



소프트웨어 : 그 본질 및 자질

가을, 2024

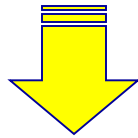


제홍@충북.ac.kr

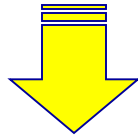
다시 방문: 소프트웨어 엔지니어링의 목표

생산하라고 품질 소프트웨어
사용자의 요구를 충족하다

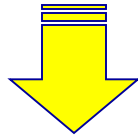
예산 내에서 정해진 시간에



소프트웨어의 품질과 생산성을 향상시키세요



시스템, 제품의 품질과 생산성을 향상시키세요



사업 성과 개선

소프트웨어 품질

정의

- DoD, 1985, "소프트웨어의 속성이 의도한 최종 용도를 수행하는 정도."
- ISO, 1986, "지정되거나 암시된 필요를 충족시키는 능력에 영향을 미치는 제품 또는 서비스의 전체적 특징 및 특성."
- 키친햄, 1986, "필요에 대한 적합성".
 - 사양에 대한 적합성: 좋은 솔루션인가?
 - 의도된 목적에 대한 적합성
: 올바른 문제를 다루고 있나요?

→ **사용자의 요구에 맞게 소프트웨어 제품이 적응할 수 있는 능력**

소프트웨어의 품질...

어떤 것을 원하시나요?



제품을 선택하는 기준은 무엇인가요?

품질에 대한 몇 가지 통찰력

품질은 절대적이지 않습니다. 품

질은 다차원적입니다.

품질은 제약(사람, 돈, 시간, 도구)의 영향을 받습니다. 품질은 수용 가능한 타협에 관한 것입니다.

품질 기준은 독립적이지 않습니다

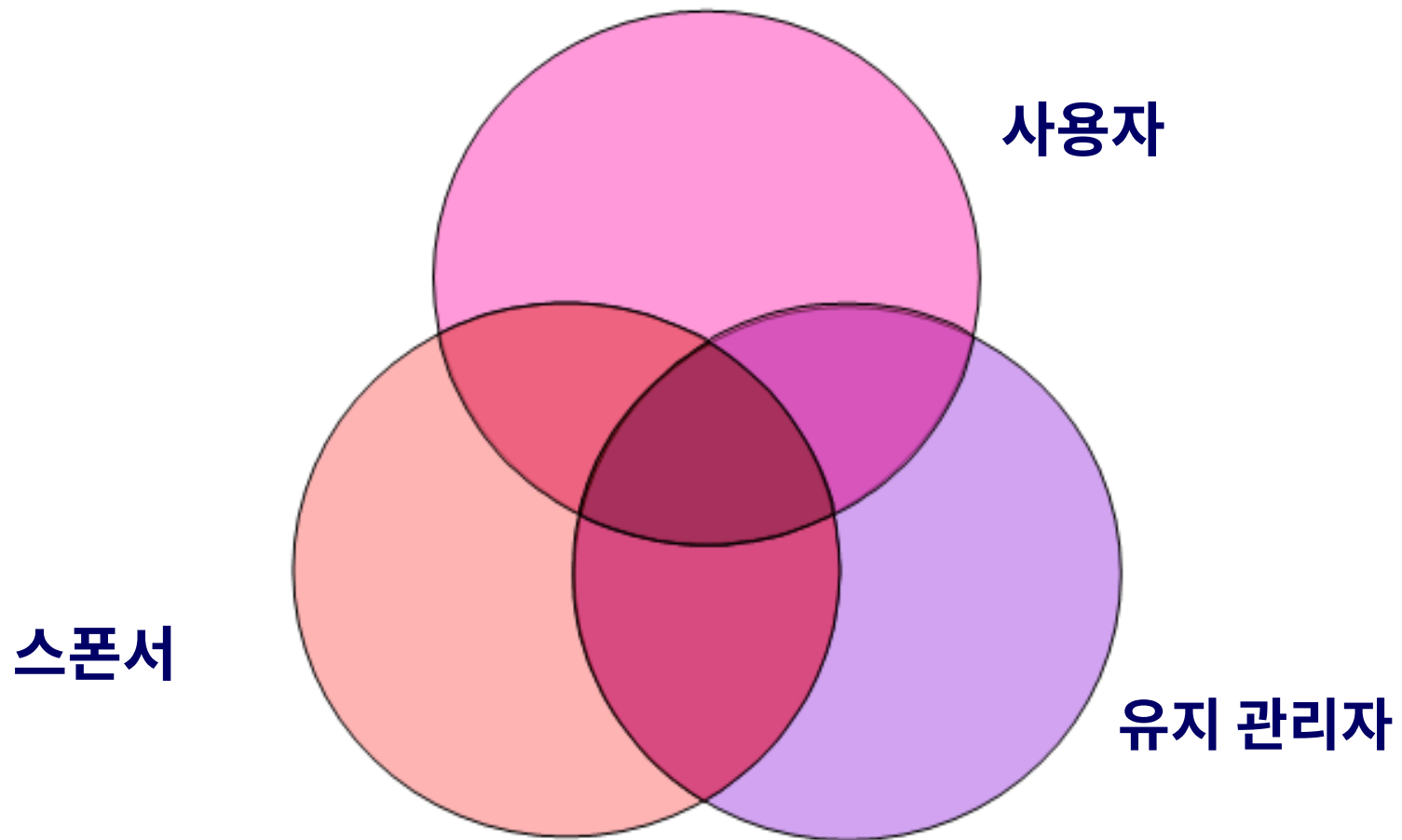
소프트웨어 품질이 다른 것과 다른 이유 품질의 유형?

소프트웨어는 물리적으로 존재하지 않습니다.

시작 시 고객 요구 사항에 대한 지식 부족. 시간이 지남
에 따라 고객 요구 사항 변경.

하드웨어와 소프트웨어 모두 변화 속도가 빠릅니다. 고객의
기대도 높습니다.

품질 요인



품질 분류

품질

- 외부 및 내부 품질
- 제품 및 프로세스 품질

외부 품질과 내부 품질

- 구별이 뚜렷하지 않다
- 외부 품질: 시스템 사용자에게 표시됩니다.
- 내부 품질: 시스템 개발자와 관련된 사항입니다.

제품 및 프로세스 품질

밀접한 관련: SW 제품을 생산하기 위해 프로세스를 사용합니다.

다. 제품 품질:

- 기능성
- 사용성
- 능률
- 신뢰성 등

공정 품질

- 방법, 도구의 효과
- 표준의 사용(표준 준수)
- 경영 등

대표적인 특성

정확성, 신뢰성 및 견고성 성능

사용자 친화성

검증 가능성

유지 보수성

재사용성

휴대성

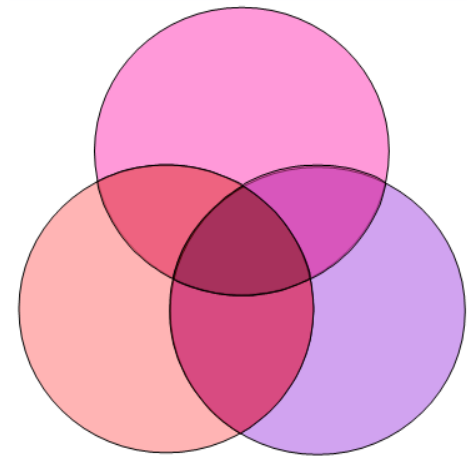
이해성

상호 운용성

생산력

적시

시계



정확성, 신뢰성 및 견고성

Open은 상호 교환적으로 사용되며, 이는 정도를 의미합니다.
응용 프로그램이 예상대로 기능을 수행하고 있습니다.

단정

프로그램이 사양에 따라 동작하는 경우 수정하세요.
기능.

추정 :

- 시스템 사양을 확인할 수 있습니다.
- 프로그램이 사양을 충족하는지 여부는 명확하게 판단할 수 있습니다.

신뢰할 수 있음

= 신뢰성: 사용자가 의지할 수 있다면 신뢰할 수 있음.

제품 실패의 빈도 및 심각도 측정

실패: 허용 가능한 범위 내에서 허용할 수 없는 효과 또는 동작 작동 조건.

통계적 동작 측면에서 정의할 수 있습니다: MTTF, MTBF 상대적 값 (Q: 정확성은 어떻습니까?)

견고성

합리적으로 동작한다면 견고합니다. 심지어 다음과 같은 상황에서도 마찬가지입니다.
요구사항 사양에 예상되지 않음.

다음과 같은 여러 요소의 기능

- 작동 조건의 범위.
- 유효한 입력에도 불구하고 받아들일 수 없는 결과가 발생할 가능성이 있습니다.
- 제품에 잘못된 입력이 주어졌을 때 효과의 수용 가능성.

Q: 정확성과의 관계

Q: 신뢰성과의 관계

성능

성능을 효율성(공간, 시간)과 동일시합니다. 시스템의 사용성에 영향을 미칩니다.

평가

- 측정(모니터링)
- 분석
- 시뮬레이션

질문: 언제 성과를 추정해야 할까요?

성능의 프로세스 적용 → 생산력

사용자 친화성

사용하기 편리함

시스템을 구성하고 적용할 수 있는 용이성
개인화된 환경.

검증 가능성

해당 속성을 안전하게 검증할 수 있는 경우 검증 가능합니다.
공식적인 분석 방법이나 테스트를 통해 수행됩니다.

유지 보수성

유지

- 교정
- 적응형
- 완료형
- 예방법

소프트웨어 진화(유지관리 대신) 수리 가능성 및
진화 가능성

유지보수성(계속)

수리가능성

- 소프트웨어 시스템이 제한된 양의 작업으로 결함을 수정할 수 있는 경우 수리 가능합니다.
- 질문: 유지관리 단계에서 오류를 수정하는 데 몇 시간이 필요합니까?
- 올바른 모듈화는 수리성을 증진시킵니다. 질문: 왜 그럴까요?
- 컴퓨터 공학에서의 가용성, 서비스성
- 질문: 소프트웨어와 하드웨어의 수리 가능성의 차이점은 무엇입니까?
- 적절한 도구(HL PL, CASE 등)를 사용하여 개선되었습니다.
- Q: 신뢰성과의 관계

유지보수성(계속)

진화 가능성

- 소프트웨어는 시간이 지남에 따라 수정됩니다
 - 새로운 기능을 제공합니다.
 - 기존 기능을 변경합니다.
- 더 크고 복잡한 소프트웨어에서 중요함
- 모듈화를 통해 달성
- 새 버전이 출시될 때마다 감소
- 질문: 그러면 우리에게 필요한 것은 무엇일까요?
- 경제, 사업에 미치는 영향
- 제품군 및 소프트웨어 아키텍처를 통한 진화 가능성 강화

재사용성

기존 구성 요소를 사용하여 새 제품을 빌드합니다. 예: 과학 라이브러리, MFC, Eclipse 플러그인 등 재사용 수준

- 사람들
- 요구 사항
- 설계
- 암호

재사용의 다른 수준

- 모듈 - 기능 - _____ - _____.

재사용의 프로세스 적용

- 소프트웨어 방법론
- 라이프 사이클 모델

휴대성

다양한 환경에서 실행할 수 있다면 휴대성이 뛰어납니다.

- 하드웨어 플랫폼
- 소프트웨어 플랫폼

모듈화를 통해 달성

- 질문: 이러한 모듈화를 어떻게 할 수 있나요?

이해성

내부적인 제품 품질.

객체 지향 패러다임은 이해하기 쉽다고 주장합니다. 추상화와 모듈성으로 강화되었습니다.

질문: 유지보수성과의 관계는?

상호 운용성

다른 시스템과 공존하고 협력할 수 있는 시스템의 능력 인터페이스 표준화를 통해 달성
개방형 시스템 개념

생산력

효율성과 관련된 소프트웨어 생산 프로세스의 품질
성능

측정하기 어려움:

- 간단한 메트릭: SLOC
- 기능 기반 메트릭 : FP

적시

제품을 제때에 배송할 수 있는 능력. 출시 시간 과제

필요합니다

- 신중한 일정 조정,
- 정확한 작업 추정 및
- 명확하게 명시된 이정표.

충분적 전달로 달성됨

시계

모든 단계와 현재 상태가 문서화되어 있는 경우 표시됩니다. 투명성

- 프로젝트의 단계 및 상태는 외부 검토를 위해 사용 가능하며 접근 가능합니다.

**그들의 행동의 영향을 평가하고 이를 통해 지침을 제공합니다.
결정을 내리는 것.**

요구사항 사양 및 설계 사양 제품이 보입니다

- 모듈의 집합으로 명확하게 구조화됨
- 명확하게 이해할 수 있는 기능 및
- 이용 가능하고 정확한 문서

질문: 가시성이 왜 중요한가요?

보안

소프트웨어를 보호하기 위해 구현된 아이디어가 안전하다면
악의적인 공격 및 기타 해커 위험

소프트웨어는 이러한 상황에서도 계속해서 정상적으로 작동합니다.
잠재적 위험.

무결성, 인증 및 가용성을 제공하는 데 필요합니다.

보안 취약점: 실제로 발생할 수 있는 약점/결함
공격자가 악용하려면 결함이 필요합니다.

-접근 가능: 공격자는 접근할 수 있어야 합니다.

-공격 가능: 공격자는 이를 사용하여 어느 정도 피해를 입힐 수 있어야 합니다.

보안 위험, 보안 취약성 및 사고 통계
연간 손실 예상이 몇 가지 지표입니다.

안전

소프트웨어 안전

- 소프트웨어 위험으로부터 자유로움(IEEE Std-1228, 1994)
- 위험한 상황을 피하고 상황이 안전하지 않게 되면 올바른 시스템에 경고하는 데 관심이 있습니다.

소프트웨어 위험

- 사고의 전제조건인 SW(운영) 조건

사고

- 사망, 부상, 질병, 환경 피해 또는 재산 손실을 초래하는 의도치 않은 사건

소프트웨어 안전성을 어떻게 측정하는가?

특정 응용 분야의 품질 요구 사항

정보 시스템

실시간 시스템

분산 시스템

임베디드 시스템

정보 시스템

데이터의 저장 및 검색

예: 은행 시스템, 도서관 카탈로그 시스템 등.

- 데이터 무결성
- 보안
- 데이터 가용성
- 거래 성과
- 사용자 친화성

실시간 시스템

미리 정의되고 엄격한 기간 내에 응답하세요

예: 공장 모니터링 시스템, 미사일 유도 시스템
마우스 처리 소프트웨어

제어 지향 시스템 OS 레벨

에서의 스케줄링

- 사선
- 우선 사항

자질

- 응답 시간 요구 사항(정확성 기준)
- 안전

분산 시스템

병렬성과 커뮤니케이션

- 작업 할당, 파티셔닝

분포의 정도

- 데이터

- 제어

- 하드웨어

예: 클라이언트/서버 시스템의 미들웨어, 그룹웨어 등.

- 시스템 가용성

- 코드 이동성

임베디드 시스템

소프트웨어는 많은 구성 요소 중 하나입니다. 최종 사용자와의 인터페이스가 없거나 제한적입니다.

예: 비행기, 로봇, 전자레인지, 식기 세척기
자동차 등

자질

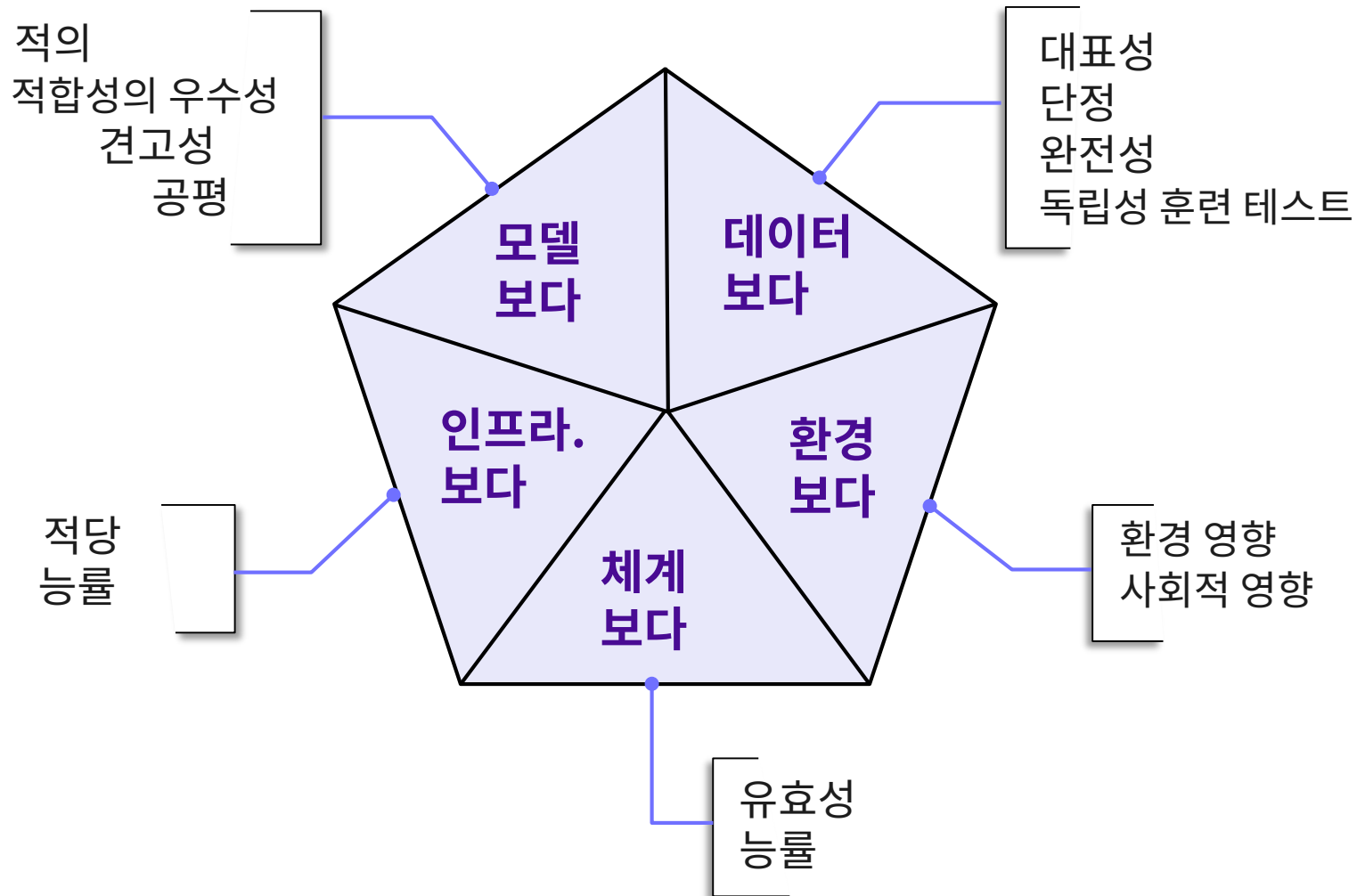
- 신뢰할 수 있음
- 적응성
- 메모리 및 성능 효율성

AI/ML 시스템을 위한 품질 요소

기존 소프트웨어와 근본적으로 다릅니다

- 그만큼 **입력과 결과 사이의 관계** 모델의 정의는 데이터의 하위 집합에 대해서만 정의되므로 이전에 보지 못한 데이터에 대한 모델 결과에 불확실성이 발생합니다.
- 캡슐화 및 모듈성과 같은 소프트웨어 엔지니어링의 공통 개발 원칙은 재고되어야 합니다. 예를 들어, **신경망은 단순히 더 작은 크기로 절단될 수 없습니다.** 서브넷을 모듈로 재사용합니다.
- ML 구성요소의 개발 및 통합은 **다학제적 접근**: 애플리케이션 도메인에 대한 지식, ML 모델을 구성하는 방법에 대한 지식, 마지막으로 소프트웨어 엔지니어링에 대한 지식이 필요합니다.
- 모델을 실행하는 알고리즘은 멀리까지 플레이합니다 **데이터보다 덜 중요한 역할** 훈련 및 테스트에 사용됨

AI/ML 시스템을 위한 품질 요소



출처 : 일본 QA4AI 컨소시엄

소프트웨어 품질에 대한 표준

품질 요소가 너무 많아요

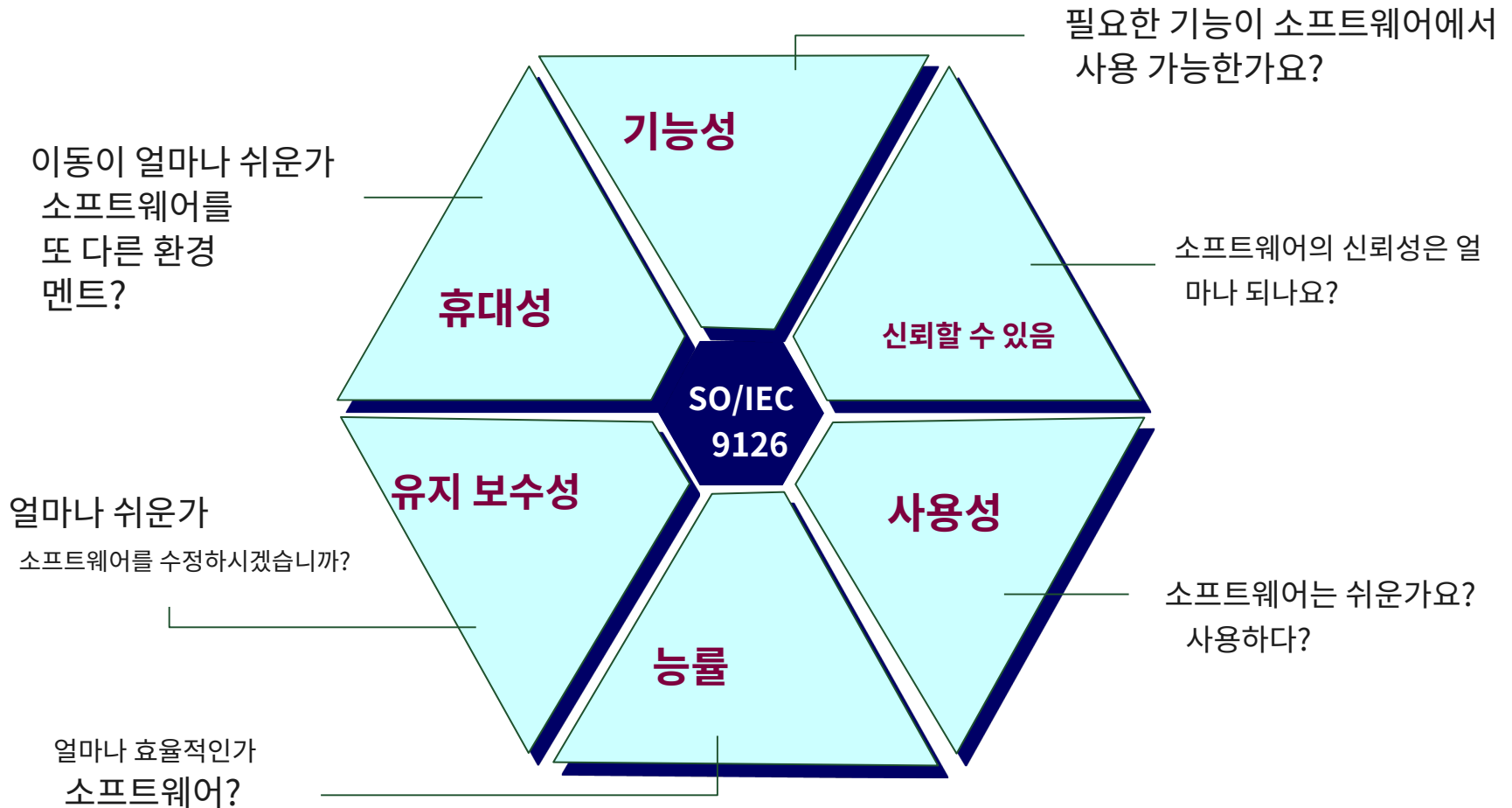
- 이러한 요소들은 서로 독립적이지 않습니다!
- 일부 품질 요인은 너무 밀접하게 관련되어 있습니다
 - 조사하기 / 평가하기 어려움
 - 시스템에 적용하기 어려움

소프트웨어 품질에 대한 표준

- ISO 9126 : 소프트웨어 엔지니어링 - 제품 품질
- ISO 25010 : 시스템 및 소프트웨어 품질 요구 사항 및 평가(SQuaRE) - 시스템 및 소프트웨어 품질 모델
- 그리고 더 많은 것...

ISO 9126: 6가지 품질 특성(2/3)

소프트웨어의 6가지 품질 특성



6가지 품질 특성 (3/3)

6가지 특성의 특징 및 하위 특징

형질	하위 특성
? 기능	적합성, 정확성, 상호 운용성, 규정 준수, 보안
? 신뢰성	성숙도, 내결함성, 복구 가능성
? 사용성	이해성, 학습성, 운용성
? 효율성	시간 행동, 자원 행동
? 유지 보수성	분석성, 변경성, 안정성, 테스트 가능성
? 휴대성	적응성, 설치성, 적합성, 교체성

ISO9926

ISO 25010: 8가지 품질 특성(1/3)

ISO/IEC 25010(2011), 시스템 및 소프트웨어 품질 요구 사항 및
평가(SQuaRE) - 현재 9126 표준을 대체하는 시스템 및 소프트웨어
품질 모델
차이점을 구별합니다

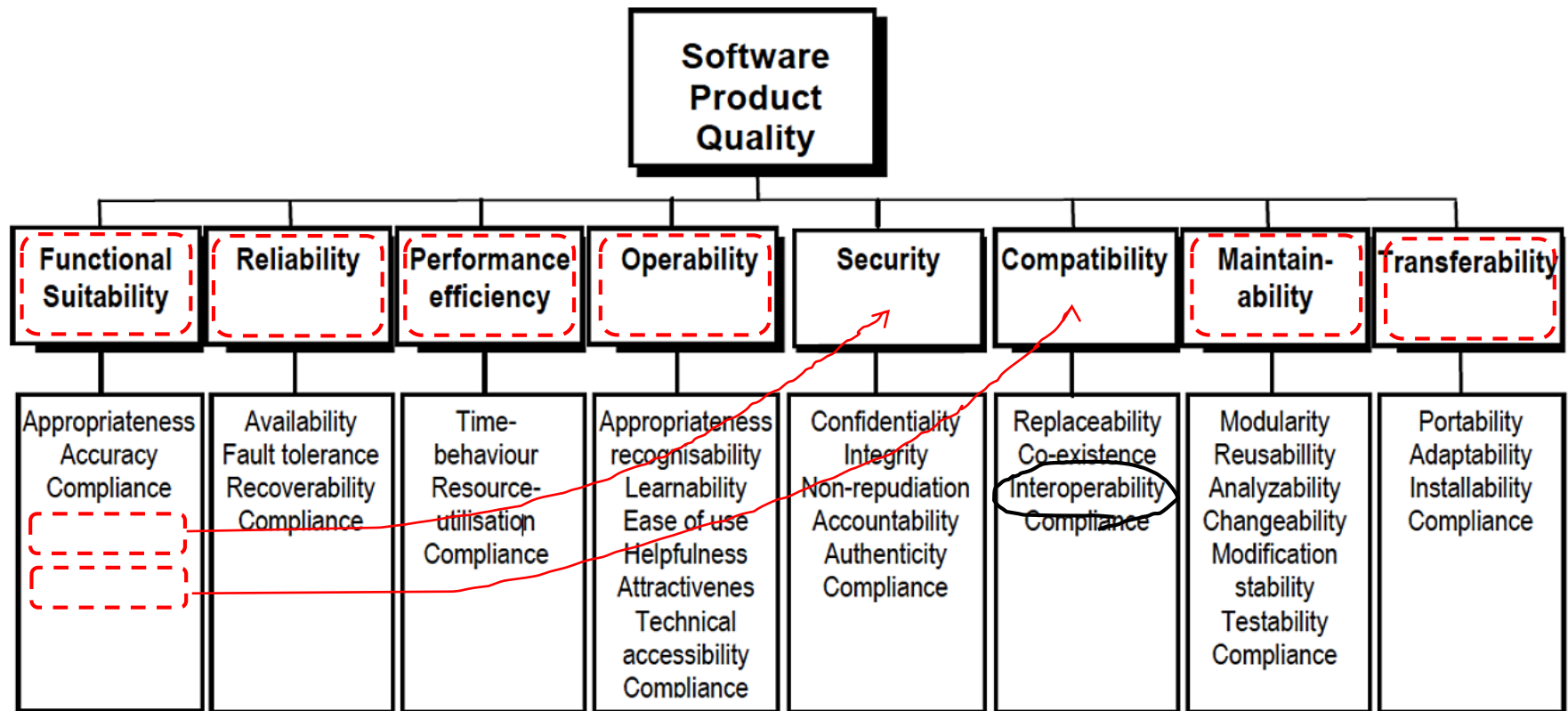
- 제품 품질 모델
- 사용 모델의 품질

ISO/IEC 25010에서는 외부 품질과 내부 품질이 특정한 의미를 갖습니다.
빠대:

- 외부 품질**블랙박스 측정을 통해 제품 품질 모델의 특성을 평가합니다.
- 내부 품질**Glass-box(즉, White-box) 측정을 통해 제품 품질 모델의 특성을 평가합니다. 즉, 소프트웨어의 내부 구조에 대한 지식을 기반으로 시스템 속성을 측정합니다.

8가지 품질 특성 (1/3)

ISO/IEC 25010(2011), 제품 품질 모델

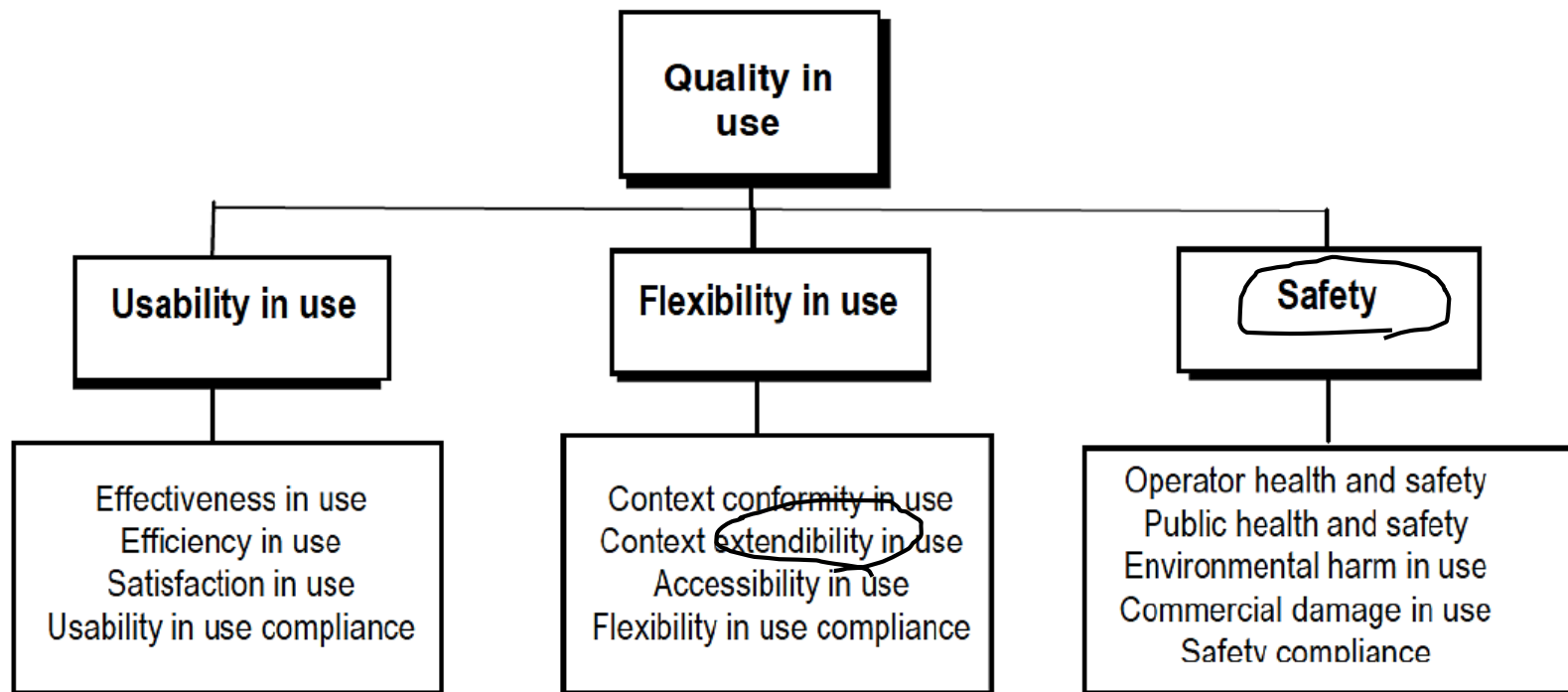


9126과의 관계는 무엇입니까?

[Red dashed box]: ISO/IEC 9126 구조

8가지 품질 특성 (1/3)

ISO/IEC 25010(2011), 사용 품질 모델



현실적인 시스템 환경(운영)에서만 달성됩니다.

요약 및 토론

소프트웨어 품질 정의

-필요에 따른 적합성

대표적인 품질 요인

- 정확성, 신뢰성, 성능, 사용자 친화성, ...
- 유지보수성, 재사용성, 이해성, 가시성, ...
- AI/ML 시스템을 위한 품질 요소

왜 소프트웨어 품질이 그렇게 중요한가요?

예를 들어 소프트웨어 품질을 측정하는 방법
유지보수성?

