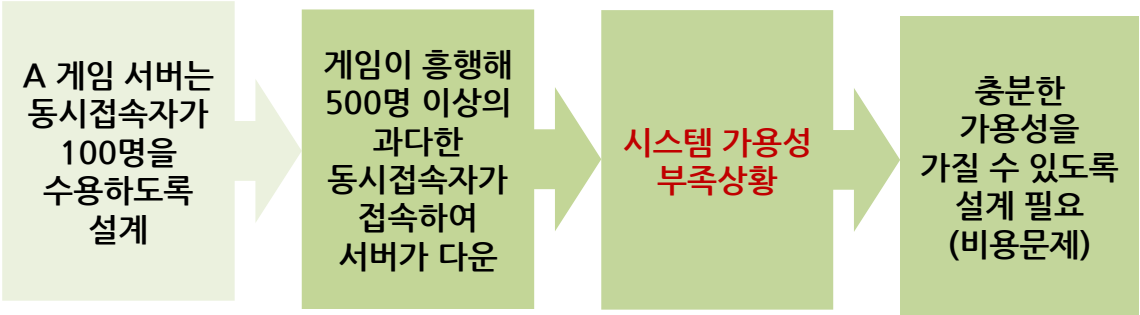
[3] 가용성 신뢰성 안정성 보안성

2. 가용성, 신뢰성(계속)

◆ 가용성(Availability)(계속)



2. 가용성 사례 - 게임 서버



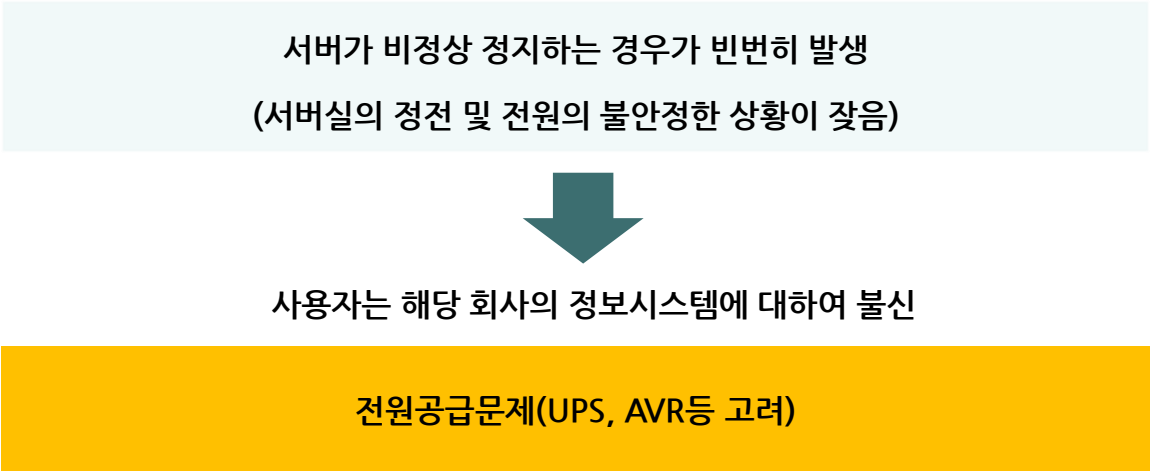
❖ 학습내용

[3] 가용성 신뢰성 안정성 보안성

2. 가용성, 신뢰성(계속)

- ◆ 신뢰성
 - 주어진 목적을 위해 지정된 기간 동안 시스템이 **고장이 안 날** 확률
 - 빈번한 고장으로 시스템에 대한 신뢰를 할 수 없음
 - 소프트웨어가 작동한 결과가 신뢰를 할 수 없는 경우

2. 신뢰성 사례 - 시스템 고장이 빈번한 경우



2. 신뢰성 사례 - 소프트웨어가 잘 못 구현된 경우



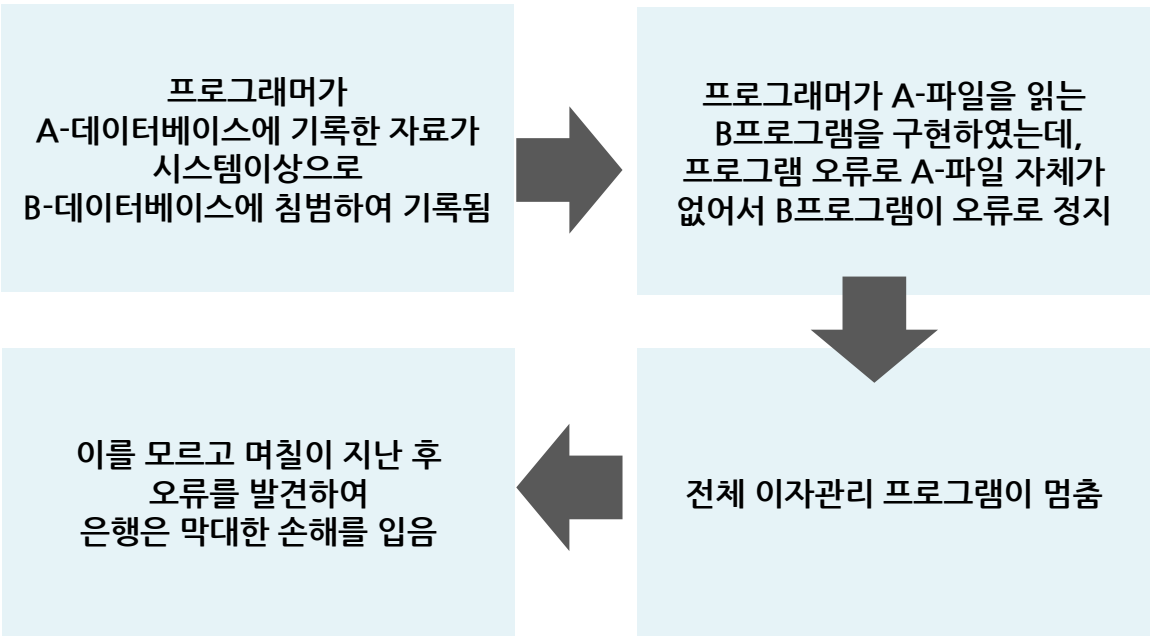
❖ 학습내용

[3] 가용성 신뢰성 안정성 보안성

3. 안정성, 보안성

- ◆ 안정성
 - 시스템이 정상적이든 비정상적이든 시스템에 **큰 손상이 없어야 함**
 - 시스템의 **사소한 문제요소**가 전체 시스템의 문제를 발생시키는 경우**안정성에 큰 문제**

3. 안정성 사례 - 은행 시스템의 오류



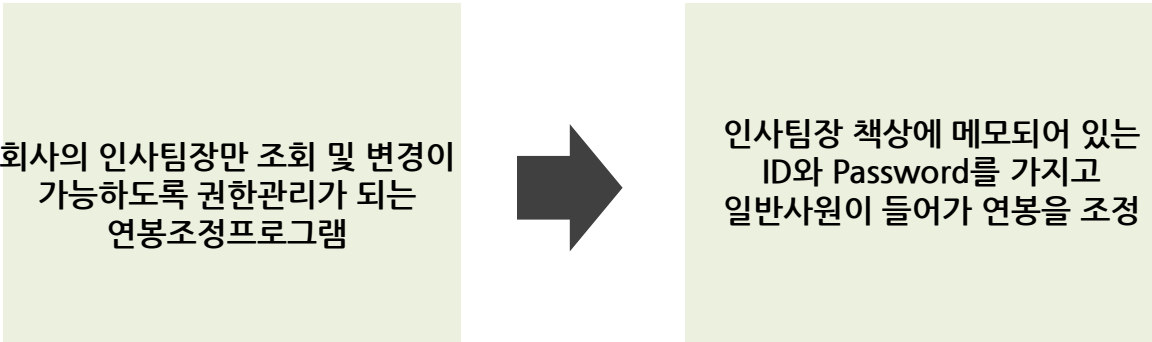
3. 안정성, 보안성(계속)

- ◆ 보안성
 - 침해사고나 외부의 공격으로부터 시스템을 **보호**하는 능력
 - 보안은 가용성, 신뢰성, 안전성보다 **기본 요건**
 - 보안성은 언급하지 않아도 당연히 지켜야 되는 **기초적 요건**

❖ 학습내용

[3] 가용성 신뢰성 안정성 보안성

3. 보안성 사례 - 관리적 보안에 치명적 오류가 있는 경우



관리적 보안의 치명적 오류

3. 보안성 사례 - 보안성이 결여되어 있는 경우



보안성 결여

누가 해당 화면을 조회하였는지 조회로그를 남기며, 캡처 등이 불가능하도록 구현하여야 함

❖ 핵심정리

1. 소프트웨어 공학의 정의

- 기업 정보시스템을 구성하고 있는 소프트웨어에 대하여 이를 잘 구축하고 운영하며, 기업의 변화요구에 맞춰 수정하고 폐기하는 방법을 배우는 것
- 소프트웨어 공학에 대한 정의는 다양한 견해가 있음

2. 중대한 시스템

- 기업의 정보시스템 중 특히 기업활동의 영향이 큰 시스템
- 유형으로는 안전이 중대한 시스템, 임무가 중대한 시스템, 업무가 중대한 시스템, 보안이 중대한 시스템이 있음

3. 가용성, 신뢰성, 안정성, 보안성

- 시스템 확실성은 중대한 시스템에서 가장 중요한 시스템 특성임
- 시스템 확실성을 위하여 가용성, 신뢰성, 안전성, 보안성이 필요함