

Sample Exam

ISTQB Foundation Level

2011 Syllabus

Version 2.3

Korean Software Testing Qualifications Board

제1장 “테스팅의 기초”

1. 다음 중 소프트웨어 테스팅의 7가지 원리를 가장 잘 설명하고 있는 것은? (K2)

- a) 완벽한 테스트(Exhaustive testing)을 위해서는 자동화된 테스트(Automated tests)가 수동 테스트(Manual tests)보다 적합하다
- b) 충분한 노력과 도구(Tool)에 의한 지원이 있으면 모든 소프트웨어에 대해 완벽한 테스팅이 가능하다
- c) 일반적으로 소프트웨어 시스템에 대해 모든 입력/출력 조합을 테스트 하는 것은 불가능하다
- d) 테스팅의 목적은 결함이 없음(Absence of defects)을 입증하는 것이다

● 해설

- a) 오답 - 수동이든 자동이든 완벽한 테스팅은 불가능하다
- b) 오답 - 테스팅에 투입하는 시간과 노력에 관계없이 완벽한 테스팅이란 불가능하다 (제2 원칙)
- c) 정답 - 제 2 원칙: “모든 가능성(입력과 사전 조건의 모든 조합)을 테스팅하는 것은 지극히 간단한 소프트웨어를 제외하고는 가능하지 않다”
- d) 오답 - 이 발언은 제1 원칙에 위배된다: 테스팅이란 결함의 존재를 보여주기 위함이다: 테스트를 통해 결함의 존재를 확인할 수 있지만, 결함의 부재를 증명할 수는 없다

2. 다음 중 테스트 팀의 가장 중요한 목표는? (K1)

- a) 시스템 테스트 중 컴포넌트 테스트(Component testing)이 충분히 실행되었는지 결정한다
- b) 결함을 식별하고 수정할 수 있도록 가능한 한 많은 장애(Failures)를 일으킨다
- c) 모든 결함이 식별됐음을 증명한다
- d) 남아있는 결함이 어떤 장애도 일으키지 않음을 증명한다

● **해설**

- a) 오답 - 컴포넌트 테스트는 시스템 테스트의 일부가 아니다
- b) 정답 - 테스트 팀의 가장 중요한 역할이다
- c) 오답 - 제 1 원칙에 의하면 완벽한 테스트이란 불가능하므로, 누구도 모든 결함이 식별됐음을 증명할 수 없다
- d) 오답 - 특정 결함이 장애로 이어질지를 알기 위해서는 우선 해당 결함을 발견해야 한다. 즉, 장애를 일으킬만한 결함이 없다고 주장하는 것은 모든 결함이 발견됐다고 말하는 것이고, 이것은 제 1 원칙에 위배된다

3. 기본적인 테스트 프로세스의 테스트 분석 및 설계(Test Analysis and Design) 단계에서 실행될 것으로 기대할 수 있는 작업은? (K1)

- a) 테스트 목표(Test objectives) 정의
- b) 테스트 베이스(Test basis) 리뷰
- c) 테스트 프로시저(Test procedures)로부터 테스트 스위트(Test suites) 생성
- d) 프로세스 개선에서 습득한 교훈(Lesson learned) 분석

● **해설**

- a) 오답 - 해당 활동은 "테스트 계획과 제어" 단계에서 이루어진다 (실라버스 1.4.1 참조)
- b) 정답 - 해당 활동은 "테스트 분석과 설계" 단계에서 이루어진다 (실라버스 1.4.2 참조)
- c) 오답 - 해당 활동은 "테스트 구현과 실행" 단계에서 이루어진다 (실라버스 1.4.3 참조)
- d) 오답 - 해당 활동은 "테스트 마감 활동" 단계에서 이루어진다 (실라버스 1.4.5 참조)

4. 다음은 테스트링 또는 운영 중에 볼 수 있는 문제들이다. 장애(Failure)에 가장 가까운 것은?
(K2)

- a) 사용자가 대화상자(Dialog box)의 특정 옵션을 선택하자 제품이 멈췄다
- b) 빌드에 포함된 소스코드의 특정 파일이 잘못된 버전이었다
- c) 계산 알고리즘이 잘못된 입력 변수를 사용했다
- d) 개발자가 알고리즘의 요구사항을 잘못 해석했다

● 해설

- a) 정답 - 장애란 결함이 외부로 표출되는 것을 말한다. 멈추는 현상은 사용자에게 당연히 노출되는 현상이다
- b) 오답 - 이런 종류의 실수는 외부로 표출되지 않는 경우도 있다 (예: 소스코드의 해당 파일이 기존에 비해서 바뀐 것은 주석 부분만일 경우)
- c) 오답 - 잘못된 입력 변수의 사용이 반드시 외부로 표출되는 장애로 이어지는 것은 아니다 (예: 아무도 해당 알고리즘을 사용하지 않는다면, 사용된 잘못된 입력 변수가 올바른 입력 변수와 비슷한 값을 가질 경우, 해당 알고리즘의 잘못된 결과를 사용하는 사람이 없을 경우). *"소프트웨어, 시스템, 문서의 결함은 장애의 원인이 되지만, 모든 결함이 장애를 일으키는 것은 아니다."*(실라버스 1.1.2 참조)
- d) 오답 - 이런 종류의 실수가 반드시 외부로 표출되는 장애로 이어지는 것은 아니다 (예: 해당 알고리즘을 사용하는 사람이 아무도 없을 경우)

5. 다음 중 개발자와 테스터가 공존하는 팀에서 리뷰나 테스트 중 팀 내 문제(또는 분쟁)로 이어질 수 있는 태도, 자격, 행동 등은 무엇인가? (K1)

- a) 테스터와 개발자가 호기심이 많고 결함 발견에 집중하고 있다
- b) 테스터와 개발자가 장애와 결함을 발견할 수 있는 충분한 자격이 있다
- c) 테스터와 개발자가 결함을 소프트웨어 제품에 대한 비판이 아닌 사람에 대한 비판으로 의사소통을 한다
- d) 테스터가 소프트웨어 제품에 개발자들이 발견하고 수정하지 못한 결함이 있을 것으로 의심한다

● **해설**

- a) 오답 - 분쟁으로 이어질만한 상황이 아니다. 테스트와 개발자는 결함 발견에 집중하고 있어야 한다
- b) 오답 - 테스트와 개발자는 장애와 결함을 발견할 수 있는 충분한 자격을 갖추고 있어야 한다
- c) 정답 - 테스터와 개발자는 서로 협조해야 한다. 결함을 사람에 대한 비판으로 전달할 경우 팀 내 갈등을 유발할 수 있다
- d) 오답 - 테스터의 역할은 개발자가 발견 및 수정하지 못한 소프트웨어의 결함을 식별하는 것이다

6. 다음 문장 중 참(True)인 것은? (K2)

- A) 소프트웨어 테스트는 계약 또는 법적 요구사항을 만족시키기 위해 요구되기도 한다
- B) 소프트웨어 테스트가 필요한 가장 중요한 이유는 개발자가 배포하는 제품의 품질 향상을 위해서다
- C) 엄격한 테스트(Rigorous testing)과 발견된 결함의 수정을 통해 운영환경에서 문제가 발생하는 리스크를 줄일 수 있다
- D) 엄격한 테스트는 간혹 모든 장애가 발견됐음을 증명하는데 사용된다

- a) A, B와 C는 참; D는 거짓
- b) A는 참; B, C와 D는 거짓
- c) A와 C는 참; B와 D는 거짓
- d) C와 D는 참; A와 B는 거짓

● 해설

- A. 참 – 계약 또는 법적 요구사항을 만족시키기 위해 소프트웨어 테스트가 요구될 수 있다
- B. 참 – 소프트웨어 테스트의 주요 목표는 개발자들이 배포하는 제품의 품질 향상이다
- C. 참 – 소프트웨어 테스트의 주요 목표 중 하나는 운영 환경에서 나타날 수 있는 문제의 리스크를 완화하는 것이다
- D. 거짓(False) – 모든 결함이 식별됐다는 것을 증명하는 것은 불가능하다

따라서...

- a) 정답
- b) 오답
- c) 오답
- d) 오답

7. 다음 중 테스트(Testing)과 디버깅(Debugging)의 차이를 가장 잘 기술한 것은? (K2)

- a) 테스트는 결함의 원인을 찾아낸다;
디버깅은 결함을 분석하고 예방활동을 제시한다
- b) 동적 테스트(Dynamic Testing)은 결함으로 인해 발생하는 장애를 보여준다;
디버깅은 소프트웨어에서 발생한 장애의 원인을 발견 및 분석하고 제거한다
- c) 테스트는 결함을 제거한다;
디버깅은 장애의 원인을 식별한다
- d) 동적 테스트는 장애의 원인을 예방한다;
디버깅은 장애를 제거한다

● **해설**

- A) 오답 - 테스트를 통해 결함의 원인을 밝힐 수는 없다
- B) 정답 - 동적 테스트는 결함으로 인한 장애를 식별한다; 디버깅은 소프트웨어 장애를 발견, 분석, 제거한다
- C) 오답 - 테스트가 결함을 제거하지는 않는다
- D) 오답 - 동적 테스트가 장애의 원인을 예방하지는 않는다

제2장 “소프트웨어 수명주기와 테스트”

8. 다음 중 비기능(Non-functional) 테스트를 가장 잘 설명한 것은? (K1)

- a) 비기능 테스트는 통합된 시스템이 요구 명세를 충족하는지 확인하기 위해 테스트하는 것이다
- b) 비기능 테스트는 시스템이 코딩 표준(Coding standards)을 준수하는지 확인하기 위해 테스트하는 것이다
- c) 비기능 테스트는 시스템의 내부 구조를 참조하지 않고 테스트하는 것이다
- d) 비기능 테스트는 사용성(Usability), 신뢰성(Reliability), 유지보수성(Maintainability)과 같은 시스템 속성을 테스트하는 것이다

● 해설

- A) 오답 - 시스템 테스트에 대한 설명이다
- B) 오답 - 화이트박스 테스트의 기능이다
- C) 오답 - 블랙박스 테스트에 대한 설명이다
- D) 정답 - 사용성, 신뢰성, 유지보수성과 같은 시스템 특성을 테스트하는 것은 비기능 테스트이다

9. 소프트웨어 개발 모델(Software development models)을 적용할 때 가장 중요한 사항은?
(K2)

- a) 프로젝트의 상황과 제품 특성에 맞는 모델을 적용해야 한다
- b) 최초로 개발되었고 가장 잘 검증된 폭포수 모델(Waterfall model)을 사용한다
- c) 우선 V-모델(V-model)로 시작하고, 반복적(Iterative) 또는 점증적(Incremental) 모델 중 하나로 이동한다
- d) 모델에 맞게 조직을 바꾼다. 하지만 그 반대는 고려하지 않는다

● 해설

- A) 정답 - 모델은 하나도 빠트리지 않고 따라야 하는 정밀한 프로세스가 아닌 전반적인 지침을 제공하는 것이다
- B) 오답 - 폭포수 모델은 팀이 선택할 수 있는 모델들 중 하나일 뿐이다
- C) 오답 - V-모델은 반복 모델과 접목이 불가능하다. 따라서 설명된 흐름은 불가능하다
- D) 오답 - 상황과 프로젝트에 맞게 모델을 선택해야 하며 반대로 해서는 안 된다

10. 다음 중 좋은 테스트의 특성이면서 모든 소프트웨어 개발 수명주기 모델(Software development life cycle model)에 적용 가능 한 것은? (K1)

- a) 인수 테스트(Acceptance testing)은 항상 최종 테스트 레벨로 적용된다
- b) 개발된 모든 기능을 대상으로 모든 테스트 레벨을 계획하고 완료한다
- c) 테스터는 실행 가능한 첫 번째 코드가 개발되는 즉시 참여한다
- d) 모든 개발 활동에는 그에 상응하는 테스트 활동이 있다

● **해설**

- A) 오답 - 인수 테스트를 하는 프로젝트에만 해당된다. 해당 테스트가 없는 프로젝트도 있다
- B) 오답 - 프로젝트에 따라 특정 테스트 레벨이 없어도 되는 경우가 있다. 예를 들어 제3자로부터 코드를 전달 받는 경우, 컴포넌트 테스트가 필요 없다
- C) 오답 - 테스터는 실행 가능한 코드가 개발되는 시점보다 훨씬 앞서서 개입해야 한다. 예를 들어, 테스터는 요구사항 명세서 리뷰에 참여해야 한다
- D) 정답 - *“성공적인 테스트를 위해서는, 그 개발 수명주기 모델에 관계 없이 다음과 같은 요건들이 필요하다: 모든 개발 활동은 이에 상응하는 테스트 활동을 동반한다”* (실라버스 2.1.3 참조)

11. 다음 중 유지보수 테스트(Maintenance testing)에 해당하는 것은? (K1)

- a) 신규 시스템의 개발 단계에서 수정된 결함 테스트
- b) 기존에 운영되던 시스템의 개선사항 테스트
- c) 인수 테스트 과정에서의 시스템 품질에 대한 불평을 조절
- d) 새로운 시스템 개발 과정에서의 기능 통합

● **해설**

- A) 오답 - “유지보수 테스트”은 새로운 시스템을 테스트하는 것이 아니다
- B) 정답 - 환경 변화 이후 시스템의 수행 능력을 테스트하는 것은 “유지보수 테스트”이다
- C) 오답 - 인수 테스트 중 발견된 장애를 처리하는 것이 “유지보수 테스트”은 아니다
- D) 오답 - 기능의 통합은 테스트 활동이 아니다

12. 다음 문장 중 참(True)인 것은? (K2)

- A. 리그레션 테스트(Regression testing)과 재테스팅(Re-testing)은 동일하다
- B. 리그레션 테스트는 모든 결함이 해결됐는지 보여준다
- C. 리그레션 테스트는 일반적으로 테스트 자동화(Test automation)에 적합하다
- D. 리그레션 테스트는 프로그램의 변경으로 인한 결함을 발견하기 위해 수행된다
- E. 리그레션 테스트는 통합 테스트(Integration testing) 도중 수행해서는 안 된다

- a) A와 B는 참
- b) A, C와 E는 참
- c) C와 D는 참
- d) B, D와 E는 참

● 해설

- A. 거짓 - 리그레션 테스트 변화로 인해 발생했거나 노출된 결함을 발견하기 위해 이미 테스트된 프로그램을 다시 테스트하는 것을 말한다. 재테스팅은 결함이 성공적으로 제거됐음을 확인하기 위해서 수행한다 (실라버스 2.3.4 참조)
- B. 거짓 - 재테스팅에 대한 설명이다
- C. 참 - 리그레션 테스트 세트는 반복적으로 여러 번 실행되고 일반적으로 점진적으로 변화한다. 따라서 리그레션 테스트는 자동화하기에 매우 적합하다 (실라버스 2.3.4 참조)
- D. 참 - 리그레션 테스트의 정의다 (실라버스 2.3.4 참조)
- E. 거짓 - "리그레션 테스트는 모든 테스트 레벨에서 수행 될 수 있으며, 기능, 비기능 그리고 구조적 테스트를 포함한다" (실라버스 2.3.4 참조)

따라서...

- a) 오답
- b) 오답
- c) 정답
- d) 오답

13. 다음 중 컴포넌트 테스트(Component testing)와 시스템 테스트(System testing)의 비교로 옳은 것은? (K2)

- A) 컴포넌트 테스트는 소프트웨어 모듈의 기능, 객체(Program objects), 독립적으로 테스트 가능한 클래스들(Classess) 등을 확인하는 반면, 시스템 테스트는 컴포넌트 간의 인터페이스와 시스템 부분들 간의 상호작용을 확인한다
- B) 컴포넌트 테스트를 위한 테스트 케이스는 컴포넌트 명세, 디자인 명세 또는 데이터 모델에서 도출되는 반면, 시스템 테스트를 위한 테스트 케이스는 요구 명세서, 기능 명세서, 유즈케이스(Use cases)에서 도출된다
- C) 컴포넌트 테스트는 기능 특성에만 초점을 맞추는 반면, 시스템 테스트는 기능적 특성과 비기능적 특성에 초점을 맞춘다
- D) 컴포넌트 테스트는 테스터의 책무이다. 반면 시스템 테스트는 일반적으로 시스템 사용자의 책무이다

● **해설**

- a) 오답 - 컴포넌트 간의 인터페이스와 시스템 부분들 간의 상호작용을 확인하는 것은 시스템 테스트의 목적이 아니라 통합 테스트(Integration tests)의 목적이다
- b) 정답
- c) 오답 - 컴포넌트 테스트는 기능적인 특성에만 초점을 맞추지는 않는다
- d) 오답 - 일반적으로 컴포넌트 테스트는 개발자의 책무다. 반면 시스템 테스트는 테스터의 책무이다

제3장 “정적 기법”

14. 다음 중 공식적인 리뷰(Formal review)의 주요 단계들을 나열한 것은? (K1)

- A) 착수(Initiation), 상태(Status), 개별 준비(Individual preparation), 리뷰 미팅(Review meeting), 재작업(Rework), 후속 처리 확인(Follow up)
- B) 계획(Planning), 개별 준비(Individual preparation), 리뷰 미팅(Review meeting), 재작업(Rework), 종결(Closure), 후속 처리 확인(Follow up)
- C) 계획(Planning), 시작(Kick off), 개별 준비(Individual preparation), 리뷰 미팅(Review meeting), 재작업(Rework), 후속 처리 확인(Follow up)
- D) 개별 준비(Individual preparation), 리뷰 미팅(Review meeting), 재작업(Rework), 종결(Closure), 후속 처리 확인(Follow up), 근본 원인 분석(Root cause analysis)

- **해설**

공식적인 리뷰의 주요 단계는 계획(Planning), 시작(Kick off), 개별 준비(Individual preparation), 리뷰 미팅(Review meeting), 재작업(Rework), 후속 처리 확인(Follow up)이다

따라서...

- a) 오답
- b) 오답
- c) 정답
- d) 오답

15. 아래의 리뷰 유형 중 소프트웨어 프로젝트의 안전 최우선 컴포넌트(Safety critical components) 리뷰를 위해 가장 적합한 것은? (K2)

- A) 비공식 리뷰(Informal review)
- B) 동료 리뷰(Peer review)
- C) 인스펙션(Inspection)
- D) 워크쓰루(Walkthrough)

● **해설**

소프트웨어 프로젝트의 안전 최우선 컴포넌트 리뷰는 좀 더 공식적이고, 문서화된 리뷰가 필요하다

따라서...

- a) 오답
- b) 오답
- c) 정답
- d) 오답

16. 다음 중 도구 지원 정적 분석(Tool-supported Static Analysis)에 대해 잘못 설명한 것은?
(K1)

- A) 도구 지원 정적 분석은 적절한 프로세스를 갖추고 진행하면 예방조치로 사용될 수 있다
- B) 도구 지원 정적 분석은 동적 테스트를 통해서는 쉽게 발견할 수 없는 결함을 찾을 수 있다
- C) 도구 지원 정적 분석은 결함을 조기에 찾아 비용 절감 효과를 가져올 수 있다
- D) 도구 지원 정적 분석은 소프트웨어 장애를 일으키는 좋은 방법이다

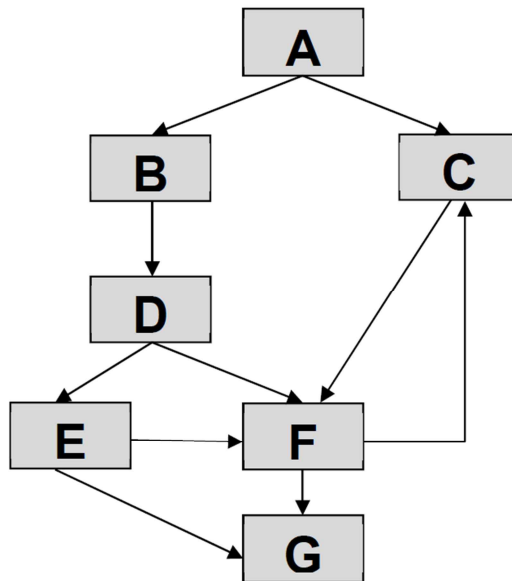
● 해설

- a) 오답 - 도구 지원 정적 분석은 예방조치로 사용될 수 있으므로 이 문장은 맞다
- b) 오답 - 도구 지원 정적 분석은 동적 테스트가 발견하지 못하는 결함을 발견할 수 있으므로 이 문장은 맞다
- c) 오답 - 도구 지원 정적 분석은 결함을 쉽게 발견하기 위한 비용 절감 방법이다
- d) 정답 - 정적 분석 중에는 코드가 실행되지 않으므로 장애를 발견할 수 없다

제4장 “테스트 설계 기법”

17. 프로젝트의 테스트 목표 중 하나는 100% 결정 커버리지 달성이다. 아래의 제어 흐름 그래프(Control flow graph)에 대해 다음 3개의 테스트 케이스가 실행됐다.

테스트 A가 커버하는 경로: A, B, D, E, G
 테스트 B가 커버하는 경로: A, B, D, E, F, G
 테스트 C가 커버하는 경로: A, C, F, C, F, C, F, G



다음 중 결정 커버리지(Decision coverage) 달성에 관해 올바르게 설명한 것은? (K4)

- a) 결정(Decision) D는 완전히 테스트 되지 않았다
- b) 100% 결정 커버리지를 달성했다
- c) 결정 E는 완전히 테스트 되지 않았다
- d) 결정 F는 완전히 테스트 되지 않았다

● 해설

그래프를 보면 조건문이 4개가 있다: A, D, E, F

테스트 A는 $A \rightarrow B$, $D \rightarrow E$, $E \rightarrow G$ 를 커버한다

테스트 B는 $A \rightarrow B$, $D \rightarrow E$, $E \rightarrow F$, $F \rightarrow G$ 를 커버한다

테스트 C는 $A \rightarrow C$, $F \rightarrow C$, $F \rightarrow G$ 를 커버한다

따라서 조건문 A ($A \rightarrow B$, $A \rightarrow C$), 조건문 E ($E \rightarrow G$, $E \rightarrow F$), 조건문 F ($F \rightarrow C$, $F \rightarrow G$)는 커버됐으나 조건문 D는 커버되지 않았다. 즉, $D \rightarrow E$ 는 커버됐으나 $D \rightarrow F$ 는 커버되지 않았다

따라서...

- a) 정답
- b) 오답
- c) 오답
- d) 오답

18. 테스트 중 결함이 발견되었다. 특정 서버로부터 사용자 데이터를 수신하는 도중 시스템 멈춤 현상(System crash)이 일어났다. 결함은 데이터를 전송하는 동안 네트워크의 가용성(Network availability)을 확인하는 코드를 수정하여 해결되었다. 기존의 테스트 케이스는 해당 모듈(Module)의 모든 구문(Statements)을 100% 커버했다. 수정사항을 확인하고 보다 광범위한 커버리지를 얻기 위해 새로운 테스트 몇 개가 설계되어 테스트 세트에 추가되고 실행됐다.

위에서 기술한 테스트는 어떤 종류의 테스트인가? (K3)

- A. 기능 테스트(Functional testing)
- B. 구조적 테스트(Structural testing)
- C. 재테스팅(Re-testing)
- D. 성능 테스트(Performance testing)

- a) A와 B 해당. C와 D는 아님
- b) A와 C 해당. B와 D는 아님
- c) A, B와 C 해당. D는 아님
- d) B, C와 D 해당. A는 아님

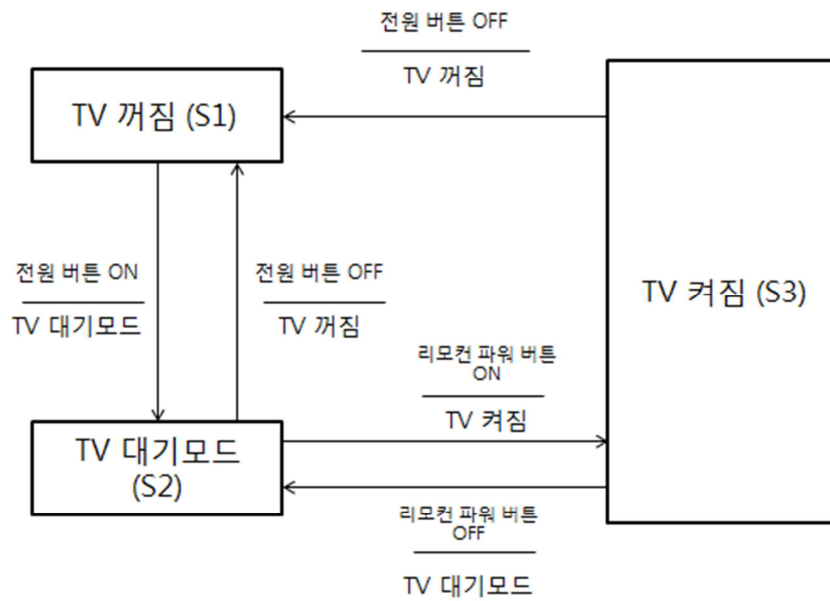
● 해설

- A. 참 - 사용자 데이터를 수신하는 것은 기능 테스트의 대표적인 예제다
- B. 참 - 질문에 "기존의 테스트 케이스는 해당 모듈의 모든 구문을 100% 커버했다"라고 명시되어 있다; 구문 테스트는 구조적 테스트이다
- C. 참 - 질문에 "수정사항을 확인하고 보다 광범위한 커버리지를 얻기 위해 새로운 테스트 몇 개가 설계되어서 테스트 세트에 추가되고 실행됐다"라고 명시되어 있다, 재테스팅을 진행 중이다
- D. 거짓 - 질문에 성능 테스트에 대한 정보는 없다.

따라서...

- a) 오답
- b) 오답
- c) 정답
- d) 오답

19. 다음 상태 전이도(State transition diagram)와 테스트 케이스에 대한 설명으로 올바른 것은? (K3)



테스트 케이스	1	2	3	4	5
시작 상태	S1	S2	S2	S3	S3
입력	전원 버튼 ON	전원 버튼 OFF	리모컨 파워 버튼 ON	리모컨 파워 버튼 OFF	전원 버튼 OFF
기대 결과	TV 대기모드	TV 꺼짐	TV 켜짐	TV 대기모드	TV 꺼짐
최종 상태	S2	S1	S3	S2	S1

- a) 위 테스트 케이스는 상태 전이도의 유효 전이와 비유효 전이를 모두 도출하는데 사용할 수 있다
- b) 위 테스트 케이스는 상태 전이도의 모든 가능한 유효 전이를 커버하고 있다
- c) 위 테스트 케이스는 상태 전이도의 가능한 유효 전이 중 일부만 커버하고 있다
- d) 위 테스트 케이스는 상태 전이도의 변환 순차 쌍(Sequential pairs of transitions)을 나타낸다

● 해설

테스트 케이스는 상태 전이도의 가능한 단일 유효 전이 5개 모두를 커버한다. ($S1 \rightarrow S2$, $S2 \rightarrow S1$, $S2 \rightarrow S3$, $S3 \rightarrow S2$, $S3 \rightarrow S1$)

따라서...

- a) 오답
- b) 정답
- c) 오답
- d) 오답

20. 다음 동등 분할 테스트(Equivalence partitioning test) 기법에 대한 설명 중 참인 것은? (K2)

동등 분할 테스트란...

- A. 같은 동작을 할 것으로 기대되는 가능한 입력들을 집합(클래스)으로 배분한다
- B. 유효하거나 유효하지 않은 분할을 모두 사용한다
- C. 모든 동등한 분할에서 적어도 두 개의 값을 포함해야 한다
- E. 그래픽 UI(Graphical User Interface)의 동등 분할 입력값을 테스트할 때만 사용 할 수 있다

- a) A, B와 D는 참; C 는 거짓
- b) A 는 참; B, C와 D는 거짓
- c) B와 C는 참; A와 D는 거짓
- d) A와 B는 참; C와 D는 거짓

● **해설**

동등 분할이란 같은 동작을 할 것으로 기대되는 가능한 입력들을 집합(클래스)으로 배분하고 유효하거나 유효하지 않은 분할을 모두 사용한다. 또한 모든 클래스의 대표 값 하나만 사용해도 된다. 그래픽 UI의 입력 값을 사용해도 되지만 가장 좋은 방법은 아니다.

따라서...

- A. 참
- B. 참
- C. 거짓
- D. 거짓

따라서...

- a) 오답
- b) 오답
- c) 오답
- d) 정답

21. 다음 중 블랙박스 설계 기법으로 분류될 수 있는 기법을 나열한 목록은? (K1)

- A) 동등 분할, 결정 테이블 테스트, 상태 전이 테스트, 경계값 분석
- B) 동등 분할, 결정 테이블 테스트, 구문 커버리지, 유즈케이스 기반 테스트
- C) 동등 분할, 결정 커버리지 테스트, 유즈케이스 기반 테스트
- D) 동등 분할, 결정 커버리지 테스트, 경계값 분석

● **해설**

- a) 정답 - 4가지 모두 블랙박스 테스트 기법이다 (실라버스 4.3 참조)
- b) 오답 - 구문 커버리지는 화이트박스 테스트 기법이다
- c) 오답 - 결정 커버리지는 화이트박스 테스트 기법이다
- d) 오답 - 결정 커버리지는 화이트박스 테스트 기법이다

22. 직원의 보너스를 계산하려고 한다. 보너스가 음수가 될 수는 없지만 0이 될 수는 있다. 보너스는 고용기간을 기반으로 한다. 2년 이하, 2년 초과 5년 미만, 5년 이상 10년 미만, 10년 이상으로 구분된다. 고용기간에 따라 직원에게 지급될 보너스의 규모가 결정된다.

보너스 계산을 테스트하기 위해 필요한 유효 동등 분할(Equivalence partitions)은 몇 개인가? (K3)

- A) 3
- B) 5
- C) 2
- D) 4

● 해설

- a) 오답 - d)에 대한 해설 참조
- b) 오답 - d)에 대한 해설 참조
- c) 오답 - d)에 대한 해설 참조
- d) 정답 - 문제에서 설명된 분할은 아래와 같다
 - $0 < \text{고용기간} \leq 2$
 - $0 < \text{고용기간} < 5$
 - $5 \leq \text{고용기간} < 10$
 - $10 \leq \text{고용기간}$

23. 다음 유즈케이스로부터 테스트 케이스를 도출하는 것의 이점에 대한 설명 중 참인 것과 거짓인 것을 맞게 표현한 것은? (K3)

- A. 유즈케이스에서 테스트 케이스를 도출하는 것은 시스템 테스트 및 인수 테스트에 도움이 된다
- B. 유즈케이스에서 테스트 케이스를 도출하는 것은 자동화 테스트(Automated testing)에만 도움이 된다
- C. 유즈케이스에서 테스트 케이스를 도출하는 것은 컴포넌트 테스트에 도움이 된다
- D. 유즈케이스에서 테스트 케이스를 도출하는 것은 통합 테스트에 도움이 된다

- a) A와 D는 참; B와 C는 거짓
- b) A는 참; B, C, D는 거짓
- c) B와 D는 참; A와 C는 거짓
- d) A, C와 D는 참; B는 거짓

● 해설

- A. 참 - 유즈케이스는 사용자와 완성된 시스템 간의 상호작용을 설명하므로 시스템 수준 테스트에 가장 적합하다. 또한 "유즈케이스는 고객이나 사용자 그룹이 참여하는 인수테스트(Acceptance Test)를 설계 할 때 매우 유용하다"(실라버스 4.3.5 참조)
- B. 거짓 - 유즈케이스는 자동 뿐만 아니라 수동으로 실행될 수 있다
- C. 거짓 - 컴포넌트 레벨에서의 테스트 케이스는 코드의 커버리지를 높이기 위해 작성된다. 코드로부터 도출되며, 코드의 특정 부분을 실행시킬 유즈케이스에 따라 작성되지는 않는다.
- D. 참 - "유즈케이스는 컴포넌트 사이의 상호작용과 간섭으로 발생하는 통합 결함을 찾는 데 도움이 된다" (실라버스 4.3.5 참조)

따라서...

- a) 정답
- b) 오답
- c) 오답
- d) 오답

24. 다음 중 결함 공격 테스트(Fault attack testing)을 위한 가장 적절한 테스트 베이스는 어느 것인가? (K2)

- A) 경험, 결함과 장애 데이터, 소프트웨어 장애(Software failures)에 대한 지식
- B) 프로젝트 시작 시점에 수행된 리스크 식별
- C) 도메인 전문가가 업무 흐름(Business flow)에서 도출한 유즈케이스
- D) 기존의 시스템과의 비교를 통해 도출한 기대 결과

● 해설

- a) 정답 - "이러한 결함이나 장애 리스트는 경험, 형식에 맞지 않거나 결함을 발생시키는 데이터와 장애 데이터, 왜 소프트웨어가 실패하는지에 대한 일반적인 지식을 근거로 만들어 질 수 있다." (실라버스 4.5 참조)
- b) 오답 - 리스크 식별은 프로젝트에서 관심을 가져야 할 영역/기능을 나타내지 그것들을 어떻게 테스트 해야 하는지를 알려주지 않는다
- c) 오답 - 업무흐름에 대한 테스트는 소프트웨어의 알려진 취약 부분을 대상으로 하지 않고, 단지 특정 유즈케이스가 실행될 수 있는지를 식별하려고 한다
- d) 오답 - 특정 테스트가 실패했는지 여부를 알 수 있는 방법 중 하나에 대한 설명이며 결함 공격과는 아무런 연관이 없다

25. 명세(Specifications)가 제대로 정의되어 있지 않고 시간에 대한 압박을 받고 있는 프로젝트에 참여하고 있다.

다음 테스트 기법 중 가장 적절한 테스트 접근법(Test approach)은? (K1)

- A) 유즈케이스 테스트(Use case testing)
- B) 구문 테스트(Statement testing)
- C) 탐색적 테스트(Exploratory testing)
- D) 결정 테스트(Decision testing)

● 해설

- a) 오답 - 프로젝트의 명세가 정확히 정의되어 있지 않다. 따라서 유즈케이스가 존재할 가능성은 낮다
- b) 오답 - 구문 테스트에는 많이 시간이 필요하다. 이 프로젝트는 시간에 대한 압박을 받고 있으므로 적절하지 않다
- c) 정답 - 명세가 제대로 정의되어 있지 않고 시간에 대한 압박을 받고 있는 경우에 탐색적 테스트가 적절하다
- d) 오답 - 결정 테스트에는 많은 시간이 필요하다. 이 프로젝트는 시간에 대한 압박을 받고 있으므로 적절하지 않다

26. 다음 테스트 기법 중 화이트박스 기법(White-box technique)은? (K1)

- A) 결정 테스트
- B) 경계값 분석
- C) 동등 분할
- D) 상태 전이 테스트

● 해설

- a) 정답 - 결정 테스트는 화이트박스 기법이다
- b) 오답 - BVA는 블랙박스 기법이다
- c) 오답 - 동등 분할은 블랙박스 기법이다
- d) 오답 - 상태 전이 테스트는 블랙박스 기법이다

27. 명세 기반 테스트를 시작하려고 한다. 0보다 큰 두 정수(A와 B)의 최대 공약수(Greatest common divisor, GCD)를 계산하는 프로그램이 테스트 대상 시스템이다.

calcGCD (A, B);

다음의 테스트 케이스(TC)가 명시되었다.

테스트 케이스	A	B
1	1	1
2	INT_MAX	INT_MAX
3	1	0
4	0	1
5	INT_MAX-1	1
6	1	INT_MAX-1

INT_MAX : 가장 큰 정수

다음 중 테스트 케이스 1부터 6을 결정하기 위해 어떤 테스트 기법이 적용되었는가? (K3)

- a) 경계값 분석(Boundary value analysis)
- b) 상태 전이 테스트(State transition testing)
- c) 유즈케이스 테스트(Use case testing)
- d) 결정 테이블 테스트(Decision table testing)

● 해설

- a) 정답 – 주어진 값 (0, 1, INT_MAX-1, INT_MAX) 등은 대표적인 경계값이다
- b) 오답 – a)에 대한 해설 참조
- c) 오답 – a)에 대한 해설 참조
- d) 오답 – a)에 대한 해설 참조

28. 회사에서 1년 이상 근무했고 개별적으로 합의된 목표를 달성한 직원에게 보너스가 지급된다. 해당 시스템을 테스트하기 위해 다음 결정 테이블이 설계되었다.

		T1	T2	T3	T4	T5	T6	T7	T8
조건									
조건1	근무기간 1년 이상?	예	아니오	예	아니오	예	아니오	예	아니오
조건2	합의된 목표 존재 여부?	아니오	아니오	예	예	아니오	아니오	예	예
조건3	목표 달성?	아니오	아니오	아니오	아니오	예	예	예	예
결과									
	보너스 지급?	아니오	아니오	아니오	아니오	아니오	아니오	예	아니오

위의 결정 테이블에 표시된 테스트 케이스 중 실제로 나타날 수 없는 상황이므로 제외시켜도 되는 것은? (K3)

- a) T1, T2
- b) T3, T4
- c) T7, T8
- d) T5, T6

● 해설

테스트 케이스의 조건으로부터 추론해야 한다. 테스트 케이스 T5와 T6의 경우, 목표가 달성됐다고 나타나 있으나 합의된 목표는 없다고 되어있다. 이런 상황은 발생할 수 없으므로 해당 테스트 케이스는 제외시켜도 된다

따라서...

- a) 오답
- b) 오답
- c) 오답
- d) 정답

제5장 “테스트 관리”

29. 다음 중 테스트 매니저(Test manager)와 테스터(Tester)간의 역할 구분을 가장 잘 설명한 것은? (K1)

- A) 테스트 매니저는 테스트 활동을 계획하고 준수할 표준을 선택하고, 테스터는 사용할 도구 및 제어수단을 선택한다
- B) 테스트 매니저는 테스트 활동을 계획, 구성, 제어하고, 테스터는 테스트를 도출 및 실행한다
- C) 테스트 매니저는 테스트 활동을 계획, 모니터링, 제어하고, 테스터는 테스트 설계 및 테스트 대상 승인에 대한 결정을 한다
- D) 테스트 매니저는 테스트를 계획 및 준비하고 테스트 케이스를 작성하며, 테스터는 테스트의 우선순위를 정하고 실행한다

● 해설

- a) 오답 - 테스트 도구 선정은 테스트 매니저의 역할이다 (실라버스 5.1.2 참조)
- b) 정답 - 실라버스 5.1.2 참조
- c) 오답 - 테스트 대상 승인에 대한 결정은 테스트 매니저의 역할이다
- d) 오답 - 테스트 매니저는 테스트 케이스를 작성하지 않는다

30. 다음 중 제품 리스크(Product risk)로 분류 될 수 있는 것은? (K2)

- A) 요구사항, 설계, 코드, 테스트의 낮은 품질
- B) 정치적 문제와 제품의 매우 복잡한 영역의 개발 지연
- C) 오류가 발생하기 쉬운 영역, 사용자의 잠재적 피해, 불완전한 제품 특성
- D) 완성도 높은 요구사항을 정의하는데 있어서의 문제, 소프트웨어나 시스템의 잠재적인 장애 영역

● **해설**

- a) 오답 - 낮은 품질의 요구사항은 프로그램 리스크다 (실라버스 5.5.1 참조)
- b) 오답 - 나열된 모든 항목은 프로그램 리스크다
- c) 정답 - 나열된 모든 항목은 제품 리스크다 (실라버스 5.5.2 참조)
- d) 오답 - 요구사항에 대한 문제는 프로그램 리스크다 (실라버스 5.5.1 참조)

31. 다음 중 일반적인 테스트 완료 조건(Test exit criteria)은? (K2)

- A) 테스트 커버리지 측정치, 신뢰성 측정치, 테스트 비용, 일정, 결함의 수정 정도, 잔여 리스크
- B) 테스트 커버리지 측정치, 신뢰성 측정치, 테스터 독립성 수준, 제품의 완성도
- C) 테스트 커버리지 측정치, 신뢰성 측정치, 테스트 비용, 테스트 가능한 코드의 존재여부, 출시 시기, 제품의 완성도
- D) 출시 시기, 잔여 결함, 테스터 자격, 테스터 독립성 수준, 테스트 커버리지 측정치, 테스트 비용

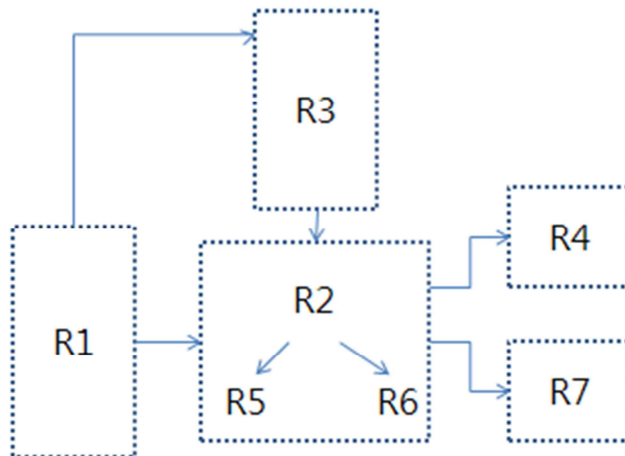
● **해설**

- a) 정답 - 실라버스 5.2.4 참조
- b) 오답 - 테스터 독립성 수준은 완료 조건과 관계없다
- c) 오답 - "테스트 가능한 코드의 존재 여부"는 진입 조건이다
- d) 오답 - 테스터 독립성 정도와 테스터 자격은 완료 조건과 관계없다

32. 당신은 테스트 매니저이며, 다음과 같은 요구사항을 테스트하려고 한다.

- R1 - 프로세스 이상 현상(Process Anomalies)
- R2 - 동기화(Synchronization)
- R3 - 확인(Confirmation)
- R4 - 이슈(Issues)
- R5 - 재무 데이터(Financial Data)
- R6 - 도표 데이터(Diagram Data)
- R7 - 사용자 프로필에 대한 변경사항(Changes to the User Profile)

테스트 요구사항 간의 논리적 종속성은 "R1->R3", 즉 "R1은 R3의 사전조건이다"와 같이 표시된다.



요구사항의 종속성을 고려했을 때, 테스트 실행 순서로 올바른 것은? (K3)

- a) R3 → R2 → R1 → R7 → R5 → R6 → R4
- b) R2 → R5 → R5 → R4 → R7 → R1 → R3
- c) R1 → R3 → R2 → R5 → R6 → R4 → R7
- d) R1 → R2 → R5 → R6 → R3 → R4 → R7

● 해설

- a) 오답 - 다른 모든 요구사항의 사전조건은 R1이다. 따라서 R1으로 시작하지 않는 것은 오답이다
- b) 오답 - 다른 모든 요구사항의 사전조건은 R1이다. 따라서 R1으로 시작하지 않는 것은 오답이다
- c) 정답 - 종속성 고려해서 테스트 케이스가 나열되어 있다
- d) 오답 - R2의 사전조건은 R3이다. 따라서 R2 이전에 R3가 테스트되어야 한다

33. 독립적인 테스트(Independent testing)이 갖는 이점은? (K1)

- A) 테스터가 개발자를 방해하지 않기 때문에 더 많은 일이 처리된다
- B) 독립적인 테스터는 선입견을 갖지 않으며, 개발자와는 다른 결함을 찾는다
- C) 독립적인 테스터는 별도의 교육과 훈련이 필요 없다
- D) 독립적인 테스터는 인시던트(Incident) 관리 프로세스의 병목현상을 완화시킨다

● 해설

- a) 오답 - 독립성이 협조의 상실을 의미하지는 않는다
- b) 정답 - 독립성이 필요한 이유 중 하나다
- c) 오답 - 교육과 훈련이 테스터에게 필요하다
- d) 오답 - 독립적인 테스터와 인시던트 관리 프로세스 사이에는 아무런 연관성이 없다

34. 다음 중 프로젝트 리스크(Project risk)로 분류 될 수 있는 것은? (K1)

- A) 스킬과 인력 부족
- B) 불완전한 소프트웨어 특성 (예: 사용성)
- C) 장애가 발생하기 쉬운 소프트웨어 출하
- D) 신뢰성 결함(버그)의 가능성

● **해설**

- a) 정답 - 스킬과 인력 부족은 프로젝트 리스크의 대표적인 예이다
- b) 오답 - 제품 리스크
- c) 오답 - 제품 리스크
- d) 오답 - 제품 리스크

35. 당신은 테스트 매니저이며, 테스트 요약 보고서를 요청 받았다. IEEE 8299에 의하면, 보고서에 포함되어야 할 가장 중요한 정보는 무엇인가? (K2)

- A) 실행한 테스트 케이스의 수와 각각의 결과
- B) 목표 달성의 측면에서 주요 테스트 활동, 사건, 진행 상황 등에 대한 요약
- C) 각 개발 산출물의 전반적 평가
- D) 테스트 활동을 지원하기 위해 테스트 팀원이 받은 교육 훈련

● **해설**

- a) 오답 - IEEE 829에 의하면 테스트 요약 보고서에 실행한 테스트 케이스의 수와 각각의 결과가 포함될 수 있지만, 이것이 가장 중요한 정보는 아니다
- b) 정답 - 테스트 요약 보고서에는 목표 달성의 측면에서 주요 테스트 활동, 사건, 진행 상황 등의 정보가 반드시 포함되어야 한다
- c) 오답 - 각 개발 산출물에 대한 평가는 테스트 요약 보고서에 포함되지 않는다
- d) 오답 - 테스트 요약 보고서에 교육 관련 사항은 없다

36. 당신은 안전-최우선 소프트웨어 개발 프로젝트를 수행하는 테스터다. 테스트 실행 중 기대와 다른 결과가 나타난 테스트 케이스가 있어 인시던트 보고서(Incident report)를 작성하려고 한다. IEEE 829에 의하면 인시던트 보고서에 포함되어야 할 가장 중요한 정보는 무엇인가? (K3)

- A) 영향도(Impact), 인시던트 설명, 날짜, 당신의 이름
- B) 리포트 고유 ID, 필요한 특별 요구사항, 결함 원인 제공자
- C) 테스트 대상 전달 항목(Transmitted items), 당신의 이름, 결함의 근본원인에 대한 당신의 의견
- D) 인시던트 설명, 개발 환경, 테스트의 기대 결과

● 해설

- a) 정답 - 안전 최우선 소프트웨어의 인시던트 보고서에 포함되어야 할 가장 중요한 정보는 영향도이다.
- b) 오답 - 인시던트 보고서에 포함되어야 할 내용이지만 영향도 만큼 중요하지는 않다.
a)에 대한 해설 참조
- c) 오답 -인시던트 보고서에는 사실을 기반으로 한 정보가 포함되어야지 테스터의 근본 원인에 대한 '의견'을 기록하는 것은 아니다. a)에 대한 해설 참조
- d) 오답 - 인시던트 보고서에 포함되어야 할 내용이지만 영향도 만큼 중요하지는 않다.
a)에 대한 해설 참조

제6장 “테스트 지원 도구”

37. 조직에서 테스트 도구 도입 시 권장할 수 있는 원칙을 아래 목록에서 고르시오. (K1)

1. 도구 도입을 동시에 전체 조직을 대상으로 한다
2. 파일럿 프로젝트로 시작한다
3. 도구 사용에 적합하도록 프로세스를 변경하고 개선한다
4. 새로운 사용자를 위한 교육과 코칭을 제공한다
5. 각 팀이 도구의 사용 방법을 각자 결정하도록 한다
6. 도구 사용 비용이 초기 구매(획득) 비용을 초과하지 않는지 모니터링한다
7. 모든 팀으로부터 각 팀에서 배운 교훈을 수집한다

- a) 1, 2, 4, 5
- b) 2, 5, 6
- c) 2, 3, 4, 7
- d) 1, 6, 7

● 해설

- 1- 거짓 - 전체 조직으로 확대하기 전에 우선 파일럿 배포를 하는 것이 권장된다 (실라버스 6.3 참조)
- 2- 참 - 1에 대한 해설 참조
- 3- 참 - “*현 프로세스나 업무에 도구를 어떻게 적용할 수 있는지 평가하고 나서 개선되어야 할 부분을 결정하라*” (실라버스 6.3 참조)
- 4- 참 - 교육 제공은 배포의 주요 성공 요인 중 하나다 (실라버스 6.3 참조)
- 5- 거짓 - 모든 사람들에게 각자 도구를 어떻게 활용할지를 결정하도록 하면 엉망이 될 수 있다. “*사용 가이드라인 정의*”가 배포의 성공 요인 중 하나다 (실라버스 6.3 참조)
- 6- 거짓 - 도구의 배포에 드는 비용이 도구의 구매 비용만이 아니다. 이 점을 인지하지 못하는 것은 도구 배포의 리스크 중 하나다 (실라버스 6.1 참조)
- 7- 참 - “*모든 팀으로부터 사용 후 교훈 수집*”은 배포의 성공 요인 중 하나다 (실라버스 6.3 참조)

따라서...

- a) 오답
- b) 오답
- c) 정답
- d) 오답

38. 다음 중 키워드 주도 테스트(Keyword-driven test) 실행 도구의 특성을 가장 잘 설명한 것은? (K1)

- A) 테스트 입력 데이터, 액션 워드(Action word), 기대 결과를 포함하는 테이블로 테스트 중인 시스템의 실행을 제어한다
- B) 테스트의 행동을 여러 번 재실행되는 스크립트로 자동화한다
- C) 테스트의 행동을 여러 세트의 테스트 입력데이터로 실행되는 스크립트로 자동화한다
- D) 테스트 결과를 기록하고, 테스트 결과와 텍스트 파일로 저장된 기대결과를 비교할 수 있다

● 해설

- a) 정답 - "키워드 주도 테스트 접근법에서 사용되는 스프레드시트에는 수행할 동작을 나타내는 키워드(액션 워드)와 테스트 데이터가 포함되어 있다" (실라버스 6.2.2 참조)
- b) 오답 - 스크립트 테스트 자동화에 대한 설명이다
- c) 오답 - 데이터 주도 테스트 자동화에 대한 설명이다
- d) 오답 - 테스트 자동화 프레임워크 중 일부에 대한 설명이다

39. 도구 평가(Tool evaluation)를 위한 파일럿 프로젝트의 목표가 아닌 것은? (K1)

- A) 도구가 기존의 프로세스와 방법에 어떻게 부합하는지 평가하기 위해
- B) 도구와 테스트웨어의 사용, 관리, 저장, 유지보수를 결정하기 위해
- C) 적절한 비용으로 도구 사용의 기대 효과를 달성 할 수 있는지 평가하기 위해
- D) 파일럿 프로젝트에서의 결함률을 줄이기 위해

● **해설**

- a) 오답 - 참 (실라버스 6.3 참조)
- b) 오답 - 참 (실라버스 6.3 참조)
- c) 오답 - 참 (실라버스 6.3 참조)
- d) 정답 - 결함의 수를 줄이는 것이 파일럿 프로젝트의 목표는 아니다

40. 다음은 소프트웨어 개발 및 테스트 조직이 달성하려는 테스트 효율 개선 목표들이다. 이러한 목표 중 테스트 관리 도구(Test management tool)의 사용이 가장 많은 도움을 줄 수 있는 것은? (K2)

- A) 요구사항, 테스트, 결함(버그) 간의 추적성 구축
- B) 장애(Failure)를 식별하는 테스트 능력 최적화
- C) 보다 신속한 결함 해결
- D) 실행 대상 테스트 케이스 선택의 자동화

● **해설**

- a) 정답 - 요구사항과 테스트 간의 추적성은 테스트 관리 도구의 기능이다
- b) 오답 - 테스트 관리 도구로 가능하지 않다
- c) 오답 - 테스트 관리 도구가 보다 신속한 결함 해결의 주요 원인은 아니다
- d) 오답 - 테스트 관리 도구가 테스트 케이스를 선택하지는 않는다