ISTQB® Certified Tester Syllabus Foundation Level

Compatible with Syllabus version 4.0

International Software Testing Qualifications Board

Sample Exam B - Answers

(샘플문제 B - 해설)

Released Version 1.4

Translated Version 1.1

(사)케이에스티큐비 KSTQB

Korean Software Testing Qualifications Board

서문 Introduction

이 문서의 목적

이 샘플 시험문제지의 예시 문항과 답안 및 관련 정답은 다음과 같은 목적으로 주제별 전문가와 숙련된 문제 작성자로 구성된 팀에 의해 작성되었습니다:

- ISTQB® 회원 위원회 및 시험 기관의 문제 출제 활동 지원
- 교육 제공자 및 시험 응시자에게 시험 문제의 예시 제공

이 문제들은 공식 시험에서 그대로 사용할 수 없습니다.

실제 시험에는 다양한 문제가 포함될 수 있으며, 이 샘플 시험문제지는 출제 가능한 모든 문제 유형, 스타일 또는 길이의 예시를 갖고 있지 않으며, 공식 시험보다 더 어렵거나 쉬울 수 있습니다.

지침 Instructions

이 문서에서는 아래와 같은 사항을 제공합니다:

- 각 정답에 대해 다음을 포함하는 정답 키워드 표:
  - K-레벨, 학습 목표(LO: Learning Objective), 및 문제 당 점수(Point value)
- 각 정답에 대해 다음을 포함하는 추가 문제에 대한 정답 키워드 표:
  - K-레벨, 학습 목표(LO), 및 문제 당 점수
- 모든 문제를 포함하는 답안 세트:
  - 정답
  - 각 옵션 답안에 대한 근거 해설
  - K-레벨, 학습 목표, 및 문제 당 점수
- 모든 문제를 포함하는 추가 답안 세트 [모든 샘플 시험에 적용되는 것은 아님*]:
  - 정답
  - 각 옵션 답안에 대한 근거
  - K-레벨, 학습 목표, 및 문제 당 점수
- 처음 40개의 문항과 그 정답은 시험 구조 및 규칙에 따라 배열되어 있으므로 샘플 시험과 유사합니다. "추가 샘플 문제에 대한 정답" 블록에는 샘플 시험의 일부가 아니지만 학습자가 관련 분야에서 더 깊은 지식을 얻는 데 도움이 될 수 있는 추가 문제에 대한 답변이 포함되어 있습니다.

- 문제은 별도의 문서에 들어 있습니다.

정답표

문제 번호(#)	정답	학습 목표(LO)	K-레벨	배점
1	d	FL-1.2.1	K2	1
2	b	FL-1.2.2	K1	1
3	d	FL-1.3.1	K2	1
4	a	FL-1.4.1	K2	1
5	c	FL-1.4.2	K2	1
6	b	FL-1.4.4	K2	1
7	b	FL-1.5.1	K2	1
8	d	FL-1.5.2	K1	1
9	b	FL-2.1.1	K2	1
10	b	FL-2.1.2	K1	1
11	a	FL-2.1.3	K1	1
12	b	FL-2.1.4	K2	1
13	a	FL-2.2.1	K2	1
14	d	FL-2.2.3	K2	1
15	b	FL-3.1.3	K2	1
16	c	FL-3.2.1	K1	1
17	d	FL-3.2.2	K2	1
18	c	FL-3.2.3	K1	1
19	d	FL-4.1.1	K2	1
20	a	FL-4.2.1	K3	1

문제 번호(#)	정답	학습 목표(LO)	K-레벨	배점
21	d	FL-4.2.2	K3	1
22	b	FL-4.2.3	K3	1
23	c	FL-4.2.4	K3	1
24	b	FL-4.3.1	K2	1
25	c	FL-4.3.2	K2	1
26	a, e	FL-4.4.2	K2	1
27	d	FL-4.4.3	K2	1
28	b	FL-4.5.2	K2	1
29	d	FL-4.5.3	K3	1
30	a	FL-5.1.3	K2	1
31	b	FL-5.1.4	K3	1
32	a	FL-5.1.5	K3	1
33	d	FL-5.1.7	K2	1
34	c	FL-5.2.4	K2	1
35	a	FL-5.3.1	K1	1
36	a	FL-5.3.3	K2	1
37	a	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	c	FL-6.1.1	K2	1
40	a	FL-6.2.1	K1	1

정답

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
1	d	a) 정답이 아닙니다. 결함 주입 등 동적 테스팅으로 사용자는 절대 일어나게 할 수 없는 상황을 일어나게 해서 테스트 대상에 문제가 생기게 하는 경우가 많습니다. 실제 사용자는 절대 접하지 않을 장애라면 그것을 식별해 내는 것이 그다지 유용하지 않을 수 있습니다. 테스팅의 궁극적인 목적은 최종 사용자를 위해 작업 산출물의 품질을 개선하는 것입니다. 실제 사용자에게 나타날 수 없는 장애를 찾기 위해 테스트에 시간을 쓴다면 테스터의 시간을 효율적으로 사용하는 것이 아닙니다. b) 정답이 아닙니다. 정적 분석 등 정적 테스팅은 개발자가 동적 테스팅을 통해 찾을 수 있는 것보다 일찍 프로그램 코드에서 결함을 식별하기 위해 사용합니다. 정적 테스팅 (그리고 정적 분석)은 동적 테스팅으로 발견할 수 있는 장애가 아닌 결함을 찾는다는 점을 유의해야 합니다. 여기서는 '장애'라는 용어의 사용이 이 선택지를 틀리게 만듭니다. c) 정답이 아닙니다. 정적 분석은 코드에서 이상 현상을 직접적으로 식별합니다. 이런 이상 현상은 결함일 수 있습니다. 이런 이상 현상 식별은 보통 개발자를 위한 것이지 고객을 위한 것이 아닙니다. 출력을 제공하지 않는 요소에 정적 분석을 적용해서 출시에 대한 근거를 찾는다는 것은 말이 되지 않습니다. d) 정답입니다. 리뷰는 소프트웨어 개발 수명주기 초기부터 적용할 수 있는 정적 테스팅으로 이후 개발 활동에서 잘못된 요구사항으로 인해 낭비되는 노력을 막기 위해 결함을 찾는데 사용됩니다. 결함을 조기에 발견하고 제거하지 않으면 잘못된 요구사항을 기반으로 만들어진 설계와 코드 등 작업 산출물도 수정해야 할 것입니다.	FL-1.2.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
2	b	a) 정답이 아닙니다. QA는 오류와 결함을 방지하기 위해 예방적 접근 방식을 사용해 프로세스 개선과 구현에 중점을 두는 반면, 테스팅은 결함을 탐지하는 데 사용하는 QC의 한 유형입니다. b) 정답입니다. QC는 제품 결함을 식별하고 수정하는 데 중점을 두어 적절한 품질 수준을 달성하는 것을 목표로 합니다. 테스팅은 QC의 중요한 부분이며 결함을 발견하는 데 도움이 됩니다. c) 정답이 아닙니다. 테스팅이 QC의 중요한 부분이고 결함을 발견하는 데 도움이 되지만, 모델 검증이나 정확성 증명과 같은 정형 기법, 시뮬레이션, 프로토타이핑 등 QC에 활용되는 다른 (테스팅이 아닌) 기법도 있습니다. d) 정답이 아닙니다. QA는 오류와 결함을 방지하기 위해 예방적 접근 방식을 사용해 프로세스 개선과 구현에 중점을 두는 반면, 테스팅은 결함을 탐지하는 데 사용하는 QC의 한 유형입니다.	FL-1.2.2	K1	1

3	d	'완벽한 테스팅은 불가능하다'는 원리는 간단한 경우를 제외하고 모든 상황에서 가능한 모든 테스트 입력값의 변형을 테스트하는 것이 불가능하다는 사실과 관련이 있습니다. 대신 테스팅은 테스트 기법, 테스트 케이스 우선순위화, 리스크 기반 테스팅을 활용해 가능한 입력값 중 표본을 추출하고 테스트 노력을 집중시키게 됩니다. a) 정답이 아닙니다. 이 원리는 간단한 경우를 제외하고는 모든 것을 테스트하는 것이 불가능하다고 말합니다. 모든 것을 테스트한다는 것은 모든 상황에서 가능한 모든 테스트 입력값의 변형을 테스트해야 한다는 의미인데, 이는 일반적으로 불가능합니다. 가능한 테스트 케이스의 수가 무한하기 때문입니다. 명시된 모든 가능한 출력값을 테스트한다고 해서 이 문제가 해결되는 것은 아닙니다. 입력값과 명시된 출력값의 관계는 테스트 대상마다 다를 수 있습니다. 때로는 명시된 가능 출력값의 수가 무한할 수도 있고(예: 실수를 나타내는 여러 변수가 있는 경우), 또 때로는 단일 변수가 참 또는 거짓인 것처럼 명시된 출력값이 두 개뿐일 수도 있습니다. b) 정답이 아닙니다. 이 원리는 모든 상황에서 가능한 모든 테스트 입력값의 변형을 테스트하는 것이 불가능하다고 말합니다. 간단한 시스템을 제외하고 실제로 가능한 수가 무한하기 때문입니다. 따라서 모든 가능한 테스트 입력값 변형을 문서화하는 것은 불가능합니다. 그렇게 하려면 무한한 시간이 걸릴 것입니다. c) 정답이 아닙니다. 리뷰와 다른 정적 테스팅 접근법으로 테스팅을 최대한 빨리 시작하는 것은 가능한 테스트 케이스가 너무 많다는 문제를 해결하지 못합니다. '조기 테스팅은 시간과 비용을 절약한다'는 원칙은 결함을 조기에 수정해 파생 작업 산출물에서 후속 결함이 발생하는 것을 방지함으로써 비용을 줄이고 장애 발생 가능성을 낮추는 것과 관련이 있습니다. d) 정답입니다. 테스트 케이스를 도출하기 위해 동등 분할과 경계값 분석을 사용하는 것은 해당 원리에 대응하는 한 가지 방법인 모든 가능한 테스트 케이스에서 유한한 부분 집합을 체계적으로 도출하는 방법을 이런 기법이 제공하기 때문입니다.	FL-1.3.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
4	a	a) 정답입니다. 테스트 설계는 테스트 컨디션을 사용해 테스트 케이스와 기타 필요한 테스트웨어(예: 테스트 데이터 요구사항, 탐색적 테스팅을 위한 테스트 차터)를 만드는 것을 포함합니다. 필요 인프라와 도구를 포함한 테스트 환경 요구사항도 명시합니다. b) 정답이 아닙니다. 테스트 실행은 테스트 케이스를 실행하는 것을 포함하지만 (테스트 절차의 일부로서), 문제에서 언급한 테스트 데이터 요구사항, 테스트 환경 요구사항, 테스트 컨디션 등 다른 테스트웨어를 직접적으로 다루지는 않습니다. c) 정답이 아닙니다. 테스트 분석은 테스트가 필요한 기능을 식별하는 데 사용됩니다. 테스트 베이시스를 분석해 테스트 컨디션을 정의하고, 관련 리스크와 함께 우선순위를 정합니다. 이 활동은 테스트 컨디션은 다루지만, 문제에서 언급한 테스트 데이터 요구사항, 테스트 환경 요구사항, 테스트 케이스 등 다른 테스트웨어는 다루지 않습니다. d) 정답이 아닙니다. 테스트 구현은 테스트 케이스로 테스트 절차(예: 수동 및 자동화된 테스트 스크립트)를 만드는 것을 포함하며, 절차를 테스트 스위트로 조합하기도 합니다. 테스트 절차는 우선순위가 지정되고 테스트 실행 일정에 따라 배치됩니다. 테스트 데이터가 생성되고, 테스트 환경이 구축되며, 설정이 검증됩니다. 이 활동은 테스트 케이스를 분명히 다루고, 테스트 데이터와 테스트 환경을 생성하기 위해 테스트 데이터 요구사항과 테스트 환경 요구사항을 사용할 수 있지만, 테스트 컨디션은 다루지 않습니다.	FL-1.4.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
5	c	a) 정답이 아닙니다. 조직의 마케팅팀은 테스팅을 많이 수행하지 않을 가능성이 높습니다(일부 조직에서는 인수 테스팅에 참여하기도 합니다). 따라서 그들의 평균 경험 수준(대부분 마케팅 분야의 경험)이 주어진 테스트 대상에 대한 테스팅 수행 방식에 영향을 미칠 가능성은 낮습니다. b) 정답이 아닙니다. 새로운 시스템이 자신들을 위해 구축되고 있다는 사용자의 인식 수준은 테스팅 수행 방식에 영향을 미치지 않을 가능성이 높습니다. 테스팅 수행 방식에 영향을 미칠 수 있는 사용자 참여는 테스터, 고객, 프로젝트 관리자가 내린 결정의 결과일 가능성이 더 높습니다. c) 정답입니다. 성능 테스팅팀 구성원의 경력 연수는 팀원이 테스트할 때 적용할 역량과 지식(예: 다양한 도구 및 결함 유형에 대한)을 결정하는 데 도움이 될 것입니다. d) 정답이 아닙니다. 최종 사용자의 조직 구조는 사용자마다(다양할 수 있음) 달라질 것입니다. 따라서 애플리케이션을 테스트할 때 알지 못할 수 있으며, 최종 사용자의 조직 구조는 테스팅 수행 방식에 미치는 영향이 거의 없을 것입니다.	FL-1.4.2	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
6	b	a) 정답이 아닙니다. 완화된 리스크와 합격한 테스트 케이스 간의 추적성은 거의 아무런 정보를 제공하지 않습니다. (테스트로) 완화되기 위해서는 리스크는 연관된 합격한 테스트 케이스가 있어야 합니다. 잔여 리스크를 평가하기 위해서는 모든 리스크와 테스트 결과 간의 추적성이 필요합니다. 그래야 연관된 합격한 테스트가 없는 리스크를 잔여 리스크로 식별할 수 있기 때문입니다. b) 정답입니다. 사용자 요구사항과 테스트 실행 결과 간의 추적성은 어떤 사용자 요구사항이 테스트되었는지 나타내므로 비즈니스 목표 대비 프로젝트 진행 상황(테스팅 관점에서)을 측정하는 수단을 제공합니다. c) 정답이 아닙니다. 실패한 테스트 케이스가 합격한 테스트 케이스보다 테스터의 기술 수준을 더욱 잘 나타낸다고 보기는 어렵습니다. 이는 부분적으로 테스트 목적(예: 신뢰 구축 또는 장애 유발)에 따라 달라질 것입니다. 또한 합격 및 실패한 테스트 케이스에 기반한 이런 식의 테스터 평가는 테스터가 테스트 목적이 아닌 해당 메트릭에 테스팅을 최적화하게 할 수 있어 역효과를 낳을 수 있습니다. d) 정답이 아닙니다. 식별된 리스크와 작성된 테스트 컨디션 간의 추적성은 어떤 테스트 컨디션을 추가로 작성해야 하는지 판단하는 수단을 제공합니다. 테스트할 가치가 있는 리스크를 판단하는 것은 리스크 관리, 특히 리스크 완화 활동의 일부입니다.	FL-1.4.4	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
7	b	a) 정답이 아닙니다. 좋은 의사소통 기술, 경청 기술, 팀워크 능력은 테스터가 모든 이해관계자와 효과적으로 상호작용할 수 있게 해주지만, 한 명의 개발자와 잘 지낼 수 있게 해준 여러 컴퓨터 게임에 대한 해박한 지식은 테스터에게 유용한 일반적인 기술의 예는 아닙니다. b) 정답입니다. 최종 사용자 및 비즈니스 담당자와 소통하고 그들을 이해하는 데 사용할 수 있는 도메인 지식은 테스터에게 필요한 일반적인 기술 중 하나입니다. 조종사 경력이 있는 테스터는 헬리콥터 제어 시스템의 인수 조건을 더 잘 이해할 수 있을 것입니다. c) 정답이 아닙니다. 프로그래밍 기술은 일부 테스트 도구를 활용할 때 효율성을 높일 수 있는 기술 지식으로 간주될 수 있지만, 이런 기술이 비즈니스 분석가와의 의사소통에 도움이 될 가능성이 높지 않습니다. d) 정답이 아닙니다. 철저함, 세부사항에 대한 주의력, 호기심, 찾기 어려운 결함을 식별하기 위한 체계적인 접근법은 모두 테스터에게 유용한 보편적 기술이지만, 탐색적 테스팅을 시작하기 전 테스트 케이스를 만들어낼 가능성은 높지 않습니다. 탐색적 테스팅의 주요 원칙 중 하나는 테스트 케이스를 테스팅 중 생성하는 것이지 사전에 스크립트로 작성하는 것이 아니라는 것입니다.	FL-1.5.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
8	d	a) 정답이 아닙니다. 전체 팀 접근법에서 필요 기술과 지식을 가진 팀원은 어떤 일이든 맡을 수 있지만, 그렇다고 해서 팀원이 언제든 모든 역할을 수행할 수 있는 것은 아닙니다. 일반적으로 자신이 잘하는 역할을 맡으며 모든 팀원이 모든 역할을 수행할 수 있다는 얘기는 없습니다. b) 정답이 아닙니다. 전체 팀 접근법은 단일 팀이 (일반적으로 애자일 소프트웨어 개발에서) 일하는 방식에 적용되며, 큰 프로젝트에서 여러 팀이 함께 일하는 것을 다루지 않습니다. 또한 전체 프로젝트에 하나의 '전체' 팀만 필요하다는 것을 시사하지 않습니다. c) 정답이 아닙니다. 전체 팀 접근법은 모든 팀원이 매 중요한 의사 결정에 참여할 것을 기대하지 않습니다. 예를 들어, 비즈니스 결과에 영향을 미치지 않는 기술적 의사 결정에 비즈니스 담당자(즉, 제품 소유자)가 관여할 필요가 없으며 그렇게 하면 일의 진행 속도가 불필요하게 느려집니다. d) 정답입니다. 전체 팀 접근법은 각 팀원이 가진 다양한 기술을 가장 효과적으로 활용함으로써 팀의 활력을 높이고, 팀 내 의사소통과 협업을 강화하며, 전체 프로젝트에 도움이 되는 팀 시너지를 창출합니다.	FL-1.5.2	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
9	b	a) 정답이 아닙니다. 애자일 소프트웨어 개발에서는 각 반복 주기마다 결과물이 생성되며, 잦은 증분 전달로 인해 광범위한 리그레션 테스팅이 필요합니다. 이 리그레션 테스팅의 일부(또는 전부)가 자동화될 수 있지만, (자동화되든 그렇지 않든) 리그레션 테스팅이 시스템 테스트 자동화로 대체될 수는 없습니다. b) 정답입니다. 순차적 개발 모델을 사용하면 수명주기 초기에는 실행할 수 있는 코드가 없으므로 이 시기에는 정적 테스팅(예: 리뷰)이 수행됩니다. 수명주기 후반에 실행할 수 있는 코드가 생기면 동적 테스팅이 가능합니다. 참고로 동적 테스팅을 위한 준비는 모든 소프트웨어 개발 수명주기에서 초기에 이루어지는 경우가 많습니다. c) 정답이 아닙니다. 애자일 소프트웨어 개발과 같은 반복적 개발 모델을 사용하면 컴포넌트 테스트를 각 반복주기의 리그레션 테스팅에 사용할 수 있습니다. 이 경우 자주 실행해야 하는 이런 컴포넌트 테스트를 자동화하는 것이 좋습니다. 개발자가 이런 컴포넌트 테스트를 수동으로 수행해야 한다는 주장은 설득력이 없습니다. d) 정답이 아닙니다. 대부분의 점진적 개발 모델에서는 각 증분마다 결과물이 생성되므로, 전달되는 각 증분에 대해 모든 테스트 레벨에서 정적 및 동적 테스팅이 모두 필요합니다.	FL-2.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
10	b	a) 정답이 아닙니다. 테스터는 시프트-레프트 접근법의 일환으로 조기 테스팅을 위해 초안이 가용해지는 즉시 작업 산출물을 리뷰해야 합니다. 다음 개발 단계까지 기다린다면 리뷰되지 않고 결함을 포함한 작업 산출물로 불필요한 개발 (및 테스트) 작업이 이루어질 수 있습니다. b) 정답입니다. 테스터는 시프트-레프트 접근법의 일환으로 조기 테스팅을 위해 초안이 가용해지는 즉시 작업 산출물을 리뷰해야 합니다. c) 정답이 아닙니다. 테스터는 일반적으로 테스트 분석의 일환으로 테스트 베이시스를 구성하는 작업 산출물을 리뷰하지, 테스트 분석 및 설계 이전에 리뷰하지는 않습니다. d) 정답이 아닙니다. 테스터는 시프트-레프트 접근법의 일환으로 조기 테스팅을 위해 초안이 가용해지는 즉시 작업 산출물을 리뷰해야 합니다. 작업 산출물이 배포될 때까지 기다린다는 것은 테스터의 리뷰로 발견될 수 있는 결함을 배포된 문서에 둔 채로 배포한다는 것을 의미합니다.	FL-2.1.2	K1	1
11	a	a) 정답입니다. 테스트 주도 개발(TDD)은 널리 알려진 테스트 우선 개발 접근법입니다. b) 정답이 아닙니다. 커버리지 주도 개발(Coverage-Driven Development)은 테스트 우선 개발 접근법이 아닙니다. c) 정답이 아닙니다. 품질 주도 개발(Quality-Driven Development)은 테스트 우선 개발 접근법이 아닙니다. d) 정답이 아닙니다. 기능 주도 개발(Feature-Driven Development)은 (스크럼에서의 사용자 스토리와 다르게) 기능 배포를 중심으로 하는 애자일 소프트웨어 개발 방법론이지만, 테스트 우선 개발 접근법이 아닙니다.	FL-2.1.3	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
12	b	a) 정답이 아닙니다. 데브옵스는 여러 가지 방식으로 테스팅을 향상시킵니다. 예를 들어, 코드 품질에 대한 빠른 피드백 제공, 리그레션 리스크를 최소화하는 자동화된 리그레션 테스팅, 높은 품질의 코드 제출과 컴포넌트 테스트를 통한 시프트-레프트 접근법 촉진 등이 있습니다. 이는 주로 개발자가 새로운 코드와 함께 컴포넌트(단위) 테스트를 제출해야 코드가 빌드에 포함될 수 있는 지속적 통합을 통해 제공됩니다. 따라서 개발자는 컴포넌트 테스팅을 완료해야 합니다. b) 정답입니다. 데브옵스는 여러 가지 방식으로 테스팅을 향상시킵니다. 예를 들어, 코드 품질에 대한 빠른 피드백 제공, 리그레션 리스크를 최소화하는 자동화된 리그레션 테스팅, 높은 품질의 코드 제출과 컴포넌트 테스트를 통한 시프트-레프트 접근법 촉진 등이 있습니다. c) 정답이 아닙니다. 데브옵스는 여러 가지 방식으로 테스팅을 향상시킵니다. 예를 들어, 코드 품질에 대한 빠른 피드백 제공, 리그레션 리스크를 최소화하는 자동화된 리그레션 테스팅, 높은 품질의 코드 제출과 컴포넌트 테스트를 통한 시프트-레프트 접근법 촉진 등이 있습니다. 테스터는 릴리스 테스팅에 더 많은 시간을 할애해 개발자와 운영자를 동등하게 대우하려고 하지 않습니다. 하지만 시프트-라이트 접근법(운영 환경에서의 테스팅)은 사용될 수 있습니다. d) 정답이 아닙니다. 데브옵스의 지속적 통합/지속적 전달(CI/CD)과 같은 자동화된 프로세스는 안정적인 테스트 환경을 용이하게 하고 수동 테스팅의 필요성을 줄여주지만, 사용자 관점에서 수동 테스팅의 중요성을 간과할 위험이 있습니다.	FL-2.1.4	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
13	a	a) 정답입니다. 시스템 테스팅은 완전한 시스템의 동작과 기능을 확인하며 보안 테스팅을 포함한 품질 특성에 대한 비기능 테스팅을 포함합니다. 이런 유형의 테스팅은 시스템 명세를 기반으로 독립적인 테스트팀이 수행하는 경우가 많습니다. b) 정답이 아닙니다. 시스템 통합 테스팅은 다른 시스템 또는 외부 서비스와의 인터페이스를 확인합니다. c) 정답이 아닙니다. 베타 테스팅은 개발 조직 외부의 사람에 의해 외부 사이트에서 수행되는 인수 테스팅의 한 유형입니다. d) 정답이 아닙니다. 컴포넌트 통합 테스팅은 사용자 인터페이스와 데이터베이스와 같은 시스템의 컴포넌트 간 (인터페이스와) 상호작용을 테스트하는 것과 관련이 있습니다.	FL-2.2.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
14	d	a) 정답이 아닙니다. 리그레션 테스트는 프로젝트가 진행됨에 따라 개수가 늘어납니다. 보통 시스템에 변경이 이루어질 때마다 새로운 리그레션 테스트가 필요하기 때문입니다. 마찬가지로 확인 테스트도 일반적으로 프로젝트가 진행됨에 따라 개수가 늘어나는데, 시스템에 수정이 이루어질 때마다 새로운 확인 테스트가 필요하기 때문입니다. b) 정답이 아닙니다. 반대입니다. 테스트 대상을 수정할 때 확인 테스트를 작성하고 실행하며, (이상적인 경우에는) 테스트 대상을 개선(변경)할 때마다 리그레션 테스트를 실행합니다. c) 정답이 아닙니다. 확인 테스팅은 결함이 올바르게 수정되었는지 검증하므로 테스트 대상의 변경사항을 테스트하는 것과 관련이 있습니다. 그러나 리그레션 테스팅은 변경사항(운영 환경의 변경 포함)으로 인해 바뀌지 않은 소프트웨어에 부정적인 영향을 미치지 않았다는 것을 보장하는 것이지 운영 환경이 변경되지 않았음을 확인하는 것이 아닙니다. d) 정답입니다. 리그레션 테스팅은 변경사항이 바뀌지 않은 소프트웨어에 부정적인 영향을 미치지 않았다는 것을 보장합니다. 확인 테스팅은 결함이 올바르게 수정되었는지 검증하며 변경된 코드와 관련이 있습니다.	FL-2.2.3	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
15	b	a) 정답이 아닙니다. 사용자 인터페이스가 제공하는 사용성이 부족한 것은 적절한 체크리스트를 사용한 리뷰로도 탐지할 수 있지만, 소수의 일반 사용자가 사용자 인터페이스를 동적으로 테스트하고 그 사용성에 대한 피드백을 제공해서 식별할 수도 있습니다. b) 정답입니다. 코드 리뷰는 어떤 경로로도 도달할 수 없는 코드를 탐지할 수 있지만, 동적 테스트는 도달 가능한 코드만 실행할 수 있으며 모든 가능한 입력값과 입력 상태의 조합을 실행하지 않고는 코드의 어느 부분에 도달할 수 없다는 것을 판단할 수 없습니다. 실제 코드를 대상으로 이렇게 하는 것은 현실적이지 않습니다. c) 정답이 아닙니다. 예상 사용자 대부분이 느낄 응답 시간 저하는 코드를 실행하지 않고(즉, 정적 테스팅으로) 판단하기 어려우므로, 이 상황은 동적 테스팅으로 결함을 찾을 수 있지만 정적 테스팅은 결함을 찾기 어려울 것입니다. d) 정답이 아닙니다. 필요한 기능을 아는 사람이 코드를 리뷰하면 필요한 기능이 코드에 구현되지 않았음을 탐지할 수 있으며, 동적 테스팅을 사용해서도 이런 필요한 기능이 구현되지 않았음을 판단할 수 있습니다.	FL-3.1.3	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
16	c	a) 정답이 아닙니다. 피드백은 이해관계자(예: 비즈니스 담당자, 최종 사용자)로부터 받는 것이지 개발자로부터 받는 것이 아니므로, 피드백이 어떤 개발자의 생산성이 높거나 낮은지를 관리자에게 알려줄 가능성은 낮습니다. b) 정답이 아닙니다. 이해관계자에게 받는 조기의 잦은 피드백은 프로젝트 관리자가 다양한 이해관계자와의 상호작용 우선순위를 정하는 데 사용되지 않습니다. c) 정답입니다. 소프트웨어 개발 프로세스 초기에 이해관계자로부터 피드백을 조기에 자주 받는 것은 매우 유익할 수 있습니다. 잠재적인 품질 문제에 대한 빠른 의사소통을 용이하게 하고, 요구사항에 대한 오해를 방지하며, 이해관계자 요구사항의 변경사항이 더 빨리 이해되고 구현되도록 보장하기 때문입니다. d) 정답이 아닙니다. 조기에 자주 피드백을 받으면 비용이 많이 드는 재작업과 납기 지연으로 이어질 수 있는 이해관계자의 요구사항을 충족하지 못하는 제품의 개발을 방지할 수 있습니다. 그래서 이상적으로는 지연이 없어야 합니다. 또한 피드백은 이해관계자로부터 받는 것이지 그들에게 주는 것이 아닙니다. 이해관계자에는 최종 사용자도 포함되므로, 최종 사용자가 피드백을 제공하는 것이 최종 사용자의 이해를 돕지는 않을 것입니다.	FL-3.2.1	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
17	d	나열된 각 과업에 대한 설명을 고려한다면: 1. 평가할 품질 특성과 완료 조건 선택 - (계획(Planning, C): 리뷰 범위, 목적, 리뷰할 작업 산출물, 평가할 품질 특성, 중점을 둘 영역, 완료 조건, 표준과 같은 지원 정보, 공수, 일정을 정의한다.) 2. 모든 사람이 작업 산출물에 접근할 수 있음 - (리뷰 착수(Review initiation, B): 모든 참가자가 작업 산출물과 필요한 자원에 접근할 수 있으며, 자신의 역할과 책임을 명확히 이해하고 있는지 확인한다.) 3. 작업 산출물에서 이상 사항을 식별함 - (개별 리뷰(Individual review, A): 체크리스트 기반 또는 시나리오 기반 리뷰와 같은 리뷰 기법을 사용해 작업 산출물의 품질을 평가하고, 이상 사항, 권고 사항, 의문 사항을 식별 및 기록한다.) 4. 이상 사항을 분석하고 논의함 - (논의 및 분석(Communication and analysis, D): 각 이상 사항을 분석하고 논의해 상태, 소유권, 필요한 조치를 결정하고, 리뷰 결정을 내린다. 일반적으로 회의를 통해 이루어진다. 여기에는 후속 리뷰의 필요성을 판단하는 것이 포함될 수 있다.) 따라서: a) 정답이 아닙니다. b) 정답이 아닙니다. c) 정답이 아닙니다. d) 정답입니다. 올바르게 연결된 것은: 1C, 2B, 3A, 4D입니다.	FL-3.2.2	K2	1

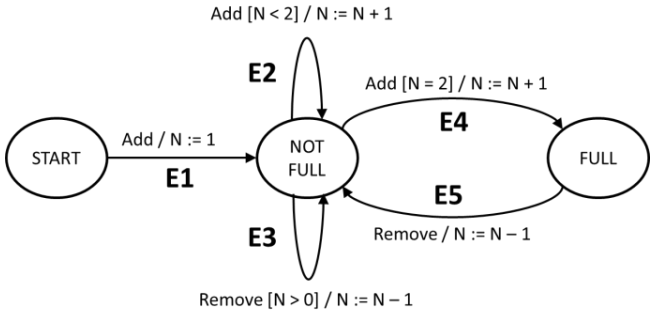
문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
18	c	나열된 각 역할을 고려한다면: 1. 서기(또는 기록자) - 리뷰어로부터 피드백을 수집하고 리뷰 회의 중 내려진 결정사항 및 식별된 새로운 이상 사항과 같은 리뷰 정보를 문서화할 책임이 있다. (리뷰 회의에서 내려진 결정과 새로 발견된 이상 사항과 같은 리뷰 정보를 기록한다 - B) 2. 리뷰 리더 - 리뷰팀 구성원 선정, 리뷰 회의 일정 잡기, 리뷰가 성공적으로 완료되도록 보장하는 등 리뷰 프로세스를 감독할 책임이 있다. (리뷰가 언제 어디서 이루어질지 조직하는 것과 같이 리뷰에 대한 전반적인 책임을 진다 - D) 3. 중재자(또는 퍼실리테이터) - 시간 관리, 토론 중재, 모든 사람이 자유롭게 의견을 말할 수 있는 안전한 환경 조성 등 리뷰 회의가 효과적으로 진행되도록 할 책임이 있다. (리뷰 회의를 효과적으로 진행하고 안전한 리뷰 환경을 조성한다 - A) 4. 관리자 - 무엇을 리뷰할 필요가 있는지 결정하고 인력과 시간과 같은 자원을 할당할 책임이 있다. (리뷰 대상을 결정하고 인력과 시간 등 리뷰에 필요한 자원을 제공한다 - C) 따라서: a) 정답이 아닙니다. b) 정답이 아닙니다 c) 정답입니다. 올바르게 연결된 것은: 1B, 2D, 3A, 4C입니다. d) 정답이 아닙니다	FL-3.2.3	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
19	d	a) 정답이 아닙니다. 결정 테이블 테스팅은 블랙박스 테스트 기법이므로 명세 기반이지 구조 기반이 아닙니다. 테스트 케이스는 소스 코드 안에 있는 결정문을 기반으로 하지 않습니다. 분기 테스팅에서 테스트 케이스는 테스트 대상의 제어 흐름에 대한 지식에서 도출됩니다. b) 정답이 아닙니다. 잠재적 결함의 예측은 오류 추정(경험 기반 테스트 기법)에 사용되지, 분기 테스팅(구조 기반 기법)에 사용되지 않습니다. 결정 테이블 테스팅에서 테스트 케이스는 비즈니스 로직을 설명하는 명세에서 도출됩니다. c) 정답이 아닙니다. 테스트 케이스가 테스트 대상의 제어 흐름에 대한 지식을 기반으로 한다면 이는 화이트박스 테스트 기법입니다. 결정 테이블 테스팅은 일반적으로 비즈니스 로직 분석을 기반으로 하므로 블랙박스 테스트 기법입니다. 분기 테스팅에서 테스트 케이스는 명세에서 도출하지 않습니다. 그랬다면 블랙박스 테스트 기법이었을 것입니다. 분기 테스팅은 화이트박스 테스트 기법으로 테스트 케이스는 소스 코드 구조를 기반으로 도출됩니다. d) 정답입니다. 결정 테이블 테스팅은 블랙박스 테스트 기법이므로 내부 구조를 참조하지 않고 테스트 대상의 명시된 동작에 대한 분석을 기반으로 합니다. 따라서 테스트 케이스는 소프트웨어가 구현되는 방식과 무관합니다. 분기 테스팅은 화이트박스 테스트 기법이므로 테스트 케이스는 테스트 대상의 내부 구조와 처리 과정에 대한 분석을 기반으로 합니다. 테스트 케이스는 소프트웨어 설계와 코딩 방식에 의존하므로 테스트 대상의 설계나 구현이 끝난 후에만 만들 수 있습니다.	FL-4.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
20	a	a) 정답입니다. 19는 "할인 없음" 분할을, 20은 "50% 할인" 분할을, 30은 "10% 할인" 분할을 다룹니다. 이 세 값으로 세 개의 유효한 동등 분할은 모두 커버됩니다. b) 정답이 아닙니다. 11과 12는 "할인 없음" 분할을, 20은 "50% 할인" 분할을 다루므로 세 개의 유효한 동등 분할 중 두 개가 커버됩니다. c) 정답이 아닙니다. 1은 "할인 없음" 분할을, 10과 50은 "10% 할인" 분할을 다룹니다. "50% 할인" 분할은 다루지 않으므로 전체적으로 세 개의 유효한 동등 분할 중 두 개를 커버합니다. d) 정답이 아닙니다. 29와 31은 "할인 없음" 분할을, 10과 30은 "10% 할인" 분할을 커버합니다. "50% 할인" 분할은 다루지 않으므로 전체적으로 세 개의 유효한 동등 분할 중 두 개가 커버됩니다.	FL-4.2.1	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
21	d	비밀번호 길이는 세 개의 동등 분할이 있습니다: ● 너무 짧은 비밀번호 {0, 1, ..., 4, 5} ● 가능한 비밀번호 {6, 7, ..., 11, 12} ● 너무 긴 비밀번호 {13, 14, ...} 3가지 선택 경계값 분석으로 100% 커버리지를 달성하려면 다음 값을 테스트해야 합니다: 0, 1, 4, 5, 6, 7, 11, 12, 13, 14. 2가지 선택 경계값 분석은 이미 했으므로 다음 길이의 비밀번호를 테스트했다는 것을 의미합니다: 0, 5, 6, 12, 13. 즉, 2가지 선택에서 3가지 선택으로 넘어가기 위해 추가로 다뤄야 할 길이는 다음과 같습니다: 1, 4, 7, 11, 14 따라서: a) 정답이 아닙니다 b) 정답이 아닙니다 c) 정답이 아닙니다 d) 정답입니다.	FL-4.2.2	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
22	b	결정 테이블에 5개의 열이 있습니다. 각 테스트 케이스는 그 중 하나를 다룹니다. TC1과 TC2는 모두 규칙 4를 다룹니다. TC3과 TC4는 모두 규칙 2를 다룹니다. TC5는 규칙 5를 다룹니다. 따라서 이 5개의 테스트 케이스는 5개 열 중 3개를 다루어 (3/5)*100% = 60%의 커버리지를 달성합니다. 그러므로 b)가 맞습니다. 따라서: a) 정답이 아닙니다. b) 정답입니다. c) 정답이 아닙니다. d) 정답이 아닙니다.	FL-4.2.3	K3	1

23	c	 그림에서 E1, …, E5로 명시된 전이를 봅시다. 변수 N은 현재 저장된 요소의 수를 나타냅니다. 각 "Add" 이벤트는 N을 1 증가시키고, 각 "Remove" 이벤트는 N을 1 감소시킵니다. NOT FULL 상태에서 "Add" 이벤트가 발생할 때 N=2 인 경우에만 상태가 FULL로 변경됩니다. N<2이면 시스템은 NOT FULL 상태에 머뭅니다. N=0이면 "Remove" 동작이 불가능합니다. 마찬가지로 N=3이면 "Add" 동작이 불가능합니다. ● 테스트 a)는 E1, E3, E3, E2, E4로 작성할 수 있습니다(5개 유효 전이 중 4개를 다루어 80%의 유효 전이 커버리지를 달성). ● 테스트 b)는 실현 불가능한데, 처음 세 번의 "Add" 동작 후 시스템이 FULL 상태가 되면 "Add" 이벤트에 의해 트리거되는 FULL에서 출발하는 유효 전이가 없기 때문입니다. 처음 세 번의 전이 후에는 60%의 유효 전이 커버리지가 달성됩니다. ● 테스트 c)는 E1, E2, E4, E5, E3로 작성할 수 있습니다(5개 유효 전이 중 5개를 다루어 100%의 유효 전이 커버리지를 달성). ● 테스트 d)는 E1, E2, E4, E5, E4로 작성할 수 있습니다(5개 유효 전이 중 4개를 다루어 80%의 유효 전이 커버리지를 달성). 따라서: a) 정답이 아닙니다. b) 정답이 아닙니다. c) 정답입니다. d) 정답이 아닙니다.	FL-4.2.4	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
24	b	a) 정답이 아닙니다. 커버리지는 항상 커버된 요소의 비율을 백분율로 표시합니다. 따라서 100%를 초과할 수 없습니다. b) 정답입니다. T1과 T2로 실행된 구문이 서로 겹치지 않는다면 {T1, T2} 테스트 스위트의 커버리지는 105%가 될 것인데, 이는 불가능합니다(답변 a 참조). 따라서 T1과 T2 모두에 의해 실행된 실행 가능 구문이 최소한 5% 이상 존재해야 합니다. c) 정답이 아닙니다. 구문 커버리지는 코드에 실행 불가능한 구문의 수에 대해 아무것도 알려주지 않습니다. d) 정답이 아닙니다. 테스트 스위트가 100% 구문 커버리지를 달성했다고 해서 100% 분기 커버리지를 달성했다는 의미는 아닙니다.	FL-4.3.1	K2	1
25	c	분기 테스팅은 커버리지 항목이 분기인 화이트박스 테스트 기법입니다. 분기는 테스트 대상에서 소스 코드 구문이 실행될 수 있는 가능한 순서를 보여주는 제어 흐름 그래프에서 두 노드 간의 제어 이동입니다. 각 제어 이동은 무조건적(즉, 직선 코드)이거나 조건적(즉, 결정문의 결과)일 수 있습니다. 커버리지는 테스트 케이스에 의해 실행된 분기 수를 총 분기 수로 나눈 값으로 백분율로 표시됩니다. 따라서: a) 정답이 아닙니다. 결정문의 결과는 조건 분기입니다. 분기 테스팅에서 X는 조건 분기뿐만 아니라 무조건 분기도 계산합니다. b) 정답이 아닙니다. 분기 커버리지는 조건 분기뿐만 아니라 무조건 분기도 계산합니다. c) 정답입니다. 분기 커버리지는 테스트 케이스에 의해 실행된 분기 수를 총 분기 수로 나눈 값으로 측정되며 백분율로 표시됩니다. d) 정답이 아닙니다. X와 Y 모두 조건 분기만 계산하고 무조건 분기는 고려하지 않습니다.	FL-4.3.2	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
26	a, e	탐색적 테스팅은 명세가 부족하거나 부적절할 때, 또는 테스팅에 상당한 시간적 압박이 있을 때 유용합니다. 탐색적 테스팅은 다른 공식적인 테스트 기법을 보완하는 데에도 유용합니다. 탐색적 테스팅은 테스터가 경험이 풍부하고, 도메인 지식이 있으며, 분석 기술, 호기심, 창의성과 같은 필수 기술을 높은 수준으로 갖추고 있을 때 더 효과적입니다. 따라서: a) 정답입니다. 탐색적 테스팅은 명세가 부족하거나 부적절할 때, 또는 테스팅에 상당한 시간적 압박이 있을 때 유용합니다. b) 정답이 아닙니다. 탐색적 테스팅은 블랙박스 테스트 기법이 아닙니다. c) 정답이 아닙니다. 탐색적 테스팅은 명세가 제대로 작성되지 않았을 때 유용합니다. d) 정답이 아닙니다. 프로그래밍 기술은 원칙적으로 탐색적 테스팅과 아무 관련이 없습니다. e) 정답입니다. 탐색적 테스팅은 테스터가 경험이 풍부하고, 도메인 지식이 있으며, 분석 기술, 호기심, 창의성과 같은 필수 기술을 높은 수준으로 갖추고 있을 때 더 효과적입니다.	FL-4.4.2	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
27	d	a) 정답이 아닙니다. 체크리스트에는 검증해야 할 테스트 컨디션을 포함해야 합니다. 이것은 오류이지 테스트 컨디션이 아닙니다. 테스터가 오류에서 잠재적인 테스트 컨디션을 추론할 수 있다 하더라도 이 오류에 대한 설명은 너무 일반적입니다. b) 정답이 아닙니다. 체크리스트에 완료 조건으로 더 적합한 항목이 포함되어서는 안 됩니다. 이것은 완료 조건의 예입니다. c) 정답이 아닙니다. 체크리스트에 너무 일반적인 항목이 포함되어서는 안 됩니다. 이것은 매우 일반적인 항목으로, 사실상 테스팅의 목표를 설명하고 있습니다. d) 정답입니다. 이것은 사람이 확인할 수 있는 테스트 컨디션의 예입니다.	FL-4.4.3	K2	1
28	b	a) 정답이 아닙니다. 규칙 중심 형식에는 검증 항목 목록이나 입출력 매핑을 담은 표와 같은 형식이 포함되며, 따라야 할 규칙을 명시적으로 보여줍니다. Given/When/Then은 검증해야 할 시나리오를 설명하기 때문에 시나리오 중심 형식입니다. b) 정답입니다. 이것은 시나리오 중심인 Given/When/Then 형식입니다. c) 정답이 아닙니다. "제품 중심" 형식의 인수 조건은 없습니다. d) 정답이 아닙니다. "프로세스 중심" 형식의 인수 조건은 없습니다.	FL-4.5.2	K2	1
29	d	a) 정답이 아닙니다. 이 테스트 케이스는 이전 주문 내역 보기와 관련이 있습니다. b) 정답이 아닙니다. 이 테스트 케이스는 이전 주문 보기와 관련이 있습니다. c) 정답이 아닙니다. 이 테스트 케이스는 주문 내역에서 이전 주문 보기와 관련이 있습니다. d) 정답입니다. 이 테스트 케이스는 등록 프로세스와 관련이 있는데, 이는 사용자 스토리에서 다루지 않습니다. 사용자 스토리는 이전 주문 보기에 관한 것입니다.	FL-4.5.3	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
30	a	a) 정답입니다. 이것은 코드를 버전 관리 시스템에 제출하기 전에 확인할 수 있는(그리고 확인해야 하는) 것입니다. b) 정답이 아닙니다. 이것은 (2) 단계를 수행한 후에 확인할 수 있는 것인데, 병합 충돌 보고는 코드를 제출하고 병합한 후에 할 수 있기 때문입니다. c) 정답이 아닙니다. 이것은 (3) 단계의 시작 조건으로 더 적합합니다. d) 정답이 아닙니다. 이것은 (3) 단계의 완료 조건으로 더 적합합니다.	FL-5.1.3	K2	1
31	b	개발 노력의 평균은 $900,000이고 테스트 노력의 평균은 $90,000입니다(4개 프로젝트에서 계산). 평균 테스트 대비 개발 노력 비율은 1:10($90,000 : $900,000)인데, 이는 과거 데이터를 기반으로 했을 때 평균적으로 테스트 노력이 개발 노력의 10%라는 것을 의미합니다. 그래서 개발 노력이 $800,000으로 추정된다면, 추정 테스트 노력은 다음과 같이 계산됩니다: 10% * $800,000 = 0.1 * $800,000 = $80,000 따라서: a) 정답이 아닙니다. b) 정답입니다. c) 정답이 아닙니다 d) 정답이 아닙니다.	FL-5.1.4	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
32	a	종속성에 따르면 SEARCH 테스트가 먼저 실행되어야 하고, 그 다음 VIEW 테스트, 그 다음 ADD 테스트, 마지막으로 ORDER 테스트가 실행되어야 합니다. 각 그룹 내에서의 순서는 테스트 케이스의 우선순위에 의해 결정됩니다. 그래서 TC1이 가장 먼저 실행되어야 하고, 그 다음 TC2, 그 다음 TC4, 그 다음 TC3, 그리고 마지막으로 TC5가 실행되어야 합니다. 즉, 순서는 TC1, TC2, TC4, TC3, TC5입니다. 따라서: a) 정답입니다. TC3이 네 번째로 실행되는 테스트 케이스입니다. b) 정답이 아닙니다. c) 정답이 아닙니다. d) 정답이 아닙니다.	FL-5.1.5	K3	1
33	d	a) 정답이 아닙니다. 사용성 테스팅은 제품을 평가하는 비즈니스와 관련된 테스팅입니다(Q3) b) 정답이 아닙니다. 기능 테스팅은 비즈니스와 관련된 테스팅입니다(Q2) c) 정답이 아닙니다. 사용자 인수 테스팅은 제품을 평가하는 비즈니스와 관련된 테스팅입니다(Q3) d) 정답입니다. 컴포넌트 통합 테스팅은 팀을 지원하는(개발을 가이드하는) 기술과 관련된 테스팅입니다(Q1)	FL-5.1.7	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
34	c	나열된 리스크와 그에 대한 완화 활동을 각각 고려해 보면: 1. 긴 시스템 응답 시간(1)은 성능 테스팅(B) 중 테스트할 수 있습니다 2. 소비자 선호도 변화(2)는 보통 우리가 통제할 수 없으므로 이는 리스크로 수용합니다(A) 3. 서버룸 침수(3)는 상당한 손실을 초래할 수 있으므로 보험에 가입하는 등 리스크를 이전해야 합니다(D) 4. 특정 연령 이상의 환자가 부정확한 보고서를 받는다(4)는 것은 경계를 잘못 설정한 것일 수 있는데, 이는 경계값 분석(BVA)과 같은 기법으로 효과적으로 탐지할 수 있습니다(C) 따라서: a) 정답이 아닙니다. b) 정답이 아닙니다. c) 정답입니다. 리스크와 완화 방법을 적절하게 조합한 것: 1B, 2A, 3D, 4C입니다. d) 정답이 아닙니다.	FL-5.2.4	K2	1
35	**a**	a) 정답입니다. 제품 품질 메트릭은 품질 특성을 측정합니다. 평균 고장 시간(MTTF)은 성숙도를 측정하므로 제품 품질 메트릭입니다. b) 정답이 아닙니다. 이것은 결함 메트릭이지 제품 품질 메트릭이 아닙니다. c) 정답이 아닙니다. 이것은 커버리지 메트릭이지 제품 품질 메트릭이 아닙니다. d) 정답이 아닙니다. 이것은 결함 메트릭이지 제품 품질 메트릭이 아닙니다.	FL-5.3.1	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
36	a	a) 정답입니다. 고객은 위치와 시간대가 다른데 있어 대면 의사소통이 어려울 수 있습니다. b) 정답이 아닙니다. 대시보드는 보통 모든 사용자가 언제든지 사용할 수 있으므로 시차가 대화 또는 대면 의사소통만큼 의사소통에 방해가 되지는 않을 것입니다. c) 정답이 아닙니다. 유럽과 미국 간 시차는 여러 시간이고 이는 다소 불편할 수 있지만, 대면 의사소통만큼 불편하지는 않습니다. d) 정답이 아닙니다. 화상회의 도구는 편리한 의사소통 수단입니다. 유럽과 미국 간 근무 시간 동안 의사소통을 하려면 보통 한쪽이 매우 이른 시간이나 늦은 시간에 접속해야 하지만, 이는 대면 의사소통만큼 불편하지는 않습니다.	FL-5.3.3	K2	1
37	a	a) 정답입니다. 복잡한 형상 항목(예: 테스트 환경)의 경우, 형상관리(CM)는 항목의 구성요소, 그 관계 그리고 각각의 버전을 기록합니다. b) 정답이 아닙니다. 형상관리(CM) 도구는 테스트 케이스를 실행하지 않으며 커버리지를 계산하지 않습니다. c) 정답이 아닙니다. 형상관리(CM) 도구는 라이선스 관리 도구가 아닙니다. d) 정답이 아닙니다. 형상관리(CM) 도구는 테스트 데이터를 생성하지 않습니다.	FL-5.4.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
38	b	a) 정답이 아닙니다. 문장은 맞지만 개발자에게 큰 가치를 제공하지는 않습니다. b) 정답입니다. 테스트 결과를 보면 시스템이 중복을 무시하고 반복을 제외하고 목록을 정렬하는 것 같습니다. 이것이 아마도 TC3, TC4, TC5의 실패 원인일 것입니다. 이런 정보는 개발자가 결함을 찾아 더 효율적으로 수정하는 데 도움이 될 수 있습니다. c) 정답이 아닙니다. 시스템은 음수 정렬에 실패하지 않습니다. 문제는 오히려 중복을 무시하는 것입니다. d) 정답이 아닙니다. TC3, TC4, TC5가 실패했지만 테스트 케이스에 결함이 있다는 것을 알지는 못합니다.	FL-5.5.1	K3	1
39	c	나열된 도구 범주와 그 설명을 각각 고려해보면: A. 정적 테스팅 도구 - 리뷰와 정적 분석 수행 시 테스터를 지원 (4) B. 확장성 및 배포 표준화 지원 도구 - 예를 들어, 가상 머신, 컨테이너화 도구 (3) C. 데브옵스 도구 - 데브옵스 배포 파이프라인, 작업흐름 추적, 자동화된 빌드 프로세스, 지속적 통합/지속적 전달(CI/CD)을 지원 (1) D. 협업 도구 - 의사소통을 용이하게 함 (2) 따라서: a) 정답이 아닙니다. b) 정답이 아닙니다. c) 정답입니다. 옳게 연결된 것은: 1C, 2D, 3B, 4A입니다 d) 정답이 아닙니다.	FL-6.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
40	a	a) 정답입니다. 테스트 자동화는 사람이 도출하기에는 너무 복잡한 측정치를 제공할 수 있습니다. 화이트박스 커버리지 메트릭 등의 경우 정말 간단한 코드인 경우를 제외하고 여기에 해당하게 됩니다. b) 정답이 아닙니다. 테스트 도구를 사용한다고 해서 테스팅에 대한 책임이 도구 공급업체와 공유되는 것은 아닙니다. 공급업체는 테스팅에 관여하지 않으며, 테스팅은 테스터의 책임입니다. 도구 공급업체에 부여할 수 있는 유일한 책임은 도구가 예상대로 작동하지 않아 잘못된 테스트 결과를 제공하는 경우뿐입니다. c) 정답이 아닙니다. 테스터는 여전히 테스트 결과의 이상 현상을 분석해서 가능한 원인을 판단할 때 비판적 사고를 해야 합니다. d) 정답이 아닙니다. 테스터나 도구 모두 프로그램 코드를 분석하는 것만으로 테스트 케이스를 생성할 수 없습니다. 코드는 구현이며 기대 결과에 대한 정보를 제공하지 않기 때문입니다. 이는 설계 명세와 같이 테스트 베이시스의 다른 부분에서 가져와야 합니다.	FL-6.2.1	K1	1