

4강. 소프트웨어 품질

메 뉴

학습에 앞서

■ 학습개요

소프트웨어 품질 개념과 중요성을 알아보고 품질 관련 표준들을 소개한다. 소프트웨어 제품의 품질 평가를 위한 모델들과 개발 프로세스의 성숙도를 평가하는 모델들을 살펴본다. 이와 함께 품질 목표 달성을 위한 관리 활동인 소프트웨어 품질 보증, 결함의 발견과 수정을 위한 품질 제어, 그리고 변경을 통제하기 위한 형상 관리 활동의 의미를 살펴본다.

■ 학습목표

1	소프트웨어 품질의 중요성을 설명할 수 있다.
2	소프트웨어 제품의 품질 특성을 나열할 수 있다.
3	소프트웨어 개발 프로세스의 성숙도를 비교할 수 있다.
4	소프트웨어의 품질 보증 계획을 설명할 수 있다.
5	소프트웨어 품질 제어를 위한 검토 방법을 구별하여 설명할 수 있다.
6	신뢰도 표현에 사용되는 메트릭들을 구별할 수 있다.

■ 주요용어

용어	해설
소프트웨어 품질	소프트웨어가 가지는 수월성에 관한 특성으로 요구사항의 만족, 개발 과정에서 표준의 준수 및 묵시적 좋은 특성을 가지는 것
ISO/IEC9126	소프트웨어 제품의 품질 특성들과 메트릭을 정의한 표준
CMMI	조직의 개발 프로세스 성숙도를 평가하기 위한 통합 모델
SQA(소프트웨어 품질 보증)	소프트웨어 개발 과정에서 품질 수준을 보증하기 위해 수행하는 품질 관리 활동
QC(품질 제어)	제품이나 개발 프로세스가 품질 보증 절차를 따르고 품질 목표를 만족하는가를 확인하는 활동
V&V(확인과 검증)	전체 생명 주기에 걸쳐 소프트웨어가 명세에 맞게 작성되었는지 확인하고 고객의 요구를 충족하는지 검증하는 활동
인스펙션	설계 문서나 코드를 작성자가 아닌 동료나 전문가 팀이 검사하여 오류를 찾아내는 검토 방법
소프트웨어 신뢰도	일정 시간 동안 주어진 환경에서 의도된 기능을 고장없이 수행할 수 있는 소프트웨어 능력

4.1 소프트웨어 품질 개요

품질

- 제품이나 서비스가 가지는 수월성에 관한 특성
 - 생산자 입장의 품질: 명시된 요구사항을 만족시키는 정도
 - 고객 입장의 품질: 고객의 기대나 사용 목적에 부합하는 정도
- 소프트웨어 품질
 - 요구사항의 만족
 - 개발 표준을 준수하여 소프트웨어를 개발
 - 일반적인 묵시적 품질 요구사항을 가지는 것

4.2 소프트웨어 품질의 분류

품질 관점

- 사용자: 제품의 신뢰성, 효율성, 사용 용이성 등
- 개발자: 검증가능성, 유지보수성, 이식성 등
- 프로젝트 관리자: 프로세스의 생산성과 제어 용이성

외부 특성과 내부 특성

- 외부 특성
 - 사용자 관점의 품질 특성
 - 신뢰성이나 사용성
 - 실행되는 코드의 행위를 측정
- 내부 특성
 - 개발자 관점의 품질 특성
 - 개발자가 외부 특성을 개선할 때 도움을 줌
 - 개발 문서나 코드에 대해 정적으로 측정

제품 특성과 프로세스 특성

- 프로세스 품질 관리와 개선을 위한 노력은 소프트웨어 제품의 품질에 영향을 줌
- 고객과 개발자 관점에서 제품의 의미가 다름

4.3 소프트웨어 제품의 품질 관련 표준

제품 품질의 표준 - ISO/IEC 9126

- 소프트웨어 제품의 품질 특성들과 메트릭을 정의한 표준

- 사용자 관점에 따라 제품의 품질 특성을 6가지로 분류
 - 기능성: 명시적 또는 묵시적 사용자 요구를 만족시키기 위한 기능의 제공 능력
 - 신뢰성: 명시된 기간과 조건에서 성능 수준을 유지하는 소프트웨어 능력
 - 사용성: 시스템을 사용하는데 드는 노력과 사용자의 평가를 나타내는 특성
 - 효율성: 소프트웨어의 성능 수준과 필요한 자원 사이의 관계를 나타내는 특성
 - 유지보수성: 소프트웨어 제품의 수정에 드는 노력을 나타내는 특성
 - 이식성: 한 환경에서 다른 환경으로 이식될 수 있는 소프트웨어 제품의 능력
- 6가지 주 품질 특성들은 각각 여러 부특성들로 구성됨

<표 4-1> ISO/IEC 9126에서 부특성의 설명

품질 특성	부특성	설명
기능성	적합성	소프트웨어가 요구된 작업을 수행하는가?
	정확성	결과가 기대와 같은가?
	상호운용성	시스템이 다른 시스템과 연동되는가?
	보안성	시스템이 불법적 접근을 방지하는가?
	준수성	시스템이 표준을 준수하는가?
신뢰성	성숙성	시간이 지나면서 소프트웨어 결함이 제거되는가?
	결함 내구성	소프트웨어가 오류를 처리할 능력을 가지는가?
	복구성	고장으로 인한 작업과 데이터 손실을 복구할 수 있는가?
사용성	이해성	시스템의 개념과 응용 분야를 쉽게 이해하는가?
	학습성	사용자가 시스템 사용법을 쉽게 배울 수 있는가?
	운용성	사용자가 큰 노력없이 시스템을 사용할 수 있는가?
	매력성	인터페이스가 보기 좋은가?
효율성	시간 효율성	시스템이 빠르게 응답하는가?
	자원 효율성	시스템이 효율적으로 자원을 사용하는가?
유지보수성	분석성	결함에 관해 빠른 진단이 가능한가?
	변경성	쉽게 소프트웨어를 수정할 수 있는가?
	안정성	변경 후에도 소프트웨어가 계속 기능을 할 수 있는가?
	시험성	쉽게 테스트할 수 있는가?
이식성	적응성	다른 환경으로 쉽게 옮겨질 수 있는가?
	설치성	소프트웨어를 쉽게 설치할 수 있는가?
	공존성	소프트웨어가 이식을 위한 표준을 준수하는가?
	대체성	다른 소프트웨어를 쉽게 대체할 수 있는가?

외부 메트릭(ISO/IEC 9126-2)과 내부 메트릭(ISO/IEC 9126-3)

- 외부 메트릭: 실행 중인 소프트웨어의 행위를 측정
- 내부 메트릭: 소프트웨어 산출물 자체를 측정
- 예) 신뢰성을 높이려면 검토 과정에서 많은 결함을 발견하여 사용 중에 고장으로 연결되지 않도록

외부 메트릭으로 평균 고장 간격(MTBF), 내부 메트릭으로 검토 중 발견된 결함의 밀도를 사용

사용 품질(ISO/IEC 9126-4)

- 6가지 주 품질특성을 조합하여 사용자가 느끼는 제품의 실제 효과를 의미
- 실제 사용 환경에서 효율성, 생산성, 안전성 및 만족성을 가지고 명시된 목표를 달성하는

소프트웨어 제품의 능력

- 프로세스 품질 -> 내부 품질 -> 외부 품질 -> 사용 품질

제품 품질의 표준 - 맥콜의 품질 특성

- 11개의 품질 요인: 사용자 관점의 외부 품질 요인

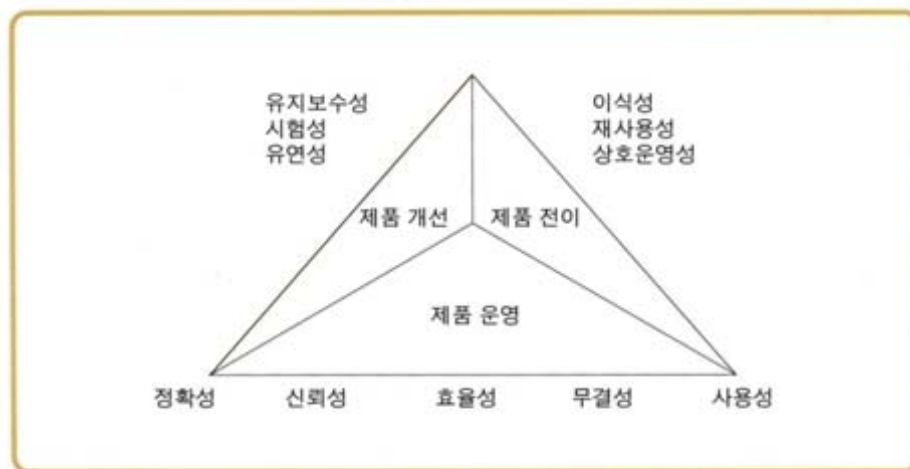
제품 운영: 정확성, 신뢰성, 효율성, 무결성, 사용성

제품 개선: 유지보수성, 시험성, 유연성

제품 전이: 이식성, 재사용성, 상호운영성

- 23개의 품질 기준: 개발자 관점의 내부 품질 기준

- 매트릭: 품질 기준을 정량적으로 표현한 측정 방법



[그림 4-5] 맥콜 모델에서 품질 요인들의 분류

4.4 프로세스 품질 표준

소프트웨어 품질 관련 표준

- 제품 품질의 평가

- 프로세스 품질의 평가

소프트웨어 개발 조직의 능력을 평가하고 개발 공정을 개선하기 위함

예) SPICE, CMMI

- 품질 시스템 구축

품질 경영체제를 도입하고 프로세스 품질을 인증

예) ISO 9001

ISO 9000 시리즈

- 품질 경영을 위한 기본 요소를 규정하고 실천을 위한 활동지침을 제시

- ISO 9000: 품질 관리 시스템의 기본과 관련 용어들을 설명

- ISO 9001

품질 관리 시스템의 요건을 자세히 설명

ISO 9001 인증은 정형화된 업무 프로세스가 적용되어 품질 문화가 정착되었다는 의미

- 특징

프로세스 평가 보다 품질 관리 자체에 중점을 둠
평가를 등급화하지 않고 ISO 9001의 인증 여부만을 결정

CMMI(Capability Maturity Model Integration)

- 조직의 개발 프로세스 성숙도를 평가하는 모델
- 다양한 기업에 프로세스 개선을 위한 프레임워크를 제공
- ISO 9001과 ISO 15504(SPICE)의 관심사를 포함한 통합 모델
- CMMI 모델의 구조
 - 단계적 모델: 조직 전체의 프로세스 성숙도를 1~5 수준으로 평가
 - 연속적 모델: 20여 개의 프로세스 영역에 대해 각각 0~5 사이의 점수를 부여

CMMI 모델의 구성 요소

- 평가 영역을 여러 프로세스 영역들로 구분
- 각 프로세스 영역은 달성해야 하는 목표를 가짐
- 각 목표는 그것을 달성하기 위한 기술들로 구성됨

프로세스 영역, 목표, 기술

- 프로세스 영역(process area)
 - '개발을 위한 CMMI'는 22개의 프로세스 영역으로 구성됨
 - 성숙도 수준으로 분류되며 4개의 그룹으로도 분류될 수 있음
- 목표(goal)
 - 조직이 달성해야 하는 바람직한 상태를 표현
 - 성숙도 수준별로 공통 목표가 존재
 - 각 프로세스 영역별로 특정 목표가 존재
- 기술(practices)
 - 목표를 달성하기 위한 구체적 표현

CMMI 평가

- 연속적 모델: 각 프로세스 영역별로 개별 평가하여 0~5 사이의 점수를 매김
 - 실행되지 않음, 실행됨, 관리됨, 정의됨, 양적으로 관리됨, 최적화됨
- 단계적 모델: 조직의 전체 프로세스 능력을 하나의 등급으로 평가
 - 수준 1: 초기 상태
 - 수준 2: 관리됨. 기본적 프로세스가 구축되어 프로젝트가 관리됨
 - 수준 3: 정의됨. 프로세스 표준화, 조직의 특성에 맞게 재단될 수 있음
 - 수준 4: 양적으로 관리됨. 프로젝트 활동이 정량적으로 관리되고 통제됨
 - 수준 5: 최적화됨. 지속적인 개선 활동이 정착화됨

<표 4-2> CMMI - 수준별 키 프로세스 영역

성숙도 수준	초점	키 프로세스 영역	결과
수준 5 (Optimizing)	지속적 프로세스 개선	조직 혁신 및 이행(OID: Organization Innovation & Deployment) 분석과 해결(CAR: Casual Analysis & Revolution)	생산성 과 품 질
수준 4 (Quantitatively Managed)	정량적 관리(프로세스의 측정과 제어가 가능)	조직 프로세스 성능(OPP: Organizational Process Performance) 정량적인 프로젝트 관리(QPM: Quantitative Project Management)	
수준 3 (Defined)	프로세스 표준화(조직의 표준에 맞춰 프로세스를 재단, 능동적으로 프로세스를 변경)	요구사항 개발 (RD: Requirement Development) 기술적 해결 (TS: Technical Solution) 제품 통합 (PI: Product Integration) 검증(Verification) 확인(Validation) 조직 프로세스 중점(OPF: Organization Process Focus) 조직 프로세스 정의(OPD: Organization Process Definition) 조직 훈련(OT: Organization Training) 통합된 프로젝트 관리(IPM: Integrated Project Management) 위험관리 (RM: Risk Management) 의사결정 분석 및 해결(DAR: Decision Analysis & Revolution)	
수준 2 (Managed)	기본 프로젝트 관리(프로젝트 중심의 프로세스)	요구사항 관리(RM: Requirement Management) 프로젝트 계획(PP: Project Planning) 프로젝트 감시 및 제어(PMC: Project Monitoring & Control) 공급자 합의 관리(SA: Supplier Agreement Management) 측정과 분석(MA: Measurement & Analysis) 프로세스와 제품 품질 보증(PPQA: Process & Product Quality Assurance) 형상 관리(CM: Configuration Management)	
수준 1 (Initial)			

4.5 소프트웨어 품질 보증

품질 보증(QA)

- 사용자가 제품을 사용했을 때 기대했던 기능과 성능을 발휘할 수 있음을 고객에게 보증하는 것

“제품이 공식적으로 인정된 기술적 요건을 충족하고 있음을 확신하는 데 필요한 계획되고 체계적인 행동 패턴”

소프트웨어 품질 보증(SQA)

- 개발자나 품질 보증 팀이 수행하는 품질 관리 활동의 체계로 개발 과정 전체에 걸쳐 수행

제품과 개발 프로세스의 표준을 정하여 시스템에 요구되는 품질 속성을 정의
품질 제어는 표준과 비교하고 필요한 활동을 취하는 것

품질 보증 계획

- SQA 작업 수행을 위한 로드맵으로 품질 보증 절차를 정하는 일
적정 시점에 산출물들이 만들어지는지 확인

검토 방법

- 정적 테스트라고 하며 적은 노력으로 빠르게 결함을 발견
- 목적
 - 명세서와 일치하는지 확인
 - 계획, 표준 및 지침에 맞게 개발되었는지 확인
 - 변경이 적절히 구현되었는지 확인

공식 기술 검토(FTR)

- 제품의 완전성, 정확성, 일관성, 기술적 타당성, 효율성 및 표준과 지침의 준수 여부를 조사하는 공식적 회의

인스펙션

- 공식 기술 검토 전이나 테스트를 수행하기 전에 설계 문서나 코드를 작성자가 아닌 동료나 전문가 팀이 검토하는 것
- 회의 전에 필요한 자료를 참가자에게 나누어 사전 검토 작업을 하고 회의 참석
- 코드 인스펙션의 경우 범하기 쉬운 흔한 오류의 발견에 초점을 둠

코드 워크스루

- 알고리즘이나 코드상의 경로를 따라가면서 결함을 찾을 목적의 비공식적 검토 방법
- 작성자 본인이 다른 구성원들과 질의 응답함

4.7 소프트웨어 형상 관리

소프트웨어 형상 관리

- 소프트웨어 요소들의 관련성을 파악하고 그것들의 버전들을 일관성있게 관리함으로써 소프트웨어 시스템의 진화를 통제하는 일
- 소프트웨어 시스템에는 여러 형상이 존재할 수 있음
 - 상이한 컴퓨터와 운영체제, 특별한 기능을 수용하기 위한 여러 릴리스가 존재
 - 소프트웨어 구성 요소(요구사항, 설계 문서, 코드, 테스트 케이스 등)들의 관계를 파악하여 묶어 놓은 것을 형상이라 함
- 베이스 라인이 설정되면 이후 발생하는 변화는 통제를 받게 됨
 - 문서나 코드가 작성 완료된 후에는 변경 효과에 대한 공식적 검토를 거쳐 변경될 수 있음

4.8 신뢰도

신뢰도

- 일정 시간 동안 의도된 기능을 고장 없이 실행할 수 있는 프로그램 능력
- 내재된 결함이 있더라도 고장으로 연결되지 않거나 결과가 심각하지 않다면 신뢰성이 있을 수 있음

결함은 시스템 고장을 야기시킬 수 있는 잠재적 요인

고장은 결함이 노출되어 오동작을 일으키는 것

- 소프트웨어 고장은 일시적일 수 있으며 특정 입력에 대해서만 잘못된 결과를 내기도 함

신뢰도 메트릭

- MTTF(Mean Time To Failures)

가동되어 고장이 발생할 때까지의 평균 시간(평균 수명)

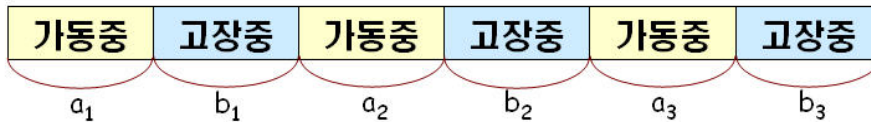
고장의 복구를 고려하지 않음

예) 워드 프로세서의 MTTF가 500이라면 500시간 단위마다 한 번의 고장이 발생

- MTBF(Mean Time Between Failures)

고장과 고장 사이의 평균 간격

$$MTTF + MTTR = (a_1 + a_2 + a_3)/3 + (b_1 + b_2 + b_3)/3$$



- AVAIL(가용성)

시스템이 가동될 확률,

논스톱 시스템의 신뢰도 측정에 사용

고장이 일어나더라도 빠른 복구가 가능하면 가용성이 높아질 수 있음

$$MTTF/(MTTF+MTTR) = (a_1 + a_2 + a_3)/(a_1 + a_2 + a_3 + b_1 + b_2 + b_3)$$

- ROCOF(Rate of Occurrences of Failures)

고장 발생 비율

규칙적인 서비스 요청이 들어오는 시스템의 신뢰도 측정

예) 예약 시스템의 ROCOF가 0.002라면 1000회의 요청 중 2회의 오류가 발생

메뉴	연습문제
-----------	-------------

1. 소프트웨어 품질 요인 중 일반 사용자가 중요시 하는 것은?

- ① 사용 용이성
- ② 유지보수성
- ③ 프로세스의 생산성
- ④ 적은 개발 비용

<정답> ①

<해설> 프로젝트 발주자, 프로젝트 관리자, 개발자, 유지보수자 및 사용자에게 따라 중요시하는 품질 요인이 다르다. 일반적으로 사용자는 기능의 정확성이나 사용의 용이성을 중시한다.

2. 소프트웨어 제품의 품질 표준인 ISO/IEC 9126에서 정의한 여섯 가지 품질 특성에 해당하지 않는 것은?

- ① 신뢰성
- ② 유지보수성
- ③ 기능성
- ④ 재사용성

<정답> ④

<해설> 기능성: 명시적 또는 묵시적 사용자 요구를 만족시키기 위한 기능의 제공 능력
신뢰성: 명시된 기간과 조건에서 성능 수준을 유지하는 소프트웨어 능력
사용성: 시스템을 사용하는데 드는 노력과 사용자의 평가를 나타내는 특성
효율성: 소프트웨어의 성능 수준과 필요한 자원 사이의 관계를 나타내는 특성
유지보수성: 소프트웨어 제품의 수정에 드는 노력을 나타내는 특성
이식성: 한 환경에서 다른 환경으로 이식될 수 있는 소프트웨어 제품의 능력

3. CMMI에서 조직의 프로세스 성숙도 단계를 낮은 수준부터 나열하면?

ㄱ.defined ㄴ.managed ㄷ.optimizing ㄹ.quantitatively managed

- ① ㄱ-ㄴ-ㄷ-ㄹ
- ② ㄴ-ㄱ-ㄷ-ㄹ
- ③ ㄴ-ㄱ-ㄹ-ㄷ
- ④ ㄱ-ㄴ-ㄹ-ㄷ

<정답> ③

<해설> 초기 상태-관리됨-정의됨-양적으로 관리됨-최적화됨

4. 공식 기술 검토(FTR)에 관한 설명이 아닌 것은?

- ① 품질 보증 활동이다
- ② 소프트웨어가 요구 명세와 일치하는지 검토한다
- ③ 소프트웨어가 미리 정한 기준에 따라 표현되었는지 확인한다.
- ④ 산출물이 아닌 제작자를 검토한다.

<정답> ④

<해설> 공식 기술 검토는 요구사항 명세서, 설계 문서, 프로젝트 계획서 등을 검토하여 산출물의 오류를 발견하기 위한 활동이다.

5. 신뢰도 메트릭에 관한 설명으로 잘못된 것은?
- ① 평균 고장 간격은 MTTF와 MTTR의 합이다
 - ② 평균 복구 시간이 빠르다고 가용성이 좋아지지 않는다
 - ③ MTTF는 고장 복구 시간을 고려하지 않는다
 - ④ ROCOF는 규칙적인 서비스 요청이 존재하는 시스템의 신뢰도 측정에 적합하다

<정답> ②

<해설> 자주 고장이 일어나더라도 복구 시간이 짧으면 가용성이 높아질 수 있다.

메 뉴	정리하기
----------------	-------------

1. 소프트웨어 품질의 의미는 무엇인가?

- 개발 표준을 준수하여 개발된 소프트웨어가 기능적/비기능적 요구사항을 만족하고 목시적 좋은 특성들을 가지는 것

2. 소프트웨어 제품의 외부 품질 특성과 내부 품질 특성의 차이는 무엇인가?

- 외부 품질 특성은 신뢰성이나 사용성과 같이 사용자 관점에서 본 품질 특성이며 내부 품질 특성은 개발자 관점의 특성으로 산출물 자체의 품질임

3. CMMI란 무엇인가?

- 개발 조직이 가지고 있는 개발 프로세스의 성숙도를 평가하는 모델로 여러 CMM 모델들을 통합한 것

4. ISO 9001과 CMMI는 인증 방법에서 어떤 차이가 있는가?

- ISO 9001은 평가 결과를 등급화하지 않고 인증 여부만을 결정함

5. 확인(verification)과 검증(validation)을 설명하라?

- V&V는 소프트웨어가 요구사항에 맞게 작성되었는지(확인) 그리고 고객의 실제

요구를 충족하는지(검증) 검사하는 생명주기 활동 전체를 말함

6. 신뢰도를 측정하기 위한 메트릭의 종류를 나열하라

□ MTTF, MTBF, AVAIL, ROCOF 등