

2 주차

2장. 소프트웨어 품질

소프트웨어 품질은 시스템, 구성 요소, 프로세스가 명시된 요구사항을 충족시키는 정도를 의미한다.

1. 소프트웨어 품질의 정의

SW 품질의 특징

■ 소프트웨어 품질의 특징

- 소프트웨어는 물리적으로 존재하지 않아 품질을 눈으로 확인하기 어렵다.
- 개발 초기에 사용자의 요구를 건축물의 설계 도면과 같이 정확히 알 수 없다.
- 시간이 지날수록 사용자가 원하는 품질 수준이 점점 높아진다.
- 소프트웨어와 하드웨어 기술이 급진적으로 발전한다.
- 특정 제품의 품질을 100점, 90점과 같이 절대적으로 평가할 수 없다.
- 품질은 다양한 관련자들을 고려해야 하는 다차원적인 것이다.
- 소프트웨어 품질은 개발자의 능력, 예산, 개발 기간 등에 따라 달라질 수 있다.
- 품질은 스테이크홀더가 수락할 수 있는 수준에 존재해야 한다.

SW 품질의 중요성

모든 요소를 모두 만족시키지 못하기 때문에 대상 소프트웨어의 특성을 고려한 적절한 품질 목표 설정이 중요하다. (완벽한 SW는 없기에)

사용자가 만족하는 품질의 소프트웨어를 품질이 좋은 소프트웨어라고 한다. 품질이 좋은 제품은 시장에서 선호도가 올라가고 이는 기업의 이익 창출을 기대한다.

2. 소프트웨어 품질 요소

스테이크 홀더: 사용자, 개발자, 운영자, 관리자 등등의 주체

역할	기대치	대응 품질 요소
스폰서 (Sponsor)	<ul style="list-style-type: none"> • 적은 비용으로 소프트웨어가 개발되기를 원한다. • 개발된 소프트웨어가 다른 응용에 사용되기를 원한다. • 기존 컴포넌트의 재사용을 통해 개발되기를 원한다. • 비즈니스 과정에 이익이 되기를 원한다. 	<ul style="list-style-type: none"> - 낮은 비용 - 적응성(Adaptability) - 재사용성 - 비용 효율성
사용자 (User)	<ul style="list-style-type: none"> • 소프트웨어의 기능이 정확히 동작하기를 원한다. • 고장이 나지 않기를 바란다. • 사용하기 쉬워야 한다. • 적은 비용으로 구매하기를 원한다. 	<ul style="list-style-type: none"> - 정확성 - 신뢰성 - 사용성(Usability) - 낮은 비용
유지보수자 (Maintainer)	<ul style="list-style-type: none"> • 소스 코드를 이해하기 쉬워야 한다. • 표준 코딩 스타일에 따라 코드가 개발되기를 원한다. • 변경 영향이 한 부분으로 국한되기를 원한다. • 수정된 코드를 쉽게 테스트하기를 원한다. • 코드와 일관성 있는 문서가 제공되어야 한다. 	<ul style="list-style-type: none"> - 가동성 - 코딩 표준 준수성 - 프로그램 구조 - 검증 가능성 - 문서화

각각 스테이크 홀더의 관심사.

소프트웨어 품질 요소: 외적 품질 요소와 내적 품질 요소(아이스 버그 그림 참고)

1. 외적 품질 요소 (보여지는 증상, 기능적 측면, 사용자 위주 측면)

정확성, 신뢰성, 견고성, 성능, 사용자 친숙성, 가용성, 보안성, 안전성, etc..

1. 정확성: 주어진 명세서의 내용을 하나씩 테스트하여 원하는 결과를 생성하는지 여부로 판
2. 신뢰성: SW를 사용하는 동안 나타나는 오류 발생 정도
3. 견고성: 사용자가 제시한 요구 사항 명세에 정의하지 않은 조건이나 환경에서도 소프트웨어가 합리적으로 동작해야 견고하다고 한다.
4. 성능: SW의 효율성 - 메모리 사용량, 총 실행 시간 등으로 측정한다.
5. 사용자 친숙성: 소프트웨어가 사용하기 편리한 지를 나타내는 품질 요소
6. 가용성: 서버 네트워크 프로그램의 정보 시스템이 정상적으로 사용 가능한 정도 (SW를 정상적으로 사용 가능한 시간 / SW 전체 운영 시간)
7. 무결성: 프로그램이나 데이터에 대한 불법, 잘못된 접근을 막는 정도
8. 외부의 악의적인 공격이나 해커의 위협을 소프트웨어가 막아낼 수 있도록 구현 해 잠재적인 공격이 예측되는 상황에서도 SW가 올바르게 동작

2. 내적 품질 요소 (감춰진 증상, 개발자 위주 측면)

검증 가능성, 유지 보수성, 재사용성, 이식성, 가독성, 생산성, 상호 운용성, 적시성, 가시성, 코드 규칙, etc..

1. 검증 가능성: 소프트웨어가 지닌 속성이 올바르다는 것을 안전하게 확인할 수 있다면 검증 가능성을 갖는다고 한다.
정형 검증과 테스트로 평가한다.
2. 유지 보수성
 - a. 수정 유지 보수: 개발된 소프트웨어를 사용하는 동안 오류가 발생하는 경우에 이루어지는 활동
 - b. 적응 유지보수: 소프트웨어 운영 조건에 대한 변화를 수용하는 유지보수 활동
 - c. 완전 유지 보수(리뉴얼하기): 사용 중인 SW에 대한 가독성 및 이해성을 높이는 재구조화를 목적으로 수행
 - d. 예방 유지 보수: 소프트웨어가 정확하게 동작할 것인지를 테스트하고 점검

임무 중심 및 안정 중심 소프트웨어의 경우에 매우 중요한 요소이다.

3. 재사용성: 새로운 소프트웨어를 개발하기 위해 기존 SW 컴포넌트를 사용하는 정도
 $\text{재사용성} = \text{LOC(R)} / \text{LOC(S)} = \text{재사용에 의해 개발된 부분의 양} / \text{전체 개발된 산출물의 양}$
4. 생산성: 외적 품질 요소인 성능의 속성을 적용한 내적 품질 속성, 주어진 시간 내에 얼마만큼의 성과를 내는가
5. 상호 운용성: 서로 다른 소프트웨어들이 협업을 수행할 수 있는 능력을 충분히 제공하는 것을 의미한다.
6. 가시성: SW 개발 단계의 상태 정보와 산출물 등을 체계적으로 유지 및 관리 (ex 문서화해서 SW 개발 과정을 관리)

3. 프로세스 품질

- a. 프로세스 모델 적합성: 어떤 프로세스 모델을 적용할 것인가
- b. 개발 방법론 적합성: 객체 지향 방법, 구조적 방법, 정보 공학 방법등이 활용된다.
- c. 도구 적합성: SW 개발 과정에서 많은 지원 도구(CASE 툴)이 개발 환경에 적합한 지
- d. 표준 준수성: 선택한 표준이 적절한 지, 선택한 표준을 준수하여 프로젝트가 진행되고 있는지
- e. 프로젝트 데이터 관리 수준: 프로젝트 진행과 관련된 상세 데이터가 정보 저장소에 저장되어 프로젝트 내에서 혹은 추후 신규 프로젝트의 예측 활동에서 활용할 수 있는지

3. 인공지능 소프트웨어 품질

전통 SW vs 인공지능 SW 의 차이점

인공지능 SW는 학습 모델의 입력과 결과 사이의 관계가 입력 데이터 일부에 대해서만 정의된다.
즉,

학습한 데이터에 의해서 입력과 결과 사이의 관계가 결정된다는 뜻.

인공지능 SW은 캡슐화, 모듈화 같은 개발 원칙을 적용하기 힘들다.

인공지능 SW은 컴포넌트 개발과 통합의 접근 방법이 매우 다양하다.

인공지능 SW에서 학습 용 알고리즘 보다 학습에 사용되는 데이터와 테스트 데이터가 더 중요하다.

인공지능 SW 품질 특성

1. 투명성과 책임: 기계학습 기반 시스템은 동일한 입력에 대하여 서로 다른 결과가 나올 수 있다.(학습 방식에 의해 결정된다)
2. 다양성, 공정성 그리고 사회적 웰빙
 - a. 다양한 데이터를 수용하여 출력할 수 있어야 한다.
 - b. 특정 요소에 편향되어 출력을 제공하는 오류가 있으면 안된다.
 - c. 이해 관계자의 참여 과정을 통해 인공지능 시스템이 환경 친화적 방향으로 구축되도록 한다.
3. 보안과 안전성: 인공지능 시스템의 응용 범위가 넓어져 개인 정보의 누출이나 프라이버시 침해를 고려해야한다.
4. 기술적 견고성과 신뢰성: 유해한 입력, 오류 있는 입력에 대해서도 믿을 만하고 이해할 만한 결과 (=타당한 결과)가 제공 되어야 한다.
5. 법적 + 윤리적 측면: 인간이 해온 일은 AI가 할 때 법적, 윤리적 책임에 대해 고려해야 한다.

ML(머신러닝) 시스템의 구성 관점 (개념만 이해하자, 사진 외우지마)

1. 모델 관점: 학습 모델 관련, 분류, 차원 축소 같은 작업에 대한 데이터 훈련
2. 데이터 관점: 모델에 입력되는 실제 데이터
3. 시스템 관점: 기계학습 컴포넌트들을 연결하고 형상을 정의하는 부분(출력, 범주 제어)
4. 인프라 관점: 어떻게 구현되는가에 초점을 맞춘 부분
5. 환경 관점: 시스템과 사용자의 상호작용을 표현한 부분

표 2-2 모델 관점의 품질 요소

측정 대상	모델 관점 품질 요소
모델 유형	모델 타당성: 정의된 기능을 수행하기 위해 선정된 모델 유형이 적합한가
훈련된 모델	적합성(Fitness): 정의된 기능이 개발된 데이터를 기반으로 정확하게 수행될 수 있는가
	견고성(Robustness): 누락 혹은 오류가 있는 데이터를 잘 처리할 수 있는가
	안정성(Stability): 서로 다른 데이터에 대하여 반복적인 결과를 생성할 수 있는가
	공정성(Fairness): 모델의 출력이 공정한 결정을 제시하는가
	해석능력(Interpretability): 훈련된 모델의 내용을 사람이 해석할 수 있는가

표 2-3 데이터 관점의 품질 요소

측정 대상	데이터 관점 품질 요소
개발 데이터	대표성(Representativeness): 데이터가 모집단을 대표하는가
	정확성(Correctness): 데이터가 오류 없이 개발되었는가
	완전성(Completeness): 누락된 데이터가 없는가
	유통성(Currentness): 데이터가 최신 내용을 포함하는가
	독립성(Independence): 훈련 데이터와 테스트 데이터가 상호 독립적인가
개발 및 운용 데이터	일관성(Consistency): 서로 다른 데이터셋에 대하여 형식, 표본 추출 등이 일관성이 있는가

표 2-6 시스템 관점의 품질 요소

측정 대상	시스템 관점 품질 요소
출력 제어	효과성(Effectiveness): 출력 제어 알고리즘이 오류 출력을 탐지하는 능력
	제어 효율성(Supervision Efficiency): ML 컴포넌트를 모니터링하기 위해 사용되는 자원이 유용한가
범주 제어	효과성(Effectiveness): 범주 제어 알고리즘이 문맥 변경을 탐지하는 능력
	제어 효율성(Supervision Efficiency): 응용 범주를 모니터링하기 위해 사용되는 자원이 유용한가

표 2-5 인프라 관점의 품질 요소

측정 대상	인프라 관점 품질 요소
인프라구조	인프라 적합성(Suitability): 인프라 구성 요소가 ML 컴포넌트의 요구를 충족하는가
훈련 알고리즘	훈련 효율성(Training Efficiency): 학습 모델을 훈련하기 위한 자원이 유용한가
실행 알고리즘	실행 효율성(Execution Efficiency): 훈련된 모델을 실행시키기 위한 자원이 유용한가

표 2-4 환경 관점의 품질 요소

측정 대상	환경 관점 품질 요소
훈련 과정	환경 영향(Env. Impact): 훈련 과정이 환경에 영향을 미치는 정도
활용 집단	사회 영향(Social Impact): ML 컴포넌트가 사회에 영향을 미치는 정도
범위	범위 준수(Scope Compliance): ML 컴포넌트가 의도된 사용 범위 내에 존재하는 정도

page 78 실습 2-2 GQM 방법 구글링, SNS 플랫폼 '가상 가족'을 개발하는 데 고려해야 하는 새로운 품질 요소 설계

4. 소프트웨어 품질 모델 및 표준

FCM 기본 개념

1. Factors: 사용자가 보는 SW 외적 특성
2. Criteria: 개발자가 보는 SW 내적 특성
3. Metrics: SW의 내적, 외적 특성을 측정하기 위한 기본 단위와 방법

ISO 9126 품질 모델

이식성, 기능성, 신뢰성, 유지보수성, 사용성, 효율성,
이기효신유사(이기 박효신이랑 유사하노)

5. 소프트웨어 품질 관리

정량적 품질 개선 프로세스

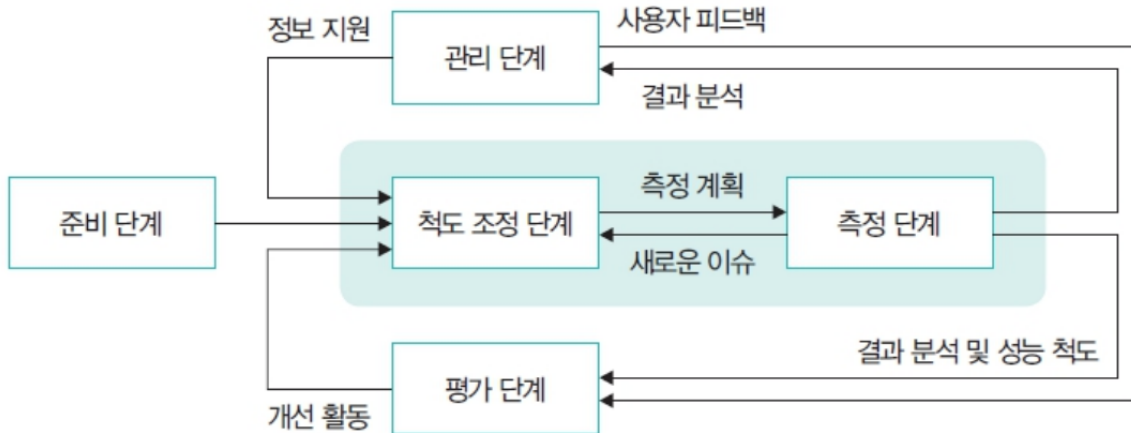


그림 2-13 소프트웨어 품질 측정을 위한 상위 프로세스

1. 준비 단계: 프로젝트 조직원에게 알리고 품질 관리 팀 구성, 담당장 지정하고 품질 관리 활동을 위한 준비를 구축하는 단계
2. 척도 조정 단계: 프로젝트의 품질 요소 중 어떤 요소를 측정할 지 결정
3. 측정 단계: 품질 측정 계획서를 바탕으로 프로젝트 데이터를 수집 및 정량화 해 분석한다.
4. 평가 단계: 측정된 품질 요소들이 올바른 품질 정보를 제공하는지 점검하고, 품질 수준을 평가
5. 관리 단계: 조직 차원에서 수행하는 품질 관리 및 기술 관리 프로세스

요약

소프트웨어 품질이란: 사용자의 기대치에 대한 충족 정도

소프트웨어 품질 요소:

1. 외적 품질 요소 vs 내적 품질 요소
2. 프로덕트(제품) 품질 요소 vs 프로세스 품질 요소

인공지능 소프트웨어의 품질 요소

1. 인공지능 모델 관점의 품질 요소: 적합성, 견고성 etc
2. 데이터 관점의 품질요소: 대표성, 정확성 etc
3. 환경 관점의 품질 요소 와 인프라 관점의 품질 요소
4. 시스템 관점의 품질 요소: 효과성, 제어 효율성

소프트웨어 품질 관리 및 표준

프로젝트2 - 스테이크 홀더 관련 질문

프로젝트 II

대상 시스템 품질 요소 추정하기

1장의 프로젝트 정의서에서 제시한 핵심 기능과 유사 시스템 분석 결과를 기반으로, 대상 시스템에서 중요하게 고려해야 할 품질 요소를 추정하고 정리합니다. 이 사항은 추후 프로젝트 관리 계획서 작성에 반영됩니다.

- (1) 사용자가 원하는 기대치는 무엇인가?
- (2) 개발자 혹은 유지보수 담당자가 원하는 기대치는 무엇인가?
- (3) 개발 조직의 관리자가 원하는 기대치는 무엇인가?

나의 프로젝트는 글씨체 딥러닝 일기 프로젝트이다.

1. 사용자가 원하는 기대치는 무엇인가?
 - a. 사용하기 쉬움
 - b. 값싼 가격
 - c. 신뢰성
2. 개발자 혹은 유지 보수 담당자가 원하는 기대치는 무엇인가?
 - a. 코드가 어렵지 않아야 한다.
 - b. 테스트 케이스를 작성하기 쉬워야 한다.
 - c. 학습 데이터를 잘 선별해야한다.
3. 개발 조직의 관리자가 원하는 기대치는 무엇인가?
 - a. 적은 비용으로 개발
 - b. 다른 SW에도 응용되어 사용할 수 있기를 (적용성)
 - c. 비즈니스 과정에서 이득 (비용 효율성)

스테이크 홀더 기대치와 품질요소