

# 3 주차

## Ch3 연습문제 자료조사

### 학습 목표

소프트웨어 개발 프로젝트가 실현 가능한 지를 진단하는 고려 사항으로 무엇이 있는지 살펴본다.

소프트웨어 개발 프로세스에는 어떠한 모델이 있는지 이해하고 비교한다.

소프트웨어 개발 능력을 향상하기 위한 CMMI와 SPICE에 대하여 이해한다.

소프트웨어 프로세스 관련 표준을 살펴본다.

## 1. 실현 가능성 분석

사용자의 요구 사항에 따라 SW 시스템을 개발하고자 할 때 비용, 일정, 기술적 수준등을 고려해 충분히 실현 가능한 가를 살펴보는 것. → **개발 비용, 이득, 대안 세 가지를 평가.**

- 개발 비용: 인건비를 제외한 하드웨어 비용, SW 비용, 유지보수 비용, 문서화 비용 등등.. etc
- 이득
- 대안: 비용과 이득을 산정할 때 다양한 측면을 고려 하는 것.

### 고려할 측면 (이건 안외워도 될듯, 개념만 알면 된다했으니)

- 경제적 측면: 비용 대비 효과를 분석해 프로젝트에 들인 비용에 비해 정당한 효과를 얻는 지
- 기술적 측면: 기술적 제약사항이 없는가
- 스케줄 측면: 프로젝트를 주어진 기간 내에 완료할 수 있는가
- 운영적 측면:
- 동기적 측면:
- 법적, 윤리적 측면:

## 2. 전통적인 소프트웨어 프로세스

### 2 - 1. 소프트웨어 프로세스란?

SW 개발 과정을 체계적으로 계획, 관리할 의도로 서로 관련 있는 활동을 단계로 분리하여 정의한 것

- 블랙박스 프로세스: SW 프로세스를 적용하지 않고 하나의 통합된 개발 과정 → 사용자 요구 사항을 만족하기 어렵다.
- 투명한 프로세스: 개발 과정을 여러 단계로 분리해 요구 사항을 만족할 수 있다.

## 소프트웨어 프로세스의 필요성

요구사항 만족도 향상 → 수정 및 재개발 비용 절감 → 품질 관리 가능 → 프로젝트 성공

## 2 - 2. 전통적인 프로세스 모델!!!!(암기)

### 폭포수 모델

폭포의 물이 아래로 떨어지는 것처럼 순차적인 단계로 SW를 개발하는 것. 각 단계마다 정해진 소프트웨어 문서를 작성한다.

실현 가능성 분석 → 요구사항 도출 및 분석 → 설계 → 코딩 및 단위 시험 → 통합 및 시스템 시험 → 배포, 설치 및 운영

#### 단점

1. 개발 과정에서 요구되는 변경을 수용하기 어렵다.
2. 이전 단계의 결과물에 오류가 있을 경우(입력 오류), 다음 단계에 영향을 준다.
3. 피드백이 없을 만큼 요구사항 분석을 제대로 하지 않은 이상 피드백을 수용하기 어렵다. - 폭포수 모델

이런 단점을 보완한 (피드백이 가능한 모델) 점진적 모델이 개발.

### 점진적 모델

전체 SW 개발 범위를 관리 가능한 일정 크기의 그룹으로 분할 후 순차적으로 개발한다.

#### 장점

1. 새로운 제품 개발 시 사용자 요구 사항이나 변경을 반영할 수 있다. (피드백이 가능하므로)
2. 변경을 쉽게 수용할 수 있다.
3. 점진적, 단계적 개발을 통해 사용자에게 SW 시스템을 제공할 수 있다.

#### 단점

1. 테스트와 통합의 반복적 수행으로 오버헤드가 발생할 수 있다.
2. 사용자 피드백을 너무 반영하면 이미 개발된 시스템의 구조가 무너질 수 있다.

### 프로토타입 모델(중)

간단한 기능만 포함하는 사용자 인터페이스의 원형을 보여주거나(이렇게 만들면 됨?), 핵심 모듈만을 우선적으로 개발해서 사용자에게 제공하고 이를 통해 사용자의 요구 사항을 만족했는지 여부를 판단한다. → 사용자 피드백을 바탕으로 실제 최종 시스템을 구현한다.

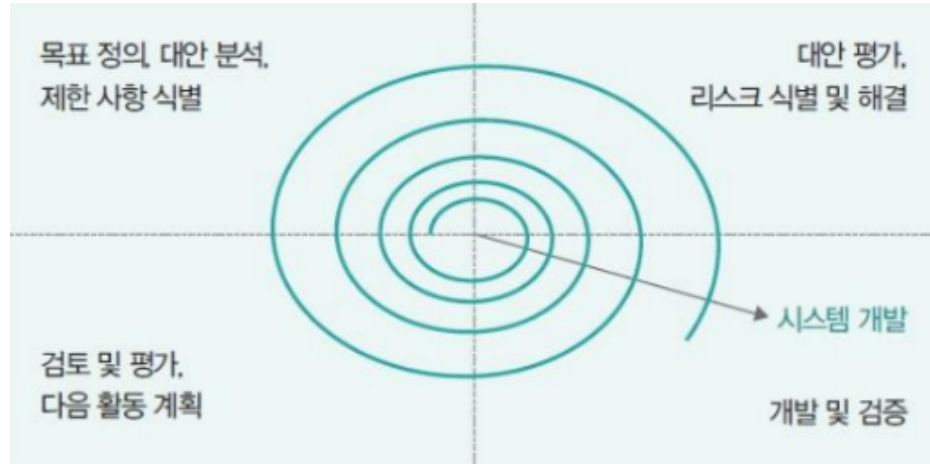
#### 장점

1. 사용자 요구 사항을 검증하여 개발 비용과 시간을 단축한다.
2. 사용자 ↔ 개발자 or 개발자 ↔ 개발자 간의 의사소통이 이루어져 상호 동일한 개념을 확보할 수 있다.

## 단점

1. 사용자 검증 과정 이후에 변경이나 바뀐 요구 사항에 대해 고려할 수 없다.

## 나선형 모델(중)



SW 시스템 개발 시 **위험을 최소화 하기 위해** 점진적으로 전체 시스템으로 개발해 나가는 모델

목표 설정 → 위험 분석 → 구현 및 테스트 → 평가 및 다음 단계 수립

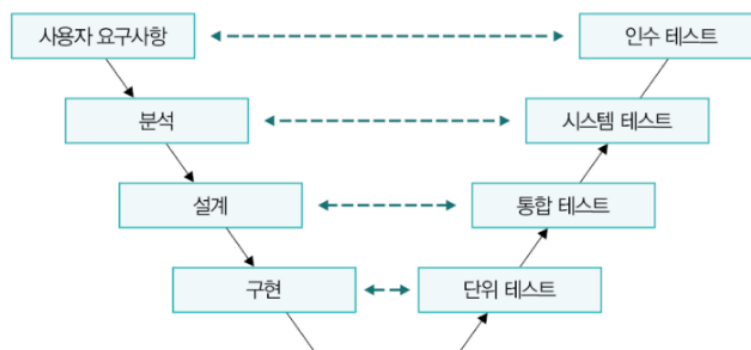
## 장점

1. 체계적인 위험 관리 → 위험성이 큰 프로젝트에 적합하다.
2. 사용자 요구 사항을 반영, 적용하기 쉽다.
3. 최종 개발 SW 시스템의 사용자 만족도와 품질이 높다 (사용자 요구 사항을 잘 반영하니까)

## 단점

1. 프로젝트 기간이 길다.
2. 프로젝트 관리가 어렵다 (사이클이 많아지면)
3. 위험 관리를 위한 전문적 지식이 필요하다.

## V 모델



폭포수 모델의 확장안 모델이다.

폭포수 모델에 따른 SW개발 과정을 진행 후 다 시 올라가면서 테스트 과정을 단계별로 수행한다.

#### 장점

1. 개발 결과물에 대한 단계적인 검증과 확인 과정을 거쳐 오류를 줄인다.
2. 요구사항의 정확한 이해를 필요로 하는 작은 시스템 개발에 유리하다.
3. 폭포수 모델을 적용함으로써 반복이 허용되지 않아 변경을 다루기 어렵다.

## 3. 애자일 개발 프로세스 (코드 중심)

계획 없이 무작정 개발하는 Build-and-Fix 방식과 구체적 계획을 수립하여 진행 하는 개발 방법 사이에서 타협점을 찾는 방법론

좋은 것을 빠르고 낭비 없이 만드는 개발을 가능하게 해주는 방법론

### 3 - 1. 애자일 적용 전제(원리)

(SW개발을 위한 시간과 비용 범위 안에서)사용자 요구 사항은 불완전하고 변경 예측이 불가능하다(자주 바뀌고, 언제 무엇으로 변경될지 모른다)는 전제가 있다. → 개발 과정에서 늦은 시점이어도 요구사항 변경을 적극적으로 수용한다.

고객 만족을 가능한 한 빨리 이끌어내고, 유연한 SW를 지속적으로 사용자에게 제공한다.

### 3 - 2. 애자일: XP 프로세스

반복 개발이라는 접근 방법 기반으로 한다. 새로운 SW가 하루에도 몇번씩 개발되며, 개발 SW의 증분이 적어도 2주에 1번씩 사용자에게 배포된다. → 철저한 테스트 수행을 통해 성공적으로 통과한 빌드에 대해서만 배포를 허용한다.

#### 개발 사이클

1. 엔지니어링 사이클(개발자 측면)  
팀 내 SW 개발 작업에 대한 사이클이다. 사용자 요구사항에 대한 코딩과 테스트를 통해 개발된 SW가 사용자에게 배포한다.
2. 배포 사이클(사용자 요구사항을 태스크로 분할하여 식별한 다음 계획 수립)  
개발하고자 하는 사용자 스토리(요구사항), 스토리 분할 및 태스크 식별, 태스크 기준 배포계획 수립, 수립 계획에 따라 SW개발/통합/테스트, SW 배포, 평가 및 피드백

### 3 - 3. 애자일: 스크럼 프로세스

반복적 개발의 관리 관점을 강조했다. 사용자에게 배포 가능한 소프트웨어를 반복적으로 개발하고 배포하면서 개발 과정에 **사용자의 요구 사항을 지속적으로 반영하도록 하는 피드백 중심의 경험적 접근 방식**, 개발 SW 프로세스이다.

스크럼 특성을 활용해 SW를 신속하고 반복적으로 개발하고자 하는 프로세스

매일 스크럼 미팅을 하는 이유 → 사용자의 요구사항은 예측 할 수 없고 사용자의 요구사항과 제품의 결과물이 일치하는지에 대한 불확실성이 존재하기 때문에 지속적인 미팅을 통해서 피드백을 지속적으로 반영하기 위함이라 생각.

### 3 - 4. Chaos와 DevOps

**Chaos Engineering:** 다양한 환경에 적응, 새로운 기능을 중단 없이 제공하기 위해서 기존 서비스에 임부러 결함을 주입하고 이런 결함이 어떤 행동을 보이는지 검사하며 시스템 운영에 대응하는 방법을 찾는다.

**DevOps:** 개발 팀 간의 협업, 의사소통의 미흡으로 인한 문제를 해결하기 위해 **SW 개발자와 운영자** 간의 소통과 협업, 통합을 강조하는 SW 개발방법

#### DevOps 핵심 원리!!

- 자동화:
- 반복: 개발 ↔ 운영을 순환구조로 연결해 반복 개발하는 형식
- 자기 서비스: 개발자와 운영자가 독립적으로 일할수 있도록 진행한다.
- 지속적 개선: 사용자의 피드백을 SW를 개선하기 위해 활용
- 협업: 개발, 운영 팀 간의 지속적인 협업 능력을 요구한다.
- 지속적 테스트: 단위 테스트, 통합 테스트와 같은 많은 종류의 테스트 들을 통합과 배포 과정에서 반복한다.

#### DevOps의 이점

1. 제품 출시 기간 단축
2. 낮은 위험과 유연한 배포
3. 신속한 복구
4. 고객 만족도 및 시작 적합도 향상

## 4. 소프트웨어 프로세스 개선

**CMM:** SW 개발 활동에 대한 데이터를 수집하고 이를 체계화한 시스템 개발 프로세스의 성숙모델이다.

**CMMI:** CMM 모델의 형식, 용어, 측정 방법의 차이점이 존재해 통합적으로 적용하기 힘든 단점이 있다. 이런 단점을 해결한 프로세스 성숙모델

**CMMI 성숙도 수준(수준과 관심사 매핑만 해서 외우자)!!!**

수준	관심사	프로세스 영역
5. 최적화 수준 (Optimizing)	지속적인 프로세스 개선	CAR - Causal Analysis and Resolution OPM - Organizational Performance Management
4. 정량적 관리 수준 (Quantitatively managed)	정량적 자료 관리	OPP - Organizational Process Performance QPM - Quantitative Project Management
3. 정의 수준 (Defined)	표준화된 프로세스	DAR - Decision Analysis and Resolution IPM - Integrated Project Management OPD - Organizational Process Definition OPF - Organizational Process Focus OT - Organizational Training PI - Product Integration RD - Requirements Development RSKM - Risk Management TS - Technical Solution VAL - Validation VER - Verification
2. 관리 수준 (Managed)	기본적인 프로젝트 관리	CM - Configuration Management MA - Measurement and Analysis PMC - Project Monitoring and Control PP - Project Planning PPQA - Process and Product Quality Assurance REQM - Requirements Management SAM - Supplier Agreement Management
1. 수행 수준 (Performed)	초기 소프트웨어 개발	해당 없음

## SPICE - 6개의 성숙도 수준!!!

레벨5 ~ 레벨0: 6개

- 6개의 성숙도 수준 정의
  - 레벨 5-최적화 프로세스
  - 레벨 4-예측 가능한 프로세스
  - 레벨 3-구축된 프로세스
  - 레벨 2-관리되는 프로세스
  - 레벨 1-적용하는 프로세스
  - 레벨 0-불완전한 프로세스
- 15504 표준에서 정의하는 프로세스 속성은 (1) 프로세스 성능, (2) 성과 관리, (3) 산출물 관리, (4) 프로세스 정의, (5) 프로세스 배포, (6) 프로세스 측정, (7) 프로세스 제어, (8) 프로세스 혁신, (9) 프로세스 최적화

## A-SPICE(SPICE를 자동차에 접목)

제공하는 역량 수준 6가지 알아 두자!!!

**식스 시그마:** 프로세스 개선을 위한 공학적 기술 및 지원 도구를 제공하는 프로세스 관점의 접근 방법론, 결함의 원인을 식별 및 제거 하고 변경을 최소화한다.

식스 시그마(Six Sigma)와 애자일(Agile)은 둘 다 프로세스 개선과 효율성 향상을 목표로 하는 방법론이다.

### 1. 식스 시그마 (Six Sigma):

- **목표:** 프로세스 내 변동성을 최소화하여 품질 향상과 결함률 감소를 달성하는 것입니다.
- **방법:** DMAIC 사이클 (Define, Measure, Analyze, Improve, Control)을 따릅니다.
- **도구:** 통계적 기법과 품질 관리 도구를 활용합니다.
- **적용 분야:** 제조업 및 공정 중심의 환경에서 주로 사용됩니다.

### 2. 애자일 (Agile):

- **목표:** 변화에 빠르게 대응하고 **고객 만족을 극대화**하는 것입니다.
- **방법:** 반복적이고 점진적인 개발 방식을 채택합니다.
- **특징:** 고객 중심의 접근, 자기 조직화된 팀, 유연한 변경 수용 등이 있습니다.
- **적용 분야:** 주로 소프트웨어 개발 및 프로젝트 관리에 사용되지만, 다른 분야에도 확장되고 있습니다.

간단히 말하자면, 식스 시그마는 안정성과 품질을 중시하는 반면, 애자일은 변화에 유연하게 대응하고 고객의 요구를 신속하게 충족시키는 것에 중점을 둡니다.