

부록 D

소프트웨어 품질 및 테스트 표준

1. ISO/IEC 9126	1
1.1. 개요	1
1.2. 범위	1
1.3. 준수 사항	3
1.4. 참조 표준	4
1.5. 품질 접근방법과 관련성	4
1.5.1. 제품 품질과 그 생명주기	4
1.6. 메트릭	11
1.6.1. 소프트웨어 제품 메트릭(Metric)	11
1.7. 소프트웨어 품질 특성	17
1.7.1. 기능성(Functionality)	17
1.7.2. 신뢰성(Reliability)	19
1.7.3. 사용성(Usability)	22
1.7.4. 효율성(Efficiency)	32
1.7.5. 유지보수성(Maintainability)	32
1.7.6. 이식성(Portability)	42
1.8. 사용 품질 특성	26
1.9. ISO/IEC 9126용어 정의	2
2. ISO/IEC 12119	35
2.1. 적용 범위	35
2.2. 정의	35
2.3. 품질 요구사항	38
2.3.1. 제품 설명서	9
2.4. 시험을 위한 지침	51
2.4.1. 선행조건 시험	3
2.4.2. 시험 활동	5
2.4.3. 테스트 기록	5
2.4.4. 테스트 보고서	5
2.4.5. 추후 테스트	6
2.5. ISO/IEC 12119 용어 정의	7

2.5.1. 부속서 A: 다른 규격에서 정의된 용어	5
2.5.2. 부속서 B: 제품설명서의 예	6
2.5.3. 해설서	6

3. ISO/IEC 14598 64

3.1. 개요	4
3.2. 평가 프로세스	4
3.2.1. 개발자를 위한 프로세스 (Process for Developers)	46
3.2.2. 획득자를 위한 프로세스 (Process for Acquirers)	46
3.2.3. 평가자를 위한 프로세스 (Process for Evaluators)	56
3.3. 평가를 위한 지원	5
3.3.1. 계획과 관리(Planning and Management)	6
3.3.2. 평가 모듈(Evaluation Modules)	6
3.4. 소프트웨어 품질 특성과 메트릭	6
3.5. 평가 프로세스	7
3.6. 평가 요구사항 설정	8
3.6.1. 평가 목적 설정	8
3.6.2. 평가될 제품의 타입을 식별	8
3.6.3. 품질 모델을 명세	9
3.7. 평가 명세화	9
3.7.1. 메트릭 선택	9
3.7.2. 메트릭을 위한 등급 레벨 성립	9
3.7.3. 평가를 위한 범주 성립	9
3.8. 평가 설계	9
3.9. 평가 실행	7
3.10. 프로세스 지원	7

4. IEEE1008 72

4.1. 범위	2
4.2. 단위 테스트 활동	2

5. IEEE1012	81
5.1. 개요	8
5.1.1. 검증	8
5.1.2. 확인	8
5.2. 소프트웨어 검증 및 확인 계획 가이드	8
5.3. 생명주기에 따른 검증 및 확인 활동	8

부록 - 표준

본 장에서는 ISO/IEC 9126, ISO/IEC 12119, ISO/IEC 14598, IEEE 1008, 그리고 IEEE 1012 등 다섯 개의 테스트 표준(Testing Standard)에 대하여 살펴본다.

1. ISO/IEC 9126

1.1. 개요

본 절에서는 ISO/IEC 9126에 대해 설명한다. 오늘날 컴퓨터는 점차 다양한 응용분야에서 광범위하게 사용되고 있으며, 정확한 작동은 기업 성공이나 인간 안전과 깊은 관련을 갖는다. 그러므로, 고품질의 소프트웨어 제품을 개발하거나 선정하는 일이 매우 중요하다. 소프트웨어 품질에 대한 포괄적인 명세 및 평가 작업은 적절한 품질을 확보하기 위한 주요 요인이다. 이러한 작업은 소프트웨어의 사용목적에 고려하여 적합한 품질 특성을 정의함으로써 달성된다. 소프트웨어 제품은 이미 검증되었거나 널리 인정된 메트릭을 사용하여 관련된 모든 특성에 따라 평가되어야 한다.

본 표준은 이러한 요구를 지원하기 위해 개발되었으며, 여섯 가지 품질 특성을 정의하고, 소프트웨어 제품 평가 프로세스 모델을 설명하고 있다.

1.2. 범위

본 표준의 범위(Scope)에서는 소프트웨어 품질을 여섯 가지 특성(characteristics)으로 구분하고, 이를 다시 부특성(sub-characteristics)로 세분화한 품질 모델을 명세한다. 이러한 부특성은 소프트웨어가 컴퓨터

시스템의 일부로 사용될 때 외부로 나타나며, 내부적인 소프트웨어 속성들의 결과이다. 최종 사용자를 위한 소프트웨어 품질 특성의 결합된 결과는 사용 품질(QIU: Quality In Use)로 정의된다.

정의된 특성은 펌웨어(firmware)에 수록된 컴퓨터 프로그램과 데이터를 포함하여 모든 종류의 소프트웨어에 적용할 수 있다. 특성과 부특성은 소프트웨어 품질에 대한 일관성 있는 용어를 제공한다. 또한 특성과 부특성은 소프트웨어에 대한 품질 요구사항을 명세하기 위해서, 또는 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 그리고 이식성과 같은 소프트웨어 능력 사이의 균형을 맞추기 위한 기본틀을 제공한다. 프로그래밍 환경의 재사용성과 같은 오직 개발자만이 관심을 갖는 소프트웨어 속성은 본 표준의 범위를 벗어난다.

구매, 요구명세, 개발, 사용, 평가, 지원, 유지보수, 품질 보증 및 소프트웨어 감사 등과 관련된 사람들이 서로 다른 관점에서 소프트웨어 제품 품질을 명세하고 평가할 수 있도록 한다. 본 표준에서 정의된 품질 모델의 사용 예는 다음과 같다.

- ◎ 요구사항 정의의 완전성 확인
- ◎ 소프트웨어 요구사항 확인
- ◎ 소프트웨어 설계 목적 확인
- ◎ 시험 목적 확인
- ◎ 완성된 소프트웨어 제품의 사용자 인수 기준 확인

본 표준은 다음 사항을 제공하기 위해 ISO/IEC 15504(소프트웨어 프로세스 심사에 관한 표준)와 함께 사용될 수 있다.

- ◎ 고객-공급자 프로세스에서 소프트웨어 제품 품질을 정의하기 위한 기본 틀
- ◎ 지원 프로세스에서 검토, 검증 및 확인을 위한 지원, 그리고 정량

적품질 평가를 위한 기본 틀

- ◎ 관리 프로세스에서 조직 품질 목표를 설정하기 위한 지원

본 표준은 다음 사항을 제공하기 위해 ISO/IEC 12207(소프트웨어 생명주기에 관한 표준)과 함께 사용될 수 있다.

- ◎ 주요생명주기 프로세스 내에서 소프트웨어 품질 요구사항을 정의하기 위한 기본 틀
- ◎ 생명주기 프로세스를 지원하는데 있어서 검토, 검증 및 확인을 위한 지원

본 표준은 다음 사항을 제공하기 위해 ISO 9001과 함께 사용될 수 있다.

- ◎ 품질목표를 설정하기 위한 지원
- ◎ 설계 검토, 검증 및 확인을 위한 지원

1.3. 준수 사항

본 항에서는 본 표준의 준수사항(Constraint)에 대하여 살펴본다. 본 표준을 준수하는 소프트웨어 품질에 대한 모든 명세서는 다음과 같은 사항에 대한 적절한 기준을 식별하기 위해 1.1.5절 4)에 정의된 품질 모델을 사용할 것을 권고한다.

- ◎ 1.1.7절에 있는 특성 및 부특성을 위한 내부 메트릭에 대한 기준
- ◎ 1.1.7절에 있는 특성 및 부특성을 위한 외부 메트릭에 대한 기준
- ◎ 1.1.8절에 있는 사용 품질 메트릭에 대한 기준

기준설정이나 비교를 위한 어떠한 메트릭도 1.1.6절 4)의 요구사항을 충족할 것을 권고한다.

1.4. 참조 표준

본 표준의 참조 표준(Reference Standard)은 다음과 같다.

- ◎ ISO 9001 (1994) Model for quality assurance in design, development, production, installation and servicing.
- ◎ ISO/IEC 12207 (1995) Information Technology – Software life-cycle processes.
- ◎ ISO/IEC PDTR 15504 (1996) Information Technology – Software Process Assessment.
- ◎ ISO/IEC DIS 14598-1 (1996) Information Technology – Software product evaluation – Part 1: General overview.

1.5. 품질 접근방법과 관련성

본 항에서는 소프트웨어에 대한 품질 접근방법과 관련성에 대해 살펴본다.

1.5.1. 제품 품질과 그 생명주기

품질은 소프트웨어 생명주기와 함께 변화한다. 즉, 생명주기의 초기에 요구한 제품 품질이 실제로 인도된 제품 품질과 다르다. 품질 관정도 또한 여러 관점에서 반영된다. 생명주기의 각 단계에서 품질을 적절히 관리하기 위해서는 이러한 품질의 다양한 관점과 생명주기에 따른 품질 변화에 대한 정의가 필수적이다. 다음은 생명주기의 서로 다른 단계에서 바라본 다양한 제품 품질 관점들이다.

목표 품질(GQ: Goal Quality)은 실제 사용자의 요구를 반영하는 필요충분한 품질이다. ISO 8402는 품질을, 명시된 요구와 내재된 요구를 만족시키는 능력이라고 정의하고 있다. 그러나, 사용자가 명시한 요구사항이

항상 실제 사용자의 요구를 반영하는 것은 아니다. 왜냐하면 사용자는 자신의 실제 요구사항들을 잘 모를 수 있으며, 명시된후에도 요구사항이 바뀔 수 있기 때문이다. 따라서, GQ는 설계 초기에 완벽히 정의될 수 없는 개념적인 개체이다. 하지만 개발자는 이 목표를 염두에 두고 이에 가까이 가도록 노력해야 한다. 사용자의 요구사항과 잠재적인 어떤 문제점을 좀 더 명확히 이해하기 위해서는 고찰된 특정 사용 시나리오를 문서화해 두는 것이 유용할 수 있다. GQ란 반드시 완전한 품질을 의미하는 게 아니라 필요 충분한 품질을 말한다. GQ에 대한 일부 요구사항은 사용 품질(QIU)로 측정할 수 있으며, QIU에 대한 요구사항은 가능한 한 품질 요구 명세서에 포함시켜야 한다.

요구 제품 품질(RPQ: Required Product Quality)은 품질 요구 명세서에 실질적으로 기술된 품질이다. RPQ는 초기 확인을 위한 목표로 사용되어야 한다. ISO/IEC 9126에 정의된 모든 품질 특성에 대한 품질 요구사항은 품질 요구 명세서에 명시되어야 한다. 최적의 요구사항뿐만 아니라 최소한의 요구사항도 명시되어 사용자나 개발자 모두 불필요한 비용이나 스케줄 지연 등을 피해야 한다.

설계 품질(DQ: Design Quality)은 소프트웨어 설계의 핵심 부분이나 중추적인 부분, 예를 들어, 소프트웨어 구조, 프로그램 구조 및 사용자 인터페이스 설계 전략 등에 나타나는 품질이다. DQ는 설계 원칙과 전략을 반영한다. 소프트웨어 품질의 상세한 내용은 코드 구현이나 시험 과정에서 개선될 수 있지만, DQ에 의해 나타나는 소프트웨어 제품 품질의 근본적인 속성은 바뀌지 않는다.

예상 제품 품질(EPQ: Estimated Product Quality)은 개발 각 단계에서 최종 소프트웨어 제품의 품질에 대한 예상 또는 예측 품질로서 DQ를 기반으로 한다. 제품 품질은 개발 도중에 ISO/IEC 9126에서 정의된 각 품질 특성에 대해 예상하거나 기대할 수 있다. 이러한 예상을 위해 DQ와

EPQ간의 상관관계를 나타낼 수 있는 기법이 개발되어야 한다.

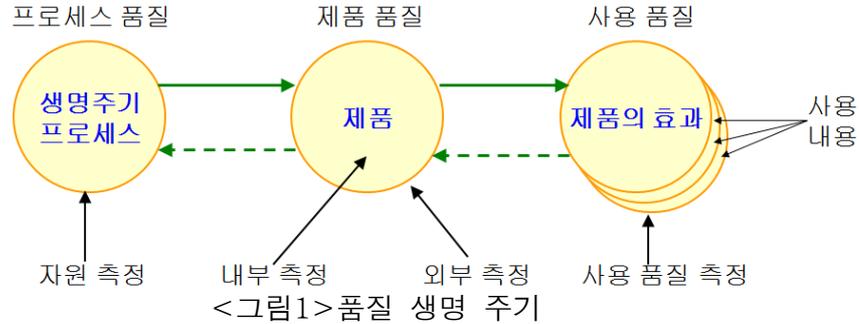
인도 제품 품질(DPQ: Delivered Product Quality)은 전형적으로 모의 실험 데이터를 가지고, 모의 환경에서 시험된 인도된 제품의 품질이다. 시험 도중에, 대부분의 결함이 발견되고 제거되어야 한다. 그러나, 일부 결함은 시험단계 이후에도 남아있을 수 있다. 소프트웨어 구조나 소프트웨어에 대한 기본적인 설계 관점은 변경하기 어렵기 때문에 기본적인 설계는 통상적으로 시험과정에서 바뀌지 않고 유지된다.

사용 품질은 소프트웨어를 포함하고 있는 시스템에 대한 사용자 관점의 품질이며, 소프트웨어 자체의 성질보다는 그 소프트웨어를 사용한 결과의 관점에서 측정된다.

참고 - "사용자"는 최종 사용자와 유지보수자 모두를 말하며, 그들의 요구사항은 다를 것이다.

사용자 환경에서의 품질은 개발자 환경에서의 품질과 다를 수 있다. 왜냐하면, 일부 기능은 사용자가 볼 수 없거나 혹은 사용자가 사용하지 않을 수 있기 때문이다. 사용자는 실제로 사용 도중에 자신에게 보이는 소프트웨어 속성만을 평가한다. 때로는요구분석 단계에서 사용자가 명시한 소프트웨어 속성들이 바람직하지 않을 경우도 있다.

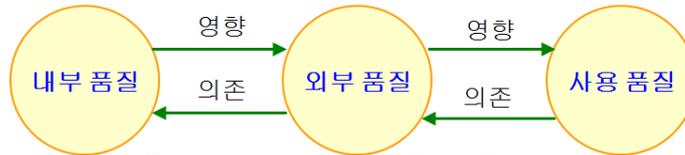
가. 품질 접근방안



소프트웨어 제품 품질을 달성하기 위해 소프트웨어 제품을 평가하는 것은 소프트웨어 개발 생명주기의 한 과정이다. 그림 8.1에서와 같이, 소프트웨어 제품 품질은 내부 속성을 측정함으로써, 혹은 외부 속성에 의해 평가될 수 있다. 그 목적은 제품이 특정한 사용에서 요구한 효과를 내도록 하자는 것이다.

프로세스 품질(ISO/IEC 12207에 정의된 모든 생명주기 프로세스의 품질)은 제품 품질의 개선을 가져오고, 제품 품질은 사용 품질의 개선을 가져온다. 그러므로, 프로세스를 평가하고 개선하는 것이 제품 품질을 개선하는 한 방법이며, 제품 품질을 평가하고 개선하는 일이 사용 품질을 개선하는 한 방법이다.

적절한 품질 평가 프로세스는 개발 동안에 품질측정을 지원하는 데 필요하다. <그림2>와 같이, 소프트웨어의 적절한 내부 속성은 요구된 외부 행위를 성취하기 위한 선결과제이며, 적절한 외부 행위는 사용 품질을 성취하기 위한 선결과제이다.



<그림2> 서로 다른 품질 관점간의 관련성

소프트웨어 제품 품질에 대한 요구사항은 일반적으로 개발자, 유지보수자, 구매자 및 최종사용자의 요구를 충족시키기 위한 내부 품질, 외부 품질 및 사용 품질에 대한 기준을 포함한다.

나. 평가 항목

항목들은 직접 측정 방법이나 혹은 그 결과를 측정하여 간접적으로 평가할 수 있다. 예를 들어, 프로세스는 그 제품을 측정하고 평가함으로써 판정할 수 있고, 제품은 사용자의 작업 성능을 측정해 봄으로써 평가할 수 있다.

소프트웨어는 단독으로 수행되는 것이 아니라 인터페이스, 하드웨어, 사용자 및 작업 흐름 등을 포함하는 다른 소프트웨어 제품으로 구성된, 보다 큰 시스템의 일부로서 수행된다. 인도된 소프트웨어 제품은 선정된 외부 메트릭 수준에서 평가된다. 이러한 메트릭은 그 환경과의 상호작용을 나타내며 운영 중에 있는 소프트웨어를 관찰함으로써 판정된다. 외부 품질 (제품이 명시되거나 내재된 요구를 만족하는 정도)는 운영환경에서 사용 품질을 평가함으로써 측정할 수 있다. 이러한 것은 일반적으로 좀 더 특정한 소프트웨어 특성을 측정함으로써 이루어지며, 개발 과정의 초기에서도 가능하다.

초기 개발 단계에서는 자원이나 프로세스만이 측정 가능하다. 중간 제품(명세서, 원시 코드, 등)이 나오기 시작하면, 선정된 내부 메트릭 수준으로 이러한 것들을 평가할 수 있다. 이러한 메트릭은 외부 메트릭의 값을

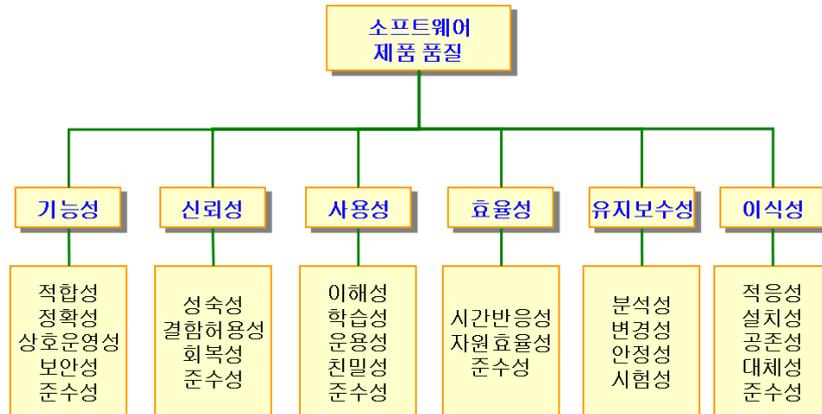
예측하는 데 사용할 수 있다. 또한, 내부 메트릭은 외부 메트릭에 대한 필수적인 선결과제로써 그 자체로 측정될 수 있다.

또 다른 구분으로는 소프트웨어 제품에 대한 평가와 그 제품이 실행되는 시스템에 대한 평가로 나누어 볼 수 있다. 예를 들어 시스템의 신뢰성은 어떤 원인(하드웨어, 소프트웨어, 사람 실수 등)으로 인한 모든 문제를 관찰해 봄으로써 판단 할 수 있다. 반면에, 소프트웨어 제품의 신뢰성은 단지 소프트웨어 내부에 있는 결함(요구 분석, 설계 혹은 구현으로부터 나온)으로 인한 고장 예를 관찰해 봄으로써 판정할 수 있다. 또한, 시스템의 경계는 평가 목적이나 사용자에 따라 다르게 판단될 수 있다.

참고 - 예를 들어, 컴퓨터를 기반으로 하는 비행 제어 시스템을 장착한 항공기의 사용자를 승객이라 한다면, 그에 따른 시스템에는 비행 승무원, 항공기 및 비행 제어 시스템내의 하드웨어와 소프트웨어 등이 포함되지만, 비행 승무원을 사용자로 본다면, 그에 따른 시스템에는 단지 항공기와 비행 제어 시스템만이 포함된다.

다. 품질 모델

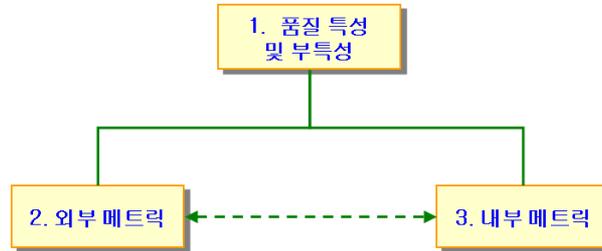
소프트웨어 품질은 정의된 품질 모델을 사용하여 평가해야 한다. [그림 8-3]과 같이, 품질 모델은 소프트웨어 제품이나 중간 제품의 품질 목표를 설정할 경우에 사용해야 한다. ISO/IEC 9126의 이 부분은 품질과 관련된 문제점에 대한 점검표(checklist)로서 사용 가능한 품질 모델(특정한 경우에는 더 적절한 품질 분류 방법이 있을 수 있지만)을 제시한다.



<그림3> 소프트웨어 제품 품질

소프트웨어 품질 속성은 여섯 가지 특성(기능성, 신뢰성, 사용성, 효율성, 유지보수성 및 이식성)으로 구분되며, 이러한 특성들은 다시 부특성들로 세분된다. 부특성은 내부 메트릭이나 외부 메트릭에 의해 측정 가능하다. 프로그램 크기와 같은 기본적인 내부 메트릭은 보통 그 하나만으로 소프트웨어 품질 메트릭으로 사용되는 것이 아니라 메트릭을 생성하기 위해 다른 측정과 함께 사용되는 소프트웨어 측정치이다.

사용 품질은 사용자 관점의 품질을 말하며, 효율성, 생산성, 안전성 그리고 만족도 등 네 가지 특성을 포함한다. 그림 8.4와 같이, 사용 품질의 달성은 관련된 소프트웨어 제품의 품질 부특성에 대한 외부 측정의 기준치 도달 여부에 달려 있으며, 외부 측정은 다시 관련된 내부 측정의 연관 기준치 도달 여부에 달려 있다. 측정은 보통 세 단계 모두에서 필요하다. 왜냐하면, 내부 측정의 기준치 만족이 외부 측정의 기준치 달성 여부를 확인하는데 일반적으로 충분하지 않으며, 부특성에 대한 외부 측정의 기준치 만족이 사용 품질에 대한 기준치 만족 여부를 확인하는 데 보통 충분하지 않기 때문이다.



<그림 4> 사용 품질

대형소프트웨어 제품의 모든 부분에 대해 내부적이거나 외부적으로 모든 부특성을 측정한다는 것은 실질적으로 불가능하다. 또한 사용자 작업의 모든 가능한 시나리오에 대해 사용 품질을 측정하는 것도 통상적으로 현실적이지 않다. 평가 요구를 위해 서로 다른 측정 유형 간에 할당되는 자원도 사업 목적과 제품이나 설계 프로세스의 속성에 따라 다르다.

1.6. 메트릭

1.6.1. 소프트웨어 제품 메트릭(Metric)

가. 속성 및 특성

어떤 내부 속성의 수준은 일부 외부 측정의 수준에 영향을 미친다고 알려져 있다. 따라서 대부분의 특성에는 외부적인 측면과 내부적인 측면의 양면성이 있다. 예를 들어, 신뢰성에 대해서는 그 소프트웨어를 시험하면서 주어진 실행시간 동안에 결함의 횟수를 관찰함으로써 외부적으로 측정할 수 있으며, 내부적으로는 고장허용 수준을 평가하기 위해 상세한 명세서와 원시 코드를 조사해 볼 수 있다. 내부 속성을 외부 속성에 대한 표시

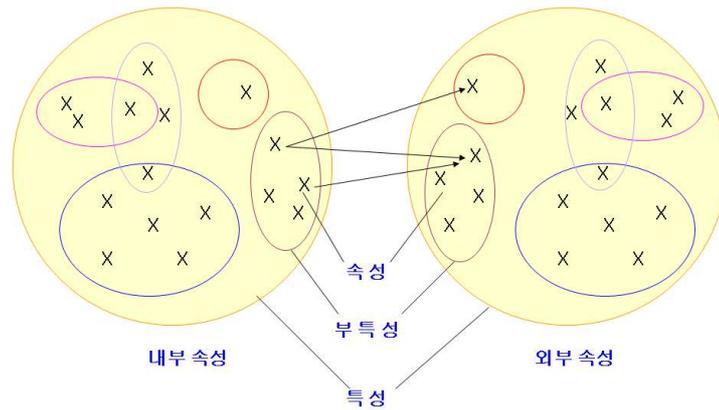
자라고 말한다. 그림 8.5와 같이, 하나의 특성이 여러 개의 속성에 의해 영향을 받는 것과 마찬가지로 하나의 속성이 여러 개의 특성에 영향을 줄 수 있다. 본 모델에서 전체적인 품질 속성은 특성과 부특성으로 이루어진 계층적인 나무 구조 형태로 구분되어 있다. 이 구조의 최상층은 품질 특성들로 이루어지며 최하층은 소프트웨어 품질 속성들로 구성되어 있다. 일부 속성은 한 개 이상의 부특성에 관련이 있기 때문에 이 계층구조는 완벽한 것은 아니다.

참고 - 내부 속성과 외부 측정간의 상관관계는 완벽한 것이 아니며, 어떤 내부 속성이 연관된 외부 측정에 끼치는 영향은 경험에 의해 결정되며 소프트웨어가 사용되는 특정 환경에 따라 달라질 것이다.

마찬가지로, 외부 성질(적합성, 정확성, 결함 허용성이나 시간 반응성과 같은)은 나타나는 품질에 영향을 줄 수 있다. 사용상의 고장(예를 들어, 사용자가 작업을 마칠 수 없다)은 외부 품질(적합성 혹은 운영성) 및 고쳐야 할 연관된 내부 속성으로 추적해 볼 수 있다.

나. 내부메트릭

내부메트릭은 설계나 코딩 도중에 실행할 수 없는 소프트웨어 제품(명세서나 원시 코드와 같은)에 적용할 수 있다. 소프트웨어 제품을 개발할 경우에 그 중간 제품에 대해서는 모의실험 행위로부터 추출 가능한 본질적인 성질을 측정하는 내부 메트릭을 이용하여 평가해야 한다. 이러한 내부 메트릭의 주된 목적은 요구된 외부 품질이 성취되었는가를 확인하는 것이며 그 예는 ISO/IEC 9126-3에 나와 있다. 내부 메트릭은 사용자, 평가자, 시험자 및 개발자가 소프트웨어 제품 품질을 평가할 수 있도록 도와주며 그 소프트웨어 제품을 만들기 전에 미리 품질 문제점들을 지적해준다.



<그림5> 품질 특성, 부특성 및 속성

내부메트릭은 중간 제품이나 인도된 소프트웨어 제품의 정적인 성질을 분석함으로써 내부 속성을 측정하거나 외부 속성을 보여준다. 내부 메트릭의 측정은 예를 들어 원시 코드 명령문, 제어 그래프 및 상태 전이 표현 등에 나타나는 소프트웨어 구성 요소들의 개수나 빈도수를 사용한다.

참고 - 문서들에 대해서도 내부 메트릭을 사용하여 평가할 수 있다.

다. 외부메트릭

외부메트릭은 실행 가능한 소프트웨어나 시스템을 시험, 운영, 관찰함으로써 소프트웨어를 구성하는 시스템 형태에 대한 측정을 위하여 사용되는 메트릭이다. 소프트웨어 제품을 구매하거나 사용하기 전에, 특정한 조직이나 기술적 환경에서 제품의 사용, 개발 및 관리와 관련된 기업 목적을 토대로 메트릭을 사용하여 평가해야 한다. 이러한 것들이 주로 외부 메트릭이며 그 예는 ISO/IEC 9126-2를 따른다. 외부 메트릭은 사용자, 평가

자, 시험자 및 개발자가 시험 수행이나 운영 중에 소프트웨어 제품 품질을 평가할 수 있도록 도와준다.

라. 외부와 내부 메트릭간의 관계

소프트웨어 품질 요구사항이 정의되면, 품질 요구사항을 표현하는 소프트웨어 품질 특성이나 부특성이 열거된다. 그 다음은, 그 소프트웨어가 사용자 요구를 만족하는 지를 확인하는 품질 기준을 정량화하기 위해 적절한 외부 메트릭 및 수용 가능한 범위가 명시된다. 그 다음은, 요구된 외부 품질과 사용 품질이 결국에는 성취될 수 있도록 계획하고 개발 도중에 중간 제품에 그 품질을 구축할 수 있도록 소프트웨어에 대한 내부 품질 속성이 정의되고 명시된다. 내부 품질 특성의 정량화를 위해 적절한 내부 메트릭 및 수용 가능한 범위를 명시하여 개발 도중에 중간 제품이 내부 품질 명세를 만족하는 지를 검증하는 데 사용될 수 있도록 한다.

목표하는 외부 메트릭과 밀접한 관계를 가지는 내부 메트릭의 사용을 권장하며, 외부 메트릭의 값을 예측하는 데도 사용할 수 있다. 그러나 내부 메트릭과 외부 메트릭간에 강한 관련성을 제공하는 이론적 모델을 설계한다는 것은 어려운 문제이다.

마. 사용 품질 메트릭

사용 품질 메트릭은 제품이 사용될 경우에 정해진 목표를 달성하기 위하여 효율성, 생산성 및 만족도 측면에서 어느 정도 특정 사용자의 요구를 충족하는가를 측정한다. 사용 품질을 평가하는 일은 특정 사용자 작업 시나리오 내에서 소프트웨어 품질을 확인하는 것이다.

사용 품질은 소프트웨어를 포함하고 있는 시스템의 품질을 사용자 관

점에서 본 것이며, 소프트웨어 자체의 성질보다는 그 소프트웨어를 사용한 결과의 관점에서 측정한다. 사용 품질은 사용자를 위한 소프트웨어 품질 특성의 복합적인 효과이다. 사용 품질과 다른 소프트웨어 품질 특성과의 관련성은 사용자 유형에 따라 다르다.

- ⊙ 최종사용자에 있어서 사용 품질은 기능성, 신뢰성, 사용성 및 효율성의 결과이다.
- ⊙ 소프트웨어 유지보수 담당자에 있어서 사용 품질은 유지보수성의 결과이다.
- ⊙ 소프트웨어 전이 담당자에게는 사용 품질은 이식성의 결과이다.

바. 메트릭과 측정 기준의 선정

메트릭 선정의 기반은 제품에 대한 기업 목표와 평가자의 요구에 따라 다를 것이다. ISO/IEC 9126의 이 부분에서의 모델은 다음 예와 같은 다양한 평가 요구사항을 지원한다.

- ⊙ 사용자나 사용자의 기업 단위에서는 사용 품질에 대한 메트릭을 사용하여 소프트웨어 제품의 적합성을 평가 한다.
- ⊙ 구매자는 기능성, 신뢰성, 사용성 및 효율성에 대한 외부 측정 기준 값이나 사용 품질 기준 값을 고려하여 소프트웨어 제품을 평가 한다.
- ⊙ 유지보수자는 유지보수성에 대한 메트릭을 사용하여 소프트웨어 제품을 평가 한다.
- ⊙ 서로 다른 환경에서 소프트웨어 설치를 담당한 사람은 이식성에 대한 메트릭을 사용하여 소프트웨어 제품을 평가 한다.
- ⊙ 개발자는 모든 품질 특성에 대한 내부 측정을 사용한 기준 값에 대비하여 소프트웨어 제품을 평가 한다.
- ⊙ 참고- ISO/IEC 14598-1은 메트릭 선정을 위한 지침과 소프트웨어 제품 평가를 위한 측정 기준을 제공한다.

사. 제품간의 비교를 위한 측정 요구사항

제품간이나 기준 값간의 신뢰성 있는 비교를 하기 위해서는 엄격한 메트릭이 필요하다. 측정 절차는 소프트웨어 품질 특성 (혹은 부특성)에 대해 기준을 설정하고 비교를 할 수 있도록 충분한 정확성을 가지고 측정해야 한다. 점검표에 의한 자료와 전문가 의견 등은 서로 다른 속성을 가진 제품을 비교할 경우에는 신뢰성이 낮을 수 있다. 측정 도구나 사람의 실수로 인한 측정 오류의 허용치도 있어야 한다.

비교를 위해 사용되는 메트릭은 신뢰성 있는 비교를 수행하는 데 사용할 수 있기 위해 타당해야 하며 정확해야 한다. 이는 측정이 객관적이며, 간격 또는 보다 나은 측정 범위를 사용한 경험적이며, 재생성할 수 있어야 함을 요구한다.

- ◎ 객관적이기 위해서, 제품의 속성에 따라 적용될 수치나 범주에 대해 문서로 작성된 합의 절차가 있어야 한다.
- ◎ 경험적이기 위해서, 자료들은 관찰이나 혹은 타당한 설문조사에 의해 수집되어야 하며, 간격, 비율, 또는 절대치 범주로 측정되어야 한다.
- ◎ 재생성 할 수 있기 위해서, 측정 절차는 서로 다른 사람이 상이한 경우에 소프트웨어 제품에 대해 동일한 측정을 시행한 결과가 동일하게(적절한 오차한계 내에서) 나와야 한다.

내부 메트릭은 또한 예언적인 타당성을 가져야 한다. 즉 이들은 일부 원하는 외부 측정과 상관관계가 있어야 한다. 예를 들어, 특정 소프트웨어의 속성에 대한 내부 측정은 그 소프트웨어가 사용될 경우에 측정 가능한 일부 품질 측면과 상관관계가 있어야 한다. 측정으로 주어지는 값이 통상적인 기대치와 일치되는 것이 중요하다. 예를 들어, 측정에서 제품의 품질

이 높은 것으로 나왔다면, 이는 특정 사용자 요구사항에 따라 만족하는 제품이 나와야 한다.

1.7. 소프트웨어 품질 특성

본 항에서는 소프트웨어 품질 특성(Software Quality Characteristics)에 대하여 살펴본다. 앞서 언급한 품질 모델은 소프트웨어 제품의 품질 속성을 여섯 가지 특성(기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성)으로 구분하고 있다. 이들은 다시 그 아래에 부특성들로 세분된다. 부특성들은 내부 혹은 외부 메트릭으로 측정할 수 있다. 또한 순수한 외부 메트릭이 있다.

각 품질 특성과 그 품질 특성에 영향을 미치는 소프트웨어 부특성에 대해서는 정의된다. 측정 가능한 내부 속성에 의해 각 특성과 부특성에 대한 그 소프트웨어 능력이 결정된다. 내부 메트릭의 예는 ISO/IEC 9126-3에 나와 있다. 특성과부특성은 그 소프트웨어를 포함하고 있는 시스템이 제공하는 능력 정도에 의해 외부적으로 측정될 수 있다. 외부 메트릭의 예는 ISO/IEC 9126-2에 나와 있다.

참고 - ISO/IEC 9126의 이 부분에 있는 일부 특성은 의존성(dependability)과 연관이 있다. 의존성 특성은 모든 유형의 시스템에 대해 IEC 50(191)에 정의되어 있으며, ISO/IEC 9126의 이 부분에 있는 용어가 IEC 50(191)에 정의되어 있고, 이 정의는 폭넓은 호환성이 있다.

1.7.1. 기능성(Functionality)

소프트웨어가 특정 조건에서 사용될 때, 명시된 요구와 내재된 요구를

만족하는 기능에 대한 소프트웨어 제품의 능력.

참고- 다른 특성들은 주로 소프트웨어가 언제, 그리고 어떻게 동작 하는가에 초점을 두는 반면, 이 특성은 요구를 충족하기 위해서 소프트웨어가 무엇을 하는가에 관심을 둔다.

특성의 명시된 요구와 내재된 요구에 대해서는 품질 정의 부분에 기술된 참고를 적용한다.

사용자에 의해 운영되는 시스템에 대해서 기능성, 신뢰성, 사용성, 그리고 효율성의 복합적인 사항은 사용 품질에 의해 외부적으로 측정될 수 있다.

가. 적합성(Suitability)

지정된 작업과 사용자 목적에 따라 적절한 기능들을 제공하는 소프트웨어 제품의 능력.

◎ 참고

- 적합성의 예로는 구성 요소가 되는 부기능, 테이블 용량 등으로부터 나오는 작업 위주의 복합 기능이 있다.
- 적합성은 ISO 92410-10에 있는 작업을 위한 적합성에 대응된다.
- 적합성은 운영성에 영향을 준다.

나. 정확성(Accuracy)

올바른 혹은 합의된 결과를 제공할 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 이것은 계산된 값이 필요한 정밀도를 가지는 예상 데이터를 포함한다.

다. 상호운영성(Interoperability)

하나이상의 명세된 시스템과 상호작용할 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 상호운영성은 대체성(Replaceability)과 혼동될 수 있으므로 이를 피하기 위해서 호환성(Compatibility) 대신에 사용되었다.

라. 보안성(Security)

권한이 없는 사람 혹은 시스템은 정보를 읽거나 변경하지 못하게 하고, 권한이 있는 사람 혹은 시스템은 정보에 대한 접근이 거부되지 않도록 정보를 보호하는 소프트웨어 제품의 능력.

◎ 참고

- 이것은 전송중인 데이터에도 적용된다.
- 안전성(Safety)은 소프트웨어에만 관련된 것이 아니라 전체 시스템에 관련되기 때문에 사용 품질의 부특성으로 정의된다.

마. 준수성(Compliance)

응용과 관련된 표준, 관례 또는 법적 규제 및 유사한 규정을 준수하는 소프트웨어 제품의 능력.

1.7.2. 신뢰성(Reliability)

소프트웨어가 명세된 조건에서 사용될 때, 성능 수준을 유지할 수 있는

소프트웨어 제품의 능력.

◎ 참고

- 소프트웨어는 해지거나 낡지 않는다. 신뢰성의 한계는 요구사항, 설계 및 결함에 원인을 제공한다. 이러한 결함으로 인한 고장은 사용 경과 시간보다는 소프트웨어 제품의 사용 방법과 선정된 프로그램 선택 사항에 따라 달라질 수 있다.
- ISO/IEC DIS 2382-14:1994에서는 신뢰성의 정의를 " ... 요구된 기능을 수행하는 기능적 단위의 능력"으로 하고 있다. 이 문서에서, 기능성은 단지 소프트웨어 품질 특성중의 하나일 뿐이다. 그러므로 기능성의 정의를 "... 요구된 기능을 수행하는..." 대신에 "... 성능 수준을 유지하는 ..." 으로 확장한 것이다.

가. 성숙성(Maturity)

소프트웨어 내의 결함으로 인한 고장을 피해 가는 소프트웨어 제품의 능력.

나. 결함허용성(Fault Tolerance)

명세된 인터페이스의 위반 혹은 소프트웨어 결함이 발생했을 때 명세된 성능 수준을 유지할 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 성능의 명세된 수준은 고장 안정 능력(fail safe capability)을 포함한다.

다. 복구성(Recoverability)

고장발생시 명세된 성능 수준을 다시 유지하고 직접적으로 영향 받은 데이터를 복구하는 소프트웨어 제품의 능력.

◎ 참고

- 소프트웨어 제품은 고장 난 후에 때때로 복구성으로 판정될 수 있는 일정 시간 동안 작동이 멈출 수 있다.
 - 가용성(Availability)이란 명시된 사용환경에서 특정 시점에 요구되는 기능을 수행할 수 있는 소프트웨어제품의 능력이다. 외부적으로 가용성은 소프트웨어 제품이 정상 작동하는 전체 시간의 비율로써 판정된다. 그러므로 가용성은 성숙성(고장율), 결함 허용성, 복구성(각 고장의 작동 중지 시간) 등의 복합체라 할 수 있다.
- ◎ 준수성(Compliance)
- 신뢰성과 관련된 표준, 관례 또는 규제를 준수하는 소프트웨어 제품의 능력.

1.7.3. 사용성(Usability)

명시된 조건에서 사용될 경우, 사용자에게 의해 이해되고, 학습되고, 사용되고 선호될 수 있는 소프트웨어 제품의 능력.

- ◎ 참고
- 기능성, 신뢰성, 효율성 등의 몇몇 특징들은 사용성에 영향을 줄 수 있지만 ISO/IEC 9126의 목적상 사용성으로써 분류되지 않는다.
 - 사용자에는 소프트웨어 사용에 영향을 받거나 의존하는 운영자, 최종 사용자, 그리고 간접 사용자 등이 포함된다. 사용성은 사용 준비나 결과 평가 등 소프트웨어가 영향을 줄 수 있는 모든 사용자 환경에 대처해야 한다.

가. 이해성(Understandability)

소프트웨어가 적합한지, 그리고 특정 작업과 사용 조건에서 어떻게 사용될 수 있는지를 사용자가 이해할 수 있도록 하는 소프트웨어 제품의 능력.

⊙ 참고

- 이것은 소프트웨어의 문서가 중요하다.

나. 학습성(Learnability)

사용자가 그 응용을 학습할 수 있도록 하는 소프트웨어 제품의 능력.

⊙ 참고

- 학습을 위한 적합성에 대응하는 내부 속성들은 ISO 9241-10에 정의되어 있다.

다. 운영성(Operability)

사용자가 소프트웨어 제품을 운영하고 제어할 수 있도록 하는 소프트웨어 제품의 능력.

⊙ 참고

- 적합성, 변경성, 적응성, 설치성 등의 특징들은 운영성에 영향을 미칠 수 있다.
- ISO 9241-10에 정의된 바와 같이 제어성, 결합 허용성, 사용자 기대 부합성이 운영성에 대응된다.
- 사용자에게 의해 운영되는 시스템에 대해서 기능성, 신뢰성, 사용성, 그리고 효율성 등의 복합체는 사용 품질에 의해 외부적으로 측정될 수 있다.

라. 선호도(Attractiveness)

사용자에게 의해 선호되는 소프트웨어 제품의 능력.

⊙ 참고

- 이것은 사용자가 그 소프트웨어에 좀 더 호감을 가질 수 있도록 만드는 것이다. 예를 들면 색깔의 사용이나 그래픽 설계 등을 말한다.

마. 준수성(Compliance)

사용성과 관련된 표준, 관례, 유형 안내 및 규제를 준수하는 소프트웨어 제품의 능력.

1.7.4. 효율성(Efficiency)

명시된 조건에서 사용되는 자원의 양에 따라 요구된 성능을 제공하는 소프트웨어 제품의 능력.

◎ 참고

- 자원은 다른 소프트웨어 제품, 하드웨어 장비, 재료(예, 인쇄 용지, 디스켓) 등을 포함한다.
- 사용자에 의해 운영되는 시스템에 대해서 기능성, 신뢰성, 운영성, 그리고 효율성 등의 복합체는 사용 품질에 의해 외부적으로 측정될 수 있다.

가. 시간 반응성(Time Behaviour)

명시된 조건에서 그 기능을 수행할 때 적절한 반응 및 처리시간과 처리율을 제공하는 소프트웨어 제품의 능력.

나. 자원활용성(Resource Utilization)

명시된 조건에서 소프트웨어가 그 기능을 수행할 때 적절한 양과 종류의 자원을 사용하는 소프트웨어 제품의 능력.

다. 준수성(Compliance)

효율성과 관련된 표준 및 관례를 준수하는 소프트웨어 제품의 능력.

1.7.5. 유지보수성(Maintainability)

소프트웨어 제품이 변경되는 능력. 변경에는 환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정, 개선, 혹은 개작(adaptation) 등이 포함된다.

가. 분석성(Analyzability)

소프트웨어의 결함이나 고장의 원인 혹은 변경될 부분들의 식별에 대한 진단을 가능하게 하는 소프트웨어 제품의 능력.

나. 변경성(Changeability)

변경명세가 구현될 수 있도록 하는 소프트웨어 제품의 능력.

◎ 참고

- 구현은 코딩, 설계, 문서화 등의 변경을 포함한다.
- 소프트웨어가 최종 사용자에게 의해 변경된다면 변경성은 운영성에 영향을 미칠 수 있다.

다. 안정성(Stability)

소프트웨어 변경으로 인한 예상치 않은 결과를 최소화하는 소프트웨어 능력.

라. 시험성(Testability)

변경된 소프트웨어가 확인될 수 있는 소프트웨어 제품의 능력.

마. 준수성(Compliance)

유지보수성과 관련된 표준 및 관례를 준수하는 소프트웨어 능력.

1.7.6. 이식성(Portability)

한 환경에서 다른 환경으로 전이될 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 환경은 조직, 하드웨어 혹은 소프트웨어 환경을 말한다.

가. 적응성(Adaptability)

소프트웨어에서 원래 목적으로 제공되는 것 이외의 다른 명세된 환경으로 변경될 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 적응성은 내부 용량(예, 화면 영역, 테이블, 트랜잭션 크기, 보고서 형식 등)의 확장성을 포함한다.
- 소프트웨어가 최종 사용자에게 의해 개작된다면 적응성은 ISO 9241-10에 정의된 바와 같이 개별화를 위한 적합성에 대응되며, 운영성에 영향을 줄 수 있다.

나. 설치성(Installability)

명세된 환경에 설치될 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 소프트웨어가 최종사용자에게 의해 설치된다면 설치성은 적합성과 운영성에 영향을 줄 수 있다.

다. 공존성(Co-existence)

공통자원을 공유하는 공동환경에서 다른 독립적인 소프트웨어와 공존할 수 있는 소프트웨어 제품의 능력.

라. 대체성(Replaceability)

동일한 환경에서 동일한 목적으로 다른 지정된 소프트웨어 제품을 대

신하여 사용될 수 있는 소프트웨어 제품의 능력.

◎ 참고

- 대체성은 사용자가 소프트웨어 제품을 새로운 버전으로 업그레이드할 때 중요하다.
- 대체성은 상호운영성과의 혼동을 피하기 위해 호환성을 대신하여 사용될 수 있다.
- 대체성은 설치성과 적합성 두 가지 모두의 속성을 포함한다.

마. 준수성(Compliance)

이식성과 관련된 표준 및 관례를 준수하는 소프트웨어 제품의 능력.

1.8. 사용 품질 특성

본 항에서는 본 표준의 사용 품질 특성(Quality in Use Characteristics)에 대하여 살펴본다. 특정 사용자에 의해 사용되는 제품이 지정된 사용과 관련하여 유효성, 생산성, 안전성, 만족도 등의 명시된 목표를 달성하기 위해 사용자 요구를 충족하는 정도.

◎ 참고

- 사용품질은 소프트웨어를 포함하고 있는 시스템의 품질을 사용자 관점에서 본 것이며, 소프트웨어 자체의 성질보다는 그 소프트웨어를 사용한 결과에 의해 측정된다.
- 사용 품질에 대한 메트릭의 예는 ISO/IEC 9126-2에 나와 있다.
- 사용성은 ISO 9241-11에서 ISO/IEC 9126의 이 부분에서 정의된 사용 품질과 유사하게 정의되어 있다. 사용 품질은 모든 품질 특성에 의해 영향을 받을 수 있으며, 따라서 ISO/IEC 9126-1에서 이해성, 학습성, 운영성 및 호감성의 관점에서 정의된 사용성보다 더 넓은 의미를 갖는다.

가. 유효성(Effectiveness)

명시된 조건하에서 사용자가 정해진 목표를 달성할 수 있게 해주는 소프트웨어 제품의 정확성과 완벽성 정도.

나. 생산성(Productivity)

소프트웨어 제품이 명시된 조건하에서 사용될 경우에 유효성과 관련하여 시스템이나 사용자가 소비하는 자원의 정도.

- ⊙ 참고 - 관련되는 자원에는 시간, 노력, 재료, 혹은 재정 비용 등이 있다.

다. 안전성(Safety)

소프트웨어 제품이 명시된 조건하에서 (사람에 대한)상해나 피해의 위험을 수용 가능한 수준으로 제한하는 정도.

- ⊙ 참고 - 이 정의는 ISO 8402에 기반 한다.

라. 만족도(Satisfaction)

지정된 조건하에서 소프트웨어 제품이 사용자를 만족시키는 정도.

- ⊙ 참고 - 만족도에 대한 신뢰성 있는 데이터를 수집하기 위해 설문지를 사용할 수 있다.

1.9. ISO/IEC 9126용어 정의

1) 구매자(Acquirer)

공급자로부터 시스템, 소프트웨어 제품, 또는 소프트웨어 서비스를 구매 혹은 조달하는 조직

2) 속성(Attribute)

한 개체의 측정 가능한 물리적 혹은 추상적 성질

- ⊙ 참고- 속성은 내부적 또는 외부적일 수 있다.

3) 개발자(Developer)

소프트웨어 생명 주기 동안 개발 활동(요구 분석, 설계, 시험에서 인수까지)을 수행하는 조직

4) 직접 측정(Direct Measure)

다른속성의 측정에 의존하지 않는 속성의 측정

5) 평가 모듈(Evaluation Module)

특정소프트웨어 품질 특성 및 부특성에 대한 평가 기술을 모아놓은 패키지

6) 외부 측정(External Measure)

해당제품을 일부분으로 하는 시스템의 행태를 측정함으로써 유추되는 간접적인 제품 측정

⊙ 참고

- 시스템에는 모든 관련된 하드웨어, 소프트웨어(주문형 소프트웨어 또는 기성형 소프트웨어) 및 사용자 등이 포함된다.
- 해당 프로그램을 실행하는 컴퓨터 시스템이 운영되는 동안에 고장 횟수를 누적하므로, 시험기간 동안 발견되는 고장 횟수는 그 프로그램에 존재하는 결함의 개수에 대한 외부 측정이 된다.
- 부 측정은 설계의 궁극적 목적에 보다 가까운 품질 속성을 평가하는데 사용되어질 수 있다.

7) 외부 품질(External Quality)

제품이 지정된 조건하에서 사용될 경우에, 명시된 요구와 내재된 요구를 충족하는 정도

8) 실패(Failure)

필요한 기능을 수행하는 동안 예상치 못한 기능 종료 혹은 미리 지정된 내에서의 수행 불능

9) 결함(Fault)

컴퓨터 프로그램 내부의 부정확한 절차, 처리 혹은 데이터 정의

◎ 참고- 이 정의는 IEEE 610.12-1990에서 인용하였다.

10) 내재된 요구(Implied Needs)

명시되지는 않았지만 해당 개체가 특정 조건에서 사용될 경우의 실질적인 요구

◎ 참고- 내재된 요구는 문서화되지 않을 수 있지만 실제로 필요한 요구이다.

11) 표시자(Indicator)

다른측정을 예상하거나 예측하는데 사용 가능한 측정

◎ 참고

- 예상된 측정은 동일한 또는 서로 다른 소프트웨어의 품질 특성에 대한 측정이 될 수 있다.
- 표시자는 소프트웨어 품질 속성과 생산 프로세스의 속성, 양쪽 모두를 예측하는 데 사용될 수 있다. 이들은 속성에 대한 간접 측정이다.

12) 간접 측정(Indirect Measure)

하나이상의 다른 속성으로부터 유추 가능한 속성 측정

◎ 참고

- 측정이 소프트웨어 속성뿐만 아니라 컴퓨팅 환경의 속성에 의해서도 영향을 받으므로, (사용자 입력에 대한 응답 시간과 같은) 컴퓨팅 시스템 속성에 대한 외부 측정은 소프트웨어 속성에 대한 간접 측정이 된다.

13) 중간 소프트웨어 제품(Intermediate Software Product)

소프트웨어 개발 프로세스에서 다른 단계의 입력으로 사용되는 소프트웨어 개발 프로세스중의 제품

◎ 참고- 어떤 경우에는 중간 제품이 또한 최종 제품이 될 수도 있다.

14) 내부 측정(Internal Measure)

제품자체에 대한 직접 혹은 간접 측정

◎ 참고

- 코드라인 수, 복잡도, 워크스루(walkthrough)에서 발견되는 결함의 개수 및 포그 지수(fog index) 등은 모두 제품 자체에 대한 내부 측정들이다.

15) 내부 품질(Internal Quality)

제품이 지정된 조건에서 사용될 경우에, 명시된 요구와 내재된 요구를 충족할 수 있는 능력을 결정하는 제품 속성의 총체

◎ 참고

- 본 표준에서 “외부 품질”과 대비되는 의미로 사용되는 “내부 품질”이라는 용어는 ISO 8402에서의 “품질”과 근본적으로 같은 의미이다.
- ISO/IEC 9126에서 “특성”이라는 용어는 보다 특정한 의미로 사용되므로, “속성”이라는 용어가 “특성”과 같은 의미로 사용되었다.

16) 유지보수자(Maintainer)

유지보수 활동을 수행하는 조직

17) 측정하다(Measure (동사))

측정을 수행하다.

18) 측정치(Measure (명사))

측정을 통해 개체의 속성에 배정되는 수치나 범주

19) 측정(Measurement)

개체의 속성에 등급 값(수치나 범주일 수 있음)을 배정하기 위한 메트

릭의 사용

◎ 참고

- 범주를 사용할 경우에, 측정은 정성적(qualitative)일 수 있다. 예를 들어, 원시 프로그램 언어(ADA, C, COBOL 등)와 같은 소프트웨어 제품의 일부 중요한 속성은 정량적인 범주이다.

20) 메트릭

정의된 측정 방법 및 측정 등급

◎ 참고

- 메트릭은 내부 혹은 외부, 또는 직접 혹은 간접적일 수 있다.
- 메트릭은 정성적인 자료의 분류 방법을 포함한다.

21) 품질(Quality)

명시된 요구와 내재된 요구를 충족할 수 있는 능력을 가진 개체 특성들의 총체

◎ 참고

- 계약의 경우나 혹은 핵 안전성 분야와 같은 통제된 환경에서는 요구가 명세되어야 하며, 반면에 다른 환경에서는 내재된 요구들을 구분하고 정의해야 한다 (ISO 8402:1994, 참고 1).
- ISO/IEC 14598에서 관련된 개체는 소프트웨어 제품이다.

22) 품질 평가(Quality Evaluation)

한 개체가 명세된 요구사항을 충족시킬 수 있는 정도에 대한 체계적인 조사

◎ 참고

- 요구사항은 제품이 계약 하에 어떤 특정 사용자를 위해 개발될 경우처럼, 공식적으로 명시될 수도 있고, 또는 소비자 소프트웨어와 같이 불특정 사용자들을 위해 개발될 경우처럼 개발 조직이 명세할 수도 있으며, 혹은 사용자가 비교나 선정을 목적으로 제품을 평가하는 경우처럼 그 요구사항이 보다 일반적일 수도 있다.

23) 품질 모델(Quality Model)

품질요구사항을 명세하고 품질을 평가하는 기준을 제공하는 특성 집합과 그들간의 상호관계

24) 사용 품질(Quality In Use)

특정사용자에 의해 사용되는 제품이 지정된 사용의 경우에 유효성, 작업 효율, 만족도 등과 관련하여 명시된 목표를 달성하기 위해 사용자 요구를 충족하는 정도

25) 판정(Rating)

측정된 값을 적절한 판정 등급으로 결정하는 활동. 특정 품질 특성을 갖는 소프트웨어와 관련된 판정 등급을 결정하기 위해 사용된다.

26) 판정 등급(Rating Level)

측정범주를 구분하기 위해 사용되는 순차적인 범주상의 지점

◎ 참고

- 판정등급은 명시되거나 내재된 요구에 따라 소프트웨어를 분류(판정)할 수 있게 한다.
- 적절한 판정 등급은 사용자, 관리자 및 개발자 등과 같이 품질을 보는 관점과 연관될 수 있다.

27) 범위(Scale)

정의된 성질을 갖는 값들의 집합

◎ 참고

- 범위에 대한 예는 다음과 같다.
- 일련의 구분에 해당되는 명목적 범위
- 순서화된 일련의 범위점(Scale Point)에 해당되는 순서적 범위
- 같은 간격의 범위점을 갖는 순서적 범위에 해당되는 간격 범위 (Interval Scale)
- 같은 간격의 범위점뿐만 아니라 절대값 '0'을 갖는 비율 범위

- 명목적 또는 순서적 범위를 사용하는 메트릭은 정성적 자료를 생성하고, 간격 범위와 비율 범위를 사용하는 메트릭은 정량적 자료를 생성한다.

28) 소프트웨어(Software)

정보처리 시스템의 모든 프로그램, 절차, 규칙 및 관련 문서 혹은 그 일부분

◎ 참고

- 소프트웨어는 기록되는 매체에 독립적인 지적 생산물이다.

29) 소프트웨어 제품(Software Product)

컴퓨터 프로그램, 절차, 또는 관련 문서 및 데이터의 집합

◎ 참고

- 제품에는 중간 제품과 개발자나 유지 보수자와 같은 사용자를 위해 만들어진 제품도 포함된다.

30) 공급자(Supplier)

계약조건하에 시스템, 소프트웨어 제품 또는 소프트웨어 서비스의 공급에 대해 구매자와 계약을 맺는 조직

31) 시스템(System)

명시된 요구나 목적을 충족시키는 능력을 제공하는 하나 이상의 프로세스, 하드웨어, 소프트웨어, 장비 및 사람으로 구성되는 결합체

32) 사용자(User)

특정기능을 수행하기 위해 소프트웨어 제품을 사용하는 개인

◎ 참고

- 사용자에는 운영 담당자, 소프트웨어 결과 수령자, 또는 소프트웨어 개발자 및 유지보수자 등이 포함될 수 있다.

33) 확인(Validation)

지정된 목적의 사용을 위해 특정 요구사항을 충족하는 객관적 증거에

대한 검사 및 규정에 의한 확인.

◎ 참고

- 설계 및 개발 단계에서 확인은 사용자 요구에 대한 부합성을 판단하기 위해 제품을 검사하는 프로세스를 말한다.
- 확인은 일반적으로 정의된 운영 조건하에서 최종 제품을 대상으로 수행된다. 확인은 그 이전 단계에서도 필요할 수 있다.
- "확인된(Validated)"이란 단어는 상응되는 상태를 나타내기 위해 사용된다.
- 사용 목적이 다양하다면 다중 확인이 수행될 수 있다.

34) 검증(Verification)

명세된 요구사항을 충족시키는 객관적 증거에 대한 조사 및 규정에 의한 확인

◎ 참고

- 설계 및 개발 과정에서 검증은 특정 활동의 결과가 그 활동에 대해 명시된 요구사항에 적합한가를 판단하기 위해 조사하는 프로세스를 말한다.
- "검증된(Verified)"이란 용어는 상응되는 상태를 가리키기 위해 사용된다.

2. ISO/IEC 12119

2.1. 적용 범위

본 항에서는 ISO/IEC12119의 적용 범위(Scope)에 대하여 살펴본다. 이 규격은 소프트웨어 패키지들에 적용할 수 있다. 예를 들면 문서 처리기, 스프레드시트, 데이터베이스 프로그램, 그래픽 패키지, 기술적 또는 과학적인 기능들을 위한 프로그램, 그리고 유틸리티 프로그램들에 적용될 수 있다.

이 규격은

- ⊙ 소프트웨어 패키지들에 대한 요구사항(품질 요구사항)
 - ⊙ 이러한 요구사항들에 대해 소프트웨어 패키지를 시험하는 방법에 관한 지침(시험을 위한 지침, 특히 제 3자 시험에 대한 지침)
- 을 수립하고 있다.

이 규격은 단지 제공되고 인도된 상태의 소프트웨어 패키지만을 다룬다. 이 규격은(각 활동과, 예를 들어 명세서와 같은 중간 산출물을 포함한) 소프트웨어 패키지의 생산공정을 취급하지 않는다. 공급자의 품질시스템(Quality System)은 이 규격의 범위 밖이다.

주 - 일부 소프트웨어(예, 안전성이 중요한 소프트웨어)는 추가적인 요구사항을 필요로 한다.

⊙ 규격의 사용자

이 규격의 사용자는 다음과 같은 사람들이 될 수 있다.

공급자가

- 소프트웨어 패키지에 대한 요구사항을 명세화 할 때

- 제품 설명 형식을 설계할 때
- 자신의 제품을 심사할 때
- 적합성을 선언할 때 [ISO/IEC Guide 22]
- 적합성 인증서나 마크를 신청할 때 [ISO/IEC Guide 23]
- 국제, 지역, 국내 인증기관에서 제3자 인증체계를 수립하려고 할 때 [ISO/IEC Guide 16, 28, 44]
- 시험 지침을 따라야 하는 시험소에서 적합성 인증서나 마크를 위한 시험을 할 때 [ISO/IEC Guide 25];
- 인정기관이 인증기관과 시험소를 인정할 때 [ISO/IEC Guide 40, 58];
- 시험소 심사원이 시험소의 능력을 심사할 때 [ISO/IEC Guide 58];

구매자가

- 자신들의 요구사항과 이 규격에서 기술된 요구사항을 비교할 때
- 의도하는 업무 요구와 기존 제품의 제품설명서의 정보를 비교할 때
- 인증된 제품을 찾을 때
- 요구사항이 충족될 때 다른 방법으로 체크할 때
- 보다 나은 제품에서 이익을 얻고자 하는 사용자.

2.2. 정의

이 규격에서는 다음과 같은 정의(Definition)가 사용된다. 참조하기 편하도록 다른 규격들에서 사용되고 있는 정의들은 부속서 A에 수록되어 있다.

1) 기능

사용자나 프로그램이 업무의 일부 또는 전부를 수행할 수 있도록 하는 프로그램내의 알고리즘 구현.

- ⊙ 참고1. 기능은 반드시 사용자에게 의해 호출될 수 있어야만 하는 것은 아니다 (예, 백업이나 저장의 자동화).
- ⊙ 참고 2. 여기 기능의 개념은 ISO 2382-14:1978(고장, 결점, 유지보

수 그리고 신뢰성의 정의)에서 사용된 것보다 더 좁은 의미이지만, ISO 2382-2와 ISO 2382-15에서 정의된 것보다는 더 넓은 의미를 갖는다.

2) 요구사항 문서

소프트웨어 패키지가 만족시켜야 할 권고사항이나 요구사항 또는 규제 사항을 포함하고 있는 문서.

- ◎ 참고- 그룹(시장 부문이나 기술협회 또는 사용자 협회), 법률 또는 공고에서 나온 기술 또는 인간공학 규격, 요구사항 목록(또는 모델 요구사항 명세서) 등이 그 예이다.

3) 제품설명서

소프트웨어 패키지의 속성을 설명하는 문서로서, 이의 주된 목적은 잠재적인 구매자가 제품 구입에 앞서 스스로 그 제품이 합당한지 (suitability) 평가할 수 있도록 도움을 주는 데 있다.

- ◎ 참고- 이 용어는 ISO/IEC 2382-20:1990의 "시스템 설명서"라는 용어보다는 더 자세하다. 제품설명서의 목적은 ISO 9127의 "표지 정보"의 제품설명서를 포함하고 있다. 제품설명서는 명세서가 아니라 다른 목적으로 사용된다.

4) 사용자문서

인쇄또는 비인쇄 형태의 사용 가능한 전체 문서들의 집합으로, 제품의 적용을 위해 제공되며, 또한 그 자체가 제품의 중요한 부분이다.

5) 패키지 문서

패키지 제품설명서 및 사용자문서.

6) 시험 사례

시험자를 위해 문서화된 지침으로서, 한 기능이나 여러 기능의 조합이 어떻게 시험되어야 하는가를 명세한다. 하나의 시험 사례는 다음과 같은

주제에 대해 세부적인 정보를 포함한다

- ⊙ 시험목적
- ⊙ 시험될 기능들
- ⊙ 시험 환경과 여타 조건들(구성의 상세 내용 및 예비 작업)
- ⊙ 시험 데이터
- ⊙ 절차
- ⊙ 시스템의 예측되는 행동

7) 유지보수

소프트웨어 패키지 수정과 관련된 시스템 유지보수의 한 부분.

2.3. 품질 요구사항

본 표준의 품질 요구사항(Quality Requirements)은 다음과 같다.

세부조항 (1)부터 (3)까지는

- ⊙ 각 소프트웨어 패키지가 제품설명서와 사용자문서를 가지고 있어야 한다는 요구사항
- ⊙ 제품설명서에 대한 요구사항. 특히 제품설명서가 명시된 정보를 포함시켜야 하며 각 문장은 시험가능하고 정확해야 한다는 요구사항이 있다
- ⊙ 사용자문서에 대한 요구사항
- ⊙ 소프트웨어 패키지 내에 포함된 프로그램이나 데이터가 있을 경우에 대한 요구사항
 - 참고1 사용자문서, 프로그램과 데이터에 대한 요구사항은 많은 일반적 요구사항(제품설명서에서 약속하고 있는 것과는 다른 것)을 담고 있으나 사용자가 바랄 수 있는 모든 속성을 포함하지는 않는다.
 - 참고 2 특정 속성, 예를 들어 사용자문서와 프로그램 메시지의 "이해가

능성"과 "개괄용의성" 등은 명백히 사용자관점에서 요구된다. 그러나 명쾌하고 반복적인 결과를 얻을 수 있는 시험이 어렵기 때문에, 이러한 속성들은 현재로서는 단지 권고로서 규정될 뿐이다.

- 참고 3 세부조항 (1)부터 (3)까지는 ISO/IEC 9126에 나오는 특성과 같은 순서로 배열되어 있다.

어떤 소프트웨어 패키지가 (1)부터 (3)까지의 모든 요구사항에 따른다면, 이는 이 규격에 적합한 것이다. 권고("--하는 것이 바람직하다"는 단어 형식을 써서 지시하고 있는)들은 선택 사항이다.

- 참고4 - 어떤 제품이 (1)부터 (3)까지의 요구사항에 제품이 적합한지 여부를 증명하기는 매우 어렵거나 불가능할 수도 있다.

2.3.1. 제품 설명서

각 소프트웨어 패키지는 제품설명서를 갖고 있어야 한다.

제품설명서는 제품을 정의한다. 이는 제품의 패키지 문서화의 일부분이다. 이는 사용자문서와 프로그램, 그리고 데이터에 대해 정보가 있을 경우 이를 제공한다.

제품설명서의 주 목적은 첫째, 사용자나 잠재적 구매자가 그 제품이 합당한지를 스스로 평가할 수 있도록 하기 위함이다. 여기까지는 판매 정보이기도 하다. 둘째, 시험을 위한 토대로 사용하기 위함이다. 셋째, 제품설명서는 그 제품에 관심을 갖는 사람들에게 이용 가능해야 한다.

가. 내용에 관한 일반적인 요구사항

제품설명서는 잠재적 구매자가 제품을 구입하기 앞서 스스로 이의 합당함을 평가할 수 있도록 충분히 이해할 수 있고, 완전하며 개괄하기 쉬운 것이 바람직하다.

제품설명서는 내부 불일치성이 없어야 한다. 용어는 모든 곳에서 동일한 의미를 지녀야 한다.

제품설명서의 문장은 시험 가능하고 정확해야 한다.

참고- 이 요구사항은 만일 제기된 외적인 요구사항 문서들이 있다면 그것의 설명문까지 해당된다.

다음의 세부조항 2)부터 8)까지는 제품설명서에 꼭 포함해야 하거나 포함하도록 권고되는 것을 명시한다. 여기에는 제품에 대한 추가적인 문장을 포함할 수 있다.

나. 식별과 지시

1) 제품설명서의 식별

제품설명서는 유일한 문서 식별을 포함해야 한다. 제품설명서는 예를 들면 "기능 설명서", "제품 정보", "제품서" 와 같이 "제품설명서"와 다르게 불릴 수 있다.

2) 제품의 식별

제품설명서는 제품을 식별해야 한다. 제품 식별은 최소한 제품 이름과 버전 또는 날짜를 갖고 있어야 한다.

3) 공급자

제품설명서는 최소한 한 공급자의 이름과 주소를 담고 있어야 한다.

주 - 이름과 주소는 반드시 인쇄될 필요는 없다 딜러의 고무도장으로도 충분하다.

4) 업무 (work task)

제품설명서는 제품을 가지고 수행될 수 있는 의도된 작업을 식별해야 한다.

5) 요구사항 문서에 적합성

제품설명서는 제품이 따르고 있는 요구사항 문서들을 참고할 수 있는데 이 경우 관련된 간행물(editions)은 식별되어야 한다.

6) 요구되는 시스템

제품을 사용할 때 요구되는 시스템(하드웨어, 소프트웨어 그리고 이의 구성)은 제조업자 이름과 모든 구성부분의 형태 식별자를 포함하여 명시되어야 한다. 예를 들면 다음과 같다.

- ⊙ 코프로세서를 포함하는 프로세스 유닛
- ⊙ 주 메모리 크기
- ⊙ 주변(보조) 저장의 유형과 크기
- ⊙ 확장 카드
- ⊙ 입출력 장비
- ⊙ 네트워크 환경
- ⊙ 시스템 소프트웨어와 기타 소프트웨어

예를들면, 다른 업무, 다른 범위값 또는 다른 효율성 요구사항을 위해 다른 요구되는 시스템이 명시될 수 있다.

이전에 하드웨어나 소프트웨어 제품이 식별되어 있다면, "(또는, 상호운용이 가능하다면, 또 다른)" 이라는 문장이 제품설명서에 나타날 수 있다. 과거에 제품의 버전이 식별되어 있다면 "또는 상호 운영 가능한 갱신된 버전"이라는 설명이 들어있을 수 있다. "버전 X부터 최소한 버전Y 까지"라는 설명이 있을 수 있지만, "버전 X부터"라는 설명은 있어서는 안 된다.

주 - "버전 X부터"라는 설명은 소프트웨어 패키지가 운영에 실패할 수 있는 버전 X+3의 등장에 의하여 후에 잘못된 것으로 될 수 있다.

7) 다른 제품과의 인터페이스

만일제품설명서가 다른 제품을 참조한다면, 그 인터페이스 또는 제품이 식별되어야 한다.

8) 인도될 항목

제공되는 제품의 모든 물리적 구성 부분, 특히 모든 인쇄된 문서와 모든 데이터 미디어는 식별되어야 한다.

예를 들면, 원시 프로그램, 목적 모듈, 또는 실행 모듈과 같이 공급되는 프로그램의 형태가 설명되어야 한다.

주 - 미디어 포맷(예를 들어 디스켓 포맷)을 표시할 필요는 없다. 요구되는 시스템에 의해 가능한 포맷이 결정되기 때문이다.

9) 설치

제품설명서에는 제품의 설치가 사용자에게 의해 수행될 수 있는지 그 여부가 언급되어야 한다.

10) 지원

제품설명서에는 제품 운영을 위한 지원이 제공되고 있는지 언급되어야 한다.

11) 유지보수

제품설명서에는 유지보수의 제공 여부가 언급되어야 한다. 유지보수가 제공된다면, 무엇이 정확히 포함되는지를 언급해야 한다.

다. 기능성에 대한 설명

1) 기능의 개괄

제품설명서는 제품이 갖고 있는 사용자가 호출 가능한 기능, 필요한 데이터, 그리고 제공되는 설비에 대한 전반적인 개괄을 제공해야 한다.

각 언급된 기능(특히 선택사항이나 변형)에 대해 그것이

- ⊙ 제품의 부분인지
- ⊙ 제품설명서에서 완전히 설명된 제품 확장 부분인지
- ⊙ 제품설명서에서 참조된 제품 확장의 부분인지
- ⊙ 보증이 안된 추가(보충) 부분인지

명백히 설명되어야 한다.

참고 - 사용자가 호출 가능한 모든 기능이 언급될 필요는 없고, 모든 기능에 대해 그것이 어떻게 호출되는지에 대해 상세한 내용을 제시할 필요는 없다.

2) 범위값

만일제품 사용이 제품의 특정 범위값에 의하여 제한 받는다면 이러한 값들은 제공되어야 한다. 예를 들면 다음과 같다.

- ⊙ 최소또는 최대 값
- ⊙ 키의 길이
- ⊙ 파일 내 레코드의 최대 수
- ⊙ 탐색 기준의 최대 수
- ⊙ 최소 샘플 크기

고정된 범위값을 제공할 수 없는 경우(예를 들어, 이 값들이 어플리케이션(application)의 형태나 입력 데이터에 의존할 때)에는 그 한계가 언급되어야 한다. 허용될 수 있는 값의 조합이 제공될 수 있으며 사용자문서 내에 보다 상세한 정보가 참조될 수 있어야 한다.

3) 보안

제품설명서는 우연이든 의도적이든 프로그램과 데이터에 권한이 없는 접근을 방어할 수 있는 수단에 대한 정보를 포함하는 것이 바람직하다.

라. 신뢰성에 대한 설명

제품설명서는 데이터 저장 절차에 대한 정보를 포함해야 한다.

참고 - 예를 들어, 운영체제에 의해 백업이 가능하다고 언급하는 것으로 충분하다.

제품의 기능적 능력을 보장하는 추가적인 제품 속성이 설명되는 것이 바람직하다. 예를 들면, 다음과 같은 것이 있다.

- ⊙ 입력이 적절한지 여부의 체크
- ⊙ 사용자 잘못으로 인해 빗어지는 중대한 결과의 방어
- ⊙ 에러 복구

마. 사용성에 대한 설명

1) 사용자 인터페이스

명령라인, 메뉴, 윈도우, 기능 키, 도움말 기능 같이 사용자 인터페이스 유형은 이름이 붙여져야 한다.

2) 요구되는 지식

제품을 적용하는 데 요구되는 특정 지식은 명시되어야 한다. 예를 들면 다음과 같은 것이 있다.

- ⊙ 기술적 영역에 대한 지식
- ⊙ 운영체제에 대한 지식
- ⊙ 특정 훈련에 의해 획득될 수 있는 지식
- ⊙ 제품설명서에 쓰여진 것 이외의 언어에 대한 지식(에러 메시지와 비주얼 데이터를 포함하여) 사용자문서 및 사용자 인터페이스에서 사용된 언어는 소프트웨어 패키지 자체와 제품설명서에서 참조한 다른 제품에서 사용되어야 한다.
 - 참고 - 이 요구사항은 ISO/IEC 9127:1988, 6.1.7의 요구사항을 넘어서는 것이다. 그 규격에서는 언어에 관한 언급은 선택적 사항이다.

⊙ 사용자 필요에 따른 개조(adaptation)

만일 제품이 사용자에게 의해 개조될 수 있다면, 개조를 위한 도구와 그 사용에 대한 조건이 식별되어야 한다. 예를 들면 다음과 같다.

- 매개변수의 변경
- 연산 알고리즘의 변경
- 기능 키에 대한 할당

3) 저작권 침해에 대한 방지

저작권 침해에 대한 기술적인 보호가 사용성을 해칠 수 있다면, 이 보호 조치는 언급되어야 한다. 예를 들면 다음과 같다.

- ⊙ 복사에 대한 기술적 보호
- ⊙ 프로그램된 사용 만료 기간
- ⊙ 복사(copy)에 대해 지불을 하라는 대화식 경고(reminder)

4) 사용 효율성 및 사용자 만족

제품설명서는 사용 효율성 및 사용자 만족에 관한 데이터를 포함할 수 있다.

주 - 그와 같은 데이터는 ISO 9241-11의 지침을 따를 수 있다.

바. 효율성에 대한 설명

제품설명서는 명시된 조건(예를 들어 시스템 구성과 로드 프로파일) 하에서 주어진 기능의 반응시간 및 처리율 등과 같은 제품의 시간적 행위에 대한 데이터를 포함할 수 있다.

사. 유지보수성에 대한 설명

제품설명서는 유지보수성에 관한 설명을 포함할 수 있다.

아. 이식성에 대한 설명

제품설명서는 이식성에 관한 설명을 포함할 수 있다.

1) 사용자문서

◎ 완전성

사용자문서는 제품 사용에 필요한 정보를 담고 있어야 한다. 제품설명서에서 언급한 모든 기능과 프로그램 내에서 사용자가 호출 가능한 모든 기능은 사용자문서에서 완전하게 기술되어 있어야 한다.

제품설명서에서 주어진 모든 범위값은 사용자문서에서도 반복되어야 한다.

만일 설치가 사용자에게 의해 수행될 수 있다면, 사용자문서는 필요한 모든 정보를 담고 있는 설치 매뉴얼을 포함해야 한다. 설치 매뉴얼은 설치시 파일의 최소/최대 크기를 말 해 주는 것이 바람직하다.

만일 유지보수가 사용자에게 의해 수행될 수 있다면, 사용자문서는 관련된 유지보수 유형에 대해 필요한 모든 정보를 담고 있는 프로그램 유지보수 매뉴얼을 포함해야 한다.

◎ 정확성

사용자문서내의 모든 정보는 정확해야 한다. 또한 사용자문서의 표현은 모호하지 않고 애러가 없어야 한다.

◎ 일관성

사용자문서의 문서들은 스스로, 상호간, 또한 제품설명서와 불일치하지 않아야 한다. 각 용어는 모든 곳에서 같은 의미를 지니는 것이 바람직하다.

◎ 이해성

사용자문서는 언급된 업무를 일상적으로 수행하는 사용자들에게 이해될 수 있어야 바람직하다. 예를 들어, 적절한 용어 선택, 그래픽 형태 제시,

상세한 설명 사용, 유용한 정보를 참조함으로써 가능하다.

◎ 개괄용의성

사용자문서는 관계성을 쉽게 알 수 있도록 개괄하기 쉬운 것이 바람직하다.

모든 문서는 목차와 색인을 갖추는 것이 바람직하다.

만일 어떤 문서가 인쇄된 형태로 제공되지 않는다면, 이를 인쇄할 수 있는 절차가 설명되어야 바람직하다.

2) 프로그램 및 데이터

◎ 기능성

▪ 설치

만일설치가 사용자에게 의해 수행될 수 있다면, 프로그램을 설치 메뉴얼에 있는 정보를 따름으로써 성공적으로 프로그램을 설치할 수 있다. 제품 설명서에 제시된 요구되는 사항은 시스템의 각각에 프로그램을 설치하기에 충분해야 한다.

설치에 이어, 제공된 시험 사례를 사용하거나 대응되는 메시지로 자체 시험을 하는 방식으로 프로그램이 제대로 기능할 수 있는지의 여부를 알 수 있어야 한다.

▪ 기능의 존재

사용자문서에 언급된 모든 기능은 관련 설비, 속성 및 데이터와 함께, 그리고 주어진 범위 값 내에서, 사용자문서에서 주어진 형태로 실행될 수 있어야 한다.

참고 - 제품설명서에서 언급된 모든 기능은 사용자 문서에서도 있어야 하기 때문에, 그 기능들이 실행될 수 있어야 한다는 것은 당연하다.

- 정확성

프로그램과 데이터는 제품설명서 및 사용자문서에 있는 모든 문장에 대응되어야 한다. 기능들은 업무에 정확히 실행되어야 한다. 특히, 프로그램과 데이터는 제품설명서에 의해 참조된 요구사항문서 내의 모든 요구사항에 따라야만 한다.

- 일관성

프로그램과 데이터는 그 자체내에, 그리고 제품설명서 및 사용자 문서와 불일치가 없어야 한다. 각각의 용어는 모든 곳에서 동일한 의미를 지니는 것이 바람직하다.

프로그램 운영 통제와 사용자에 의한 프로그램 행위(예를 들어 메시지, 입력 스크린 포맷, 인쇄 된 보고서)는 균일하게 구조화되는 것이 바람직하다.

- 신뢰성

하드웨어, 요구되는 소프트웨어와 그 제품에 속하는 프로그램으로 이루어진 시스템은 사용자가 이를 통제할 수 없는 상태에 빠지지 않아야 하며, 데이터의 훼손이나 손실이 일어나지 않아야 한다.

이 요구는 다음과 같은 경우에도 충족되어야 한다.

- 처리능력이 명시된 한계까지 올라간 경우
- 명시된 한계 이상으로 처리 능력을 끌어올리려는 시도가 가해지는 경우
- 사용자에 의해서 또는 제품설명서에 나열된 프로그램의 다른 곳으로부터 잘못된 입력이 주어진 경우
- 사용자문서화 내의 명확한 지침을 어긴 경우

어떠한 프로그램도 감지 할 수 없는 하드웨어와 운영체제 인터럽션(예를 들어 시스템 운영 리셋을 위한 키나 키의 조합)의 가능성 만은 제외한다.

프로그램은 입력 구문 조건을 어기는 것을 인식해야 한다. 프로그램이 입력에 대하여 에러 또는 정의되지 않은 것으로 인식하는 경우, 프로그램은 이것을 허용 가능한 입력으로서 처리해서는 안 된다.

◎ 사용성

사용성에 관해서는, 이 규격에 근거해 합의하는 당사자는 ISO 9241 시리즈의 가장 최신판을 적용할 가능성을 조사할 것을 권고한다.

참고 - 특히 ISO 9241 시리즈의 part 10과 13을 고려하는 것이 좋다.

▪ 이해가능성

프로그램의 질문, 메시지 그리고 결과는 이해할 수 있어야 바람직하다. 예를 들면 다음과 같다.

- 용어의 적절한 선택에 의해
- 그래픽 제시에 의해
- 배경 정보의 제공에 의해
- 도움말 기능의 설명에 의해

에러메시지는 (예를 들어 사용자문서에 있는 항목을 참조함으로써) 사용 에러에 상응하는 원인의 설명 또는 시정 방법을 설명하는 상세한 정보를 제공해야 한다.

▪ 개괄용의성

각각의 데이터 매체는 제품 식별을 지녀야 하며, 만일 매체가 하나 이

상일 때에는 구별되는 숫자나 문장이 있어야 한다.

사용자를 위해 프로그램으로 작업을 할 때 어떤 기능이 실행되고 있는지를 항상 알 수 있어야 한다.

프로그램은 쉽게 눈으로 볼 수 있거나 읽기 쉬운 형태로 정보를 사용자에게 제공하는 것이 바람직하다. 사용자는 정보의 적절한 코딩 및 그룹화에 의해 안내 받는 것이 바람직하다. 필요하다면, 프로그램은 사용자에게 경고를 하는 것이 바람직하다.

프로그램으로부터의 메시지는 사용자가 유형에 의해 쉽게 구별할 수 있도록 설계되는 것이 바람직하다. 예를 들면 다음과 같다.

- 승인(acknowledgement)
- 프로그램으로부터의 질의
- 경고
- 에러 메시지

입력 스크린 포맷, 보고서, 기타 입출력은 명백하고 개괄하기 쉽게 설계되는 것이 바람직하다. 이를 위해서는 다음과 같은 것이 있다.

- 문자 필드는 왼쪽 정렬된다.
- 숫자 필드는 오른쪽 정렬된다.
- 테이블 내의 소수점이나 콤마는 같은 수직선 상에 정렬된다.
- 필드 제한을 알 수 있어야 한다.
- 필드의 사용은 필수적이며, 따라서 그렇게 알 수 있어야 한다.
- 입력 실패는 식별된 입력 스크린 포맷 내에서 강조되어야 한다.

- 시각적이거나 청각적인 표시로써 사용자의 관심을 스크린의 내용 변화로 이끌어야 한다.

- 운영가능성

심각한 결과를 초래하는 기능의 실행은 취소 가능하거나, 프로그램이 그 명령을 실행하기 전에 결과에 대해 분명한 경고를 하고 확인을 요청하도록 해야 한다. 특히, 데이터 삭제 및 덮어쓰기, 장기적인 프로세스 수행을 인터럽트 하는 경우는 심각한 결과를 빚게 된다.

만일 문서화하고 있는 문장이 대화에 제시되면, 예를 들어 전개된 목차 테이블을 선택하거나 키워드에 기반한 탐색 기능에 의해서, 사용자가 직접 그 문장의 하위항목을 접근할 수 있는 것이 바람직하다.

- ◎ 효율성

아무것도 요구되는 것은 없다. 그러나 제품설명서에서 효율성에 대해 서술하였다면 이를 따라야 한다.

- ◎ 유지보수성

아무것도 요구되는 것은 없다. 그러나 제품설명서에서 유지보수성에 대해 서술하였다면 이를 따라야 한다.

- ◎ 이식성

아무것도 요구되는 것은 없다. 그러나 제품설명서에서 이식성에 대해 서술하였다면 이를 따라야 한다.

2.4. 시험을 위한 지침

본 항에서는 시험을 위한 지침(Instructions for Testing)에 대하여 살

펴본다. (1)부터 (5)까지의 지침(instruction)은 품질 요구사항에 대해 제품이 어떻게 시험되어야 하는가를 명시하고 있다. 이 지침에는 모든 적합한 제품에서 요구되는 특성에 대한 시험과 제품설명서에서 약속한 특성에 대한 시험을 포함한다. 이 지침에는 문서 검사에 의한 시험과 프로그램 및 데이터의 블랙박스 시험을 포함한다.

이 지침들은 기능 시험(블랙박스 시험)을 기술한다. 구조적 시험은 원시 코드를 이용해야 하기 때문에 포함되지 않는다.

제품이 필요한 시스템 내의 제품만이 시험된다. 컴퓨터 작업장에서의 인간공학적 평가는 이 규격에서는 고려되지 않는다.

참고1. 이 지침들은 인증 체계에 따르는 제 3자 시험을 일차적인 목적으로 한다. 생산하는 동안에는 구조 시험을 하는 것이 더 경제적이고 효과적일 것이다.

참고 2. 조항 4는 소프트웨어 패키지에 대한 요구사항을 담고 있지 않다. 어떤 소프트웨어 패키지는 조항 4에 따른 시험을 받지 않고서도 인정된다. 그러므로 그 시험은 비적합성의 존재를 발견하지 못할 수도 있다.

참고 3. 요구되는 시스템이 제품설명서에 의해 결정되기 때문에, 요구되는 시스템과 제품의 어떠한 비적합성도 제품의 비적합성으로 취급된다.

참고 4. 인증 체계는 선택적인 권고사항에 대한 시험을 실시할 수 있다.

참고 5. 인간공학적 평가에 대한 지침은 ISO 9241-11에 있다.

2.4.1. 선행조건 시험

가. 제품 항목의 존재

소프트웨어 패키지를 시험하기 위해서는 제품설명서에서 식별된 요구

사항 문서들과 아울러 인도된 모든 항목이 존재해야 한다.

나. 시스템 구성물의 존재

소프트웨어 패키지를 시험하기 위해서는 제품설명서에서 명명된 모든 컴퓨터 시스템의 구성 부분이 이용될 수 있어야 한다.

다. 훈련

만일 제품설명서에 훈련이 언급되어 있으면, 시험자는 훈련 자료와 훈련 프로그램을 이용해야 한다.

2.4.2. 시험 활동

제품설명서, 사용자문서, 소프트웨어 패키지의 부분으로서 인도되는 프로그램과 어떠한 데이터도

- ⊙ 조항 4의 요구사항에 따라서 시험되는 것이 바람직하다.
- ⊙ 조항 4의 권고에 따라서 시험되는 것이 바람직하다.
- ⊙ 시험대상은 조항 4의 요구사항으로부터 도출되며 그 모든 것을 포함한다(완벽성, 일관성 등).
- ⊙ 만일 제품설명서에 다른 제품이 언급되어 있다면, 이 다른 제품은 시험 대상 제품의 제품설명서의 내용만을 위해서 시험될 필요가 있다.
- ⊙ 제품설명서, 사용자문서, 제품의 기능과 데이터에 있는 세부사항은 다음과 같은 경우 시험자의 판단에 따라 반드시 시험될 필요는 없을 수 있다.
 - 규정된 업무의 적절성에 무시할 정도로 작은 영향을 미치거나
 - 원칙적으로는 시험될 수 있지만 비용에서 타당하지 못할 때
 - 시험되지 않은 세부사항은 시험 기록과 시험 보고서에 언급되어야 한다. 이것들을 시험하지 않은 이유를 시험 기록에 문서로 남겨야 한다.

가. 제품설명서

조항 4의 요구사항을 충족하고 있는가 시험되어야 하며 조항 4의 권고를 충족하고 있는가도 시험되는 것이 바람직하다.

나. 사용자문서

조항 4에 있는 요구사항을 충족하고 있는가 시험되어야 하며 조항 4의 권고를 충족하고 있는가도 시험되는 것이 바람직하다.

다. 프로그램과 데이터

조항 4에 있는 요구사항을 충족하고 있는가 시험되어야 하며 조항 4의 권고를 충족하고 있는가도 시험되는 것이 바람직하다.

제품설명서에 명명된 모든 컴퓨터 시스템에서 프로그램은 시험되어야 한다.

만일 몇 개의 프로그램 변형(상이)이 있다면, 그 각각은 시험되어야 한다. 제품설명서와 사용자 문서에 따라, 여러 변형(상이)에서 동일한 기능은 하나의 변형에서 각각 시험될 수도 있다.

프로그램 및 이와 함께 공급되는 데이터는 제품설명서 사용자 문서에 근거하여 구성된 시험 사례를 이용하여 시험되어야 한다. 추가적인 자료(예를 들어 원시 코드)는 제품 설명서와 사용자 문서의 주장을 테스트함에 있어서 필요하지 않는 한 굳이 고려될 필요는 없다.

테스트 사례는 방법론적으로 그리고 체계적으로 구성되어야 한다.

만일 사용자문서에서 예가 주어진다면 테스트 사례로서 이용되어야 하지만, 테스트는 이 예들에 한정되어서는 안된다.

소프트웨어 패키지의 공급자에 의해서 제공된 테스트 사례는 이용될

수 있지만, 테스트는 이 사례들에 한정되어서는 안된다.

1) 설치

만일제품설명서에 따라 사용자에게 의해 설치가 수행될 수 있다면, 설치 매뉴얼에 서술되어있는 대로 프로그램이 설치될 수 있고 또 성공적으로 설치되는가 테스트되어야 한다.

그렇지 않으면 설치된 프로그램의 하드웨어와 소프트웨어 환경이 고려 중인 컴퓨터 시스템을 위한 제품설명서의 서술과 상응하는지 확인되어야 한다.

2) 프로그램 실행

테스트 사례는 제품설명서와 사용자문서에서 기술된 모든 기능을 포괄해야 한다. 그리고 업무를 위해 대표되는 기능의 조합을 고려해야 한다.

프로그램은 범위값(제품설명서와 사용자문서에 따라)이 적용되는 요구 시스템에서 모든 범위값에 대해 테스트되어야 한다.

사용자문서에서 금지한다고 명백하게 반대하고 있거나 선언하고 있는 입력과 명령 절차는 테스트에서 사용되어야 한다.

2.4.3. 테스트 기록

각 테스트에 대한 기록은 반복적인 테스트를 허용할 수 있도록 다음과 같은 충분한 정보를 갖추어야 한다[ISO/IEC Guide 25].

- ⊙ 테스트 계획이나 테스트 사례를 포함한 테스트 명세
- ⊙ 테스트하는 동안 발생한 고장을 포함하여 테스트 사례와 관련된 모든 결과
- ⊙ 테스트에 참가한 사람들의 식별

2.4.4. 테스트 보고서

테스트의 대상과 결과는 (테스트 기록에 기록된 데로) 테스트 보고서에 요약되어야 한다. 테스트 보고서는 다음과 같은 구조를 갖춰야 한다.

- ⊙ 제품식별
- ⊙ 테스트에 이용된 컴퓨터 시스템(하드웨어, 소프트웨어 그리고 그 구성)
- ⊙ 사용된 문서(식별도 함께)
- ⊙ 제품설명서, 사용자문서, 프로그램과 데이터의 테스트의 결과
- ⊙ 요구와 일치하지 않는 것의 목록
- ⊙ 권고와 일치하지 않는 것의 목록 또는 준수되지 않은 권고의 목록, 또는 제품이 권고를 테스트하지 않았다는 설명문
- ⊙ 테스트가 완료된 날짜
- ⊙ 테스트 보고서의 결과는 각 제목에 상응하는 설명을 담고 있어야 한다.
- ⊙ 제품을 권고에 대한 적합성 테스트를 하지 않았다는 설명과 더불어, 테스트 보고서는 권고에 부적합한 목록도 제공할 수 있다.
- ⊙ 테스트 보고서의 식별(테스트한 연구소, 제품 식별, 테스트보고서의 데이터)과 보고서 페이지의 총 숫자가 테스트보고서의 각 페이지에 존재해야 한다.
- ⊙ 테스트 보고서는 테스트한 연구소의 문서화된 동의가 없이는 그 전체가 복제(재생)될 수 없다는 설명 [ISO/IEC Guide 25]

- ⊙ 테스트 보고서는 ISO/IEC Guide 25의 테스트 보고서를 위한 규정을 따르는 것이 바람직하다.

2.4.5. 추후 테스트

이미 테스트를 받은 제품이 다시 테스트될 때에는, (이전의 테스트를

고려하여) 문서 그리고 기능과 데이터의 변경된 부분 모두는 새로운 제품 처럼 테스트되어야 한다.

변경된 부분에 의해서 또는 요구되는 시스템의 변화에 의해서 (시험자의 전문적 지식에 의하면) 영향을 받을 것으로 예견되는 변경되지 않은 부분은 새로운 제품처럼 테스트되어야 한다.

여타 모든 부분은 최소한 샘플로 테스트되어야 한다.

2.5. ISO/IEC 12119 용어 정의

2.5.1. 부속서 A: 다른 규격에서 정의된 용어

이 규격에서 사용된 일부 용어는 참조를 쉽게 하기 위해서 다른 규격에서 정의되어 있는 것을 인용하고 있다. 공표 당시, 지시된판들은 유효하였고 이로부터 인용된 정의는 사용되고 있었거나 고려되고 있었다.

가. A.1 일반 용어

- 1) A.1.1 소프트웨어: 정보처리 시스템의 프로그램, 절차, 규정 그리고 관련된 문서. [ISO/IEC 2382-1:1993]
- 2) A.1.2 소프트웨어 패키지: 일반적인(범용) 애플리케이션 또는 기능을 위해서 몇몇 사용자에게 제공된 완비되고 문서화된 프로그램 집합. [ISO/IEC 2382-20:1990]
- 3) A.1.3 시스템 소프트웨어: 애플리케이션 소프트웨어의 실행을 지원하는 애플리케이션에 독립적인 소프트웨어. [ISO/IEC 2382-20:1990]
- 4) A.1.4 유틸리티 루틴, 유틸리티 프로그램: 컴퓨터사용자와 서비스 직원에게 일반적이고 빈번하게 요구되는 루틴(컴퓨터 프로그램).

[ISO/IEC 2382-7:1989]

- 5) A.1.5 기능 단위: 규정된 목적을 수행할 수 있는 하드웨어나 소프트웨어, 또는 이 둘의 개체. [ISO/IEC 2382-1:1993]
- 6) A.1.6 (컴퓨터) 프로그램: 특정 프로그래밍 언어의 규칙에 따르고 어떤 기능, 과업, 또는 문제를 해결하는 데 필요한 선언문과 설명문 또는 지침으로 구성된 구문적 단위. [ISO/IEC 2382-1:1993]
- 7) A.1.7 인터페이스: 기능적 특성, 공통의 물리적 상호연결 특성, 신호 특성, 그리고 여타 다른 특성에 의해서 정의된 두 기능 단위 사이의 공유된 범위. [ISO/IEC 2382-9:1984]
- 8) A.1.8 사용자 인터페이스: 사람인 사용자와 컴퓨터 시스템의 하드웨어나 소프트웨어 구성물 사이에 정보가 전달될 수 있게 하는 인터페이스. [ANSI/IEEE Std 610.12-1990]
- 9) A.1.9 구성: 정보처리 시스템의 하드웨어와 소프트웨어가 조직되고 상호 연결되는 방식. [ISO/IEC 2382-1:1993]

나. A.2 제품의 특성

- 1) A.2.1 기능성: 일련의 기능 존재와 이들의 명세된 특성과 관련된 일련의 속성들의 집합. 명시적 또는 묵시적 필요를 만족하는 것이 기능이다. [ISO/IEC 9126:1991]
- 2) A.2.2 신뢰성: 명시된 기간 동안 명시된 조건에서 소프트웨어의 성능 수준을 유지하는 능력과 관련된 속성들의 집합. [ISO/IEC 9126:1991]
- 3) A.2.3 사용성: 사용자(실제 사용자나 묵시적인 사용자)가 사용을 위해 요구하는 노력과 그러한 사용에 대한 개개인의 판단과 관련된 속성들의 집합. [ISO/IEC 9126:1991]
- 4) A.2.4 효율성: 명시된 조건하에서 소프트웨어 성능 수준과 사용된 자원의 양 사이에 관계된 속성들의 집합. [ISO/IEC 9126:1991]
- 5) A.2.5 유지보수성: 규정된 수정을 수행하기 위하여 필요한 노력과

관련된 속성들의 집합. [ISO/IEC 9126:1991]

- 6) A.2.6 이식성: 소프트웨어가 다른 환경으로 이전되는 능력과 관련된 속성들의 집합. [ISO/IEC 9126:1991]

다. A.3 데이터

- 1) A.3.1 데이터: 의사소통, 해석, 또는 처리에 적합하도록 형식화된 방식으로 정보를 재해석할 수 있는 표현. [ISO/IEC 2382-1:1993]
- 2) A.3.2 데이터 매체: 데이터가 내부 또는 표면에 기록될 수 있고 그로부터 데이터가 복원될 수 있는 물질. [ISO/IEC 2382-1:1993]

라. A.4 테스트

- 1) A.4.1 테스트: 규정된 절차에 따라 주어진 제품, 공정, 또는 서비스의 한 개 이상의 특성을 결정하는 기술적인 작업(operation). [ISO/IEC Guide 2:1991]
- 2) A.4.2 테스트 데이터: 점검 문제를 위해 사용되는 데이터. [ISO 2382-8:1986]
- 3) A.4.3 점검 문제: 기능 단위가 올바르게 작동되는 판단하기 위하여 사용되는 문제로 알려진 해결책을 가지고 있는 문제. [ISO 2382-8:1986]
- 4) A.4.4 테스트 방법: 테스트를 수행하기 위한 규정된 기술적인 절차. [ISO/IEC Guide 2:1991]
- 5) A.4.5 테스트 계획, 시스템 테스트, 평가 계획: 시스템의 테스트와 평가를 위한 세부 요구사항, 기준, 일반적 방법론, 책임, 일반 계획을 수립한 계획. [ISO 2382-20:1990]
- 6) A.4.6 테스트 보고서: 테스트 결과 및 테스트와 관련된 관련 정보를 포함한 문서. [ISO 2382-20:1990]

마. A.5 기타 용어

- 1) A.5.1 프로그램 유지보수 매뉴얼: 프로그램의 유지보수에 필요한 모든 정보를 제공하는 문서. [ISO 2382-20:1990]
- 2) A.5.2 시스템 유지보수: 결함을 고치거나, 성능을 개선하거나 또는 시스템을 변환된 환경이나 요구사항에 맞추기 위한 시스템의 수정. [ISO 2382-20:1990]
- 3) A.5.3 업무(work task): 작업 시스템의 의도된 결과. [ISO 6385]
- 4) A.5.4 작업 시스템(work system): 작업 시스템은 사람과 작업 도구로 구성되어 작업장에서, 작업환경 안에서, 그리고 업무에 의해 부여된 조건하에서 업무를 수행하기 위해 작업공정에서 함께 기능을 수행하여 작업을 수행함. [ISO 1395:1981]

2.5.2. 부속서 B: 제품설명서의 예

다음의 예는 모든 제품설명서에서 제시되어야 할 정보를 보여주기 위하여 이 규격에 따라서 단지 가상의 소프트웨어 패키지를 설명한다.

<p>제품설명서</p> <p style="text-align: right;">작업불꽃 버전 2.6</p>
<p>작업불꽃 - 화면 보호기와 암호 보호</p> <p>프로그램 작업불꽃은 당신이 컴퓨터에 작업을 하고 있지 않을 때 적색-녹색-청색의 색깔을 띤 화면 위로 - 대단히 근사한 불꽃을 펼쳐보이면서 작업중인 화면을 보호해준다. 암호를 걸어놓으면, 당신의 부재 중 누군가 당신의 컴퓨터를 조작하면 당신은 이에 대한 통고를 받게 된다.</p> <p>작업불꽃은 기억장치에 설치가 된다. 이 프로그램은 일정 시간 동안 아무런 키가 눌러지지 않거나 마우스가 움직여지지 않으면 언제든지 자동으로 활성화 된다. 그 시간은 조정할 수 있다. 이것은 당신이 아무 키를 누르거나 마우스를 움직이자마자 실행을 멈춘다. 그러나 당신이 암호를 정해놓으면, 작업불꽃은 암호가 입력될 때까지 중단되지 않고 대기할 것이다.</p>

당신은 가장 좋아하는 구성을 하여

작업불꽃이 자동으로 활성화되기 전에 기다리는 시간을 조정할 수 있다(1부터 999분까지, 아니면 즉시).

동시에 일어나는 불꽃의 수를 조정할 수 있다(1부터 19까지).

이를위해 작업불꽃은 명령라인 대화나 윈도우 대화를 할 것이다(당신의 운영체계가 시스템 날짜와 시간을 조정하는 방식으로).

당신은 패스워드도 같은 방식으로 정의할 수 있다(6부터 45 문자까지). 그러므로 작업불꽃이 임의의 문자에 의해 작동이 멈추거나 아니면 암호를 입력했는데 작동이 멈추지 않는다면, 예를 들어 전원을 껐다가 켜는 식으로 누구인가 작업불꽃을 가로채서 암호 없이 또는 다른 암호로 작업불꽃을 재시작 해놓은 것이다.

운영체계에 의해서 이 프로그램과 그 구성을 백업 복사할 수 있다. 암호는 저장되지 않는다.

몇 가지 기술적 세부사항:

작업불꽃은 최소한 1MB 주기억장치와 최소한 720 KB의 90 mm (3.5 in) 또는 130 mm(5.25 in)의 디스켓 드라이브를 갖춘 Quince Hardcore 119xi 개인용 컴퓨터(그리고 이와 호환되는 컴퓨터)에서 실행된다. 하드디스크를 필요로 하지 않는다. 작업불꽃은 직렬 또는 병렬 Mini-Rat 마우스(또는 이와 호환되는 모든 마우스)를 지원하지만, 마우스를 요구하지는 않는다.

작업불꽃은 B.I.T.S 1.01 또는 Gnome 3.0(또는 이 둘과 호환되는 모든 운영체계) 하에서 실행된다.

작업불꽃을 주문할 때에는

B.I.T.S의 상이(변형)을 위한 것인지 아니면 Gnome의 상이(변형)을 위한 것인지

작업불꽃을 90mm (3.5 in) 디스켓으로 원하는지 130mm(5.25 in) 디스켓으로 원하는지 여부를 본사에 연락하기 바란다.

(제품의) 꾸러미는 디스켓에 담긴 프로그램(로드 모듈)과 설치 안내를 포함하고 있는 문서화된 소책자로 구성되어 있다.

다음사항에 주목하기 바란다.

작업불꽃을 설치하거나 사용할 때 어떠한 특별한 지식을 필요로 하지 않는다.
프로그램 메시지와 문서는 영어로 쓰여 있다.

작업불꽃은 ISO/IEC 12119: 1994 정보기술 - 소프트웨어 패키지 - 품질 요구 사항과 테스트를 완전히 준수하고 있다.

제품의 운영이나 유지보수에 지원을 하지는 않는다.

작업불꽃을 구할 수 있는 곳:

PyroManiac Klaus P Schmidt Ltd

33 Bell Street

Bergheim, SU 53844

Telephone (022) 845 3902

2.5.3. 해설서

ISO/IEC JTC1의 소프트웨어공학 표준화 위원회에서는 1991년 소프트웨어 제품의 품질 평가를 위한 ISO/IEC 9126을 발간하였다. 이 표준은 불특정한 소프트웨어의 개발과정에서 나오는 중간 산출물과 최종 제품의 평가를 위해서 개발되어 실제 산업 현장에서 적용하기에 여러 가지 어려움이 존재하였다. 따라서, 위원회에서는 일반적으로 많이 사용되면서도 평가가 용이한 패키지 소프트웨어의 품질평가와 품질을 테스트할 수 있는 기준을 개발하게 되었다. 따라서 본 표준에서는 ISO 9126에서 정의된 모든 품질 특성이 소프트웨어 패키지에는 적용되지 않고 있다.

본 표준이 가지는 특성 중에 하나는 ISO 9126에서 정의한 품질특성을

사용하고 있지만 특이하게 테스트를 위한 내용을 포함하고 있다는 것이다. 경우에 따라서는 본 표준에서 정의한 품질에 대한 테스트가 부족할 수 있으나, 이는 개정 중에 있는 ISO 9126 시리즈의 일부를 보완적으로 사용하면 해결될 수 있는 간단한 문제이다.

본 표준은 대규모 구매를 하는 집단을 중심으로 구매인증을 위한 기준으로 유럽에서 사용되기 시작하였다. 또한 본 표준에 따른 적용 결과 및 방법은 여러 곳에서 실험되고 연구되어 발표되고 있다. 현재 본 표준의 적용성은 대단히 높은 편이며, 개정 중에 있는 ISO 9126이 완성되어도 본 표준의 골격에는 영향이 없을 것이다. 또한 본 표준에 따른 품질 평가의 경험이 없이는 ISO 9126에 따른 품질 평가는 어렵다는데 본 표준의 중요성이 있다.

3. ISO/IEC 14598

3.1. 개요

ISO/IEC 14598은 소프트웨어 제품 평가 프로세스(software product evaluation processes)의 개요와 평가에 대한 안내지침 및 요구사항을 제공한다.

3.2. 평가 프로세스

국제표준인 ISO/IEC 14598은 세가지 다른 상황에서의 평가 프로세스 (Evaluation Process)에 대한 안내지침과 요구사항을 제공한다.

- ⊙ 개발(development) - ISO/IEC 14598-3;
- ⊙ 획득(acquisition) - ISO/IEC 14598-4;
- ⊙ 독립 평가(independent evaluation) - ISO/IEC 14598-5.

3.2.1. 개발자를 위한 프로세스 (Process for Developers)

ISO/IEC 14598-3은 새로운 제품을 개발 또는 기존에 있는 제품을 향상 시키기 위해서 계획하는 조직과 기술 직원을 통하여 제품 평가를 수행하려는 조직에서 사용될 수 있다. ISO/IEC 14598-3은 생명주기 동안에 개발된 중간 제품을 측정(measure) 함으로써, 최종 제품의 품질을 예측할 수 있는 척도(indicator)를 제공한다.

3.2.2. 획득자를 위한 프로세스 (Process for Acquirers)

ISO/IEC 14598-4는 기존 소프트웨어 제품 또는 미리 개발된 소프트웨어 제품을 획득 또는 재사용하기 위하여 계획하는 조직에서 사용될 수 있다. ISO/IEC 14598-4는 여러 후보 제품에서 특정 제품에 대하여 승인을 결정하기 위한 목적 또는 한 제품을 선택하기 위한 목적으로 적용될 수 있다.

3.2.3. 평가자를 위한 프로세스 (Process for Evaluators)

ISO/IEC 14598-5는 소프트웨어 제품의 독립적인 평가(independent assessment)를 수행하는 평가자에 의해서 사용될 수 있다. 이러한 평가는 개발자, 획득자, 또는 다른 동료의 요청에 의해서 수행될 수 있다.

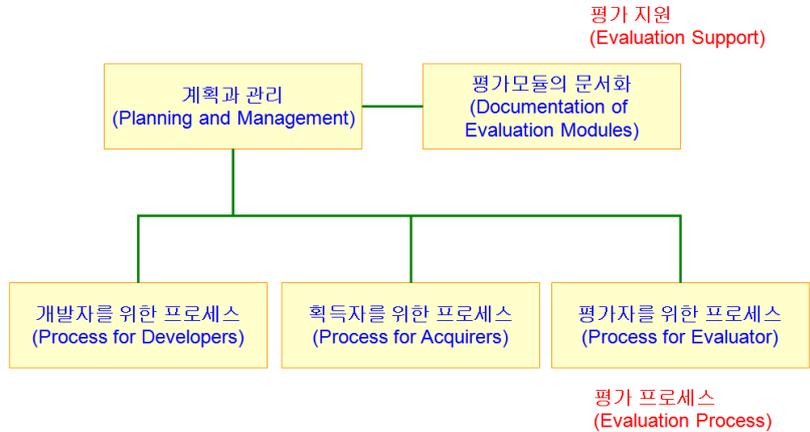
3.3. 평가를 위한 지원

본 항에서는 ISO/IEC 14598에서 평가를 위한 지원(Support for Evaluation)에 대하여 살펴본다. 그림 8.6과 같이, 각 평가 프로세스 표준은 ISO/IEC 14589-2 계획과 관리(Planning and Management) 그리고 ISO/IEC 14598-6 평가 모듈의 문서화(Documentation of evaluation modules)와 연결(conjunction)하여 사용될 수 있다.

<그림6> 평가 프로세스와 평가 지원 표준의 관계

3.3.1. 계획과 관리(Planning and Management)

ISO/IEC 14598-2 계획과 관리는 소프트웨어 제품 평가에 대한 기능을 지원하기 위해 필요한 요구사항과 안내지침을 포함하고 있다. ISO/IEC14598-2에서의 지원(support)은 소프트웨어 평가 프로세스 그리고 개발, 획득, 표준화, 통제, 조직 내 평가전문가의 피드백(feedback) 등



을 포함하여 관련된 활동(activity)의 계획과 관리에 연관되어 있다. ISO/IEC 14598의 이 부분은 정량적인(quantitative) 평가 계획을 만드는 매니저에 의해서 사용될 수 있다.

3.3.2. 평가 모듈(Evaluation Modules)

ISO/IEC 14598-6은 평가모듈을 문서화하기 위한 안내지침을 제공한다. 평가모듈은 품질 모델의 명세서(specification), 모델의 계획된 어플리케이션에 대한 관련된 데이터와 정보 그리고 실제 어플리케이션에 대한 정보를 포함한다. ISO/IEC 14598의 이 부분은 새로운 평가 모듈을 생산하는 조직에 의해서 사용될 수 있다.

3.4. 소프트웨어 품질 특성과 메트릭

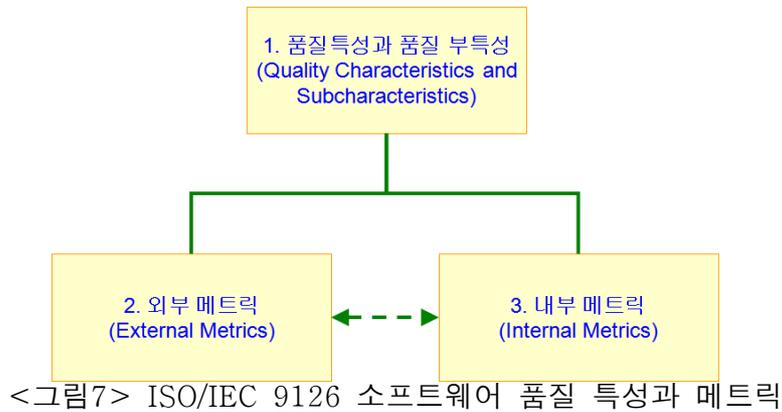
ISO/IEC 14598의 각 부분은 소프트웨어 품질 특성과 메트릭(Software Quality Characteristics and Metrics)을 기술하고 있는 ISO/IEC 9126의 부분과 연결(conjunction)하여 사용될 수 있다.

◎ 품질특성과 부특성(Quality Characteristics and

Subcharacteristics) - ISO/IEC 9126-1

- ⊙ 외부 메트릭(External Metrics) - ISO/IEC 9126-2
- ⊙ 내부 메트릭(Internal Metrics) - ISO/IEC 9126-3

[그림 8-7]과 같이, ISO/IEC 9126-1은 품질 특성, 연관된품질 부특성 그리고 ISO/IEC 9126 품질 모델의 상위 세 레벨 사이의 관계를 정의한다. ISO/IEC9126-2 그리고 ISO/IEC 9126-3은 각 메트릭에 일치하는 품질 특성과 품질 부특성의 관계를 식별한다.



3.5. 평가 프로세스

소프트웨어 품질을 평가하기 위하여, 우선 평가 요구사항을 설정하고, 명세하고, 설계하고, 마지막으로 평가를 수행해야 한다. ISO/IEC 14598의 이 부분은 평가 프로세스(Evaluation Process)의 개요를 설명하고 있다. ISO/IEC 14598의 다른 부분은 이 프로세스가 다른 환경에 적용될 수 있는 방법에 대하여 설명하고 있다.

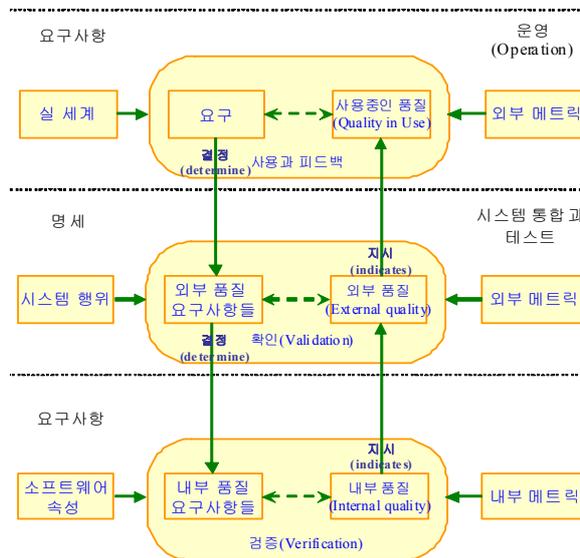
3.6. 평가 요구사항 설정

3.6.1. 평가 목적 설정

소프트웨어 품질 평가의 목적은 고객의 필요를 만족하는 소프트웨어의 개발과 획득 모두를 직접적으로 지원하는 것이다. 궁극적인 목표는 제품이 요구되는 품질을 제공한다는 것을 보증하는 것이다.

3.6.2. 평가될 제품의 타입을 식별

평가될 중간 소프트웨어 제품 또는 최종 소프트웨어 제품의 타입은 생명주기에 있는 단계와 평가 목적에 의존적일 것이다. 이 단계의 목적은 소프트웨어 제품이 사용자에게 의해 실제 사용될 때 상태를 만족하고 요구를 적용하는 것이다.



<그림8> 소프트웨어 생명주기에서의 품질

3.6.3. 품질 모델을 명세

소프트웨어 평가의 첫 번째 단계는 품질 모델을 사용함으로써 관련된 품질 특성을 선택하는 것이다. 소프트웨어 평가를 위한 품질 모델은 일반적으로 특성과 부특성의 계층적 트리 구조로 구분된 소프트웨어 품질 속성의 총합으로 표현된다.

3.7. 평가 명세화

3.7.1. 메트릭 선택

메트릭은 소프트웨어 제품 측정을 쉽고 경제적으로 만들 수 있고 측정 결과를 사용하기 쉽기 때문에 매우 중요하다. 많은 소프트웨어 측정이 편리 하려면 몇몇 종류의 도구로 구성하고 평가 모듈로써 패키징 되어야 할 것이다(ISO/IEC 14598-6).

평가의 두 가지 넓은 의미는 수정할 문제를 식별하기 위한 것과 양자택 일되는 제품이나 요구사항에 대응하는 제품의 품질을 비교하기 위한 것이다.

3.7.2. 메트릭을 위한 등급 레벨 성립

양적인 특징은 품질 메트릭을 사용함으로써 측정될 수 있다. 측정된 결과는 양과 맵핑된다. 이런 값은 그 자체가 레벨의 만족도를 보이지는 않는다. 이러한 목적 때문에 양은 요구사항의 다른 만족도에 대응하는 범위로

분할한다.



<그림9> 메트릭을 위한 등급 레벨(rating level)

3.7.3. 평가를 위한 범주 성립

소프트웨어 품질 요구사항 명세는 적절하게 잘 정의된 품질 모델을 사용함으로써 정의될 수 있다. 이런 목적을 위해 다른 모델을 사용해야 하는 특별한 이유가 없다면 ISO/IEC 9126-1의 품질 모델과 정의를 사용한다. 제품의 품질을 평가하기 위해 다른 특징의 평가 결과를 요약할 필요가 있다.

3.8. 평가 설계

평가설계(Design the evaluation) 단계는 평가 계획을 생산하는 것을 의미한다. 평가 계획은 평가 방법과 평가자의 행동이 스케줄을 설명한다(ISO/IEC 14598-3, ISO/IEC 14598-4 혹은 ISO/IEC 14598-5). 평가 계획은 측정 계획과 일치해야 한다(ISO/IEC 14598-2).

3.9. 평가 실행

가. 측정

측정을 위해서, 선택된 메트릭은 소프트웨어 제품에 적용한다. 결과는 메트릭 척도의 값이다.

나. 범주를 비교

등급단계에서, 측정된 값은 이미 결정된 범주와 비교된다.

다. 결과 평가

평가는 소프트웨어 평가 프로세스의 마지막 단계이다. 이 단계에서 등급 레벨의 집합이 도출된다.

3.10. 프로세스 지원

프로세스 지원 활동은 다음과 같은 정보를 수집함으로써 평가를 지원한다. 수집된 정보는 메트릭을 개발하고 평가하며, 평가 프로세스를 표준화하고, 메트릭과 측정을 위한 데이터로 사용된다. ISO/IEC 14598-2는 소프트웨어 제품 평가를 위한 지원 프로세스의 요구사항과 지침을 포함한다.

4. IEEE1008

4.1. 범위

본 하에서는 IEEE1008의 적용 가능한 범위(Scope)에 대하여 살펴본다. IEEE1008은 소프트웨어 단위 테스트에 대한 표준이다. 소프트웨어 단위 테스트는 테스트 계획의 성과, 테스트 집합 획득, 소프트웨어 단위의 요구사항에 대한 테스트 단위의 측정을 포함한 절차이다. 측정은 단위 요구사항 문서에 명세함으로서 단위를 실행하거나 측정에 요구한 동작과 단위의 실질적인 동작을 비교하기 위해 견본 데이터를 사용한다.

본 표준에서는 체계적이고 문서화된 단위 테스트를 위하여 조직화된 접근을 정의한다. 이 접근 방법에서 테스트의 완벽성을 결정하기 위하여 단위 디자인과 단위 구현 정보가 사용 된다. 또한 본 표준에서는 단계, 활동, 지침을 체계적으로 구성된 단위 테스트 절차를 설명한다.

일반적으로 단위 테스트 계획은 전체 테스트 계획에서 수행된다. 전체 테스트 계획 절차와 본 표준의 범위와는 적합하지 않지만, 일반적인 단위 테스트 계획 활동을 다룬다.

4.2. 단위 테스트 활동

본 항에서는 단위 테스트 활동(Unit Testing Activities)에 대하여 설명한다.

가. 테스트 계획 수행 단계

- ⊙ 일반적인 접근방법, 자원, 스케줄 계획

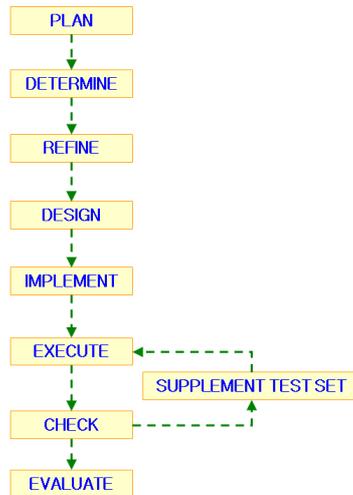
- ⊙ 테스트되기 위한 특징 결정
- ⊙ 일반적인 계획 정제

나. 테스트 집합 획득 단계

- ⊙ 테스트 집합 디자인
- ⊙ 정의된 계획과 디자인 구현

다. 테스트 단위 측정 단계

- ⊙ 테스트 절차 실행
- ⊙ 종결을 위한 확인
- ⊙ 테스트 노력과 단위 평가



[그림 8-10] 단위 테스트 활동

라. 일반적인 접근방법, 자원, 스케줄 계획

일반적으로 단위 테스트 계획은 전체 테스트 계획을 하는 동안 계획 문서를 고려하여 작성해야 한다.

마. 테스트 계획 입력물

- ⊙ 프로젝트 계획
- ⊙ 소프트웨어 요구사항 문서

바. 계획지침

- ⊙ 단위 테스트를 위한 일반적인 접근방법 명세
- ⊙ 완벽한 요구사항 명세
- ⊙ 종료 요구사항 명세
- ⊙ 자원 요구사항 명세
- ⊙ 일반적인 스케줄 명세

사. 테스트 계획 산출물

- ⊙ 일반적인 단위 테스트 계획 정보(계획 지침 1~5항목에 해당)
- ⊙ 단위 테스트 일반적인 자원 요구서(만약 계획 지침 4항목에 산출물이 있을 경우)

아. 테스트를 위한 특징 결정

- 1) 결정입력물
 - ⊙ 단위요구사항 문서
 - ⊙ 소프트웨어 아키텍처 디자인 문서(필요 시)
- 2) 결정지침
 - ⊙ 기능적 요구사항 연구
 - ⊙ 추가적 요구사항과 연관된 절차 식별

- ⊙ 단위 상대 식별
- ⊙ 입력과 출력의 데이터 특성 식별
- ⊙ 테스트 안에 포함 되기 위한 요소 선택
- 3) 결정산출물
 - ⊙ 테스트 안에 포함 되기 위한 요소의 항목(결정 지침의 5항목에 해당)
 - ⊙ 일반적인 단위 테스트 계획 정보(결정 지침의1~5항목에 해당)

자. 일반적인 계획 정제

- 1) 정제입력물
 - ⊙ 테스트 안에 포함 되어지기 위한 요소의 항목(결정 지침의 5항목에 해당)
 - ⊙ 단위 요구사항 설명 요구서(결정 지침의 1~4항목에 해당)
- 2) 정제지침
 - ⊙ 접근 방법 정제
 - ⊙ 특별한 자원 요구사항 명세
 - ⊙ 명확한 스케줄 명세
- 3) 정제산출물
 - ⊙ 명확한 단위 테스트 계획 정보 (정제 지침 1~3항목에 해당)
 - ⊙ 단위 테스트 특별한 자원 요구서(만약 정제 지침 2항목의 산출물이 있을 경우)

차. 테스트 집합 명세

- 1) 디자인 입력물
 - ⊙ 단위요구사항 문서
 - ⊙ 테스트 안에 포함되기 위한 요소의 항목 (결정 지침의 5항목에 해당)

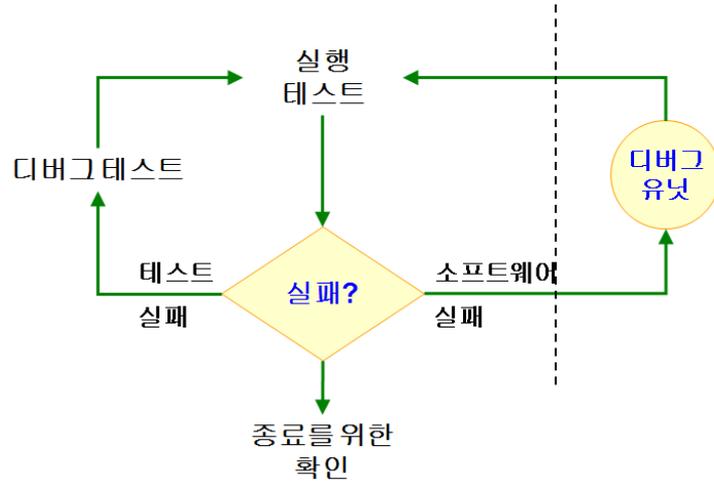
- ⊙ 단위 테스트 계획 정보 (테스트 계획 지침의 1,2항목, 정제 지침 1 항목)
- ⊙ 단위 디자인 문서
- ⊙ 이전 테스트로부터 테스트 명세서(사용가능 시)
- 2) 디자인 지침
 - ⊙ 테스트 집합의 아키텍처 디자인
 - ⊙ 명확한 테스트 절차 획득
 - ⊙ 테스트 케이스 명세서 획득
 - ⊙ 디자인 정보에 기반한 테스트 케이스 명세서 확대
 - ⊙ 테스트 디자인 명세서 완성
- 3) 디자인 산출물
 - ⊙ 단위테스트 디자인 명세서(디자인 지침 5항목에 해당)
 - ⊙ 분리된 테스트 절차 명세서(만약 디자인 지침 2항목의 산출물이 있을 경우)
 - ⊙ 분리된 테스트 케이스 명세서(만약 디자인 지침 3혹은4항목의 산출물이 있을 경우)
 - ⊙ 단위 디자인 강화 요구서(만약 디자인 지침 4항목의 산출물이 있을 경우)

카. 정제된 계획과 디자인 구현

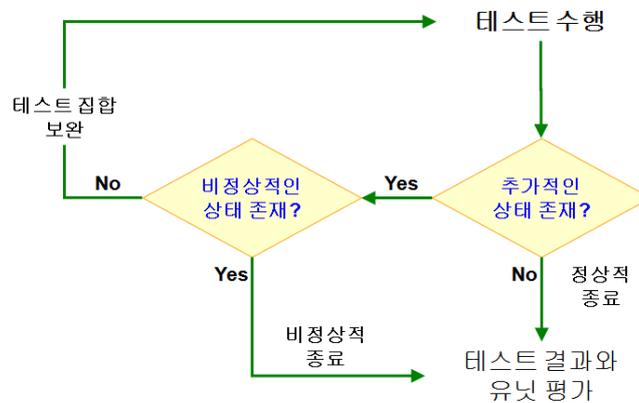
- 1) 구현입력물
 - ⊙ 단위테스트 계획 정보
 - ⊙ 단위 테스트 디자인 명세서 혹은 분리된 문서들에 테스트 케이스 명세서
 - ⊙ 소프트웨어 데이터 구조 설명서
 - ⊙ 테스트 지원 자원
 - ⊙ 테스트 아이템
 - ⊙ 이전 테스트 활동으로부터 테스트 데이터(사용 가능 시)

- ⊙ 이전 테스트 활동으로부터 테스트 도구(사용 가능 시)
- 2) 구현지침
 - ⊙ 테스트 데이터 검증과 획득
 - ⊙ 특별한 자원 획득
 - ⊙ 테스트 아이템 획득
- 3) 구현산출물
 - ⊙ 검증된 테스트 데이터(검증 지침 1항목에 해당)
 - ⊙ 테스트 지원 자원(검증 지침 2항목에 해당)
 - ⊙ 테스트 아이템 구성(검증 지침 3항목에 해당)
 - ⊙ 초기 요약 정보(검증 지침 3항목에 해당)
- 4) 테스트 절차 실행
- 5) 실행입력물
 - ⊙ 검증된 테스트 데이터(검증 지침 1항목에 해당)
 - ⊙ 테스트 지원 자원(검증 지침 2항목에 해당)
 - ⊙ 테스트 아이템 형상(검증 지침 3항목에 해당)
 - ⊙ 테스트 절차 명세서(검증 지침 2 항목에 해당, 만약 산출물이 있을 경우)
 - ⊙ 실패 분석 결과(디버깅 절차에 해당, 만약 산출물이 있을 경우)
- 6) 실행지침
 - ⊙ 테스트 실행
 - ⊙ 결과 결정
 - 테스트가 실패할 경우를 고려하여 테스트 실패 정보, 분석내용을 테스트 요약 보고서의 결과 정리 항목에 기록한다. 실패했을 경우 다음의 적합한 Case를 선택하고 Case에 알맞은 행동을 수행한다.
 - Case 1: 테스트 명세서 혹은 테스트 데이터 실패
 - Case 2: 테스트 절차 실행의 실패
 - Case 3: 테스트 환경의 실패
 - Case 4: 단위 구현의 실패

- Case 5: 단위 디자인의 실패
- 주의 - [그림 8-11]과 같이, 실행 주기와 체크 지침은 테스트 지침 3 항목에 정의된 종료 상태가 될 때까지 반복한다. 실행 활동에서 제어 흐름은 [그림 8-12]와 같이 표시된다.



<그림11> 실행활동의 제어 흐름



<그림12> 체크 활동의 제어 흐름

7) 실행산출물

- ⊙ 테스트 결과, 테스트하며 발생한 사건 설명, 실패 분석 결과, 오류 수정 활동, 오류를 수정 할 수 없는 이유, 자원 소비 데이터, 절차

언어 구현을 위한 추적 요약 정보를 포함한 테스트 요약 보고서의
실행 정보(실행 지침 1,2항목에 해당)

- ⊙ 수정된 테스트 명세서(만약 실행지침 2의 산출물이 있을 경우)

수정된 테스트 데이터(만약 실행지침 2의 산출물이 있을 경우)

- ⊙ 종료를 위한 확인

8) 확인입력물

- ⊙ 종료요구사항(테스트 계획 지침 2, 3항목에 해당)
- ⊙ 실행 정보(실행 지침 1,2항목에 해당)
- ⊙ 테스트 명세서(디자인 지침 1~3항목에 해당, 요구 시)
- ⊙ 소프트웨어 데이터 구조 설명서(요구 시)

9) 확인지침

- ⊙ 테스트 절차의 정상적인 종료를 위한 확인
- ⊙ 테스트 절차의 비정상적인 종료를 위한 확인
- ⊙ 테스트 집합 보충

10) 확인산출물

- ⊙ 테스트 케이스의 추가적인 활동과 종료 상태를 포함한 테스트 요약 보고서의 확인 정보(확인 지침 1~3항목에 해당)
- ⊙ 추가되고 혹은 수정한 테스트 명세서(확인 지침 3항목의 산출물이 있을 경우)
- ⊙ 추가된 테스트 데이터(만약 지침 3항목의 산출물이 있을 경우)

타. 테스트 노력과 단위 평가

1) 평가입력물

- ⊙ 단위테스트 디자인 명세서(디자인 지침 5항목에 해당)
- ⊙ 실행 정보(실행 지침 1,2항목에 해당)
- ⊙ 확인 정보(확인 지침 1~3항목에 해당)
- ⊙ 분리된 테스트 케이스 명세서(디자인 지침 3,4항목에 해당, 만약 산출물이 있을 경우)

2) 평가지침

- ⊙ 테스트 상태 설명
- ⊙ 단위의 상태 설명
- ⊙ 테스트 요약 보고서 완료
- ⊙ 테스트 생산물의 보존 보증

3) 평가산출물

- ⊙ 완벽한 테스트 요약 보고서(평가 지침 3항목에 해당)
- ⊙ 완벽하게 저장된 테스트 생산물의 수집물(평가 지침 4항목에 해당)

5. IEEE1012

5.1. 개요

IEEE 1012는 소프트웨어의 검증 및 확인 계획(Software Verification and Validation Plan, SVVP)에 대한 표준으로서 이를 작성하기 위해 필요한 지침이나 정보 등을 담고 있다. 소프트웨어 검증 및 확인은 애플리케이션 생명주기의 각 단계에서 소프트웨어를 평가하기 위해 잘 정의된 접근 방법이다. 소프트웨어 검증 및 확인 활동을 통해 소프트웨어는 그 품질을 보장하고, 소프트웨어가 사용자 요구사항에 부합되는지를 확인할 수 있다. 또한 소프트웨어의 검증 및 확인을 통해 프로젝트와 소프트웨어의 상태를 파악할 수 있는 통찰력을 가지고 소프트웨어를 관리할 수 있으며 이로 인해 각 단계의 산출물이나 소프트웨어 개발 프로세스 또는 지원 프로세스를 유연하게 적용할 수 있다.

소프트웨어 검증 및 확인은 소프트웨어와 생명주기에 따른 중간 산출물이 소프트웨어의 기능과 품질을 위한 요구사항에 부합되는지를 확인하기 위해 리뷰(Review), 분석(Analysis), 테스트(Testing) 기술 등을 포함한다.

소프트웨어 검증 및 확인 활동은 소프트웨어의 결점을 찾아내고 사용자들이 요구하는 기능과 속성들이 소프트웨어에 적용되었는지를 확인하는 것에 목적을 둔다. 따라서 검증 및 확인 활동들은 소프트웨어 개발 산출물들에 영향을 주며 그 내용은 다음과 같이 구분된다.

5.1.1. 검증

소프트웨어 생명주기의 모든 단계 산출물에 대한 검증으로서 검증 내용은 다음과 같다.

각 단계의 산출물은 이전 단계의 요구사항과 산출물에 대해 정확성(Correctness), 완결성(Completeness), 일관성(consistency), 정밀성(Accuracy)등을 만족시켜야 한다.

각 단계의 산출물은 그 단계의 표준, 관례, 규칙 등을 만족시켜야 한다.

각 단계의 산출물은 다음단계의 활동을 위한 정보를 포함해야 한다.

5.1.2. 확인

최종산출물이 정의된 소프트웨어 및 시스템 요구사항을 만족시키는지 에 대한 확인이다.

검증및 확인 활동들은 일반적으로 소프트웨어 개발 프로세스와 지원 프로세스에 함께 적용하며 그 중 어떤 활동은 두 프로세스 사이에서 수행 될 수 있다. 검증 및 확인 활동은 계획, 조직, 감독 등의 관리 활동과 분석, 평가 리뷰, 테스트 등의 기술활동으로 구분된다.

5.2. 소프트웨어 검증 및 확인 계획 가이드

본 항에서는 소프트웨어 검증 및 확인 계획 가이드(Software Verification and Validation Plan)에 대하여 살펴본다. 계획 수립과 계획의 문서화의 목적은 확인 및 검증을 위한 자원을 효과적으로 관리하고, 검증 및 확인 절차를 감독하고 제어하며, 활동에 참여하는 사람들의 역할과

의무를 명확하게 하는 것이다.

소프트웨어 검증 및 확인 계획은 모든 프로젝트에 적용될 수 있는 범용적인 계획을 프로젝트에 맞게 수정해야 한다. 검증 및 확인 활동의 기본적인 목적은 전반적인 생명주기 동안 바뀌지 않으나 어느 경우에는 계획에 대해 중요한 수정사항이 발생할 수 있다. 이러한 수정사항들은 소프트웨어 개발 활동이나 검증 및 확인 활동에 연결되어있다. 따라서 계획을 수정하기 위한 절차도 계획 안에 포함되어야 한다. 전반적인 프로젝트 계획을 위한 검증 및 확인 계획은 다음과 같은 세부 스텝을 포함한다.

1) 검증 및 확인 범위를 식별한다.

소프트웨어 개발 활동들을 정의하며 이 활동은 중요도, 복잡성, 이용 가능한 자원 등의 요소를 고려한 프로젝트에 맞게 적용될 수 있다.

2) 전체적인 프로젝트의 범위에서 구체적인 목표를 설정한다.

설정된 목표가 상세하고 측정 및 성취 가능하여 검증 및 확인 활동을 통해 만족될 수 있는 목표여야 한다.

3) 검증 및 확인 도구와 기술을 선택하고 계획을 수립하기 위해 선행되는 프로젝트 입력물을 분석한다.

4) 사용될 기술 및 도구를 선택한다.

프로젝트를 진행할 때 사용 가능한 도구나 기술들을 식별하고 각 도구나 기술들을 선택하기 위해 필요한 정보를 수집한다. 수집한 정보를 기반으로, 프로젝트의 특정한 제약사항이나 필요에 맞게 도구와 기술을 선택한다.

5) 검증 및 확인 계획을 수립한다.

이전단계의 결과를 리뷰하고, 이를 기반으로 검증 및 확인의 목적과 제약사항들에 맞는 상세한 활동들을 정의한다.

5.3. 생명주기에 따른 검증 및 확인 활동

본 항에서는 생명주기에 따른 검증 및 확인 활동(Life-Cycle Verification and Validation)에 대하여 살펴본다. 소프트웨어 생명주기는 소프트웨어 개발 프로세스와 검증 및 확인 활동들을 포함한다. 대부분의 소프트웨어 검증 및 확인 계획은 생명주기의 단계에 할당된 활동을 통해 관리된다. [그림 8-13]는 소프트웨어 검증 및 확인 계획의 전반적인 구성 요소를 보여준다.

생명주기는 7개의 소프트웨어 개발 및 사용 단계로 구성되어 있고 각 단계는 검증 및 확인활동을 위한 입력물을 정의하고 있으며 생명주기의 각 단계 내에는 최소한의 검증 및 확인 활동을 정의하고 해당 단계를 통해 요구되는 검증 및 확인 출력물을 정의한다. 또한 IEEE1012표준에서는 검증 및 확인활동을 위해 검증 및 확인의 관리를 한 단계로 정의하고 있다. 이 활동은 생명주기의 특정한 한 단계라기보다 여러 단계에 적용되고 함께 수행되는 단계이다.

검증 및 확인 관리를 포함한 생명주기에서의 검증 및 확인 활동은 다음과 같다.

- 1) 검증 및 확인 관리(Management of V&V)
 - ⊙ 소프트웨어 검증 및 확인 계획을 생성
 - ⊙ 한계선(기준선) 변경 평가
 - ⊙ 검증 및 확인 리뷰 관리
 - ⊙ 리뷰 검증
- 2) 구상 단계의 검증 및 확인(Concept Phase V&V)
- 3) 요구사항 도출 단계의 검증 및 확인(Requirement Phase V&V)
 - ⊙ 소프트웨어 요구사항 추적성 분석
 - ⊙ 소프트웨어 요구사항 평가
 - ⊙ 소프트웨어 요구사항 인터페이스 분석
 - ⊙ 테스트 계획 생성
 - 시스템 테스트
 - 인수 테스트
- 4) 설계단계에서의 검증 및 확인(Design Phase V&V)
 - ⊙ 소프트웨어 설계 추적성 분석
 - ⊙ 소프트웨어 설계 평가
 - ⊙ 소프트웨어 설계 인터페이스 분석
 - ⊙ 테스트계획 생성
 - 컴포넌트 테스트
 - 통합테스트
 - ⊙ 테스트 설계 생성
 - 컴포넌트 테스트
 - 통합테스트
 - 시스템 테스트
 - 인수 테스트
- 5) 구현 단계의 검증 및 확인(Implementation Phase V&V)

- ⊙ 소스코드 추적성 분석
- ⊙ 소스코드 평가
- ⊙ 소스코드 인터페이스 분석
- ⊙ 소스코드 문서 평가
- ⊙ 테스트케이스 생성
 - 컴포넌트 테스트
 - 통합테스트
 - 시스템 테스트
 - 인수 테스트
- ⊙ 테스트절차 생성
 - 컴포넌트 테스트
 - 통합 테스트
 - 시스템 테스트
- ⊙ 컴포넌트 테스트 실행
- 6) 테스트단계의 검증 및 확인(Test Phase V&V)
 - ⊙ 인수테스 생성
 - ⊙ 테스트실행
 - 통합테스트
 - 시스템 테스트
 - 인수 테스트
- 7) 설치 및 검사 단계의 검증 및 확인(Installation and checkout Phase V&V)
 - ⊙ 설치구성 감사
 - ⊙ 최종 검증 및 확인 보고서 생성
- 8) 작동 및 유지보수 단계의 검증 및 확인(Operation and Maintenance Phase V&V)
 - ⊙ 소프트웨어 검증 및 확인 계획 수정
 - ⊙ 예외처리 평가
 - ⊙ 제안된 수정 평가

◎ 단계 테스트 반복