

ISTQB® Certified Tester Syllabus Foundation Level

Compatible with Syllabus version 4.0

International Software Testing Qualifications Board



Sample Exam C - Answers (샘플문제 C - 해설)

Released Version 1.3

Translated Version 1.1

(사)케이에스티큐비 KSTQB

Korean Software Testing Qualifications Board

Copyright Notice © International Software Testing Qualifications Board (hereinafter called ISTQB®) and Korean Software Testing Qualifications Board(hereinafter called KSTQB).

All rights reserved.

This document may be copied in its entirety, or extracts made, if the source is acknowledged.

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.
- 문제는 별도의 문서에 들어 있습니다.

정답표

문제 번호(#)	정답	학습 목표(LO)	K-레벨	배점
1	b	FL-1.1.1	K1	1
2	c	FL-1.1.2	K2	1
3	b	FL-1.3.1	K2	1
4	b, e	FL-1.4.1	K2	1
5	a	FL-1.4.3	K2	1
6	c	FL-1.4.5	K2	1
7	b	FL-1.5.2	K1	1
8	a	FL-1.5.3	K2	1
9	d	FL-2.1.2	K1	1
10	d	FL-2.1.3	K1	1
11	b	FL-2.1.5	K2	1
12	c	FL-2.1.6	K2	1
13	d	FL-2.2.1	K2	1
14	b	FL-2.2.3	K2	1
15	d	FL-3.1.3	K2	1
16	a	FL-3.2.1	K1	1
17	b	FL-3.2.4	K2	1
18	b	FL-3.2.5	K1	1
19	c	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	d	FL-4.2.3	K3	1
23	a	FL-4.2.4	K3	1
24	c	FL-4.3.2	K2	1
25	a	FL-4.3.3	K2	1
26	b	FL-4.4.1	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-5.1.1	K2	1
31	c	FL-5.1.4	K3	1
32	a	FL-5.1.5	K3	1
33	b	FL-5.1.6	K1	1
34	d	FL-5.1.7	K2	1
35	c	FL-5.2.3	K2	1
36	b	FL-5.3.2	K2	1
37	d	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	d	FL-6.1.1	K2	1
40	d	FL-6.2.1	K1	1

정답

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
1	b	<p>a) 정답이 아닙니다. 문서화된 요구사항이 충족됐는지 확인하는 것은 잘못된 설명입니다. 벨리데이션은 사용자 요구사항 및 기대사항을 충족하는 것과 관련이 있지만, 베리피케이션은 주어진 요구사항을 충족하는 것과 관련이 있기 때문입니다. 그러므로 이 선택지에서 '벨리데이션'을 '베리피케이션'으로 바꾸면 정답이 됩니다.</p> <p>b) 정답입니다. 장애를 유발하고 결함을 식별하는 것은 동적 테스트의 가장 일반적인 목적입니다.</p> <p>c) 정답이 아닙니다. 오류를 유발하고 근본 원인을 식별하는 것은 잘못된 설명입니다. 이는, 테스터들은 오류를 유발하지 않으며, 장애를 유발하려고 시도하기 때문입니다. 오류는 일반적으로 개발자에 의해 만들어지며(실제로 유발할 수는 없음) 결함을 야기하는데, 이는 테스터가 정적 테스트를 통해 직접적으로 식별하거나 동적 테스트로 인한 장애를 통해 간접적으로 확인할 수 있습니다. 근본 원인을 식별하는 것은 유용하지만 디버깅의 일부이며, 이는 테스트와는 별개의 활동입니다.</p> <p>d) 정답이 아닙니다. 테스트 대상이 사용자 기대사항을 충족하는지 확인하는 베리피케이션은 틀린 설명입니다. 이는, 베리피케이션은 주어진(문서화된) 요구사항이 충족되었는지 확인하는 것인 반면에, 벨리데이션은 사용자 요구사항과 기대를 충족하는 것과 관련이 있기 때문입니다. 따라서 '베리피케이션'을 '벨리데이션'으로 바꿔야 정답이 됩니다.</p>	FL-1.1.1	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
2	c	<p>a) 정답이 아닙니다. 동적 테스트는 장애를 유발하며(이로부터 결함을 찾아내고 수정할 수 있음), 디버깅은 결함을 찾아 수정하는 것과 관련이 있습니다. 따라서, 디버깅은 장애를 수정하지 않습니다.</p> <p>b) 정답이 아닙니다. 테스트와 디버깅 모두 테스트 대상의 품질향상에 기여하므로 이 둘을 모두 긍정적으로 생각해야 합니다. 일반적으로 디버깅은 무언가를 수정하는 것이기 때문에 긍정적인 활동으로 간주됩니다. 동적 테스트는 의도적으로 테스트 대상에 장애를 일으키는 것을 수반하기 때문에 일부 사람들은 이것을 부정적인 활동으로 생각하지만, 이는 매우 협소한 관점입니다(그리고 일반적으로 테스터가 지지하는 않는 견해임). 긍정 테스트 케이스와 부정 테스트 케이스 모두 가능합니다. 긍정 테스트 케이스는 테스트 대상이 해야 할 일을 올바르게 수행하는지 확인하는 반면, 부정 테스트 케이스는 테스트 대상이 수행하면 안 되는 것을 하지 않는지 확인합니다.</p> <p>c) 정답입니다. 테스트는 리뷰에서 결함을 직접 (또는 정적 분석의 도구에 의해) 관찰하거나, 동적 테스트에서 장애를 유발해 간접적으로 결함이 존재하는지 여부를 판단합니다. 디버깅은 테스트와는 별개의 활동(일반적으로 개발자에 의해 수행)이며, 결함을 찾고 (동적 테스트에 한함) 수정하는 것과 관련이 있습니다.</p> <p>d) 정답이 아닙니다. 결함의 원인은 보통 사람의 실수입니다. 테스트는 정적 테스트를 통해 직접적으로 결함을 찾거나, 동적 테스트에서 장애를 유발해 간접적으로 찾고, 디버깅은 결함을 수정합니다. 따라서, 테스트는 결함의 원인을 찾지 않고, 디버깅은 결함의 원인을 수정하지 않습니다.</p>	FL-1.1.2	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
3	b	<p>‘결함 부재의 오류(absence-of-defects-fallacy)’은 요구사항에 따른 정확성 보장(즉, 구현상의 결함이 없음을 확인하는 것)이 시스템에 대한 사용자 만족을 보장하지 않는다는 개념과 관련이 있습니다. 이를 해결하기 위해서는, 시스템이 사용자의 요구와 기대를 충족하고, 비즈니스 목표를 달성하며, 경쟁 시스템을 능가하는 성능을 발휘하는지 검증해야 합니다.</p> <p>a) 정답이 아닙니다. ‘테스팅은 결함의 존재를 밝히는 활동이지, 결함이 없음을 증명하지는 않는다.’라는 원리는 테스팅이 테스트 대상에 결함이 존재함을 탐지할 수는 있지만, 결함의 부재를 입증할 수는 없으므로, 그 정확성을 보장할 수 없다는 것을 설명합니다. 따라서, 테스팅으로 결함의 부재를 보여주는 것이 불가능하다는 것을 설명하는 것은, 이원리를 부분적으로 다루는 것뿐이지 ‘결함 부재의 오류’를 다루는 것이 아닙니다.</p> <p>b) 정답입니다. 최종 사용자가 인수 테스팅을 수행할 수 있도록 지원함으로써 시스템이 사용자의 요구와 기대를 충족하는지 확인할 수 있어야 합니다.</p> <p>c) 정답이 아닙니다. ‘테스팅은 결함의 존재를 밝히는 활동이지, 결함이 없음을 증명하지는 않는다.’의 원리가 테스팅은 테스트 대상의 결함의 존재를 탐지할 수 있지만, 결함의 부재를 입증하는 것은 불가능하기 때문에 그 정확성을 보장할 수 없다고 설명하고 있듯이, 인도된 시스템에 구현 결함이 남아있지 않음을 보장하는 것은 불가능합니다.</p> <p>d) 정답이 아닙니다. 잔존 결함의 최소화를 위해 장애를 유발하지 않는 테스트를 수정하는 것은 ‘테스트 효과는 줄어든다’는 원리를 해결하는 한 가지 방법입니다. 이 원리는 변경되지 않은 코드에 대해 동일한 테스트를 반복하는 것은 새로운 결함을 발견할 가능성이 낮고, 따라서 테스트의 수정이 필수적일 수 있다는 개념과 관련이 있습니다. 이는 시스템이 사용자의 요구와 기대를 충족하는지 검증하지 못할 것입니다.</p>	FL-1.3.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
4	b, e	<p>테스트 분석에 대해 다음과 같은 설명이 주어졌다면: 테스트가 필요한 기능을 식별하기 위해 테스트 베이스를 분석하고 테스트 조건으로 정의한 다음, 관련 리스크와 함께 우선순위를 정합니다. 테스트 조건을 커버리지 항목으로 체계적으로 식별하는 것은 종종 테스트 분석 중, 그리고 테스트 설계 활동의 일환으로 테스트 기법을 사용하는 것을 포함합니다.</p> <p>이상의 설명으로 볼 때, 테스트 분석 및 테스트 설계 활동에서 테스트 기법이 자주 사용됨을 알 수 있으며, 경계값 분석과 동등 분할은 테스트 기법입니다.</p> <p>a) 정답이 아닙니다. 테스트 구현은 테스트 기법의 사용을 포함할 가능성이 높지 않습니다. 왜냐하면 테스트 기법이 테스트 케이스를 작성하는 반면, 테스트 구현은 테스트 케이스를 테스트 절차에 조합시키는 것과 관련이 있기 때문입니다.</p> <p>b) 정답입니다. 테스트 설계는 테스트 조건과 커버리지 항목에서 테스트 케이스를 만들기 위해 테스트 기법을 사용하는 것을 포함할 가능성이 높습니다.</p> <p>c) 정답이 아닙니다. 테스트 실행은 테스트 기법의 사용을 포함할 가능성이 높지 않습니다. 왜냐하면 테스트 기법이 테스트 케이스를 작성하는 반면, 테스트 실행은 주로 테스트 절차(및 테스트 케이스)를 실행하는 것과 관련이 있기 때문입니다.</p> <p>d) 정답이 아닙니다. 테스트 모니터링은 테스트 기법의 사용을 포함할 가능성이 높지 않습니다. 테스트 모니터링은 주로 계획이 준수되고 있는지 확인하기 위한 지속적인 점검과 관련이 있는 반면 테스트 기법은 테스트 케이스를 작성합니다.</p> <p>e) 정답입니다. 테스트 분석은 테스트 조건을 식별하기 위해 테스트 기법을 사용할 가능성이 높습니다.</p>	FL-1.4.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
5	a	<p>나열된 각 테스트 활동과 그 산출물인 테스트웨어를 고려해보면:</p> <p>A. 테스트 분석 - 우선순위가 지정된 테스트 조건 (4) (예 : 인수 기준), 테스트 베이스에서 식별된 결함에 대한 결함 보고서</p> <p>B. 테스트 설계 - (우선순위가 지정된) 테스트 케이스, 테스트 차터, 커버리지 항목(1), 테스트 데이터 요구사항, 테스트 환경 요구사항</p> <p>C. 테스트 구현 - 테스트 절차, 자동화된 테스트 스크립트, 테스트 스위트, 테스트 데이터, 테스트 실행 일정(3), 스텝(stubs), 드라이버, 시뮬레이터, 서비스 가상화 등의 테스트 환경 요소.</p> <p>D. 테스트 완료 - 테스트 완료 보고서, 문서화된 교훈, 개선을 위한 실행 항목, 변경 요청서 (2) (제품 백로그 항목으로써)</p> <p>따라서:</p> <p>a) 정답입니다. 올바른 연결은: 1B, 2D, 3C, 4A 입니다.</p> <p>b) 정답이 아닙니다.</p> <p>c) 정답이 아닙니다.</p> <p>d) 정답이 아닙니다.</p>	FL-1.4.3	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
6	c	<p>a) 정답이 아닙니다. 애자일 소프트웨어 개발에서 일부 테스트 관리 업무를 애자일 팀 자체에서 처리할 수 있다고 보는 것이 맞지만, 테스트 역할은 팀 외부의 한 개인의 책임은 아닙니다. 대신, 전체 팀 접근법에 따라 다양한 팀원이 테스트를 수행할 가능성이 높습니다.</p> <p>b) 정답이 아닙니다. 테스트 관리 역할은 주로 테스트 계획, 테스트 모니터링과 제어, 테스트 완료와 관련된 활동을 포함합니다. 따라서 이 설명은 부분적으로 맞지만, 테스트 역할이 주로 테스트 모니터링 및 제어를 담당한다는 것은 맞지 않습니다.</p> <p>c) 정답입니다. 애자일 소프트웨어 개발에서 일부 테스트 관리 업무는 애자일 팀 자체에서 처리할 수 있습니다. 단, 조직 내 여러 팀에 걸친 테스트 활동의 경우, 개발팀 외부의 테스트 관리자가 이런 작업을 수행할 수 있습니다.</p> <p>d) 정답이 아닙니다. 테스트 관리 역할은 주로 테스트 계획, 테스트 모니터링과 제어, 테스트 완료와 관련된 활동을 포함하며, 테스트 역할은 주로 테스트 분석 테스트 설계, 테스트 구현 및 테스트 실행과 같은 테스트의 공학(기술)적인 측면을 담당합니다. 따라서 테스트 관리 역할이 일반적으로 테스트 분석과 테스트 설계를 책임지지 않지만, 테스트 역할이 주로 테스트 구현 및 실행을 담당한다고 말하는 것은 옳습니다.</p>	FL-1.4.5	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
7	b	<p>a) 정답이 아닙니다. 전체 팀 접근법에서 테스터는 테스트 지식을 팀과 공유하고 제품 개발을 가이드하는 중요한 역할을 합니다. 그들은 원하는 품질 수준 달성을 위해 다른 팀원과 협력하고 비즈니스 담당자와 인수 테스트를 생성합니다. 또한 테스터는 최적의 테스트 전략과 테스트 자동화 접근법 결정을 위해 개발자와 협력합니다.</p> <p>b) 정답입니다. 전체 팀 접근법은 각 팀원의 다양한 기술을 가장 효과적으로 활용함으로써 우수한 팀 역학을 조성하고, 견고한 의사소통과 협업을 촉진하며, 전체 프로젝트에 도움이 되는 시너지 효과를 창출합니다.</p> <p>c) 정답이 아닙니다. 전체 팀 접근법은 필요한 기술과 지식을 가진 팀원이라면 누구나 어떤 작업도 수행할 수 있게 하므로, 전문가 팀원은 이 접근법의 장점은 아닙니다.</p> <p>d) 정답이 아닙니다. 전체 팀 접근법을 사용하는 팀의 최적 규모에 대한 구체적인 지침은 없으며, 규모가 큰 팀이 더 좋다는 제안도 없습니다.</p>	FL-1.5.2	K1	1

8	a	<p>a) 정답입니다. 테스트에서 독립성의 가장 큰 이점은 다양한 배경, 기술적 관점, 인지 편향 등의 잠재적 편향으로 인해 개발자에 비해 다양한 유형의 장애 및 결함을 식별할 가능성이 높다는 것입니다. 그러나 테스트에서 독립성의 가장 큰 단점은 테스터가 개발팀과 차단되어 협업과 의사소통이 어려울 수 있으므로, 잠재적으로 적대적인 관계로 이어질 수 있고, 테스터가 출시 프로세스의 지연 및 병목 현상의 원인으로 지목될 수 있다는 것입니다.</p> <p>b) 정답이 아닙니다. 개발자가 코드에 익숙하다고 해서 코드에서 결함을 거의 발견하지 못하는 것이 아니라, 이런 익숙함으로 인해 자신의 코드에서 많은 결함을 효율적으로 발견할 수 있다는 것입니다. 그리고, 개발자와 테스터가 배경을 공유하기보다는 개발자와 테스터의 배경이 다르다는 것이 서로 다른 종류의 결함을 발견하는 이유로 일반적으로 언급됩니다.</p> <p>c) 정답이 아닙니다. 테스트는 작성자에게 독립성이 전혀 없는 경우부터 조직 외부의 테스터에게 주어지는 매우 높은 독립성에 이르기까지 다양한 레벨의 독립성에서 수행될 수 있습니다. 개발자가 컴포넌트와 컴포넌트 통합 테스트를 수행하고, 테스트팀이 시스템과 시스템 통합 테스트를 수행하며, 비즈니스 담당자가 인수 테스트를 수행하는 등 대부분의 프로젝트에서 다양한 수준의 독립성이 활용됩니다. 따라서 테스터는 개발자의 팀에 있을 수 있으며 조직 외부에서 올 필요가 없습니다. 애플리케이션 도메인에 대한 지식은 경우에 따라 바뀌며, 독립성 수준에 의존하지 않습니다.</p> <p>d) 정답이 아닙니다. 테스트는 작성자에게 독립성이 전혀 없는 경우부터 조직 외부의 테스터에게 주어지는 매우 높은 독립성에 이르기까지 다양한 레벨의 독립성에서 수행될 수 있으며, 개발팀 외부의 테스터는 일반적으로 팀에 속한 테스터보다 더 독립적입니다. 그러나 팀 외부의 테스터는 개발자와 더 고립될 가능성이 높아 제품 출시 지연의 원인으로 지목될 가능성이 더 높다고 믿을만한 이유가 더 많습니다.</p>	FL-1.5.3	K2	1
---	---	---	----------	----	---

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
9	d	<p>a) 정답이 아닙니다. 품질 관리는 모든 개발 활동에 적용되며, 모든 소프트웨어 개발 활동에는 상응하는 테스트 활동이 있습니다. 그러나 여기서는 테스트 레벨을 개발 레벨과 동일시하려고 시도하고 있으며, '테스트 레벨'이 무엇을 의미하는지는 알지만 '개발 레벨'이라는 용어에 대한 공통된 이해는 없습니다.</p> <p>b) 정답이 아닙니다. 모든 소프트웨어 개발 활동에는 상응하는 테스트 활동이 있지만 테스트 목적은 상당히 다릅니다. 예를 들어, 테스트 대상을 인도하기 전에 특정 유형의 테스트를 수행해야 한다는 계약상 요구사항의 준수 확인이 테스트 목적일 수 있습니다. 이 경우 상응하는 개발 목적이 있어야 할 이유는 없습니다.</p> <p>c) 정답이 아닙니다. 품질 관리는 모든 개발 활동에 적용되며, 모든 소프트웨어 개발 활동에는 상응하는 테스트 활동이 있습니다. 그러나 테스트와 사용자 활동에는 동일한 대칭성이 적용되지 않습니다. 예를 들어, 최종 사용자를 식별하는 것조차 어려운 시스템도 있습니다. 또한 개발자 측에 중점을 두고 있어(예: 유지 보수 용이성 테스트) 사용자 측면이 전혀 없는 테스트 활동도 있습니다.</p> <p>d) 정답입니다. 품질 관리는 모든 개발 활동에 적용되며, 모든 소프트웨어 개발 활동에는 상응하는 테스트 활동이 있습니다.</p>	FL-2.1.2	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
10	d	<p>a) 정답이 아닙니다. 단위 테스트 주도 개발은 테스트 우선 개발 접근법의 올바른 예가 아닙니다.</p> <p>b) 정답이 아닙니다. 통합 테스트 주도 개발은 테스트 우선 개발 접근법의 올바른 예가 아닙니다.</p> <p>c) 정답이 아닙니다. 시스템 테스트 주도 개발은 테스트 우선 개발 접근법의 올바른 예가 아닙니다.</p> <p>d) 정답입니다. ATDD(인수 테스트 주도 개발)은 개발에 대한 테스트 우선 개발 접근법의 잘 알려진 예입니다.</p>	FL-2.1.3	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
11	b	<p>a) 정답이 아닙니다. 시프트-레프트 테스트 관련 프랙티스는 개발수명주기의 초기 단계에서 더 많은 테스트 활동을 구현하는 것을 목표로 하며, SDLC가 왼쪽에서 오른쪽으로 이동하는 것으로 묘사합니다. 테스트 프로세스의 왼쪽과 같은 것은 없습니다.</p> <p>b) 정답입니다. 시프트-레프트는 소프트웨어 개발수명주기(SDLC) 초기에 테스트를 시작하는 것이 중요하다는 점을 강조합니다. 시프트-레프트 테스트를 구현하려면 추가 교육이 필요하고 SDLC 초기 단계에서 노력과 비용이 증가하지만, 전체적인 절감 효과는 더 커야 합니다.</p> <p>c) 정답이 아닙니다. 리그레션 테스트를 위한 자동화된 단위 테스트와 단위 통합 테스트는 일반적으로 가치가 있지만, 이런 테스트의 작성은 보통 개발자의 책임이며, CI/CD (지속적 통합/지속적 배포) 접근법을 따르는 경우 이런 테스트는 코드와 함께 제출됩니다. 어떤 상황에서는 테스터가 리그레션 테스트를 위해 테스트를 자동화할 수도 있고, 때로는 단위 테스트와 단위 통합 테스트를 위해 테스트를 자동화할 수도 있지만, 이는 SDLC의 초기 단계에서 테스트를 이동하는 '시프트-레프트 접근법'의 일부가 아닙니다.</p> <p>d) 정답이 아닙니다. SDLC 초기에 작업을 수행하도록 테스터를 교육하는 것은 SDLC 초기에 테스트를 시작하는 것의 중요성을 강조함으로써 시프트-레프트 접근법을 지원합니다. 그러나 SDLC 후반부에 수행할 더 많은 테스트 활동을 자동화하는 것은 '시프트-레프트 접근법'의 일부가 아닙니다.</p>	FL-2.1.5	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
12	c	<p>a) 정답이 아닙니다. 회고의 목적 중 하나는, 잠재적인 프로세스 개선 사항을 식별하는 것이며, 이를 실행에 옮길 경우 개발 프로세스(테스트 대상)의 미래 산출물의 품질이 더 높아져야 합니다. 따라서 이는 회고의 결과로 발생할 수 있습니다.</p> <p>b) 정답이 아닙니다. 테스트를 위한 회고의 이점에는 프로세스 개선을 통한 테스트 효율성 향상이 포함됩니다. 따라서 이것은 회고의 결과로 발생할 가능성이 높습니다.</p> <p>c) 정답입니다. 회고의 참가자에는 일반적으로 테스터, 개발자, 설계자, 제품 소유자 및 비즈니스 분석가가 포함되지만, 최종 사용자가 초대되거나 참석하는 경우는 거의 없으며, 이런 회의로부터 보고를 받을 가능성도 낮습니다. 따라서, 회고를 통해 개발 및 테스트 프로세스에 대해 더 많이 배우고 이해할 가능성은 거의 없습니다.</p> <p>d) 정답이 아닙니다. 테스트를 위한 회고의 이점에는 개발자와의 합동 리뷰를 통한 테스트웨어(자동화된 테스트 스크립트 포함)의 품질 개선이 포함됩니다. 따라서, 이는 회고의 결과로 발생할 가능성이 높습니다.</p>	FL-2.1.6	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
13	d	<p>a) 정답이 아닙니다. 컴포넌트 테스트(단위 테스트이라고도 함)은 개별 컴포넌트를 격리 상태로 테스트하는 것으로, 사용자 요구에 대한 벨리데이션이 아니라 대부분 명세에 대한 베리피케이션입니다. 그러나 이 테스트는 일반적으로 테스터가 수행하지 않으며, 개발자가 개발 환경에서 이 테스트를 수행하는 경우가 많습니다.</p> <p>b) 정답이 아닙니다. 단위 통합 테스트는 컴포넌트 간 인터페이스와 상호작용을 테스트하는 것으로, 사용자 요구에 대한 벨리데이션이 아니라 대부분 명세에 대한 베리피케이션입니다. 그러나 이 테스트는 일반적으로 테스터가 수행하지 않으며, 개발자가 수행하는 경우가 많습니다.</p> <p>c) 정답이 아닙니다. 시스템 통합 테스트는 다른 시스템 및 외부 서비스의 인터페이스를 검사하며, 대부분 사용자 요구사항에 대한 벨리데이션이 아닌 명세에 대한 베리피케이션입니다. 또한 이런 유형의 테스트는 대부분 테스터가 수행합니다.</p> <p>d) 정답입니다. 인수 테스트는 시스템이 사용자의 비즈니스 요구사항을 충족하고 배포 준비가 되었는지 확인하는 데 중점을 둡니다. 이상적으로, 이 테스트는 최종 사용자가 수행합니다.</p>	FL-2.2.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
14	b	<p>a) 정답이 아닙니다. 업데이트가 올바르게 구현되었는지 확인하기 위한 확인 테스트가 필요하지만, 시스템의 변경되지 않는 부분에 결함이 도입되거나 드러나지 않았는지 확인하기 위한 리그레션 테스트를 수행하는 것이 합리적입니다.</p> <p>b) 정답입니다. 확인 테스트로 업데이트가 올바르게 구현되었는지 확인한 다음, 시스템의 변경되지 않는 부분에 결함이 도입되거나 드러나지 않았는지 확인하기 위해 리그레션 테스트를 사용합니다.</p> <p>c) 정답이 아닙니다. 리그레션 테스트를 사용해 업데이트가 이루어졌을 때 시스템의 변경되지 않는 부분에 결함이 도입되거나 드러나지 않았는지 확인해야 하지만, 업데이트가 올바르게 구현되었는지 확인하는 확인 테스트도 수행해야 합니다.</p> <p>d) 정답이 아닙니다. 확인 테스트는 업데이트가 올바르게 구현되었는지 확인하고, 시스템의 변경되지 않는 부분에 결함이 도입되거나 드러나지 않았는지 확인하기 위해 리그레션 테스트를 사용합니다. 그러나 수행할 때 (즉, 업데이트를 테스트해야 할 때), 확인 테스트는 리그레션 테스트보다 선행됩니다.</p>	FL-2.2.3	K2	1

15	d	<p>나열된 각 결함의 예시를 고려해보면:</p> <ul style="list-style-type: none"> i. 설계의 복잡성으로 인해 설계 명세서의 두 부분이 일치하지 않는다 – 이는 불일치, 모호성, 모순, 누락, 부정확성 및 중복을 포함하는 명세 결함의 예이며, 정적 테스트로 가장 쉽게 찾을 수 있습니다. ii. 응답시간이 너무 길어 사용자가 인내심이 떨어진다 – 이는 프로그램을 실행하고 응답 시간을 측정해야 탐지할 수 있는 응답시간 결함의 예시이며, 동적 테스트를 통해 가장 쉽게 찾을 수 있습니다. iii. 실행 중 도달할 수 없는 코드 경로가 있다 – 이는 코딩 결함의 예시로서, 정의되지 않은 값, 명시되지 않은 변수, 중복되거나 도달할 수 없는 코드, 그리고 과도한 코드 복잡성이 있으며, 정적 테스트를 통해 가장 쉽게 찾을 수 있습니다. iv. 명시되었지만 이후 프로그램에서 사용하지 않는 변수가 있다 – 이는 코딩 결함의 예시로서, 정의되지 않은 값, 명시되지 않은 변수, 중복되거나 도달할 수 없는 코드, 그리고 과도한 코드 복잡성이 있으며, 정적 테스트를 통해 가장 쉽게 찾을 수 있습니다. v. 보고서 작성 프로그램이 사용하는 메모리 양이 너무 크다 – 이는 성능 결함의 예시로, 프로그램을 실행하고 사용된 메모리를 측정해야만 실제로 탐지할 수 있으며, 동적 테스트를 통해 가장 쉽게 찾을 수 있습니다. <p>따라서:</p> <ul style="list-style-type: none"> a) 정답이 아닙니다. b) 정답이 아닙니다. c) 정답이 아닙니다. d) 정답입니다. 정적 테스트와 맞게 연결된 것은 i, iii, iv 입니다. 	FL-3.1.3	K2	1
----	---	--	----------	----	---

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
16	a	<p>a) 정답입니다. 소프트웨어 개발 프로세스에서 이해관계자의 피드백을 빠르게 자주 받는 것은 매우 큰 이점이 될 수 있습니다. 잠재적인 품질 문제에 대한 조기 커뮤니케이션을 용이하게 하고, 요구사항에 대한 오해를 방지할 수 있으며, 이해관계자 요구사항의 변경사항을 신속히 이해하고 구현할 수 있게 해줍니다.</p> <p>b) 정답이 아닙니다. 피드백은 이해관계자로부터 받은 것이며, 피드백을 제공한다고 해서 자신들의 사용자 요구사항에 대한 이해가 향상되지는 않습니다.</p> <p>c) 정답이 아닙니다. 소프트웨어 개발 프로세스에서 이해관계자의 피드백을 빠르게 자주 받는 것은 매우 큰 이점이 될 수 있습니다. 잠재적인 품질 문제를 조기에 전달할 수 있고, 요구사항에 대한 오해를 방지할 수 있으며, 이해관계자 요구사항에 대한 모든 변화를 더 빨리 이해하고 구현할 수 있게 해줍니다. 그러나 요구사항의 변화를 더 빨리 이해하고 구현할 수 있다고 해서, 요구사항에 대한 무제한 변경을 장려한다는 의미는 아닙니다.</p> <p>d) 정답이 아닙니다. 피드백은 이해관계자로부터 받은 것이며, 그들에 대한 의사 소통은 다루지 않습니다. 최종 사용자에게 대한 의사 소통에는 릴리스 전에 구현되지 않을 요구사항에 대해 알려주는 것이 포함될 수 있지만, 이상적으로는 이런 일이 전혀 일어나지 않아야 합니다.</p>	FL-3.2.1	K1	1

17	b	<p>나열된 각 리뷰 유형을 고려해보면:</p> <ol style="list-style-type: none"> 1. 기술 리뷰 – 기술적인 자격을 갖춘 리뷰어가 수행하고, 중재자가 주도합니다. 목적은 품질을 평가하고 작업 산출물에 대한 신뢰를 구축하는 동시에 새로운 아이디어를 창출하고, 저자가 개선할 수 있게 동기를 부여하고, 이상현상을 감지하는 동시에 기술적 문제에 대한 합의를 얻고 결정을 내리는 것입니다. 2. 비공식 리뷰 – 주 목적은 이상현상을 감지하는 것입니다. 정의된 프로세스가 없으며, 공식적인 결과 문서도 요구하지 않습니다. 3. 인스펙션 – 가장 공식적인 리뷰 유형으로, 완벽하고 보편적인 리뷰 프로세스를 따릅니다. 주 목적은 이상현상을 최대한 많이 찾는 것이며, 그 외 작업 산출물에 대한 품질 평가 및 신뢰 구축, 저자가 개선할 수 있도록 동기 부여, 인스펙션 프로세스를 포함한 SDLC 향상에 사용할 수 있는 메트릭 수집 등이 있습니다. 저자는 리뷰 리더 또는 서기 역할을 할 수 없습니다. 4. 워크쓰루 - 저자가 주도하는 이 유형의 리뷰는 품질 평가 및 작업 산출물에 대한 신뢰 구축, 리뷰어 교육, 합의 도출, 새로운 아이디어 창출, 저자가 개선할 수 있도록 동기 부여 및 지원, 이상현상 발견 등 여러 가지 목적을 달성할 수 있습니다. 리뷰어는 워크쓰루 전에 개별 리뷰를 수행할 수도 있지만, 반드시 해야 하는 것은 아닙니다. <p>A. 합의 도출, 새로운 아이디어 창출, 저자의 개선 동기 부여 등의 목적을 포함합니다 (A) B. 주 목적은 잠재적 결함을 탐지하는 것이며 공식적으로 문서화된 결과물을 생성하지 않습니다 (D) C. 주 목적은 잠재적 결함을 탐지하는 것이며 프로세스 개선을 지원하기 위해 메트릭 수집이 필요합니다 (C) D. 리뷰어 교육, 합의 도출, 새로운 아이디어 창출, 잠재적 결함 발견 등의 목적을 포함합니다 (B)</p> <p>따라서: a) 정답이 아닙니다. b) 정답입니다. 맞게 연결한 것은: 1A, 2D, 3C, 4B 입니다. c) 정답이 아닙니다. d) 정답이 아닙니다.</p>	FL-3.2.4	K2	1
----	---	---	----------	----	---

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
18	b	<p>a) 정답이 아닙니다. 성공적인 리뷰를 위해서는 리뷰 프로세스에 대한 경영진의 지원을 확보하는 것이 중요하지만, 그렇다고 해서 그들이 리뷰어로 참여해야 하는 것은 아닙니다.</p> <p>b) 정답입니다. 성공적인 리뷰 수행을 위해서는 리뷰어가 개별 리뷰 또는 리뷰 회의 중에 집중력을 잃지 않도록, 작업 산출물을 적절한 기간 내에 리뷰할 수 있도록 작은 단위로 분할하는 것이 중요합니다.</p> <p>c) 정답이 아닙니다. 성공적인 리뷰를 위해서는, 참가자를 평가하는 것이 아닌, 목표와 측정 가능한 완료 기준을 명확하게 정의하는 것이 중요합니다.</p> <p>d) 정답이 아닙니다. 리뷰를 성공적으로 수행하려면 리뷰를 더 작은 단위로 분할해 개별 리뷰 또는 리뷰 회의 중에 리뷰어가 집중력을 잃지 않도록 하는 것이 중요합니다. 따라서 리뷰 당 하나의 문서를 다루도록 계획해서는 안 됩니다.</p>	FL-3.2.5	K1	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
19	c	<p>a) 정답이 아닙니다. 대부분의 경우 블랙박스 테스트 기법과 경험 기반 테스트 기법 모두 동일한 테스트 대상에 사용할 수 있습니다.</p> <p>b) 정답이 아닙니다. 블랙박스 테스트 기법과 경험 기반 테스트 기법은 둘 다 모든 테스트 레벨에서 사용 가능합니다.</p> <p>c) 정답입니다. 블랙박스 테스트 기법(명세 기반 기법이라고도 함)은 테스트 대상의 내부 구조를 참조하지 않고, 테스트 대상의 명시된 동작에 대한 분석을 기반으로 합니다. 따라서 테스트 베이스는 보통 명세가 됩니다. 경험 기반 테스트 기법은 테스트 케이스의 설계 및 구현을 위해 테스터의 지식과 경험을 효과적으로 사용합니다. 이것은 테스트를 설계할 때 테스터가 명세를 전혀 사용하지 않을 수 있음을 의미합니다.</p> <p>d) 정답이 아닙니다. 경험 기반 테스트 기법은 블랙박스 (및 화이트박스) 테스트 기법을 사용했을 때 놓칠 수 있는 결함을 탐지할 수 있습니다. 따라서 경험 기반 테스트 기법은 블랙박스 테스트 기법과 화이트박스 테스트 기법을 보완하며, 블랙박스 테스트 기법과 경험 기반 테스트 기법은 모든 SDLC에서 사용할 수 있습니다.</p>	FL-4.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
20	a	<p>a) 정답입니다. "1"이라는 값은 "길이가 잘못됨"과 "서로 다른 숫자의 개수가 잘못됨"을 모두 커버합니다. 그러나 시스템은 첫 번째 잘못된 파티션에 도달할 때 이 파티션에 대해 유효성을 검사하지 않습니다. "1111"이라는 값은 "길이가 올바름"과 "서로 다른 숫자의 개수가 잘못됨"을 커버합니다. "1234"라는 값은 다시 "길이가 올바름"과 "서로 다른 숫자의 개수가 올바름"을 커버합니다.</p> <p>b) 정답이 아닙니다. 이 세 값은 동등 분할의 100% 커버리지를 달성하지만, 두 값으로도 100% 커버리지를 달성할 수 있습니다.</p> <p>c) 정답이 아닙니다. 이 두 개의 값은 "서로 다른 숫자의 개수"만 커버합니다.</p> <p>d) 정답이 아닙니다. 이 두 개의 값은 "서로 다른 숫자의 개수가 잘못됨" 동등 분할을 커버하지 않습니다.</p>	FL-4.2.1	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
21	d	<p>동등 분할은: {..., 99, 100}, {101, 102, ..., 198, 199}, {200, 201, ...} 입니다.</p> <p>따라서 경계값은 100, 101, 199, 200 4개입니다.</p> <p>2개 선택 경계값 분석에서는 각 경계값에 대해 2개의 커버리지 항목 (경계값과 인접한 분할에 속하는 가장 가까운 값)이 있습니다. 가장 가까운 값도 인접 분할의 경계값이므로 커버리지 항목은 4개뿐입니다.</p> <p>따라서:</p> <p>a) 정답이 아닙니다. 100과 200만 2개 선택 경계값 분석의 유효한 커버리지 항목이므로 50% 커버리지를 달성합니다.</p> <p>b) 정답이 아닙니다. 100과 200만 2개 선택 경계값 분석의 유효한 커버리지 항목이므로 50% 커버리지를 달성합니다.</p> <p>c) 정답이 아닙니다. 100과 101만 2개 선택 경계값 분석의 유효한 커버리지 항목이므로 50% 커버리지를 달성합니다.</p> <p>d) 정답입니다. 101, 199, 200이 2개 선택 경계값 분석의 유효한 커버리지 항목입니다. 따라서 75%의 커버리지를 달성합니다.</p>	FL-4.2.2	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
22	d	<p>a) 정답이 아닙니다. 조합 (T,T,F)는 어떤 규칙과도 일치하지 않습니다. 이는 모순이 아닌 누락의 예시입니다.</p> <p>b) 정답이 아닙니다. 조합 (T,F,T)는 R2라는 하나의 열하고만 일치하므로 모순이 없습니다.</p> <p>c) 정답이 아닙니다. 두 조합 (T,T,T)와 (F,T,T) 모두 R1이라는 하나의 열하고만 일치하므로 모순이 없습니다.</p> <p>d) 정답입니다. 조합 (F,F,F)가 R2와 R3 열에 모두 일치하지만, R2와 R3는 다른 동작을 가지고 있으므로, 이는 R2와 R3간의 모순을 보여줍니다.</p>	FL-4.2.3	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
23	a	<p>다음 세 가지 전이가 있다:</p> <p>"REQUESTING -> CONFIRMED"</p> <p>"WAITING LIST -> CONFIRMED"</p> <p>"WAITING LIST -> END"</p> <p>이들은 동일한 테스트 케이스에 나타날 수 없으므로, 최소 세 개의 테스트 케이스가 필요함을 시사합니다. 다른 모든 전이는 이 세 가지 전이 중 하나 이상의 조합으로 나타날 수 있으므로 최소 세 개의 테스트 케이스가 필요하며 사실상, 가능한 순서는 세 가지뿐입니다:</p> <p>TC1: START (Room request) → REQUESTING (Available) → CONFIRMED (Pay) → END</p> <p>TC2: START (Room request) → REQUESTING (Not available) → WAITING LIST (Available) → CONFIRMED (Pay) → END</p> <p>TC3: START (Room request) → REQUESTING (Not available) → WAITING LIST (No room available) → END</p> <p>따라서:</p> <p>a) 정답입니다.</p> <p>b) 정답이 아닙니다.</p> <p>c) 정답이 아닙니다.</p> <p>d) 정답이 아닙니다.</p>	FL-4.2.4	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
24	c	<p>분기 테스트에서 커버리지 항목은 분기이며, 이는 제어 흐름 그래프의 간선으로 표시됩니다. 제어 흐름 그래프에는 8개의 간선이 있습니다.</p> <p>따라서:</p> <p>a) 정답이 아닙니다.</p> <p>b) 정답이 아닙니다.</p> <p>c) 정답입니다.</p> <p>d) 정답이 아닙니다.</p>	FL-4.3.2	K2	1
25	a	<p>a) 정답입니다. 블랙박스 테스트만 수행하는 것은 실제 코드 커버리지에 대한 측정치를 제공하지 않습니다. 화이트박스 커버리지 측정치는 커버리지에 대한 객관적인 측정값을 제공하고, 이 커버리지를 높이기 위해 추가 테스트를 만들 수 있는 필요한 정보를 제공해서 결국 코드 신뢰도를 높일 수 있게 해 줍니다.</p> <p>b) 정답이 아닙니다. 맞는 설명이지만 블랙박스 테스트와는 무관합니다.</p> <p>c) 정답이 아닙니다. 일반적으로 화이트박스과 블랙박스 기법 사이에는 포함 관계가 없습니다.</p> <p>d) 정답이 아닙니다. 테스트 대상 자체를 기준으로 테스트를 설계하는 데는 화이트박스 기법이 사용되며, 명세를 기준으로 테스트를 설계하는 데는 블랙박스 기법이 사용됩니다. 따라서 이 두 가지 유형의 기법에서 파생된 커버리지 항목 간에는 아무런 관계가 없습니다.</p>	FL-4.3.3	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
26	b	<p>a) 정답이 아닙니다. 탐색적 테스트는 가능한 결함/장애의 목록이 아닌, 테스트 차터를 사용합니다. 탐색적 테스트는 다른 테스트 기법의 사용을 포함할 수 있지만, 이 경우에는 결점 공격이 가장 가능성이 높은 옵션입니다.</p> <p>b) 정답입니다. 이는 가능한 장애 목록입니다. 결점 공격은 오류 추정의 구현에 대한 체계적인 접근 방식이며, 테스터가 발생 가능한 오류, 결함, 장애 목록을 만들거나 획득해서 오류와 관련된 결함을 식별 및 노출하거나, 장애를 유발하는 테스트를 설계하도록 합니다.</p> <p>c) 정답이 아닙니다. 테스터는 테스트를 지원하기 위해 체크리스트 항목을 사용합니다. 오류 추정과 체크리스트 기반 테스트는 모두 이 리스트를 사용하지만, 이 리스트는 테스트 조건이 아닌 가능한 장애 목록이기 때문에, 가장 가능성이 높은 테스트 기법은 오류, 결함, 장애에 초점을 맞추는 결점 공격입니다.</p> <p>d) 정답이 아닙니다. 경계값 분석(BVA)은 동등 분할의 경계값 분석을 기반으로 합니다. 위 목록에는 동등 분할이나 그 경계가 언급되어 있지 않습니다.</p>	FL-4.4.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
27	d	<p>a) 정답이 아닙니다. 테스터가 체크리스트를 기반으로 상세한 테스트 케이스를 구현하고 실행할 수 있는 것은 맞지만, 이것이 어떻게 커버리지를 높일 수 있는지에 대해서는 설명하지 않습니다.</p> <p>b) 정답이 아닙니다. 체크리스트 항목을 자동화하면 안 됩니다. 만약 자동화한다 해도, 자동화된 테스트 스크립트는 항상 같은 방식으로 테스트를 실행하므로, 커버리지가 높아지지 않습니다.</p> <p>c) 정답이 아닙니다. 각 체크리스트 항목을 별도로 독립적으로 테스트해야 하는 것은 사실이지만, 이는 테스트 수행 순서에 영향을 미치며 달성된 커버리지에는 영향을 미치지 않으므로 커버리지가 높아지지 않습니다.</p> <p>d) 정답입니다. 체크리스트가 상위 수준이라면, 실제 테스트에서 어느정도 달라질 가능성이 있어 <u>잠재적으로 더 높은 커버리지</u>를 얻을 수 있지만, 재현 가능성은 떨어집니다. 두 명의 테스터가 상위 수준 체크리스트 항목을 따르는 경우, 각 테스터는 서로 다른 테스트 데이터, 테스트 단계 등을 사용할 수 있습니다. 이런 식으로, 한 테스터가 다른 테스터가 커버하지 않은 일부 영역을 커버함으로써 커버리지가 높아질 수 있습니다.</p>	FL-4.4.3	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
28	b	<p>a) 정답이 아닙니다. 이 인수 조건은 시스템이 준수해야 하는 규칙이나 규정(이 경우에는, 잊혀질 권리)을 기술한 것입니다. 이는 규칙 기반 인수 조건의 한 예시입니다.</p> <p>b) 정답입니다. 이 인수 조건은 시스템에서 실현 가능해야 하는 시나리오 예시를 기술하고 있습니다. 이는 시나리오 기반 인수 조건의 예시입니다.</p> <p>c) 정답이 아닙니다. 이 문장은 비즈니스 규칙을 구현하는 코드 라인에 더 가깝습니다. 인수 조건은 비즈니스 담당자와 협력해 작성해야 하기 때문에, 그들이 이해할 수 있는 언어로 작성해야 합니다. 이 문장은 이해관계자가 이해하지 못할 가능성이 높습니다.</p> <p>d) 정답이 아닙니다. 이 인수 조건은 시스템이 준수해야 할 규칙이나 규정이 무엇이며, 어떻게 준수를 보장할 것인지를 설명합니다. 따라서 이는 시나리오 기반 인수 조건이 아닌 규칙 기반 인수 조건입니다.</p>	FL-4.5.2	K2	1
29	d	<p>a) 정답이 아닙니다. 특별 사용자가 일반 사용자의 권한을 갖는지 확인하려면, 일반 사용자가 아닌 특별 사용자에게 대한 접근 권한을 테스트해야 합니다.</p> <p>b) 정답이 아닙니다. 특별 사용자가 일반 사용자가 갖는 권한을 확인하려면, 일반 사용자가 아닌 특별 사용자에게 대한 접근 권한을 테스트해야 합니다.</p> <p>c) 정답이 아닙니다. 인수 조건에 5층에 대한 설명은 없습니다. 테스트 케이스가 사용자 스토리의 범위를 확장해서는 안 됩니다. 하지만 부정 테스트를 수행하고 싶다 하더라도, 이 테스트는 AC3과는 직접적인 관련이 없습니다.</p> <p>d) 정답입니다. 이렇게 하면 특별 사용자가 일반 사용자가 접근 가능한 층에 출입할 수 있는지 확인할 수 있습니다.</p>	FL-4.5.3	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
30	a	<p>a) 정답입니다. 테스트 계획에는 (테스트 접근법의 일부로) 테스트 데이터 요구사항이 포함될 수 있지만, 테스트 케이스에 대한 자세한 테스트 데이터는 들어있지 않습니다. 테스트 데이터는 테스트 계획이 아닌 테스트 케이스의 일부입니다. 또한 테스트 계획을 만들 때 보통 컴포넌트가 어떻게 보일지 정확히 알 수 없기 때문에, 이런 데이터를 정의하는 것은 불가능합니다.</p> <p>b) 정답이 아닙니다. 테스트 계획의 목적 중 하나는 시작 조건과 완료 조건을 포함함으로써 수행된 테스트 활동이 수립된 기준을 충족하도록 돕는 것입니다. 코드 커버리지 기준은 단위 테스트 레벨에 대한 그런 기준의 예시입니다.</p> <p>c) 정답이 아닙니다. 보고서 양식은 테스트 계획에 들어가는 전형적인 내용입니다. 이는 의사소통 또는 보고의 표준 방식을 정의함으로써 이해관계자 간의 커뮤니케이션을 용이하게 하는데 도움이 됩니다.</p> <p>d) 정답이 아닙니다. 테스트 계획의 목적 중 하나는 테스트가 기존 테스트 정책과 테스트 전략을 준수할 것임을 보여주거나, 테스트가 왜 그것에서 벗어날지를 설명하는 것입니다. 이것은 준수할 (또는 준수하지 않을) 테스트 레벨과 관련해 편차를 설명하는 예시입니다.</p>	FL-5.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
31	c	<p>그래프를 통해 다음을 확인할 수 있습니다:</p> <p>$A(4)=6$ 이며 $A(3)=8$ (마지막 두 개의 회색 상자들).</p> <p>우리가 알게 된 공식으로부터:</p> <p>$E(5) = (3 \cdot A(4) + A(3)) / 4 = (3 \cdot 6 + 8) / 4 = 26 / 4 = 6.5$ 인/일.</p> <p>따라서:</p> <p>a) 정답이 아닙니다.</p> <p>b) 정답이 아닙니다.</p> <p>c) 정답입니다.</p> <p>d) 정답이 아닙니다.</p>	FL-5.1.4	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
32	a	<p>우선순위에 따라 테스트 케이스를 실행하되, 의존성도 고려해야 합니다.</p> <p>우선순위만 고려한다면, 먼저 TC 5와 TC 7(가장 높은 우선순위)을 실행하고, 그 다음 TC 1, TC 3, TC 4를 실행하고, 마지막으로 TC 2와 TC 6(가장 낮은 우선순위)를 실행할 것입니다.</p> <p>하지만 TC 7을 실행하기 위해서는 TC 4를 먼저 실행해야 합니다.</p> <p>TC 5를 실행하기 위해서는 TC 4와 TC 2를 실행해야 하는데 TC 1에 의해 TC 2가 차단되어, TC 2보다 먼저 실행해야 합니다.</p> <p>따라서 우선순위 1 테스트 케이스를 가능한 한 빨리 실행하기 위해서는</p> <p>TC 4- TC 7 – TC 1 – TC 2 – TC 5가 첫 다섯 개의 테스트 케이스가 되어야 합니다.</p> <p>다음으로, TC 6보다 우선순위가 높은 TC 3을 실행해야 합니다.</p> <p>따라서 전체 일정은 TC 4 – TC 7 – TC 1 – TC 2 – TC 5 – TC 3 – TC 6 이 됩니다.</p> <p>그러므로, 여섯 번째 테스트 케이스는 TC 3이 될 것입니다.</p> <p>따라서:</p> <p>a) 정답입니다.</p> <p>b) 정답이 아닙니다.</p> <p>c) 정답이 아닙니다.</p> <p>d) 정답이 아닙니다.</p>	FL-5.1.5	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
33	b	<p>a) 정답이 아닙니다. 테스트 피라미드 모델은 테스트 우선순위에 대한 정보를 제공하지 않습니다.</p> <p>b) 정답입니다. 테스트 피라미드 모델은 테스트마다 상세 수준이 다르다는 것을 보여줍니다.</p> <p>c) 정답이 아닙니다. 테스트 피라미드 모델은 커버리지 조건과 독립적입니다.</p> <p>d) 정답이 아닙니다. 테스트 피라미드 모델은 다른 테스트들 사이의 관계를 보여주지 않습니다.</p>	FL-5.1.6	K1	1
34	d	<p>a) 정답이 아닙니다. 테스트 사분면은 여러 기준에 따라 테스트 레벨과 테스트 유형을 별도로 그룹화합니다. 테스트 레벨과 테스트 유형의 조합을 나타내지 않으며 SDLC 내의 어떤 위치와도 관련이 없습니다. 테스트 사분면 모델에서는 테스트 레벨과 테스트 유형을 모두 별도로 취급합니다.</p> <p>b) 정답이 아닙니다. 테스트 사분면은 여러 기준에 따라 테스트 레벨과 테스트 유형을 그룹화합니다. 각 테스트 레벨에서 수행되는 개별 테스트 유형의 상세화 정도를 설명하지 않습니다. 테스트 레벨과 관련된 이런 모델을 테스트 피라미드라고 합니다.</p> <p>c) 정답이 아닙니다. 일반적으로 모든 테스트 유형은 모든 테스트 레벨에서 수행할 수 있기 때문에 이 설명은 잘못되었습니다.</p> <p>d) 정답입니다. 테스트 사분면은 애자일 소프트웨어 개발에서 테스트 레벨, 테스트 유형, 활동, 테스트 기법 및 작업 산출물을 그룹화합니다. 이 모델에서 테스트는 비즈니스 측면과 기술 측면이 있을 수 있습니다. 테스트는 팀을 지원하거나(즉, 개발을 이끌거나) 제품을 평가할 수 있습니다(즉, 기대치 대비 동작 측정). 이 두 관점의 조합이 4사분면을 결정합니다.</p>	FL-5.1.7	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
35	c	<p>a) 정답이 아닙니다. 리스크 모니터링은 리스크 분석이 아니라 리스크 제어의 일부입니다.</p> <p>b) 정답이 아닙니다. 리스크 식별 자체만으로는 리스크 완화 활동을 수행할 수 없습니다. 완화 조치는 리스크 제어 단계에서 정의됩니다.</p> <p>c) 정답입니다. 리스크 분석이 테스트의 철저성과 범위에 어떤 영향을 미치는지 보여주는 예시입니다.</p> <p>d) 정답이 아닙니다. 커버리지 항목은 리스크 분석이 아닌 테스트 기법을 이용해 도출합니다.</p>	FL-5.2.3	K2	1
36	b	<p>a) 정답이 아닙니다. 테스트 진행 상황 보고서는 주로 테스트 설계가 아니라, 테스트 모니터링 및 제어, 테스트 완료에 사용됩니다.</p> <p>b) 정답입니다. 테스트 완료 보고서는 프로젝트, 테스트 레벨, 또는 테스트 유형이 완료되고, 이상적으로는 종료 조건이 충족되었을 때, 테스트 완료 활동에서 작성합니다. 이 보고서는 테스트 진행 상황 보고서와 기타 데이터의 정보를 사용합니다.</p> <p>c) 정답이 아닙니다. 테스트 진행 상황 보고서는 테스트 분석이 아닌 테스트 모니터링 및 제어, 시험 완료 중에 주로 사용됩니다.</p> <p>d) 정답이 아닙니다. 테스트 진행 상황 보고서는 테스트 계획이 아닌 테스트 모니터링 및 제어, 테스트 완료 중에 주로 사용됩니다.</p>	FL-5.3.2	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
37	d	<p>a) 정답이 아닙니다. 사용자가 소프트웨어 장애를 보고할 때 커밋의 고유 식별자 덕분에 사용자가 사용한 소프트웨어 버전 (및 해당하는 테스트 스크립트 버전)의 파일을 다시 재구성해 장애를 재현하고 더 빠르게 결함을 찾을 수 있습니다.</p> <p>b) 정답이 아닙니다. 테스트 환경의 변경으로 인해 테스트 도중 예기치 않은 문제가 발생하면 형상 관리로 테스터는 이전 버전의 환경으로 롤백할 수 있습니다. 이를 통해 테스트가 변경 사항에 영향을 받지 않고 계속 진행될 수 있습니다.</p> <p>c) 정답이 아닙니다. 형상 관리는 식별된 모든 문서(예: 요구사항 명세) 및 소프트웨어 항목이 테스트 문서(예: 테스트 계획)에서 명확하게 참조되도록 보장합니다.</p> <p>d) 정답입니다. 이는 형상 관리 프로세스가 아닌 결함 관리에 의해 보장됩니다.</p>	FL-5.4.1	K2	1
38	b	<p>a) 정답이 아닙니다. 이는 중요하지만, 테스트 환경 요소만큼 중요하지는 않습니다.</p> <p>b) 정답입니다. 중요한 것은 테스트에 사용된 브라우저와 장치의 식별입니다. 이런 결함이 브라우저 또는 장치에 따라 다를 수 있기 때문에 브라우저와 장치 정보는 중요합니다. 예를 들어, 한 브라우저(또는 특정 브라우저의 한 버전)에서는 로그인 버튼이 잘 작동하지만 다른 브라우저에서는 작동하지 않을 수 있습니다. 따라서 브라우저와 장치 정보는 개발자가 문제를 재현하고 문제의 근본 원인을 더 빨리 찾을 수 있도록 도울 수 있습니다.</p> <p>c) 정답이 아닙니다. 테스트 대상이 식별되었습니다 (WebShop V0.99).</p> <p>d) 정답이 아닙니다. 영향이 포함되어 있습니다 – 이것은 심각도입니다 (높음)</p>	FL-5.5.1	K3	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
39	d	<p>a) 정답이 아닙니다. 테스트 실행 및 커버리지 도구는 테스트 케이스의 자동 실행과 해당 테스트 케이스를 실행함으로써 달성되는 커버리지 측정을 용이하게 합니다. 그러나 이런 도구는 결함 및 형상 관리의 체계화에는 도움이 되지 않습니다.</p> <p>b) 정답이 아닙니다. 테스트 설계 및 구현 도구는 테스트 케이스, 테스트 데이터, 테스트 절차의 생성을 용이하게 하지만, 결함 및 형상 관리의 체계화에는 도움이 되지 않습니다.</p> <p>c) 정답이 아닙니다. 결함 관리 도구는 결함 관리에 사용되지만 테스트 도구는 아니며, 테스트 케이스나 형상 관리의 체계화하는 데 사용되지 않습니다.</p> <p>d) 정답입니다. 테스트 관리 도구는 소프트웨어 개발수명주기(SDLC), 요구사항, 테스트, 결함, 형상 관리 등의 관리를 용이하게 하여 테스트 프로세스 효율성을 높입니다.</p>	FL-6.1.1	K2	1

문제 번호 (#)	정답	해설/근거	학습목표 (LO)	K-레벨	배점
40	d	<p>a) 정답이 아닙니다. '테스트 베이스에 접근없이 테스트 케이스를 생성'은 불가능합니다. 테스터나 도구에 의한 테스트 케이스 생성은 테스트 베이스에 대한 접근을 필요로 합니다.</p> <p>b) 정답이 아닙니다. '더욱 객관적인 평가를 통해 커버리지 높이기'은 테스트 자동화의 직접적인 이점은 아닙니다. 테스트 자동화는 커버리지에 대한 더 객관적인 평가를 제공할 수 있지만, 그 객관적인 평가가 커버리지를 높이지는 않을 것입니다. 커버리지 결과를 사용해 추가 테스트 케이스를 작성해야만 커버리지가 향상될 수 있습니다.</p> <p>c) 정답이 아닙니다. '연산 처리 능력이 높아져 테스트 실행 가능 시간 증가'는 모순된 표현으로, 처리 능력이 높아지면 일반적으로 실행 시간이 줄어들고, 실행 시간 증가는 테스트 시간을 더 오래 걸리게 만들기 때문에 이점이 없습니다.</p> <p>d) 정답입니다. 테스트 자동화의 이점은 테스트 자동화가 인적 오류에 시달리지 않기 때문에 일관성과 재현 가능성을 높여 인적 오류를 방지하는 것입니다. 예를 들어, 요구사항에서 테스트가 일관성 있게 도출되고, 테스트 데이터가 체계적으로 생성되며, 테스트가 동일한 빈도와 동일한 순서로 도구에 의해 실행된다는 것을 의미합니다.</p>	FL-6.2.1	K1	1