

10주 1강

# 소프트웨어 품질



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는바, 무단  
전재, 배포, 전송, 대여 등을 금합니다.

\*사용서체 : 나눔글꼴

# 이번 주차에는...

## 소프트웨어 품질

- 품질의 이해
- 품질 요소와 품질 평가 모델
- 제품 품질 특성 평가 모델
- 프로세스 품질 특성 평가 모델
- 대표적인 프로세스 능력 평가 모델
- 품질 관리

# 1. 품질과 소프트웨어 품질(1)

## ■ 품질

- 물건이 얼마나 좋은지, 나쁜지를 나타내는 정도

## ■ 하드웨어 품질

- 사용자의 요구 사항을 규격서에 서술하고, 이 규격서대로 만들면 큰 문제가 없다.
- 만든 제품이 규격서대로 작동하는지 테스트하는 것도 비교적 용이
- 개발된 제품이 규격서대로 만들어지면 사용자가 만족할 만한, 품질이 좋은 하드웨어



## 2. 품질과 소프트웨어 품질(2)

### ■ 소프트웨어

- HW(규격서) vs SW(요구 분석 명세서)
- 하드웨어의 규격서보다 소프트웨어의 요구 분석 명세서를 작성하는 것이 훨씬 어렵다.

### ■ 소프트웨어 품질 정의

- ‘사용자의 요구와 부합되는 정도’

- US DoD: 개발된 소프트웨어가 사용자의 요구 사항을 만족할 수 있는 능력
- IEEE: - 소프트웨어가 필요한 속성을 보유하고 있는 정도
  - 사용자의 기대 수준을 만족할 수 있는 정도를 결정하는 소프트웨어의 특성

- 개발자 관점에서의 좋은 소프트웨어

- 결함 없는 프로그램
- 요구 분석 명세서 대로 만든 소프트웨어

### 3. 관점에 따른 품질(1)



그림 9-2 같은 음식, 다른 느낌



그림 9-3 산악용 MTB 자전거



그림 9-4 로드용 사이클 자전거



그림 9-5 주부용 바구니 자전거

## 4. 관점에 따른 품질(2)

- 프로젝트 관리자 관점의 좋은 소프트웨어
  - 추가 부담(기간, 비용)이 발생하지 않는 소프트웨어
- 개발자 관점의 좋은 소프트웨어
  - 개발하기 쉽고 사용 중 내용 추가 및 코드 수정이 쉽고 편리하게 변경 가능한 소프트웨어  
→ 코딩 표준에 맞게 개발된 프로그램
- 유지보수자 관점의 좋은 소프트웨어
  - 작성된 코드가 코딩 규칙 및 표준을 따르고 주석문이 많이 포함된 소프트웨어  
→ 가독성이 높고 쉽게 이해할 수 있게 개발된 소프트웨어

## 5. 관점에 따른 품질(3)

- 구매 담당자 관점의 좋은 소프트웨어
  - 값이 싼 소프트웨어
- 사용자 관점의 좋은 소프트웨어
  - 배우기 쉽고, 사용하기 편리하며, 다양한 기능을 제공하고, 응답 시간이 빠른 소프트웨어  
→ 첫 사용자나 숙련된 사용자 모두 만족하는 소프트웨어

## 6. 품질 목표(1)

### ■ 품질 좋은 소프트웨어가 되려면

- 처음부터 품질을 고려한 계획을 세운다.
- 품질 요구 사항에 대한 명세서가 작성되어야 한다.
- 단계별로 생산되는 산출물도 검사 항목에 따라 철저히 점검해야 한다.
- 품질을 보증할 수 있는 프로세스를 마련해, 품질을 체크할 수 있는 과정을 명확히 한다.



## 7. 품질 목표(2)

표 9-1 소프트웨어 품질 목표

품질 목표	정의
정확성 <sup>correctness</sup>	<ul style="list-style-type: none"><li>• 사용자가 요구하는 기능을 충족시키는 정도</li><li>• 요구 분석 명세서와 일치하는 정도</li></ul>
신뢰성 <sup>reliability</sup>	사용자가 요구한 기능을 정확하고 일관되게 원하는 정밀도로 수행할 수 있는 정도
효율성 <sup>efficiency</sup>	사용자가 요구하는 기능을 수행하는 데, 최소의 시간과 기억 용량을 사용하여 요구되는 기능을 수행할 수 있는 정도
무결성 <sup>integrity</sup>	허가받지 않은 사용자가 데이터 접근을 통해 변경을 시도했을 때 보호할 수 있는 정도
사용성 <sup>usability</sup>	소프트웨어를 사용하는 데 있어서 혼란스러워하거나 사용하는 순간에 고민하지 않도록 편리한 기능을 제공하는 정도
유지보수 용이성 <sup>maintainability</sup>	프로그램 내에 존재하는 오류를 찾아 수정하고 패치할 때 쉽게 변경할 수 있는 정도
테스트 용이성 <sup>testability</sup>	쉽고 철저하게 테스트할 수 있는 정도
유연성 <sup>flexibility</sup>	운영 환경의 변화에 따라 새로운 기능을 추가하거나 다른 환경에 적용할 수 있게 운용되는 프로그램을 쉽게 수정할 수 있는 정도
이식성 <sup>portability</sup>	다른 하드웨어 환경에서도 운용 가능하도록 쉽게 수정하여 이식할 수 있는 정도
재사용성 <sup>reusability</sup>	시스템의 일부나 전체를 다른 애플리케이션에서도 쉽게 사용할 수 있는 정도
상호운용성 <sup>interoperability</sup>	한 소프트웨어를 다른 소프트웨어와 쉽게 연계하거나 결합하여 정보를 교환할 수 있는 정도

## 8. 품질 특성

표 9-2 품질 특성

학자	품질 요소	
McCall <sup>맥콜</sup>	제품 수정	유지보수 용이성, 유연성, 테스트(기능)성
	제품 변환	상호운용성, 재사용성, 이식성
	제품 운영	정확성, 효율성, 무결성, 신뢰성, 사용성
Garvin <sup>가빈</sup>	• 성능 <sup>performance quality</sup>	• 특성 품질 <sup>feature quality</sup>
	• 신뢰성 <sup>reliability</sup>	• 일치성 <sup>conformance</sup>
	• 내구성 <sup>durability</sup>	• 서비스 <sup>serviceability</sup>
	• 미적 <sup>aesthetics</sup>	• 지각 <sup>perception</sup>
Boehm <sup>보헴</sup>	이식성	장치 독립성, 자체 포함성
	신뢰성	자체 포함성, 정확성, 안전성, 강건성/무결성, 일관성
	효율성	책임성, 장치 효율성, 접근성
	인간공학성	강건성/무결성, 접근성, 통신성
	테스트성	책임성, 접근성, 통신성, 자체 기술성, 구조성
	이해성	일관성, 자체 기술성, 구조성, 간결성, 가독성
	수정(변경)성	구조성, 증가성

## 9. McCall의 품질 요소

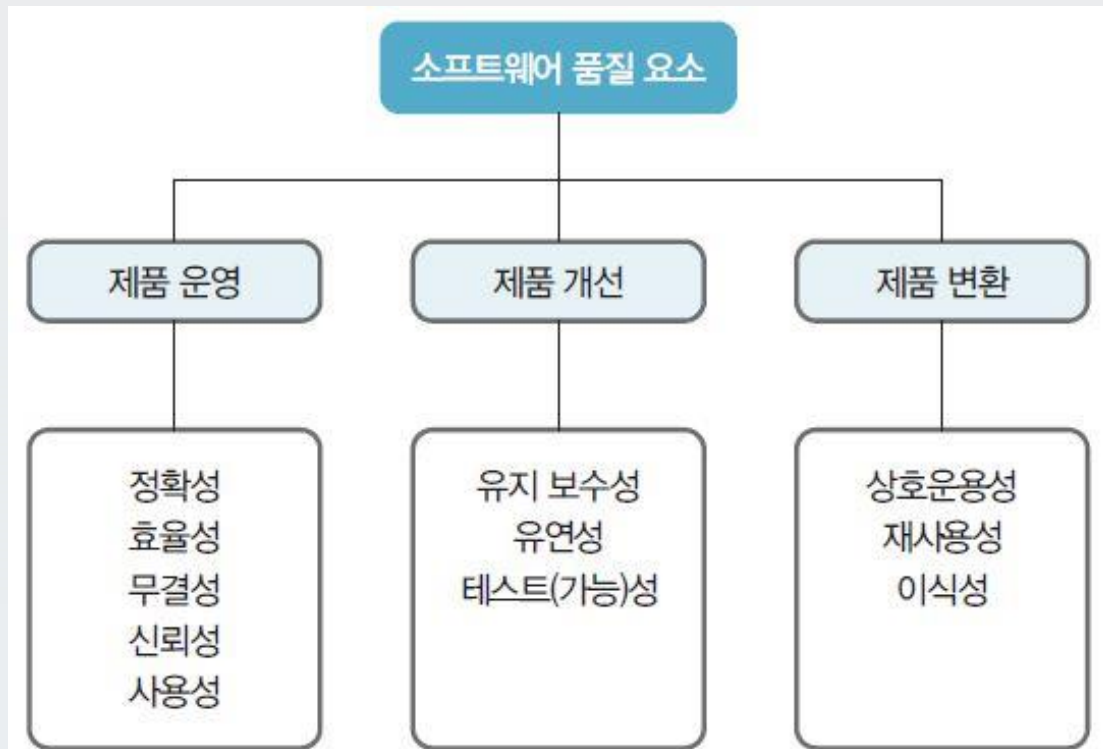


그림 9-6 McCall의 품질 모델

# 10. 제품 운영

- 제품 운영product operation

- 개발된 SW가 고객이 사용해도 될 만큼 적합한지의 여부를 판단할 수 있는 품질 요소

- 정확성correctness

- 개발된 소프트웨어가 사용자의 기능적 요구 사항을 담은 프로그램 명세서와 얼마나 일치하는지를 나타낸다.

- 효율성efficiency

- 사용자가 요구하는 기능을 수행하는 데CPU와 메모리 같은 자원을 얼마나 사용하는가와 관련된 특성이다

- 무결성integrity

- 허가 받지 않은 사용자가 데이터 접근을 통해 변경을 시도했을 때 얼마나 보호할 수 있는지를 나타낸다.

# 11. 제품 개선

- **제품 개선**product revision
  - 소프트웨어를 변경하기 쉽고 편하게 만든 정도를 나타내는 품질 요소
- **유지보수성**maintainability
  - 사용 중인 소프트웨어를 얼마나 쉽게 변경할 수 있는지, 또 변경한 후에도 문제없이 안정적으로 운영되는지를 나타낸다.
- **유연성**flexibility
  - 운영 환경의 변화에 따라 새로운 기능을 쉽게 추가할 수 있는지, 다른 환경에 적용할 수 있도록 운용되는 프로그램을 얼마나 쉽게 수정할 수 있는지를 나타낸다.
- **테스트 용이성**testability
  - 사용자가 요구하는 기능을 만족할 만큼 잘 수행하고 있는지에 대해 얼마나 쉽고, 철저하게
  - 테스트할 수 있는지를 나타낸다.

## 12. 제품 변환

- **제품 변환**product transition
  - 개발된 소프트웨어의 활용도를 높이려 할 때, 쉽게 할 수 있는 정도를 나타내는 품질 요소
- **상호운용성**interoperability
  - 다른 소프트웨어와 얼마나 쉽게 연계 또는 결합하여 정보를 교환할 수 있는지를 나타낸다.
- **재사용성**reusability
  - 시스템의 일부나 전체를 다른 애플리케이션에서도 얼마나 쉽게 사용할 수 있는지를 나타낸다.
- **이식성**portability
  - 하드웨어 또는 운영체제와 같은 환경에서 또 다른 환경으로 옮겨도 환경 변화에 무리 없이 잘 작동할 수 있도록 프로그램을 수정하여 이식하는 것이 얼마나 쉬운가를 나타낸다

# 13. 품질 평가 표준 모델(1)

- 제품 품질 특성 평가
  - 완성된 제품에 대한 평가
  - ISO/IEC 9126에서 제시하는 모델이 소프트웨어 품질에 대한 표준적인 모델
- 프로세스 품질 특성 평가
  - 소프트웨어 제품의 개발 프로세스를 평가
  - 소프트웨어 개발 과정의 각 단계마다 평가

# 14. 품질 평가 표준 모델(2)

표 9-3 소프트웨어 품질 평가와 품질 평가 모델

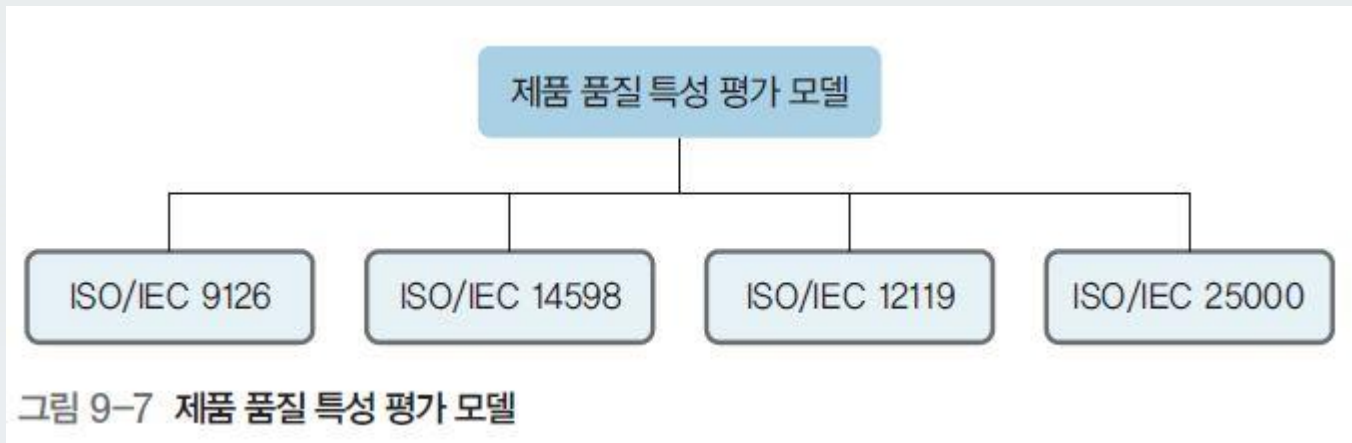
품질 평가	품질 평가 모델	세부 내용
제품 품질 특성 평가	ISO/IEC 9126	ISO/IEC 9126-1(품질 모델) ISO/IEC 9126-2(외부 품질) ISO/IEC 9126-3(내부 품질) ISO/IEC 9126-4(사용 품질) 가능성, 신뢰성, 사용성, 효율성, 유지보수 용이성, 이식성
	ISO/IEC 14598	14598-1(일반적 개요) 14598-2(계획과 관리) 14598-3(개발자용 프로세스) 14598-4(구매자용 프로세스) 14598-5(평가자용 프로세스) 14598-6(평가 모델)
	ISO/IEC 12119	소프트웨어 패키지 • 제품 설명서 • 사용자 문서 • 프로그램과 데이터
	ISO/IEC 25000	2500X(SQuaRE 개요) 2501X(품질 모델) 2502X(품질 메트릭) 2503X(품질 요구 사항) 2504X(품질 평가)
프로세스 품질 특성 평가	ISO/IEC 9000(품질 경영과 품질 보증)	9001(설계, 개발, 서비스) 9002(생산과 설치) 9003(최종 검사 및 시험) 9004(지침 표준)
	ISO/IEC 12207	5. 기본 생명주기 프로세스 6. 지원 생명주기 프로세스 7. 조직 생명주기 프로세스
	CMMI	• 단계적 표현 • 연속적 표현
	SPICE	(ISO 15504)



# 15. 제품 품질 특성 평가

## ■ 제품 품질 특성 평가 모델

- 개발된 최종 산출물인 소프트웨어 제품이 사용자의 의도대로 기능을 수행하는지 평가



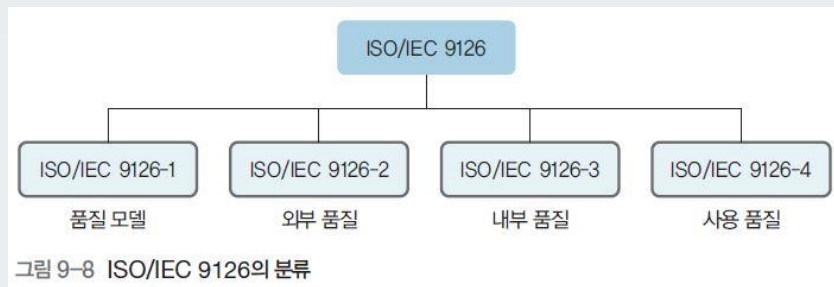
# 16. ISO/IEC9126 모델(1)

## ■ 품질 특성과 용도

표 9-4 ISO/IEC 9126 모델의 품질 특성과 용도

품질 특성	<ul style="list-style-type: none"><li>• 모든 소프트웨어 품질을 포괄할 수 있도록 매우 일반화된 개념으로 정의한다.</li><li>• 고객 관점에서 소프트웨어에 관한 품질 특성, 부특성을 정의한다.</li><li>• 품질을 계량적으로 측정 및 평가하기 위한 품질 요소, 특성, 메트릭스를 정의한다.</li><li>• 내부 척도와 외부 척도를 정의한다.</li></ul>
용도	<ul style="list-style-type: none"><li>• 소프트웨어 품질 특성과 척도에 관한 지침으로 사용한다.</li><li>• 개발되는 소프트웨어 품질을 측정하는 기준으로 사용한다.</li><li>• 프로젝트 시작 시 소프트웨어의 제품 품질에 대한 목표 요구 사항을 기술하는 데 사용한다.</li><li>• 개발 공정에서 품질을 객관적으로 정량화하는 국제 표준을 제시한다.</li></ul>

## ■ 분류



# 17. ISO/IEC9126 모델(2)

- ISO/IEC 9126-1(품질 모델)
  - 6가지 품질 특성과 소프트웨어 제품의 품질 평가를 위한 프레임워크를 정의
- ISO/IEC 9126-2(외부 품질)
  - 개발자를 위한 표준
  - 개발자/평가자/구매자가 품질 특성에 대해 사용할 수 있는 external metrics 제공
  - 사용자/평가자/구매자/개발자들이 테스트 단계와 운영 중에 SW를 평가하고 보고서를 작성할 수 있도록 도와준다.
  - 완성된 소프트웨어의 성능, 오류 발생, 사용 용이성 등이 여기에 해당

## 18. ISO/IEC9126 모델(3)

### ■ ISO/IEC 9126-3(내부 품질)

- 구매자를 위한 표준
- 개발자/평가자/구매자가 제품 품질을 평가할 수 있도록 도와주며, 제품 완성 전 미리 품질의 문제점들을 지적해준다.
- 품질 특성에 대하여 사용할 수 있는 내부 메트릭스(internal metrics)를 제공

### ■ ISO/IEC 9126-4(사용 품질)

- 사용자를 위한 표준으로 사용 품질(quality in use)을 정의
- 제품이 특정 환경에서 사용될 때 사용자의 작업 효율성, 생산성, 안정성, 만족도 등 사용자의 요구를 충족시키는 정도
- 제품이 고객에게 인도된 후 소프트웨어 자체의 특성보다 사용해본 결과를 토대로 사용자가 측정

# 19. ISO/IEC9126의 품질 특성 및 하위 특성(1)

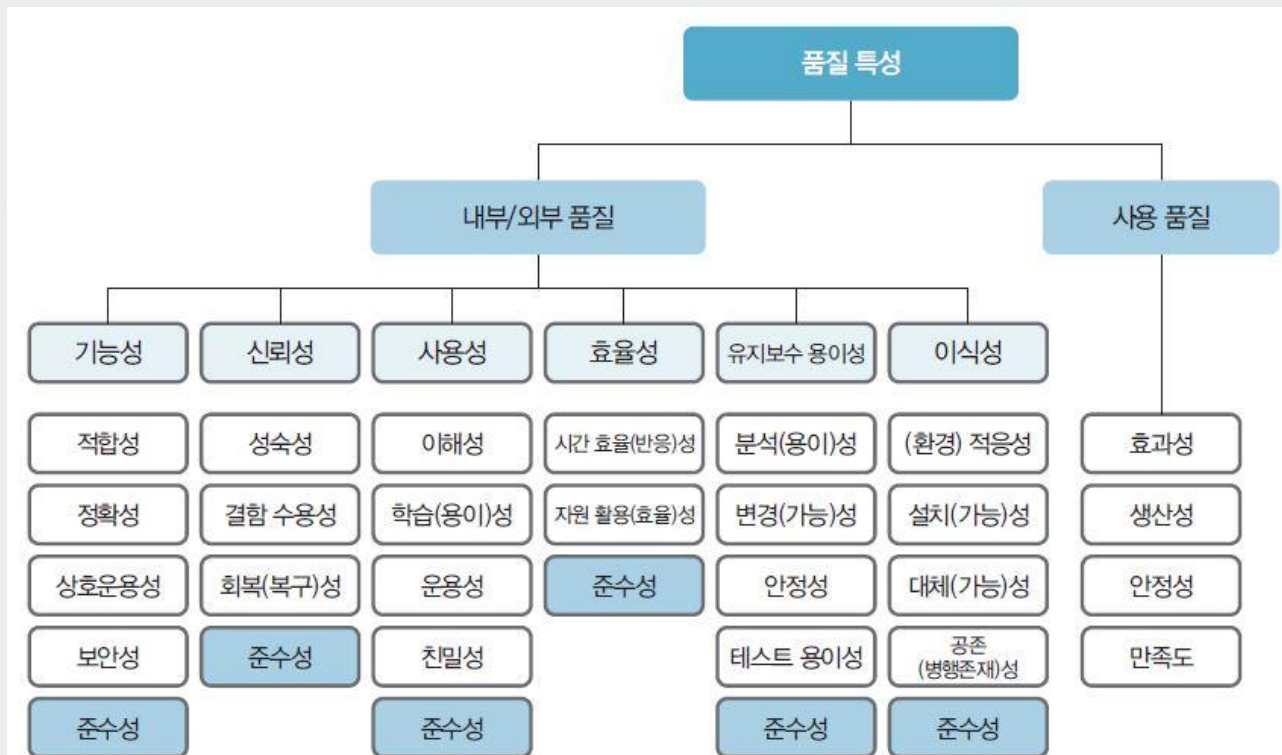


그림 9-9 ISO 9126 모델의 품질 특성

## 20. ISO/IEC9126의 품질 특성 및 하위 특성(2)

### ■ 기능성functionality

- 개발된 소프트웨어가 특정 조건에서 사용될 때 개발 전에 의도했던 대로 정확하게 사용자의 요구를 만족하는 기능을 제공하는지 여부를 나타낸다.

표 9-5 기능성의 하위 특성

특성	설명
적합성suitability	규정된 작업과 사용자의 사용 목적을 위한 적절한 기능들을 제공하는 소프트웨어 제품의 능력을 말한다.
정확성accuracy	사용자가 요구하는 정밀도를 유지하거나 허용 범위 내의 결과 값을 제공할 수 있는 소프트웨어 제품의 능력을 말한다.
상호운용성interoperability	특정한 시스템과 상호작용(데이터 교환 등)할 수 있는 소프트웨어 제품의 능력을 말한다.
보안성security	권한이 있는 사람(또는 시스템)과 권한이 없는 사람(또는 시스템)을 구별하여 정보에 대한 접근을 철저히 통제함으로써 정보를 보호하는 소프트웨어 제품의 능력을 말한다.
준수성compliance	기능성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.

## 21. ISO/IEC9126의 품질 특성 및 하위 특성(3)

- 신뢰성reliability
  - 소프트웨어를 믿고 사용할 수 있는지 여부를 나타낸다.

표 9-6 신뢰성의 하위 특성

특성	설명
성숙성 <sup>maturity</sup>	소프트웨어에 오류가 발생하여 문제가 생겼을 경우, 그 상황에서도 문제를 피해 소프트웨어가 안전한지 여부를 나타낸다.
결함 수용성 <sup>fault tolerance</sup>	소프트웨어 일부에서 고장 <sup>fault</sup> 이 발생해도 요구되는 기능을 유지할 수 있는지를 나타낸다. 따라서 예외적으로 발생할 수 있는 상황에 대해서도 처리할 수 있는 능력을 갖추어야 한다.
회복(복구)성 <sup>recoverability</sup>	소프트웨어 오류로 인해 문제가 발생했을 때 프로그램을 수정하게 된다. 수정한 후 복구되었을 때 이전 상황과 얼마나 동일하게 돌아갈 수 있는지를 나타낸다.
준수성 <sup>compliance</sup>	신뢰성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.

## 22. ISO/IEC9126의 품질 특성 및 하위 특성(4)

- 사용성usability
  - 편리한 기능을 제공하는 정도를 나타낸다

표 9-7 사용성의 하위 특성

특성	설명
이해성 <small>understandability</small>	소프트웨어를 사용할 때 사용 방법을 얼마나 쉽게 이해할 수 있게 만들었는지에 관한 속성이다. 인터페이스에 일관성이 있어 사용자가 쉽게 접근할 수 있고, 입출력 값과 관련된 예시를 볼 수 있게 만들어서 사용하는 데 고민을 적게 해주는 것이다.
학습(용이)성 <small>learnability</small>	사용자가 어떤 동작 또는 선택을 할 때 지금까지 보편적으로 생각했던 대로 할 수 있도록 해 줌으로써 다시 생각해보아야하는 고민을 덜어주는 것이다. 예를 들어 신호등 색깔이 주는 의미에 일반인들은 익숙해 있다. 초록색은 '간다'라는 진행의 의미, 빨간색은 '정지'라는 멈춤의 의미로 받아들여진다. 따라서 진행과 멈춤의 의미를 나타내는 버튼을 만든다면 신호등 색깔과 일치하는 버튼의 색깔을 만들어야 사용자가 그 색깔만 보고도 고민 없이 쉽게 판단하고 사용할 수 있다. 즉 많은 사람이 공통적으로 생각하는 방식에 따라 사용할 수 있게 제공하는가 여부를 나타낸다.
운용성 <small>operability</small>	사용자가 소프트웨어 제품을 쉽게 운용하고 제어할 수 있는지 여부를 나타낸다.
친밀성	사용자에게 호감을 줄 수 있는 소프트웨어 제품의 능력을 말한다.
준수성 <small>compliance</small>	사용성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.



## 23. ISO/IEC9126의 품질 특성 및 하위 특성(5)

### ■ 효율성 efficiency

표 9-8 효율성의 하위 특성

특성	설명
시간 효율(반응)성 <small>time behavior</small>	특정 환경에서 기능을 실행할 경우 시간을 기준으로 얼마나 빨리 응답하는지, 처리 시간이 얼마나 짧은지 등을 나타낸다.
자원 활용(효율)성 <small>resource utilization, resource behavior</small>	기능을 실행했을 때 사용된 자원을 얼마나 적절하게 사용하였는지를 나타낸다.
준수성 <small>compliance</small>	효율성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.

## 24. ISO/IEC9126의 품질 특성 및 하위 특성(6)

### ■ 유지보수 용이성 maintainability

표 9-9 유지보수 용이성의 하위 특성

특성	설명
분석(용이)성 <sup>analyzability</sup>	결함이나 고장 등으로 변경해야 하는 경우, 그 원인이나 상태에 관한 정보를 파악할 수 있는 소프트웨어 제품의 능력을 말한다.
변경(기능)성 <sup>changeability</sup>	소프트웨어를 쉽게 변경할 수 있는 소프트웨어 제품의 능력을 말한다.
안정성 <sup>stability</sup>	소프트웨어 변경으로 인한 예상치 않은 결과를 최소화하는 소프트웨어 제품의 능력을 말한다.
테스트 용이성 <sup>testability</sup>	사용자가 요구하는 기능을 만족할 만큼 잘 수행하고 있는지에 대해 쉽고 철저하게 테스트할 수 있는지 여부를 나타낸다.
준수성 <sup>compliance</sup>	유지보수 용이성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.

# 25. ISO/IEC9126의 품질 특성 및 하위 특성(7)

## ■ 이식성portability

표 9-10 이식성의 하위 특성

특성	설명
(환경) 적응성 <sup>adaptability</sup>	소프트웨어가 설치되어 있는 운영체제나 미들웨어 또는 하드웨어 환경에서 정상적으로 잘 작동할 수 있는 소프트웨어 제품의 능력을 말한다.
설치(가능)성 <sup>installability</sup>	특정한 환경에 소프트웨어를 설치할 수 있는 소프트웨어 제품의 능력을 말한다.
대체(가능)성 <sup>replaceability</sup>	동일한 환경에서 동일한 목적으로, 다른 지정된 소프트웨어 제품을 대신하여 사용할 수 있는 소프트웨어 제품의 능력을 말한다.
공존성(병행존재)성 <sup>co-existence</sup>	공통 자원을 공유하는, 동일 운영 환경에서 도입된 서로 다른 소프트웨어가 간섭이나 충돌 없이 병행하여 서로 잘 운영될 수 있는 소프트웨어 제품의 능력을 말한다.
준수성 <sup>compliance</sup>	이식성과 관련된 표준, 규약, 관례, 법적 규제 및 유사한 규정 등을 따르는 소프트웨어 제품의 능력을 말한다.

# 26. ISO/IEC 14598 모델

## ■ ISO/IEC 14598 information technology-software product evaluation

- 소프트웨어 공급자와 구매자 사이에서 소프트웨어 개발 과정 또는 개발된 제품의 품질을 객관적으로 평가하기 위한 방법과 절차를 정의한 국제 표준 규격

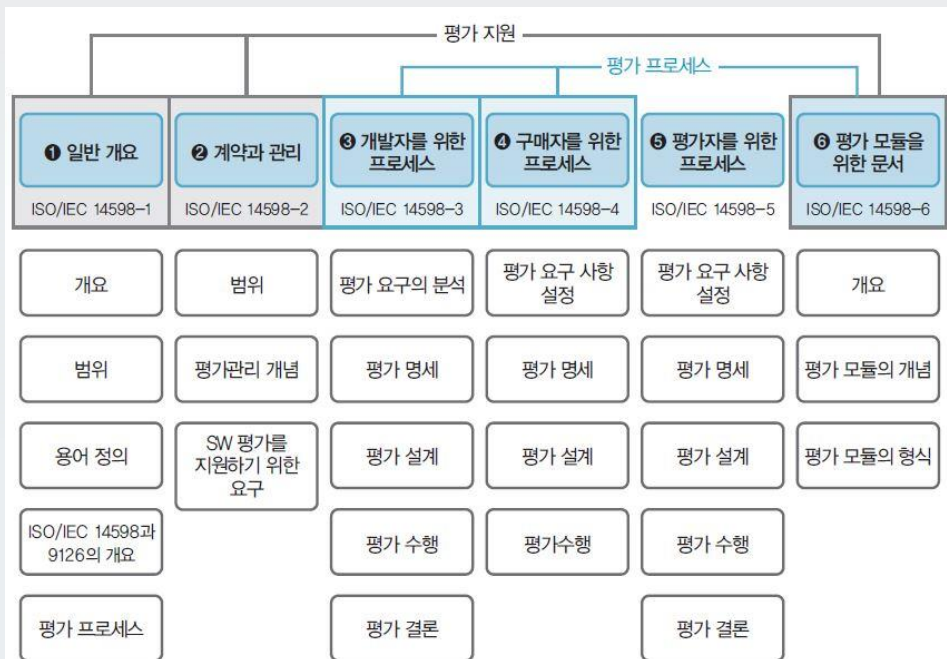


그림 9-10 ISO/IEC 14598의 전체 구성도

## 27. ISO/IEC 14598의 특성

### ■ ISO/IEC 14598의 특성

- 반복성(repeatability) : 특정 제품을 동일한 평가자(기관)가 동일 기준을 적용하여 평가 했을 때 동일 한 결과가 나와야 한다.
- 재생산성(reproducibility) : 특정 제품을 다른 평가자(기관)가 동일 기준을 적용하여 평가했을 때 동 일한 결과가 나와야 한다.
- 공정성(impartiality) : 평가가 특정한 결과를 내기 위해 불공정한 편견이 없어야 한다.
- 객관성: 평가는 주관적 판단을 최소화하고 객관적 자료를 근거로 해야 한다.
- ISO/IEC 9126 표준 준수: ISO/IEC 9126에 규정한 표준을 준수한다.
- 규정하지 않는 내용: 품질 평가의 측정 기술, 측정 결과의 해석 방법 등은 규정하고 있지 않다

## 28. ISO/IEC 14598의 제품 품질 측정 및 평가 방법(1)

- 일반 개요(ISO/IEC 14598-1)
  - 소프트웨어 제품의 품질 평가를 수행하기 위한 일반적인 개요로 범위와 용어를 정의
  - ISO/IEC 14598과 ISO/IEC 9126의 관계 설명
  - 개발자/평가자/구매자가 사용할 수 있는 평가 프로세스와 평가 모듈에 대한 로드맵 제공
- 계획과 관리(ISO/IEC 14598-2)
  - 품질 평가 척도를 프로젝트 성격에 따라 선정하고 적용하기 위한 제품 품질 측정 계획의 준비와 구현에 대해 다루고 있다.
  - 제품 평가 기능을 지원하기 위한 요구 사항과 안내 지침을 포함한 전체적인 사항을 제공
- 개발자를 위한 프로세스(ISO/IEC 14598-3)
  - 소프트웨어 개발 단계에서 알아야 할 개발자 준수 사항으로, 개발자가 개발 과정 및 최종 소프트웨어 제품의 품질 평가에 사용할 수 있는 방법을 제공

## 29. ISO/IEC 14598의 제품 품질 측정 및 평가 방법(2)

- 구매자를 위한 프로세스(ISO/IEC 14598-4)
  - 소프트웨어 제품을 구매하기 위한 계획을 수립할 때 사용하는 구매자용 프로세스
  - 구매자가 상용 소프트웨어 제품을 구매하는 과정에서 소프트웨어 제품의 품질 평가에 사용할 수 있는 방법 제공
- 평가자를 위한 프로세스(ISO/IEC 14598-5)
  - 품질 전문가를 위한 프로세스로, 평가자가 개발 과정 및 최종 소프트웨어 제품의 품질 평가를 수행할 때 사용
- 평가 모듈(ISO/IEC 14598-6)
  - 개발자/구매자/평가자가 소프트웨어 제품의 품질을 평가할 때 사용할 수 있도록 평가 모델에 대한 기본적인 가이드와 이론적인 모델 제공
  - 평가 모듈을 개발하여 문서화하고 검증하는 지침 제공

## 30. ISO/IEC 14598의 제품 품질 평가 절차



그림 9-11 ISO/IEC 14598 품질 평가 절차



# 31. ISO/IEC 12119 모델(1)

- ISO/IEC 12119 information technology-software packages-quality requirements and testing
  - 패키지 소프트웨어의 일반적인 제품 품질 요구 사항 및 테스트를 위한 국제 표준 규격
  - 규격 내용: 품질, 지침, 세부 인증
  - 요건 사항: 명확화, 유사 문서 정의, 변경(가능)성, 환경 명세, 보안성
  - 구성: 제품 설명서, 사용자 문서, 실행 프로그램
  - 첫 번째 평가 대상: - 패키지 소프트웨어
    - 제품 문서, 사용자 문서, 프로그램/데이터에 명시된 요구 사항
  - 두 번째 평가 대상: 패키지 SW와 최종 제품 및 중간 산출물을 포함하는 수주 개발 SW
  - 셋째 평가 대상: 패키지 SW와 최종 제품 및 개발/유지보수 과정을 포함한 수주 개발 SW
  - 테스트 대상: 소프트웨어 제품 명세서, 사용자 매뉴얼, 사용자 요구 테스트 결과서 등
  - 주요 테스트 평가 항목: 기능성, 신뢰성, 사용성, 이식성 등
  - 평가 척도: ISO 9126-2, ISO 9126-3(품질 메트릭), ISO 12119(테스트 방법)

## 32. ISO/IEC 12119 모델(2)

### 평가 절차와 SW 패키지 품질 요구사항 및 테스트 모델

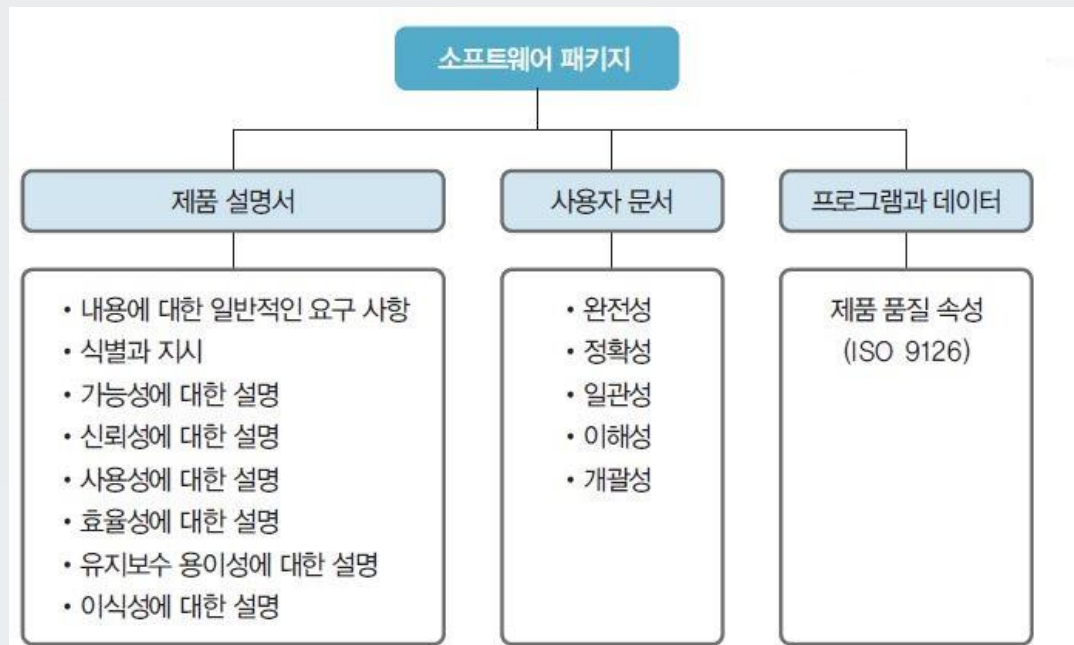


그림 9-13 ISO/IEC 9126 소프트웨어 패키지 품질 요구 사항 및 테스트 모델

# 33. ISO/IEC 25000 모델(1)

- ISO/IEC 25000, SQaRE<sup>Software Quality and Requirement Evaluation</sup>
  - 사용자들에게 유용하도록 여러 표준 문서를 통합하고 재구성하여 만든 국제 표준 문서
  - 소프트웨어 품질 평가 모델로부터 시작해 전체적인 품질 평가를 위한 표준 방안 제시

표 9-11 ISO/IEC 25000의 구성

모델 구분	구성	내용
2500×(9126-1)	제품 품질 일반 분류	SQaRE에 대한 개요, 전체에 대한 계획과 관리
2501×(9126-2)	소프트웨어 품질 모델	품질 모델 및 품질 사용 안내
2502×(9126-3)	소프트웨어 품질 측정	기본 매트릭스, 내/외부 매트릭스, 사용 중 품질 매트릭스, 평가 모듈의 문서화
2503×(9126-4)	소프트웨어 품질 요구	품질 요구 사항
2504×(9126-5)	소프트웨어 품질 평가	품질 평가 프로세스에 관한 개요 및 개발자, 구매자, 평가자 관점의 품질 평가 프로세스

## 34. ISO/IEC 25000 모델(2)

### ■ ISO/IEC 25000 구성도

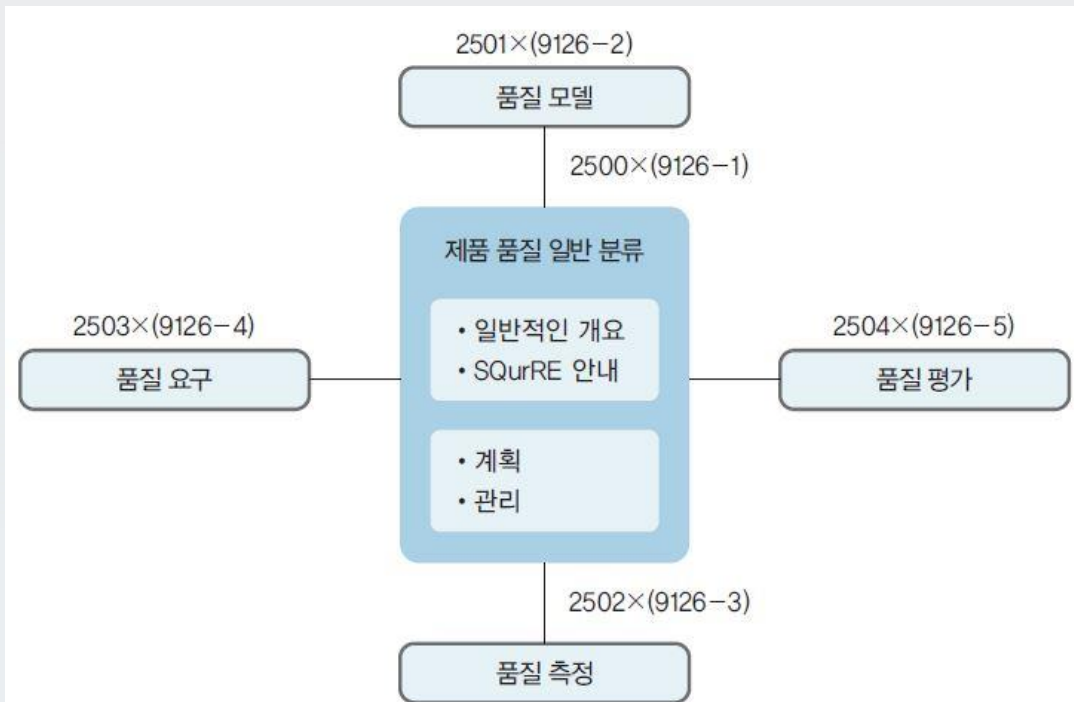


그림 9-14 ISO/IEC 25000의 구성도



다음 시간

# 프로세스 품질 특성 평가 모델



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는다, 무단  
전제, 배포, 전송, 대여 등을 금합니다.

\*사용서체 : 나눔글꼴