부록 A

**(규정)**

**기술(기법) 및 측정 방법(조치)의 선정 기준**

* 1. General

이 표준의 조항은 부속문서 표(표 A.1~표 A.11의 조항 A.2 참조)내에서 적합성 달성을 위한 방법을 설명하기 위해 관련된다. 하위 수준 표도 존재하며, 조항 표의 특정 항목을 확장하는 세부 표(예를들어, A.3절의 표 A.12 ~ 표 A.23 참조)가 존재한다. 예를들어, 표 A.2의 "모델링"은 표 A.17에서 확장된다. 또한 표에서 참조하는 부록 D에 정보가 존재한다.

표의 기법 및 측정 방법에는 각 소프트웨어 안전무결성수준(SIL)에 대한 요구사항이 있다. 이 문서에서는 소프트웨어 안전무결성수준 1과 2에 대한 요구사항은 각 기법에 대해 동일합니다. 마찬가지로, 소프트웨어 안전무결성수준 3과 4에서 동일한 요구사항을 가지고 있다. 이러한 요구사항은 다음과 같다.

|  |  |
| --- | --- |
| ‘M’ | 이 기호는 해당 기법의 사용이 필수적(mandatory)이라는 것을 의미한다. |
| ‘HR’ | 이 기호는 안전무결성수준에 대해 기술이나 조치가 적극 권장 사항(Highly Recommended)을 의미한다. 이 기법이나 측정 방법을 사용하지 않을 경우 대체 기법을 사용하는 근거는 소프트웨어 품질보증 계획서 또는 소프트웨어 품질보증 계획서에서 참조하는 다른 문서에 자세히 설명되어야 한다. |
| ‘R’ | 이 기호는 안전무결성수준을 위해 기술이나 측정 방법이 권장사항임(Recommended)을 의미한다.  이는 'HR'보다 낮은 수준의 권장사항이며 이러한 기법을 결합하여 패키지의 일부를 구성할 수 있다. |
| ‘-‘ | 이 기호는 해당 기법에 사용에 대한 권장사항 또는 반대사항이 없음을 의미한다. |
| ‘NR’ | 이 기호는 해당 기법이나 측정 방법이 안전무결성수준에 대해 권장하지 않음을 의미합니다. 해당 기법이나 측정 방법이 사용되는 경우, 이를 사용하는 근거는 소프트웨어 품질보증계획 또는 소프트웨어 품질보증 계획에서 참조하는 다른 문서에 자세히 설명되어야 한다. |

기법이나 측정 방법의 조합은 소프트웨어 품질보증 계획 또는 소프트웨어 품질보증 계획에서 참조하는 다른 문서에 명시되어야 하며 표에 첨부된 참고사항이 다른 요구사항을 나타내지 않는 한 하나 이상의 기법 또는 측정방법이 선택되어야 한다. 이러한 명시는 승인된 기법 또는 승인된 기법조합에 대한 참조가 포함될 수 있다.

모든 각각의 필수기법을 포함하여 해당 기법 또는 기법의 조합이 사용하는 경우, 평가자는 해당 기법을 유효한 것으로 승인해야하고 올바르게 적용되는지 관심을 가져야 한다. 만약, 다른 기법이 사용하고 정당화할 수 있는 경우, 평가자는 이를 허용할 수 있다.

Add, EN 50128:2011/A2:2020

NOTE 1: 다음의 예는 표(A.1 참조)와 상세 표(A.2 참조)의 적용에 대한 추가 지침을 제공한다.

* 표 A.7은 기본 무결성(Basic Integrity)을 위한 "기능 및 블랙박스 시험"에 대한 HR을 정의하며 표 A.14를 참조한다. 그러나 A.14에는 "R" 방법만 포함되어 있다. 즉, 기능 및 블랙박스 시험을 적용할 것을 강력히 권장하지만, 이 표준의 사용자는 A.14 또는 다른 기법의 기법을 자유롭게 적용할 수 있음을 의미한다.
* 표 A.5는 기본 무결성(Basic Integrity)에 대한 "정적 분석"에 대한 요구 사항이 없으며 표 A.19를 참조한다. 그러나 A.19에는 "Walkthroughs/Design Reviews"에 대한 "HR"이 포함되어 있다. 이는 정적 분석을 수행할 필요가 없음을 의미하지만, 그럼에도 불구하고 이 표준의 사용자는 표 A.19의 HR를 사용하기로 결정한 경우, 표 A.19의 HR를 적용할 수 있다.
* 표 A.7은 SIL3/4에 대한 성능시험을 위한 "M"을 정의하며 표 A.18을 참조한다. 그러나 A.18은 "HR" 방법만을 포함하고 있다. 이는 성능시험을 실시하는 것이 의무적이지만, 대체 기법을 사용하는 근거가 제공될 수 있는 경우 A.18에 언급되지 않은 기법을 적용할 수 있음을 의미한다.
  1. Clauses tables

Table A.1 – Life cycle Issues and Documentation (5.3) (1 of 2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DOCUMENTATION** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| ***Planning*** |  |  |  |  |  |
| 1. Software Quality Assurance Plan | HR | HR | HR | HR | HR |
| 2. Software Plan Verification Report | R | HR | HR | HR | HR |
| 3. Software Configuration Management Plan | HR | HR | HR | HR | HR |
| 4. Software Verification Plan | HR | HR | HR | HR | HR |
| 5. Software Validation Plan | HR | HR | HR | HR | HR |
| ***Software requirements*** |  |  |  |  |  |
| 6. Software Requirements Specification | HR | HR | HR | HR | HR |
| 7. Overall Software Test Specification | HR | HR | HR | HR | HR |
| 8. Software Requirements Verification Report | R | HR | HR | HR | HR |
| ***Architecture and design*** |  |  |  |  |  |
| 9. Software Architecture Specification | R | HR | HR | HR | HR |
| 10. Software Design Specification | R | HR | HR | HR | HR |
| 11. Software Interface Specifications | HR | HR | HR | HR | HR |
| 12. Software Integration Test Specification | R | HR | HR | HR | HR |
| 13. Software/Hardware Integration Test Specification | R | HR | HR | HR | HR |
| 14. Software Architecture and Design Verification Report | R | HR | HR | HR | HR |
| ***Component Design*** |  |  |  |  |  |
| 15. Software Component Design Specification | - | HR | HR | HR | HR |
| 16. Software Component Test Specification | - | HR | HR | HR | HR |
| 17. Software Component Design Verification Report | - | HR | HR | HR | HR |
| ***Component Implementation and Testing*** |  |  |  |  |  |
| 18. Software Source Code and supporting documentation | HR | HR | HR | HR | HR |
| 19. Software Component Test Report | - | HR | HR | HR | HR |
| 20. Software Source Code Verification Report | - | HR | HR | HR | HR |
| ***Integration*** |  |  |  |  |  |
| 21. Software Integration Test Report | R | HR | HR | HR | HR |
| 22. Software/Hardware Integration Test Report | R | HR | HR | HR | HR |
| 23. Software Integration Verification Report | R | HR | HR | HR | HR |
| ***Overall Software Testing / Final Validation*** |  |  |  |  |  |
| 24. Overall Software Test Report | HR | HR | HR | HR | HR |
| 25. Overall Software Test Verification Report(IEC 62278:2015) | HR | HR | HR | HR | HR |
| 25. Software Validation Report | HR | HR | HR | HR | HR |
| 26. Tools Validation Report | - | HR | HR | HR | HR |
| 27. Release Note | HR | HR | HR | HR | HR |

Table A.1 (2 of 2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DOCUMENTATION** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| ***Systems configured by application data/ algorithms*** |  |  |  |  |  |
| 28. Application Requirements Specification | HR | HR | HR | HR | HR |
| 29. Application Preparation Plan (see NOTE 4) | R | HR | HR | HR | HR |
| 30. Application Test Specification (see NOTE 4) | HR | HR | HR | HR | HR |
| 31. Application Architecture and Design (see NOTE 4) | R | HR | HR | HR | HR |
| 32. Application Preparation Verification Report | - | HR | HR | HR | HR |
| 33. Application Test Report | HR | HR | HR | HR | HR |
| 34. Source Code of Application Data/Algorithms | HR | HR | HR | HR | HR |
| 35. Application Data/Algorithms Verification Report | HR | HR | HR | HR | HR |
| ***Software deployment*** |  |  |  |  |  |
| 36. Software Release and Deployment Plan | R | HR | HR | HR | HR |
| 37. Software Deployment Manual | R | HR | HR | HR | HR |
| 38. Release Notes | HR | HR | HR | HR | HR |
| 39. Deployment Records | R | HR | HR | HR | HR |
| 40. Deployment Verification Report | R | HR | HR | HR | HR |
| ***Software maintenance*** |  |  |  |  |  |
| 41. Software Maintenance Plan | R | HR | HR | HR | HR |
| 42. Software Change Records | HR | HR | HR | HR | HR |
| 43. Software Maintenance Records | R | HR | HR | HR | HR |
| 44. Software Maintenance Verification Report | R | HR | HR | HR | HR |
| ***Software assessment*** |  |  |  |  |  |
| 45. Software Assessment Plan | - | HR | HR | HR | HR |
| 46. Software Assessment Report | - | HR | HR | HR | HR |
| NOTE 1 기본 무결성( Basic Integrity)의 경우 소프트웨어 검증 계획의 내용은 6.2.4.9 a), b), c), d), f)로 제한될 수 있다.  NOTE 2 기본 무결성( Basic Integrity)의 경우 소프트웨어 인터페이스 사양은 전체 소프트웨어의 경계에 대해서만 HR이다(7.3.4.18 참조).  NOTE 3 5.3.2.12 및 5.3.2.13에 따라 문서를 다르게 결합할 수 있다.  NOTE 4 HR 또는 R인 문서 29, 30 및 31은 프로세스에서 정의된 중요도와 검증이 수행되는 위치에 따라 달라진다. 예를 들어 데이터는 시스템 도메인에서만 검증할 수 있지만 더 많은 기능적 특성은 시험과 검증이 모두 필요하다. 이 경우 HR이 표시되었지만 선택적으로 R이 될 수 있다.  NOTE 5 기본 무결성을 위한 소프트웨어 아키텍처 사양 및 소프트웨어 설계 사양은 특히 장기 유지보수를 위해 수립하는 것이 중요하다. 정의된 문서는 이을 고려할 수 있다(예: 소프트웨어 요구 사항 사양 내에 관련 정보 추가). | | | | | |

Table A.2 – Software Requirements Specification ([7.2](#_bookmark110))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Formal Methods (based on a mathematical approach) | [D.28](#_bookmark303) | – | R | R | HR | HR |
| 2. Modelling | [Table A.17](#_bookmark253) | R | R | R | HR | HR |
| 3. Structured methodology | [D.52](#_bookmark341) | R | R | R | HR | HR |
| 4. Decision Tables | [D.13](#_bookmark288) | R | R | R | HR | HR |
| Requirements:   1. 소프트웨어 요구사항 명세서에는 자연어로 문제를 서술하고 필요한 정형 또는 비정형 표기법이 포함되어야 한다. 2. 표는 명세를 명확하고 정확하게 정의하기 위한 추가 요구사항이 반영되어 있다. 하나 이상의 이러한 기술은 사용 중인 소프트웨어 안전무결성수준을 충족하도록 선택되어야 한다. | | | | | | |

Table A.3 – Software Architecture ([7.3](#_bookmark121))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Defensive Programming | [D.14](#_bookmark289) | – | HR | HR | HR | HR |
| 2. Fault Detection & Diagnosis | [D.26](#_bookmark301) | – | R | R | HR | HR |
| 3. Error Correcting Codes | [D.19](#_bookmark294) | – | – | – | – | – |
| 4. Error Detecting Codes | [D.19](#_bookmark294) | – | R | R | HR | HR |
| 5. Failure Assertion Programming | [D.24](#_bookmark299) | – | R | R | HR | HR |
| 6. Safety Bag Techniques | [D.47](#_bookmark336) | – | R | R | R | R |
| 7. Diverse Programming | [D.16](#_bookmark291) | – | R | R | HR | HR |
| 8. Recovery Block | [D.44](#_bookmark333) | – | R | R | R | R |
| 9. Backward Recovery | [D.5](#_bookmark280) | – | NR | NR | NR | NR |
| 10. Forward Recovery | [D.30](#_bookmark319) | – | NR | NR | NR | NR |
| 11. Retry Fault Recovery Mechanisms | [D.46](#_bookmark335) | – | R | R | R | R |
| 12. Memorising Executed Cases | [D.36](#_bookmark325) | – | R | R | HR | HR |
| 13. Artificial Intelligence – Fault Correction | [D.1](#_bookmark276) | – | NR | NR | NR | NR |
| 14. Dynamic Reconfiguration of software | [D.17](#_bookmark292) | – | NR | NR | NR | NR |
| 15. Software Error Effect Analysis | [D.25](#_bookmark300) | – | R | R | HR | HR |
| 16. Graceful Degradation | [D.31](#_bookmark320) | – | R | R | HR | HR |
| 17. Information Hiding | [D.33](#_bookmark322) | – | – | – | – | – |
| 18. Information Encapsulation | [D.33](#_bookmark322) | R | HR | HR | HR | HR |
| 19. Fully Defined Interface | [D.38](#_bookmark327) | HR | HR | HR | M | M |
| 20. Formal Methods | [D.28](#_bookmark303) | - | R | R | HR | HR |
| 21. Modelling | [Table A.17](#_bookmark253) | R | R | R | HR | HR |
| 22. Structured Methodology | [D.52](#_bookmark341) | R | HR | HR | HR | HR |
| 23. Modelling supported by computer aided design and specification tools | [Table A.17](#_bookmark253) | R | R | R | HR | HR |
| Requirements:   1. 소프트웨어 안전 무결성 수준 3 및 4에 대한 승인된 기법의 조합은 다음과 같다.   1) 1, 7, 19, 22와 4, 5, 12 또는 21 중 하나;  2) 1, 4, 19, 22와 2, 5, 12, 15 또는 21 중 하나.   1. 소프트웨어 안전무결성수준 1 및 2에 대한 승인된 기법의 조합은 다음과 같다: 1, 19, 22와 2, 4, 5, 7, 12, 15 또는 21 중 하나. 2. 이러한 문제 중 일부는 시스템 수준에서 정의될 수 있다. 3. EN 50159의 요건에 따라 오류 검출 코드(Error detecting codes)를 사용할 수 있다.   NOTE 기법/측정 방법 19는 외부 인터페이스의 용도로 사용된다. | | | | | | |

Table A.4 – Software Design and Implementation ([7.4](#_bookmark146))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Formal Methods | [D.28](#_bookmark303) | – | R | R | HR | HR |
| 2. Modelling | [Table A.17](#_bookmark253) | R | HR | HR | HR | HR |
| 3. Structured methodology | [D.52](#_bookmark341) | R | HR | HR | HR | HR |
| 4. Modular Approach | [D.38](#_bookmark327) | HR | M | M | M | M |
| 5. Components | [Table A.20](#_bookmark256) | HR | HR | HR | HR | HR |
| 6. Design and Coding Standards | [Table A.12](#_bookmark248) | HR | HR | HR | M | M |
| 7. Analysable Programs | [D.2](#_bookmark277) | HR | HR | HR | HR | HR |
| 8. Strongly Typed Programming Language | [D.49](#_bookmark338) | R | HR | HR | HR | HR |
| 9. Structured Programming | [D.53](#_bookmark342) | R | HR | HR | HR | HR |
| 10. Programming Language | [Table A.15](#_bookmark251) | R | HR | HR | HR | HR |
| 11. Language Subset | [D.35](#_bookmark324) | – | – | – | HR | HR |
| 12. Object Oriented Programming | [Table A.22](#_bookmark258) [D.57](#_bookmark346) | R | R | R | R | R |
| 13. Procedural programming | [D.60](#_bookmark349) | R | HR | HR | HR | HR |
| 14. Metaprogramming | [D.59](#_bookmark348) | R | R | R | R | R |
| Requirements:   1. 소프트웨어 안전 무결성 수준 3, 4에 대한 승인된 기법의 조합은 4, 5, 6, 8 이며, 1 또는 2 중 하나이다. 2. 소프트웨어 안전 무결성 수준 1, 2에 대한 승인된 기술의 조합은 3, 4, 5, 6 그리고 8, 9, 10 중 하나이다. 3. 메타프로그래밍은 컴파일 전에 소프트웨어 소스의 코드 생성으로 제한된다. | | | | | | |

Table A.5 – Verification and Testing (6.2 and 7.3, 7.5)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Formal Proof | [D.29](#_bookmark318) | – | R | R | HR | HR |
| 2. Static Analysis | [Table A.19](#_bookmark255) | – | HR | HR | HR | HR |
| 3. Dynamic Analysis and Testing | [Table A.13](#_bookmark249) | – | HR | HR | HR | HR |
| 4. Metrics | [D.37](#_bookmark326) | – | R | R | R | R |
| 5. Traceability | [D.58](#_bookmark347) | R | HR | HR | M | M |
| 6. Software Error Effect Analysis | [D.25](#_bookmark300) | – | R | R | HR | HR |
| 7. Test Coverage for code | [Table A.21](#_bookmark257) | R | HR | HR | HR | HR |
| 8. Functional/ Black-box Testing | [Table A.14](#_bookmark250) | HR | HR | HR | M | M |
| 9. Performance Testing | [Table A.18](#_bookmark254) | – | HR | HR | HR | HR |
| 10. Interface Testing | [D.34](#_bookmark323) | HR | HR | HR | HR | HR |
| Requirements:   1. 소프트웨어 안전 무결성 수준 3, 4에 대한 승인된 기법의 조합은 3, 5, 7, 8 그리고, 1, 2 또는 6 중에 하나이다. 2. 소프트웨어 안전 무결성 수준 1, 2에 대한 승인된 기법의 조합은 5와 2, 3, 8 중에 하나이다. | | | | | | |
| NOTE 1 Techniques/measures의 1, 2, 4, 5, 6 그리고 7 은 검증(verification) 활동임  NOTE 2 Techniques/measures의 3, 8, 9 그리고, 10 은 테스팅 활동임. | | | | | | |

Table A.6 – Integration (7.6)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Functional and Black-box Testing | [Table A.14](#_bookmark250) | HR | HR | HR | HR | HR |
| 2. Performance Testing | [Table A.18](#_bookmark254) | - | R | R | HR | HR |

Table A.7 – Overall Software Testing ([6.2](#_bookmark43) and [7.7](#_bookmark185))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Performance Testing | [Table A.18](#_bookmark254) | – | HR | HR | M | M |
| 2. Functional and Black-box Testing | [Table A.14](#_bookmark250) | HR | HR | HR | M | M |
| 3. Modelling | [Table A.17](#_bookmark253) | – | R | R | R | R |
| Requirement:  a) 소프트웨어 안전 무결성 수준 1, 2의 경우 승인된 기법의 조합은 1, 2이다. | | | | | | |

Table A.8 – Software Analysis Techniques ([6.3](#_bookmark52))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Static Software Analysis | [D.13](#_bookmark288)  [D.37](#_bookmark326)  [Table A.19](#_bookmark255) | R | HR | HR | HR | HR |
| 2. Dynamic Software Analysis | [Table A.13](#_bookmark249) [Table A.14](#_bookmark250) | – | R | R | HR | HR |
| 3. Cause Consequence Diagrams | [D.6](#_bookmark281) | R | R | R | R | R |
| 4. Event Tree Analysis | [D.22](#_bookmark297) | – | R | R | R | R |
| 5. Software Error Effect Analysis | [D.25](#_bookmark300) | – | R | R | HR | HR |
| Requirement:  a) 기법 중 하나 이상은 사용 중인 소프트웨어 안전무결성수준을 충족하도록 선택해야 한다. | | | | | | |

Table A.9 – Software Quality Assurance ([6.5](#_bookmark71))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Accredited to ISO 9001 | [7.1](#_bookmark107) | R | HR | HR | HR | HR |
| 2. Compliant with ISO 9001 | [7.1](#_bookmark107) | M | M | M | M | M |
| 3. Compliant with ISO/IEC 90003 | [7.1](#_bookmark107) | R | R | R | R | R |
| 4. Company Quality System | [7.1](#_bookmark107) | M | M | M | M | M |
| 5. Software Configuration Management | [D.48](#_bookmark337) | M | M | M | M | M |
| 6. Checklists | [D.7](#_bookmark282) | R | HR | HR | HR | HR |
| 7. Traceability | [D.58](#_bookmark347) | R | HR | HR | M | M |
| 8. Data Recording and Analysis | [D.12](#_bookmark287) | HR | HR | HR | M | M |
| Requirement:  a) 이 표는 다른 역할과 모든 단계에 적용되어야 한다. | | | | | | |

Table A.10 – Software Maintenance ([9.2](#_bookmark225))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Impact Analysis | [D.32](#_bookmark321) | R | HR | HR | M | M |
| 2. Data Recording and Analysis | [D.12](#_bookmark287) | HR | HR | HR | M | M |

Table A.11 – Data Preparation Techniques ([8.4](#_bookmark199))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Tabular Specification Methods | [D.68](#_bookmark357) | R | R | R | R | R |
| 2. Application specific language | [D.69](#_bookmark358) | R | R | R | R | R |
| 3. Simulation | [D.42](#_bookmark331) | R | HR | HR | HR | HR |
| 4. Functional testing | [D.42](#_bookmark331) | M | M | M | M | M |
| 5. Checklists | [D.7](#_bookmark282) | R | HR | HR | M | M |
| 6. Fagan inspection | [D.23](#_bookmark298) | – | R | R | R | R |
| 7. Formal design reviews | [D.56](#_bookmark345) | R | HR | HR | HR | HR |
| 8. Formal proof of correctness (of data) | [D.29](#_bookmark318) | – | – | – | HR | HR |
| 9. Walkthrough | [D.56](#_bookmark345) | R | R | R | HR | HR |
| Requirements:  a) 소프트웨어 안전 무결성 수준 1, 2의 경우 승인된 기법의 조합은 1과 4이다.  b) 소프트웨어 안전 무결성 수준 3, 4의 경우 승인된 기법의 조합은 1, 4, 5, 7 또는 2, 3, 6이다.  Note 참조 D.29에 대한 설명은 프로그램에 있는 반면, 이 맥락에서 기법 8은 데이터의 정확성에 대한 공식적(formal) 증명에 적용된다. | | | | | | |

* 1. Detailed tables

Table A.12 – Coding Standards

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Coding Standard | [D.15](#_bookmark290) | HR | HR | HR | M | M |
| 2. Coding Style Guide | [D.15](#_bookmark290) | HR | HR | HR | HR | HR |
| 3. No Dynamic Objects | [D.15](#_bookmark290) | – | R | R | HR | HR |
| 4. No Dynamic Variables | [D.15](#_bookmark290) | – | R | R | HR | HR |
| 5. Limited Use of Pointers | [D.15](#_bookmark290) | – | R | R | R | R |
| 6. Limited Use of Recursion | [D.15](#_bookmark290) | – | R | R | HR | HR |
| 7. No Unconditional Jumps | [D.15](#_bookmark290) | – | HR | HR | HR | HR |
| 8. Limited size and complexity of Functions, Subroutines and Methods | [D.38](#_bookmark327) | HR | HR | HR | HR | HR |
| 9. Entry/Exit Point strategy for Functions, Subroutines and Methods | [D.38](#_bookmark327) | R | HR | HR | HR | HR |
| 9. Limited number of subroutine parameters | [D.38](#_bookmark327) | R | R | R | R | R |
| 10. Limited use of Global Variables | [D.38](#_bookmark327) | HR | HR | HR | M | M |
| Requirement:  a) 기법 3, 4, 5는 검증된 컴파일러 또는 번역기의 일부로 제공될 수 있다. | | | | | | |

Table A.13 – Dynamic Analysis and Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Test Case Execution from Boundary Value Analysis | [D.4](#_bookmark279) | – | HR | HR | HR | HR |
| 2. Test Case Execution from Error Guessing | [D.20](#_bookmark295) | R | R | R | R | R |
| 3. Test Case Execution from Error Seeding | [D.21](#_bookmark296) | – | R | R | R | R |
| 4. Performance Modelling | [D.39](#_bookmark328) | – | R | R | HR | HR |
| 5. Equivalence Classes and Input Partition Testing | [D.18](#_bookmark293) | R | R | R | HR | HR |
| 6. Structure-Based Testing | [D.50](#_bookmark339) | – | R | R | HR | HR |
| Requirement:  a) 테스트 케이스에 대한 분석은 하위 시스템 수준에서 수행되며 명세서 및/또는 명세서 및 코드를 기반으로 한다. | | | | | | |

Table A.14 – Functional/Black Box Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Test Case Execution from Cause Consequence Diagrams | [D.6](#_bookmark281) | – | – | – | R | R |
| 2. Prototyping/Animation | [D.43](#_bookmark332) | – | – | – | R | R |
| 3. Boundary Value Analysis | [D.4](#_bookmark279) | R | HR | HR | HR | HR |
| 4. Equivalence Classes and Input Partition Testing | [D.18](#_bookmark293) | R | HR | HR | HR | HR |
| 5. Process Simulation | [D.42](#_bookmark331) | R | R | R | R | R |
| Requirement:  a) 시뮬레이션의 완전성(completeness)은 소프트웨어 안전 무결성 수준, 복잡성 및 응용 프로그램의 정도에 따라 달라진다. | | | | | | |

Table A.15 – Textual Programming Languages

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. ADA | [D.54](#_bookmark343) | R | HR | HR | HR | HR |
| 2. MODULA-2 | [D.54](#_bookmark343) | R | HR | HR | HR | HR |
| 3. PASCAL | [D.54](#_bookmark343) | R | HR | HR | HR | HR |
| 4. C or C++ | [D.54](#_bookmark343)  [D.35](#_bookmark324) | R | R | R | R | R |
| 5. PL/M | [D.54](#_bookmark343) | R | R | R | NR | NR |
| 6. BASIC | [D.54](#_bookmark343) | R | NR | NR | NR | NR |
| 7. Assembler | [D.54](#_bookmark343) | R | R | R | R | R |
| 8. C# | [D.54](#_bookmark343)  [D.35](#_bookmark324) | R | R | R | R | R |
| 9. JAVA | [D.54](#_bookmark343)  [D.35](#_bookmark324) | R | R | R | R | R |
| 10. Statement List | [D.54](#_bookmark343) | R | R | R | R | R |
| Requirements:   1. 언어 선택은 6.7 및 7.3에 제시된 요건에 근거해야 한다. 2. 특정 프로그래밍 언어를 제외하기 위해 내린 결정을 정당화할 필요는 없다.   NOTE 1 프로그래밍 언어의 적합성 평가에 대한 내용은 D.54 조항 '적합 프로그래밍 언어' 항목을 참조한다.  NOTE 2 특정 언어가 표에 없는 경우 자동으로 제외되지 않습니다. D.54 조항을 준수할 것으로 예상된다.  NOTE 3 응용 프로그램을 실행하는 데 필요한 선택된 언어와 관련된 런타임 시스템은 소프트웨어 안전 무결성 수준에 따라 계속 사용할 수 있어야 한다. | | | | | | |

Table A 16 – Diagrammatic Languages for Application Algorithms

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Functional Block Diagrams | [D.63](#_bookmark352) | R | R | R | R | R |
| 2. Sequential Function Charts | [D.61](#_bookmark350) | – | HR | HR | HR | HR |
| 3. Ladder Diagrams | [D.62](#_bookmark351) | R | R | R | R | R |
| 4. State Charts | [D.64](#_bookmark353) | R | HR | HR | HR | HR |

Table A.17 – Modelling

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Data Modelling | [D.65](#_bookmark354) | R | R | R | HR | HR |
| 2. Data Flow Diagrams | [D.11](#_bookmark286) | – | R | R | HR | HR |
| 3. Control Flow Diagrams | [D.66](#_bookmark355) | R | R | R | HR | HR |
| 4. Finite State Machines or State Transition Diagrams | [D.27](#_bookmark302) | – | HR | HR | HR | HR |
| 5. Time Petri Nets | [D.55](#_bookmark344) | – | R | R | HR | HR |
| 6. Decision/Truth Tables | [D.13](#_bookmark288) | R | R | R | HR | HR |
| 7. Formal Methods | [D.28](#_bookmark303) | – | R | R | HR | HR |
| 8. Performance Modelling | [D.39](#_bookmark328) | – | R | R | HR | HR |
| 9. Prototyping/Animation | [D.43](#_bookmark332) | – | R | R | R | R |
| 10. Structure Diagrams | [D.51](#_bookmark340) | – | R | R | HR | HR |
| 11. Sequence Diagrams | [D.67](#_bookmark356) | R | HR | HR | HR | HR |
| Requirements:   1. 모델링 지침을 정의하고 사용해야 한다. 2. HR기법 중 하나 이상을 선택해야 한다. | | | | | | |

Table A.18 – Performance Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Avalanche/Stress Testing | [D.3](#_bookmark278) | – | R | R | HR | HR |
| 2. Response Timing and Memory Constraints | [D.45](#_bookmark334) | – | HR | HR | HR | HR |
| 3. Performance Requirements | [D.40](#_bookmark329) | – | HR | HR | HR | HR |

Table A.19 – Static Analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Boundary Value Analysis | [D.4](#_bookmark279) | – | R | R | HR | HR |
| 2. Checklists | [D.7](#_bookmark282) | – | R | R | R | R |
| 3. Control Flow Analysis | [D.8](#_bookmark283) | – | HR | HR | HR | HR |
| 4. Data Flow Analysis | [D.10](#_bookmark285) | – | HR | HR | HR | HR |
| 5. Error Guessing | [D.20](#_bookmark295) | – | R | R | R | R |
| 6. Walkthroughs/Design Reviews | [D.56](#_bookmark345) | HR | HR | HR | HR | HR |

Table A.20 – Components

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Information Hiding | [D.33](#_bookmark322) | – | – | – | – | – |
| 2. Information Encapsulation | [D.33](#_bookmark322) | R | HR | HR | HR | HR |
| 3. Parameter Number Limit | [D.38](#_bookmark327) | R | R | R | R | R |
| 4. Fully Defined Interface | [D.38](#_bookmark327) | R | HR | HR | M | M |
| Requirement:  1) 정보 숨기기 및 캡슐화는 데이터 액세스에 대한 일반적인 전략이 없는 경우에만 권장됩니다.  NOTE Technique/measure 4는 내부 인터페이스를 위한 것이다. | | | | | | |

Table A.21 – Test Coverage for Code

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test coverage criterion** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Statement | [D.50](#_bookmark339) | R | HR | HR | HR | HR |
| 2. Branch | [D.50](#_bookmark339) | – | R | R | HR | HR |
| 3. Compound Condition | [D.50](#_bookmark339) | – | R | R | HR | HR |
| 4. Data flow | [D.50](#_bookmark339) | – | R | R | HR | HR |
| 5. Path | [D.50](#_bookmark339) | – | R | R | HR | HR |
| Requirements:   1. 모든 SIL에 대하여, 테스트 실시 시 커버리지를 정량화 해야 한다. 이는 테스트에서 얻은 신뢰도와 추가 기술의 필요성에 대한 판단을 뒷받침할 수 있다. 2. SIL 3 또는 4 시험커버리지는 다음 각 호에 따라 측정하여야 한다.    * 2 와 3; or    * 2 와 4; or   – 5,  또는, 통합 수준에서 테스트 커버리지는 2, 3, 4 또는 5 중 하나 이상에 따라 측정되어야 한다.   1. 이것이 정당화될 수 있다는 점을 고려할 때, 다른 테스트커버리지 기준을 사용할 수 있다. 이러한 기준은 소프트웨어 아키텍처(표 A.3 참조)와 프로그래밍 언어(표 A.15 및 표 A 16 참조)에 따라 달라진다. 2. 테스트할 수 없는 코드는 적절한 기법을 사용하여 올바른 것으로 입증되어야 합니다. 예를 들어 표 A.19의 정적 분석.   NOTE 1 구문 커버리지(Statement coverage)는 2~ 5의 수행에 의해 자동으로 달성된다.  NOTE 2 이 표에서 테스트 커버리지의 기준은 구조 기반 테스팅(코드 기반, 화이트박스)으로 사용된다. 기능시험(명세기반, 블랙박스)의 기법은 표A.14에 제시되어 있다.  NOTE 3 높은 비율의 커버리지는 대개 달성하기 어렵습니다. 경계 값(Clause D.4)과 동등 클래스 및 입력 분할 테스트(Clause D.18)에서 테스트 케이스 실행을 사용하면 적은 수의 테스트로 충분한 커버리지를 달성할 수 있습니다.  NOTE 4 2와 3의 차이는 실제로 프로그래밍 언어의 수준과 복합 조건의 사용에 따라 달라진다. 단일 조건만 사용하는 경우(예: 컴파일 결과), 2와 3은 동일한 것으로 간주한다. | | | | | | |

Table A.22 – Object Oriented Software Architecture

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Traceability of the concept of the application domain to the classes of the architecture | – | R | R | R | HR | HR |
| 2. Use of suitable frames, commonly used  combinations of classes and design patterns | – | R | R | R | HR | HR |
| 3. Object Oriented Detailed Design | [Table](#_bookmark259) [A.23](#_bookmark259) | R | R | R | HR | HR |
| Requirement:  a) 기존 프레임 및 디자인 패턴을 사용할 때 기존 소프트웨어(pre-existing software)의 요구사항이 표 A.22와 같은 프레임 및 패턴에 적용된다.  NOTE 1 객체 지향 접근 방식은 절차적 접근 방식과 다르게 정보를 제시하며, 다음 목록에는 특정 고려 사항이 필요한 권장 사항이 포함되어 있다.  – 클래스 계층을 이해하고 (기존 클래스 라이브러리를 사용하는 경우를 포함하여) 주어진 메서드의 호출에 따라 실행될 소프트웨어 기능을 식별합니다.  – 구조 기반 테스트(표 A.13).  애플리케이션 도메인에서 클래스 아키텍처로의 추적성은 덜 중요하다.  NOTE 2 의도된 소프트웨어의 일부의 경우 유사한 작업을 성공적으로 해결했으며 개발 담당자에게 잘 알려진 기존 소프트웨어에서 프레임이 존재할 수 있다. 그런 다음 해당 프레임을 사용하는 것이 좋은 관행으로 간주된다. | | | | | | |

Table A.23 – Object Oriented Detailed Design

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TECHNIQUE/MEASURE** | **Ref** | **Basic Integrity** | **SIL 1** | **SIL 2** | **SIL 3** | **SIL 4** |
| 1. Classes should have only one objective | – | R | R | R | HR | HR |
| 2. Inheritance used only if the derived class is a refinement of its basic class | – | R | HR | HR | HR | HR |
| 3. Depth of inheritance limited by coding standards | – | R | R | R | HR | HR |
| 4. Overriding of operations (methods) under strict control | – | R | R | R | HR | HR |
| 5. Multiple inheritance used only for interface classes | – | R | HR | HR | HR | HR |
| 6. Inheritance from unknown classes | – | – | – | – | NR | NR |
| Requirements:   1. 한 클래스는 밀접하게 연결된 데이터와 이러한 데이터에 대한 운영을 관리하는 하나의 책임을 갖는 것이 특징이다. 2. 객체 간의 순환 의존을 방지하기 위해 주의가 필요하다. | | | | | | |