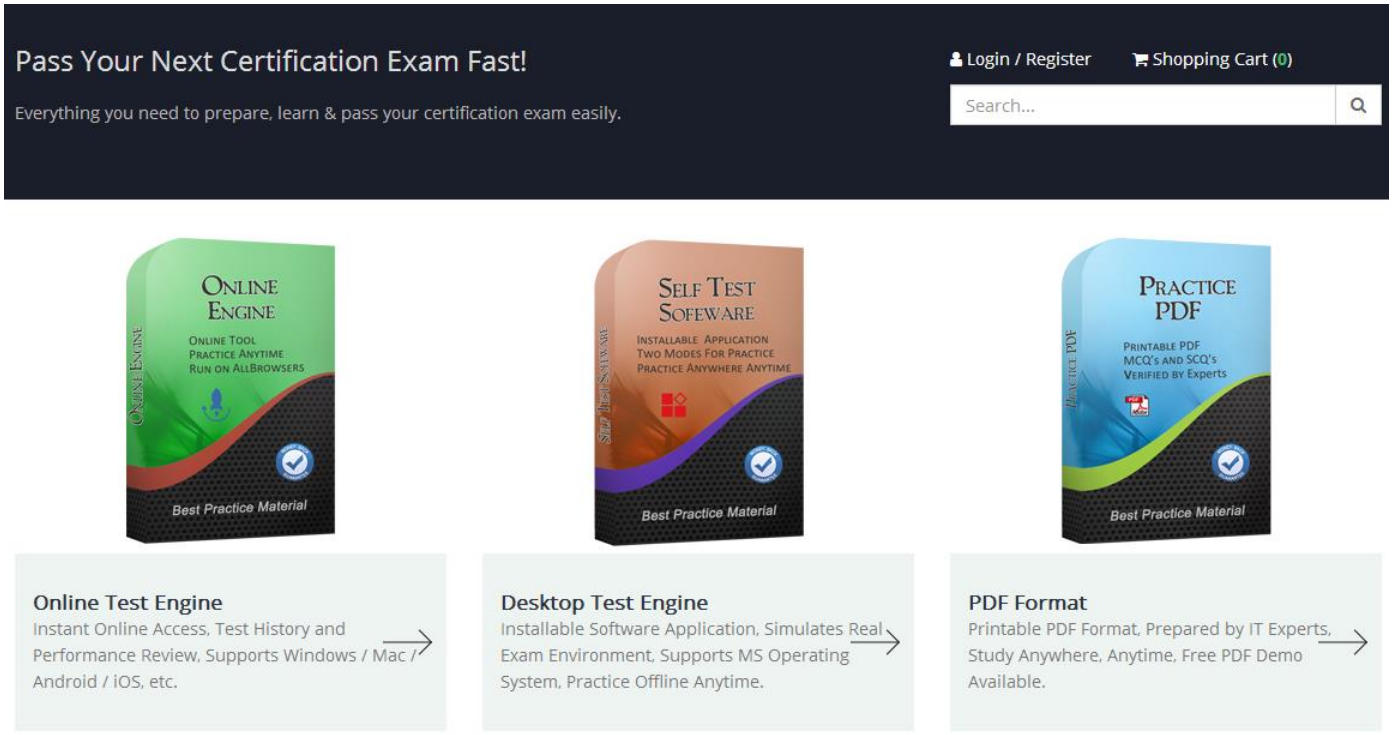
Actual4Labs



Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarante in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

http://www.actual4labs.com

Excellent Quality Exam Dumps Questions Never Let You down - Actual4Labs

Exam : **CTFL_Syll2018-KR**

Title : ISTQB Certified Tester Foundation Level (Syllabus 2018) (CTFL_Syll2018 Korean Version)

Vendor : ISQI

Version : DEMO

QUESTION NO: 1
논리적 조건으로 구성된 시스템 요구 사항을 효과적으로 테스트하기 위해 어떤 테스트 설계 기술을 사용할 수 있습니까?
A. 경계값 분석
B. 사용 사례 테스트
C. 등가 분할
D. 의사결정 테이블 테스트
Answer: D
Explanation of Correct answer: Decision table testing is a test design technique that can be used for effectively testing system requirements consisting of logical conditions. Decision table testing is based on the idea that the system behavior can be represented by a table, where each column corresponds to a possible combination of inputs and each row corresponds to the expected outputs for that combination. Decision table testing helps to identify and cover all possible scenarios and outcomes for a given requirement12.
References: 1: Certified Tester Foundation Level Syllabus, Version 2018 v3.1.1, Section 4.3.2 2: A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.3.2

QUESTION NO: 2
일반적으로 종료 기준은 다음으로 구성될 수 있습니다.
A. 테스트 문서의 양, 세부 구조 수준 및 템플릿을 정의합니다.
B. 결함 밀도 또는 신뢰성 측정 추정치.
C. 취해진 테스트 접근 방식의 적절성.
D. 재해 복구에 대한 논의.
Answer: B
Explanation:
These are examples of exit criteria, which are the conditions that must be met before testing can be completed.
Exit criteria may also include test coverage measures, residual risk assessment, or stakeholder approval. The other options are not exit criteria, but rather test planning activities .

QUESTION NO: 3
조직이 테스트 도구 사용을 고려할 때 다음을 수행해야 합니다.
A. 도구는 프로세스 반복성을 강제하고 따라서 좋은 테스트 프로세스를 시행하므로 좋은 테스트 프로세스를 정의하는 데 도움이 되는 도구를 사용하십시오.
B. 항상 자동화된 테스트 실행 도구를 도입하는 것부터 시작하십시오. 이러한 도구는 투자 수익이 가장 높으므로 먼저 도입해야 합니다.
C. 테스트 프로세스를 분석한 후 도구 지원 도입을 통해 지원 가능한지 평가합니다.
D. 도구는 기술적이며 개발자는 테스트 도구 선택 및 구성에 대해 조언할 수 있는 적절한 기술을 갖고 있으므로 개발자가 테스트 도구를 선택할 수 있도록 허용합니다.
Answer: C
Explanation:
When an organization considers the use of testing tools, they should C. Perform analysis of

the test process and then assess whether it can be supported through the introduction of tool support. Testing tools can provide many benefits to the test process, such as increased efficiency, effectiveness, consistency, quality, etc.
However, testing tools also have some challenges and risks associated with them, such as cost, learning curve, compatibility, maintenance, etc. Therefore, before introducing any testing tool, an organization should analyze their current test process and identify their needs, goals, expectations, constraints, etc., and then evaluate whether a tool can support them or not. A detailed explanation of testing tools can be found in Software Testing Foundations: A Study Guide for the Certified Tester Exam, pages 193-1983.

QUESTION NO: 4
다음 중 유지 관리 테스트를 위한 트리거의 가장 좋은 예를 제공하는 설명은 무엇입니까?
A. 고객 서비스 애플리케이션 종료
B. 전자상거래 웹 애플리케이션 개발 프로젝트에 자금을 지원하기 위한 사업 승인
C. 애플리케이션에 대한 요구사항 사양 완료
D. 애플리케이션에 대한 디자인 완성
Answer: A
Explanation:
Maintenance testing is triggered by changes, such as migration, retirement, or enhancement of a system1.
Therefore, option A is the best example of a trigger for maintenance testing. Options B, C, and D are examples of triggers for development testing, not maintenance testing1.
References: 1, Section 2.3.2

QUESTION NO: 5
통합 테스트란 무엇입니까?
A. 더 큰 구성요소나 하위 시스템에서 결함 찾기
B. 시스템 무결성 테스트의 또 다른 용어
C. 어떤 구성 요소를 어떤 순서로 통합할지 지정
D. 인터페이스가 올바르게 작동하는지 테스트
Answer: D
Explanation:
Integration testing is the process of testing the interactions between different components or subsystems of a system. Integration testing verifies that the interfaces work correctly, meaning that they pass data and control correctly between components or subsystems, and that they handle errors and exceptions properly. Integration testing can also verify the functionality and quality of the integrated system or subsystem.
References: Certified Tester Foundation Level Syllabus, Section 5.1.

QUESTION NO: 6
테스트 및 디버깅에 대한 다음 문장 중 올바른 것은 무엇입니까?
A. 디버깅이 실패를 발견하고 분석했는지 확인하는 재테스트
B. 동적 테스트를 통해 결함을 찾아내고 디버깅을 통해 오류를 제거합니다.
C. 동적 테스트를 통해 오류가 드러나고 디버깅을 통해 결함이 제거됩니다.

D. 대부분의 개발 활동과 마찬가지로 디버깅은 일반적으로 테스트가 시작되기 전에 수행됩니다.

Answer: C

Explanation:

Testing and debugging are two different activities that are related to finding and removing defects and failures in software. Testing is the process of evaluating software by applying test cases to find failures and provide information on its quality. Debugging is the process of finding, analyzing, and removing the causes of failures in software. Testing reveals failures, which are deviations of the actual behavior of the software from its expected behavior. Debugging removes defects, which are flaws in the software that cause failures.

References: Certified Tester Foundation Level Syllabus, Section 1.2.1

QUESTION NO: 7

다음 중 테스트 프로세스 전반에 걸쳐 추적성을 유지할 수 있도록 테스트웨어의 모든 항목을 식별하고, 버전을 제어하고, 변경 사항을 추적하는 프로세스는 무엇입니까?

A. 소프트웨어 추적성 프로세스

B. 사고 관리 프로세스

C. 테스트 디자인 프로세스

D. 구성 관리 프로세스

Answer: D

Explanation:

The process that ensures that all items of testware are identified, version controlled, tracked for changes, so that traceability can be maintained throughout the test process is D. Configuration management process.

Configuration management is a process that establishes and maintains the integrity and consistency of all items of testware throughout the test process. Configuration management involves identifying, storing, controlling, tracking, and auditing all items of testware (such as test plans, test cases, test scripts, test data, test results, etc.) and their versions and changes. Configuration management helps to ensure that only authorized and approved items of testware are used for testing and that traceability can be maintained between them.

A detailed explanation of configuration management can be found in A Study Guide to the ISTQB Foundation Level

2018 Syllabus, pages 101-1021.

QUESTION NO: 8

다음 소프트웨어 개발 모델 중 초기 테스트 원칙을 지원하지 않는 모델을 가장 잘 보여주는 것은 무엇입니까?

A. 폭포 모델

B. V 모델

C. 증분 개발 모델

D. 반복 개발 모델

Answer: A

Explanation:

According to the syllabus, a software development life cycle (SDLC) model is a conceptual

framework that describes the phases and activities involved in developing a software product. Different SDLC models have different advantages and disadvantages depending on the project context and objectives. The principle of early testing states that testing activities should start as early as possible in the software development life cycle and be focused on defined objectives. Early testing helps to prevent defects, reduce rework, lower costs, and improve quality. The answer A is correct because it best exemplifies a model that does not support the principle of early testing. The waterfall model is a sequential SDLC model that divides the development process into distinct phases, such as requirements analysis, design, implementation, testing, and maintenance.
Each phase must be completed before the next phase can begin, and there is no overlap or iteration between phases. Testing is done only after the implementation phase, which means that defects are detected late in the development cycle and are more expensive and difficult to fix. The other answers are incorrect because they exemplify models that support the principle of early testing. The V-model is an extension of the waterfall model that emphasizes verification and validation activities at each phase of development. Testing is done in parallel with each corresponding development phase, which means that defects are detected early and feedback is provided to the developers. The incremental development model is an iterative SDLC model that divides the development process into smaller increments or iterations, each delivering a working software product or a subset of features. Testing is done at the end of each iteration, which means that defects are detected early and feedback is provided to the developers. The iterative development model is another iterative SDLC model that repeats the development process for each iteration, with each iteration producing an improved version of the software product or a subset of features. Testing is done throughout each iteration, which means that defects are detected early and feedback is provided to the developers.
References: Certified Tester Foundation Level Syllabus, Section 1.1.1, page 9-10.

QUESTION NO: 9
기본 테스트 프로세스의 어떤 활동에 테스트 스크립트 생성이 포함됩니까?
A. 테스트 종료 활동
B. 테스트 분석 및 설계
C. 테스트 계획 및 제어
D. 테스트 구현 및 실행
Answer: D
Explanation:
Test implementation and execution is the activity in the fundamental test process that includes test script generation. Test script generation is the process of creating executable test cases from test conditions and test data2 defines this activity as follows:
Test implementation and execution has the following major tasks:
* Develop and prioritize test cases, creating test data and writing test procedures.
* Check test environment has been set up correctly.
* Execute test cases, evaluate results and document deviations from expected results.
* Use exit criteria to report on suitability of system under test.
Test closure activities (A) include finalizing and archiving test results, evaluating the test process, identifying areas for improvement, and celebrating achievements. Test analysis and

design (B) include reviewing test basis, identifying test conditions, designing high-level test cases, and defining exit criteria. Test planning and control include defining test objectives, scope, strategy, resources, schedule, risks, and metrics.

QUESTION NO: 10
다음 설명 중 테스트의 일반 원칙과 모순되는 것은 무엇입니까?
A. 새로운 결함이 발견되면 동일한 테스트 세트를 더 자주 실행해야 합니다.
B. 테스트가 수행되는 방식은 특정 프로젝트의 상황에 따라 다릅니다.
C. 테스트는 프로젝트 초기에 시작하는 것이 좋습니다.
D. 대부분의 결함은 시스템 모듈의 작은 하위 집합에서 발견됩니다.
Answer: A
Explanation:
The general principles of testing state that testing can show the presence of defects, but not their absence.
Running the same test set more often will not increase the likelihood of finding new defects, unless the system or its environment changes. Therefore, statement A contradicts the general principles of testing. Statement B is true, as testing is context-dependent and should be tailored to the specific situation of a project. Statement C is also true, as early testing can help prevent defects and reduce rework. Statement D is true, as it reflects the Pareto principle or the 80/20 rule, which states that most defects are found in a small subset of a system's modules.
References: ISTQB Certified Tester Foundation Level Syllabus 2018, Section 1.2

QUESTION NO: 11
테스트는 테스트 중인 소프트웨어 또는 시스템의 릴리스에 대해 정보에 입각한 결정을 내릴 수 있도록 이해관계자에게 충분한 정보를 제공해야 합니다. 다음 중 어떤 기본 테스트 프로세스 활동에서 테스트의 충분성과 결과 정보가 평가됩니까?
A. 구현 및 실행
B. 요구사항 사양
C. 종료 기준 평가 및 보고.
D. 분석 및 설계
Answer: C
Explanation:
The fundamental test process activity where the sufficiency of testing and resulting information are assessed is Evaluating exit criteria and reporting. This activity involves checking whether the test objectives have been met and whether there are any unresolved issues or risks that could affect the release or deployment decision.
This activity also involves preparing and communicating a test summary report that summarizes the test activities and results and provides recommendations and feedback for improvement. You can find more information about Evaluating exit criteria and reporting in [A Study Guide to the ISTQB Foundation Level
2018 Syllabus], Chapter 3, Section 3.5.

QUESTION NO: 12
다음 목록 중 기본 테스트 프로세스의 주요 활동의 올바른 순서를 나타내는 것은

무엇입니까(다른 모든 활동과 동시에 발생해야 하는 제어 활동은 제외)?
A. 계획, 분석 및 보고, 설계 및 구현, 실행, 테스트 종료 활동, 종료 기준 평가.
B. 계획, 분석, 설계 및 구현, 실행, 기록, 테스트 종료 활동, 종료 기준 평가.
C. 계획, 분석 및 설계, 실행, 로깅 및 보고, 회귀 테스트
D. 계획, 분석 및 설계, 구현 및 실행, 평가 종료 기준 및 보고, 테스트 종료 활동
Answer: D

QUESTION NO: 13
다음 문장은 소프트웨어 테스트 문서 표준' 사양(IEEE 829)을 나타냅니다.
어떤 문장이 맞나요?
A. 고품질 테스트 문서화 체제의 핵심은 이 표준을 엄격하게 준수하는 것입니다.
B. 이 표준에서 벗어나는 경우 경영진, 마케팅 및 개발 부서의 승인을 받아야 합니다.
C. 이 테스트 계획 개요는 군사 프로젝트와 관련이 있습니다. 소비자 시장 프로젝트의 경우 더 적은 수의 항목으로 다른 사양이 있습니다.
D. 대부분의 테스트 문서 체계는 특정 상황이나 조직에 맞게 변경하여 어느 정도 이 사양을 따릅니다.
Answer: D
Explanation:
The 'Standard for Software Test Documentation' specification (IEEE 829) is a standard that defines a set of documents that can be used to document the test process and its outcomes. The standard provides an outline for each document, specifying its purpose, content, and format. However, the standard does not prescribe how to apply it in different contexts or projects. It is up to each organization or project to decide how to adapt the standard to their specific needs and situation. Therefore, the standard is not a rigid or mandatory requirement that must be followed strictly by all testers. Rather, it is a flexible and adaptable guideline that can be used as a reference or a starting point for creating test documentation regimes. You can find more information about IEEE 829 and test documentation in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter
51.

QUESTION NO: 14
다음 중 회귀 테스트를 설명하는 설명은 무엇입니까?
I. 수정된 결함을 다시 테스트함
II 이미 테스트된 프로그램의 테스트
III. 프로그램의 새로운 기능 테스트
IV. 회귀 테스트는 기능 테스트에만 적용됩니다.
V 반복할 필요가 없는 테스트. 한 번만 사용되기 때문에
A. II
B. II. IV, V
C. I.IV
D. I, III, IV
Answer: A
Explanation:
Regression testing is testing of an already tested program after modification to discover any

defects introduced or uncovered as a result of changes in software or its environment. Regression testing ensures that previously working functionality still works after changes are made3 defines regression testing as follows:
Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.
Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.
I, III, IV, and V are incorrect statements about regression testing. Retesting of a fixed defect (I) is not regression testing, but confirmation testing or rework testing, which verifies that a specific defect has been resolved. Testing of new functionality in a program (III) is not regression testing, but functional testing or new feature testing, which verifies that new requirements are met by new code. Regression testing applies only to functional testing (IV) is not true, as regression testing can also apply to non-functional aspects such as performance or security. Tests that do not have to be repeatable, because they are only used once (V) are not regression testing, but exploratory testing or ad hoc testing, which are based on learning and discovery rather than predefined test cases.

QUESTION NO: 15
ISTQB 기본 테스트 프로세스는 5가지 주요 활동으로 구성됩니다.
"필요한 테스트 데이터 식별"?
A. 테스트 기준 평가 및 보고
B. 테스트 구현 및 실행
C. 테스트 계획 및 제어
D. 테스트 분석 및 설계
Answer: D
Explanation:
Test analysis and design is the activity in the fundamental test process that includes identifying necessary test data. Test data are the inputs that are used to execute the test cases and verify the expected results2 defines this activity as follows:
Test analysis and design has the following major tasks:
* Reviewing the test basis (such as requirements, risk analysis reports, design documents or code).
* Identifying test conditions based on analysis of test items, specifications, behavior and structure of the software.
* Designing high-level test cases based on test conditions and designing techniques.
* Evaluating testability of requirements and system under test.
* Defining exit criteria.
Evaluating exit criteria and reporting (A) is part of the test closure activities, where the results of testing are evaluated against the defined objectives. Test implementation and execution (B) is where the test cases are executed using the identified test data and deviations from expected results are documented. Test planning and control is where the overall approach and resources for testing are defined and monitored.

QUESTION NO: 16
다음 테스트 도구 중 테스트 실행 도구로 분류되는 두 가지는 무엇입니까? [K2] 가. 테스트

데이터 준비 도구 b. 테스트 하니스 다. 검토 도구 d. 테스트 비교기 e. 구성 관리 도구
A. a와 b
B. c와 d
C. c와 e
D. b와 d
Answer: D
Explanation:
The test tools that would be classified as test execution tools are D. b and d. Test execution tools are tools that automate the execution of test cases or test scripts, and compare the actual results with the expected results.
Test execution tools can also record and replay user actions, generate test data, and report test results. Test harness and test comparators are examples of test execution tools. A test harness is a tool that creates a test environment for a component or system under test, by simulating the required dependencies, such as stubs, drivers, or mock objects. A test comparator is a tool that compares the actual outputs of a component or system under test with the expected outputs, and reports any differences or anomalies. A detailed explanation of test execution tools can be found in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, pages
111-1121.

QUESTION NO: 17
Use Case를 기반으로 테스트 케이스를 생성할 때 어떤 유형의 테스트가 생성되나요?
A. 구조 테스트
B. 기능 테스트
C. 성능 테스트
D. 회귀 테스트
Answer: B
Explanation:
Functional test is the type of test that is created when a test case is based on a use case. A use case is a description of how a system interacts with one or more actors (users or other systems) to achieve a specific goal or function. A functional test is a test that verifies that a system or software component performs its specified functions according to its requirements. Functional tests can be derived from use cases by identifying test scenarios and test cases that cover the main flow and alternative flows of each use case2 explains this as follows:
Use cases are one of the most commonly used techniques for analyzing and modeling functional requirements for a system. A use case describes how an actor interacts with a system to accomplish a specific goal.
Functional Testing is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions are tested by feeding them input and examining the output.
Use cases can be used as a source for deriving functional tests by identifying test scenarios and test cases that cover the main flow and alternative flows of each use case.
A, C, and D are incorrect answers. Structural test, performance test, and regression test are not types of tests that are created when a test case is based on a use case. Structural test is

a type of test that is based on the internal structure and logic of the code rather than the functionality or requirements. Performance test is a type of test that measures the speed, responsiveness, scalability, or stability of a system under various workloads or conditions. Regression test is a type of test that verifies that previously working functionality still works after changes are made to the system or its environment.

QUESTION NO: 18
일반적으로 "경험 기반" 테스트 설계 기술은 다음과 같습니다.
A. 의사결정 테이블을 사용하여 실행할 부울 테스트 조건을 생성합니다.
B. 구성요소, 통합 또는 시스템 수준에서 시스템 또는 소프트웨어의 구조를 식별합니다.
C. 테스터의 기술, 직관 및 경험을 활용하여 오류 추측 및 탐색적 테스트를 통해 테스트 사례를 도출합니다.
D. 테스트 조건부터 사양 및 요구 사항까지 추적성을 확립합니다.
Answer: C
Explanation:
Experience-based test design techniques are techniques that use the skill, intuition, and experience of testers to derive test cases, using error guessing and exploratory testing1. Error guessing is a technique that uses common sense and previous experience to guess where defects might occur in a system1. Exploratory testing is an approach that involves simultaneous learning, test design, and test execution1. Experience-based test design techniques are typically used when there is insufficient information or time to apply other more formal techniques1. They do not use decision tables, identify the structure of the system or software, or establish traceability from test conditions back to the specifications and requirements.

QUESTION NO: 19
다음 중 일반적으로 도구별 정적 분석을 사용하여 식별되는 것은 무엇입니까? [K1]
A. 오류 메시지의 철자 오류
B. 잠재적인 무한 루프
C. 메모리 누수
D. 잘못된 값으로 설정된 변수
Answer: B
Explanation:
A potential infinite loop is a type of defect that can be typically identified using static analysis by tools1. Static analysis is a technique that examines the source code or other software artifacts without executing them, and can detect defects, vulnerabilities, code smells, and deviations from standards early in the development process1. Static analysis tools are software tools that automate the static analysis technique and provide various features and functionalities to support it1. Static analysis tools can identify a potential infinite loop by analyzing the control flow and data flow of the source code and checking for conditions or statements that may cause an endless repetition or iteration1. A potential infinite loop can cause serious problems in software performance, functionality, reliability, and security1. Therefore, a potential infinite loop is a type of defect that can be typically identified using static analysis by tools.

QUESTION NO: 20
다음 중 일반 시장을 대상으로 하고 Agile 방법을 사용하여 구축된 새로운 소프트웨어 제품의 첫 번째 릴리스에 대한 사용자 승인 테스트에 적합한 테스트 목표는 무엇입니까? [K2] 가. 가능한 많은 결함을 식별하기 위해 b. 코드 적용 범위를 최대화하려면 c. 제품이 예상대로 작동하는지 확인하기 위해 d. 제품의 전반적인 품질을 평가하기 위해 e. 제품의 신뢰성을 확인하기 위해
A. b와 c
B. a와 d
C. b와 e
D. c 및 d
Answer: D
Explanation:
* The test objectives for user acceptance testing of the first release of a new software product aimed at a
* general market and built using Agile methods are D. c and d. User acceptance testing is a level of testing that involves users or customers validating that the software product meets their needs and expectations before accepting it for use or deployment. User acceptance testing can have different objectives depending on the context and purpose of the software product. For a new software product aimed at a general market and built using Agile methods, some possible objectives are:
* To ensure the product works as expected, which means verifying that the product meets the functional and non-functional requirements agreed upon by the users or customers and the development team during each iteration or sprint of the Agile development process.
* To assess the overall quality of the product, which means evaluating how well the product satisfies the quality attributes that are important for the users or customers, such as usability, performance, reliability, security, etc.
A detailed explanation of user acceptance testing can be found in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], pages 79-80.

QUESTION NO: 21
웹 사이트의 로그인 페이지를 테스트해야 합니다. 페이지에 사용자 이름 및 비밀번호 필드가 포함되어 있습니다. 이 경우에 가장 적합한 테스트 설계 기술은 무엇입니까?
A. 의사결정 테이블 테스트, 상태 전환 테스트.
B. 등가 분할, 경계값 분석.
C. 탐색적 테스트, 설명 범위.
D. 의사결정 커버리지, 결함 공격.
Answer: A
Explanation:
Provides developers with information to isolate the failure is the incident report objective that this excerpt satisfies, because it gives details about how to recreate the failure using a specific test file and location . An incident report is a document that records any event that deviates from the expected or desired behaviour of the software under test . One of its objectives is to provide developers with information to isolate the failure, which means to identify and locate the cause of the failure in the software . The other options are not incident

report objectives that this excerpt satisfies. Option B is wrong, because this excerpt does not belong to an incident report, but rather to a test report. A test report is a document that summarizes the results and outcomes of testing activities . Option C is wrong, because this excerpt does not provide project managers with information on the project risks, which means to identify and assess the potential threats or uncertainties that may affect the project objectives or outcomes . Option D is wrong, because this excerpt does not provide test managers with information to report test progress, which means to measure and communicate how much of the planned testing activities have been completed or remain to be done .

QUESTION NO: 22
다음 의사 코드를 고려하십시오.
1. 시작
2. 성별 읽기
3. __"친애하는"이라고 인쇄하세요.
4. 성별 = '여성'인 경우
5. 인쇄 ("Ms")
6. 기타
7. __인쇄("미스터")
8. 엔디프
9. 종료
100% 의사결정 적용 범위를 달성하려면 몇 개의 테스트 사례가 필요합니까?
A. 1
B. 2
C. 3
D. 4
Answer: B
Explanation:
Decision coverage is a structure-based test technique that involves testing every decision point in the code by testing all possible outcomes or branches of each decision point. Decision coverage requires some knowledge of the internal structure or implementation of the software system; it focuses on how the system does what it does rather than what it does.
The pseudo code given in the question has one decision point at line 4, where the value of Gender is compared to 'female'. This decision point has two possible outcomes or branches:
* True branch: If Gender = 'female', then print "Ms" at line 5.
* False branch: If Gender != 'female', then print "Mr" at line 7.
To achieve 100 per cent decision coverage, we need to test both outcomes or branches of the decision point at least once. Therefore, we need two test cases that cover both scenarios :
* TC1: Gender = 'female' -> Print "Dear Ms"
* TC2: Gender = 'male' -> Print "Dear Mr"
The other options are not correct because they do not provide enough or too many test cases to achieve 100 per cent decision coverage. For example:
* A. 1: This option is not enough because it only provides one test case that covers one

outcome or branch of the decision point, but not the other.
* C. 3: This option is too many because it provides three test cases that cover both outcomes or branches of the decision point, but one of them is redundant or unnecessary.
* D. 4: This option is too many because it provides four test cases that cover both outcomes or branches of the decision point, but two of them are redundant or unnecessary.
You can find more information about decision coverage and structure-based testing in [A Study Guide to the ISTQB Foundation Level 2018 Syllabus], Chapter 4, Section 4.3.

QUESTION NO: 23
다음 중 통합 테스트 테스트 수준을 정확하게 정의하는 것은 무엇입니까? [K2]
A. 테스트 기반에는 소프트웨어 및 시스템 설계가 포함되고 테스트 개체에는 인터페이스가 포함되며 테스트는 시스템의 여러 부분 간의 상호 작용에 중점을 둡니다.
B. 테스트 기반에는 구성 요소 요구 사항이 포함되고, 테스트 개체에는 데이터베이스 모듈이 포함되며, 테스트는 전체 시스템 동작에 중점을 둡니다.
C. 테스트 기반에는 비즈니스 프로세스가 포함되고 테스트 개체에는 시스템 구성 및 구성 데이터가 포함되며 테스트는 시스템에 대한 신뢰도 구축에 중점을 둡니다.
D. 테스트 기반에는 사용 사례가 포함되고 테스트 개체에는 사용자 절차가 포함되며 테스트는 시스템 동작의 상위 수준 모델에 집중됩니다.
Answer: A
Explanation:
Integration testing is a test level that accurately defines the integration testing test level1. Integration testing is a type of testing that verifies that different components or systems work together correctly and consistently1. The test basis for integration testing includes software and system design documents that specify how the components or systems are integrated and interact with each other1. The test objects for integration testing include interfaces between components or systems that enable data exchange or communication1. The tests for integration testing concentrate on the interactions between different parts of a system and check for functional, performance, reliability, security, compatibility, and interoperability issues1. Therefore, integration testing is a test level that accurately defines the integration testing test level.

QUESTION NO: 24
다음 중 소프트웨어 코드에 결함이 발생하는 이유는 무엇입니까?
A. 테스트를 위해 코드를 넘겨야 하는 촉박한 기한을 맞추기 위해 서두르고 있습니다.
B. 부적절한 시스템 테스트
C. 부적절한 단위 테스트
D. 동적 테스트보다는 정적 테스트에 중점을 둡니다.
Answer: A
Explanation:
According to the syllabus, a defect is a flaw in a component or system that can cause it to fail to perform its required function. A defect can be introduced in any phase of the software development life cycle, such as requirements, design, coding, testing, or deployment. A possible reason for introducing a defect in software code is rushing to meet a tight deadline to turn code over for testing. This can lead to careless mistakes, incomplete functionality, or poor quality code. The answer A is correct because it is an example of a possible reason for

introducing a defect in software code. The other answers are incorrect because they are not reasons for introducing defects in software code, but rather consequences or detection methods of defects.
References: [Certified Tester Foundation Level Syllabus], Section 1.3.1, page 12-13.

QUESTION NO: 25
다음 중 "검증"과 "검증"이라는 용어를 가장 잘 구별하는 것은 무엇입니까?
A. 검증은 객관적인 증거 제공을 통해 지정된 요구 사항이 충족되었음을 확인하는 것이며, 검증은 객관적인 증거 제공을 통해 특정 사용 목적에 대한 요구 사항이 충족되었음을 확인하는 것입니다.
B. 검증은 주관적 증거 제공을 통해 지정된 요구 사항이 충족되었음을 확인하는 것이며, 검증은 주관적 증거 제공을 통해 특정 의도된 용도에 대한 설계가 충족되었음을 확인하는 것입니다.
C. 밸리데이션은 주관적 증거 제공을 통해 명시된 요구사항이 충족되었음을 확인하는 것이며, 검증은 주관적 증거 제공을 통해 특정 용도에 맞는 설계가 충족되었음을 확인하는 것입니다.
D. 검증은 특정 요구사항이 충족되었음을 객관적인 증거 제공을 통해 확인하는 것이고, 검증은 특정 사용 목적에 대한 요구사항이 충족되었음을 객관적인 증거 제공을 통해 확인하는 것입니다.
Answer: D
Explanation:
According to the syllabus, verification is the process of evaluating a product or component to determine whether it satisfies its specified requirements. Validation is the process of evaluating a product or component to determine whether it fulfills its intended use and user expectations. Verification answers the question "Are we building the product right?" while validation answers the question "Are we building the right product?" Both verification and validation involve providing objective evidence, which means factual, unbiased, and observable information that supports the evaluation. The answer D is correct because it best distinguishes the terms "validation" and "verification". The other answers are incorrect because they either confuse the terms
"validation" and "verification" or use subjective evidence, which means personal, biased, or unobservable information that does not support the evaluation.
References: [Certified Tester Foundation Level Syllabus], Section 1.2.1, page 10-11.

QUESTION NO: 26
웹 개발을 위한 클라이언트-서버 시스템은 시간당 최소 200개의 문의를 지원해야 합니다. 피크 시간에는 애플리케이션의 중요한 특성으로 인해 24시간 x 7일 동안 사용할 수 있어야 하며, 피크 로드 중에는 응답 시간이 20초 미만이어야 합니다.
이 진술서에 표현된 비기능적 요구사항을 검증하는 데 가장 적합한 테스트 유형은 다음 중 무엇입니까?
A. 성능, 유용성, 회귀.
B. 시스템, 로드, 장애 조치.
C. 성능, 부하, 스트레스.
D. 로드, 볼륨 및 구성요소.
Answer: C
Explanation:

Performance, Load, Stress. Performance testing is a type of non-functional testing that measures how well the system performs under various conditions, such as response time, throughput, resource utilization, etc. Load testing is a type of performance testing that simulates a high volume of user requests or transactions on the system and measures its behavior under normal and peak load conditions. Stress testing is a type of performance testing that simulates extreme or abnormal conditions on the system and measures its behavior under stress or overload conditions. These test types would help verify the requirements such as minimum number of enquiries per hour, availability, and response time under peak loads. A detailed explanation of performance, load, and stress testing can be found in Software Testing Foundations: A Study Guide for the Certified Tester Exam, pages 187-1913.

QUESTION NO: 27
다음 중 조직 내 도구 배포의 성공에 부정적인 영향을 미칠 수 있는 요소는 무엇입니까?
A. 조직의 모든 테스트 팀에 동시에 도구를 도입합니다.
B. 초기 프로젝트에서 도구의 사용 정보를 수집합니다.
C. 도구의 새로운 사용자를 위한 교육 및 코칭 제공
D. 도구 사용에 프로세스를 적용하고 개선합니다.
Answer: A
Explanation:
Introducing the tool to all the test teams of the organization at the same time is a factor that could negatively influence the success of the deployment of a tool within an organization, as it may cause resistance, confusion, or overload among the test teams. A better approach would be to introduce the tool gradually and incrementally, starting with a pilot project or a small group of users, and then expanding to other test teams based on the feedback and results. suggests this as follows:
A pilot project should be conducted before introducing a new test tool into an organization in order to learn more about how to use it effectively and efficiently in your context and how it will interact with other tools and processes. A pilot project should involve a small group of users who are willing and able to experiment with the tool and provide feedback. The pilot project should have clear objectives, scope, duration, and success criteria.
B, C, and D are factors that could positively influence the success of the deployment of a tool within an organization. Collecting usage information of the tool from the early projects (B) can help to evaluate the benefits and drawbacks of the tool, identify areas for improvement, and justify the investment. Providing training and coaching for new users of the tool can help to increase their confidence and competence in using the tool, reduce errors and frustration, and enhance their productivity and satisfaction. Adapting and improving the processes to the usage of the tool (D) can help to optimize the workflow and performance of the test teams, align the tool with the organizational goals and standards, and increase the value and quality of the testing activities.

QUESTION NO: 28
다음 테스트 기술 중 구조 기반 테스트 기술은 무엇입니까?
A. 제어 흐름 테스트
B. 사용 사례 테스트

C. 상태 전환 테스트
D. 의사결정 테이블 테스트
Answer: A
Explanation:
Test techniques are methods or procedures that can be used to design, execute, or evaluate test cases. Test techniques can be classified into two categories: specification-based and structure-based. Specification-based test techniques, also known as black-box test techniques, are based on the requirements, specifications, or expectations of the system under test. They do not require any knowledge of the internal structure or implementation of the system. Some examples of specification-based test techniques are use case testing, state transition testing, decision table testing, etc. Structure-based test techniques, also known as white-box test techniques, are based on the code, architecture, or design of the system under test. They require some knowledge of the internal structure or implementation of the system. Some examples of structure-based test techniques are control flow testing, data flow testing, branch testing, statement testing, etc. You can find more information about test techniques in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 41.

QUESTION NO: 29
다음 중 제품 위험 분석이 테스트 접근 방식에 어떤 영향을 미칠 수 있는지 보여주는 예는 무엇입니까?
A. 복잡한 비즈니스 규칙으로 인해 비용이 많이 드는 실패가 발생할 수 있으므로 의사결정 테이블 테스트는 테스트 사례 설계에 사용됩니다.
B. 성능 실패가 예상보다 훨씬 낮았으므로 이 영역에 대해 더 많은 테스트 분석이 수행됩니다.
C. 테스트 팀에 자동화 기술이 부족하여 자동화 도구에 대한 교육이 출시될 예정입니다.
D. 보안과 관련하여 기록된 제품 위험이 없으므로 비상 조치로 보안 테스트가 우선적으로 제공됩니다.
Answer: A
Explanation:
Product risk analysis is an approach to testing that aims to reduce the level of product risks and inform stakeholders of their status, starting in the initial stages of a project1. It involves the identification of product risks and the use of risk levels to guide the test process1. Product risks are uncertain situations that can affect the quality or value of the software product2. One of the factors that can influence the testing approach is the test design technique to be applied3. Test design techniques are methods to derive and select test cases based on the test objectives, test conditions, and test basis2. Decision table testing is a test design technique that is suitable for testing complex business rules that have logical conditions and outcomes2. Therefore, if the product risk analysis identifies that the complex business rules could result in costly failures, then decision table testing will be used for test case design to cover all the possible combinations of conditions and outcomes2.
The other options are not examples of how product risk analysis can influence the testing approach, because they are either not related to product risks or not based on risk levels. They are:
* Performance failures were much lower than expected, so more test analysis will be conducted in this area (B). This is not related to product risk analysis, but to test monitoring and control, which is the process of comparing actual progress against the plan and reporting

the status2. Test analysis is the activity of analyzing the test basis and defining test objectives2. If the performance failures were much lower than expected, then more test execution or evaluation might be needed, not more test analysis.
* There is a lack of automation skills in the test team, so training for the automation tool will be rolled out. This is not related to product risk analysis, but to test management, which is the process of planning, monitoring, and controlling the test activities and resources2. Test automation is the use of software to perform or support test activities2. The lack of automation skills in the test team is a project risk, not a product risk, and it should be addressed by providing adequate training, tools, and resources for the test team2.
* There are no product risks recorded around security, so security testing will be given priority as a contingency measure (D). This is not based on risk levels, but on the absence of risk identification, which is a poor practice in product risk analysis. Security testing is a type of testing that determines whether the software protects data and maintains functionality as intended2. Product risk analysis should identify the potential security risks and assign them appropriate risk levels based on their probability and impact1. Security testing should be prioritized based on the risk levels, not on the lack of risk identification, which could lead to overlooking or underestimating the security risks1.
References:
* ISTQB Foundation Level 2018 syllabus
* Risk in Software Testing
* Risk-Based Testing
* Test Approach

QUESTION NO: 30
테스트 스크립트 TransVal 3.1은 TRN 003B 화면을 통해 거래 유효성 검사를 테스트합니다. 사양(PID 버전 1.3 10b iv)에 따르면 유효성 검사 화면에서는 미래 날짜의 거래를 허용해서는 안 됩니다. 테스트 스크립트 TransVal 3.1이 통과했습니다. 테스트 스크립트 eod 1.4는 일과 종료 처리를 테스트하고 해당 테스트 중에 입력된 데이터를 사용하여 TransVal 3.1 실행 후 실행됩니다. 다음 중 사고 보고서의 가장 좋은 세부 사항은 무엇입니까? [K3]
A. 제목. 일과 종료 실패.. 재현 가능. 예. 설명. 해당 날짜의 첫 번째 거래가 미래 날짜의 거래인 경우 스크립트 eod 1.4가 실패합니다. 첨부된 오류 스크린샷.
B. 제목. 거래 입력 화면 검증..재현 가능. 예. 설명. 스크립트 eod 1..4가 실패합니다. 첨부된 오류 스크린샷. TRN-003B 화면의 거래 입력 유효성 검사는 미래 날짜의 거래를 허용해서는 안 됩니다. PID 버전 1.3 10b항 참조 iv.
C. 제목. 화면 TRN-003B 거래일자 검증.. 재현 가능. 번호 설명. 미래 날짜의 거래가 하루 종료 프로세스로 처리되면 실패가 발생할 수 있습니다. 이런 일이 항상 발생하는 것은 아닙니다. 첨부된 오류 스크린샷.
D. 제목. 거래 날짜의 TRN-003B 검증 화면. 재생할 수 있는. 예. 설명. 해당 날짜의 첫 번째 거래가 미래 날짜의 거래인 경우 스크립트 eod 1.4가 실패합니다. 첨부된 오류 스크린샷. 화면 TRN-003B의 거래 항목 확인은 미래 날짜의 거래를 허용해서는 안 됩니다.
- PID 버전 1.3 10b항 iv를 참조하세요.
Answer: D
Explanation:
* An incident report is a document that records any event occurring during testing that requires investigation1. An incident report should contain sufficient information to enable

reproduction of the incident and resolution of the defect1. According to IEEE 829 Standard for Software Test Documentation, an incident report should contain the following information:
* Identifier: A unique identifier for the incident report
* Summary: A brief summary of the incident
* Incident description: A description of the incident, including:
* Date: The date when the incident was observed
* Author: The name of the person who reported the incident
* Source: The software or system lifecycle process in which the incident was observed
* Version: The identification of configuration items of the software or system
* Test case: The identification of the test case that caused the incident
* Execution phase: The phase of test execution when the incident was observed
* Environment: The hardware and software environment in which the incident was observed
* Description: A description of the anomaly to enable reproduction of the incident
* Expected result: The expected result of the test case
* Actual result: The actual result of the test case
* Reproducibility: An indication of whether the incident can be reproduced or not
* Impact analysis: An analysis of the impact of the incident on other aspects of the software or system
* Incident resolution: A description of how the incident was resolved, including:
* Resolution date: The date when the incident was resolved
* Resolver: The name of the person who resolved the incident
* Resolution summary: A brief summary of how the incident was resolved
* Status: The current status of the incident (e.g., open, closed, deferred)
* Classification information: A classification of the cause and effect of the incident for metrics and reporting purposes Therefore, among the options given in this question, only D provides the best detail on an incident report. It contains a clear title, a reproducibility indicator, a description that includes both expected and actual results, a reference to the specification document, and a screen shot of the failure. The other options are either missing some important information or providing inaccurate or irrelevant information

QUESTION NO: 31
회사는 회사의 모든 금융 거래를 처리하는 새로운 시스템을 구입했습니다. 어떤 테스트 유형에 재무 부서의 전문가가 참여해야 합니까?
A. 구성 요소 테스트
B. 승인 테스트
C. 유지보수 테스트
D. 시스템 테스트
Answer: B
Explanation:
Acceptance tests are the test types that call for involvement of an expert from the financial department for a new system that deals with all financial transactions in a company. Acceptance tests are tests conducted to determine if the requirements of a specification or contract are met by a system or software component prior to its delivery or deployment. Acceptance tests are usually performed by end users or customers who have domain knowledge and expertise in evaluating if the system meets their needs and expectations1

defines acceptance tests as follows:
Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.
Acceptance Testing is also known as User Acceptance Testing (UAT), End-User Testing, Operational Acceptance Testing (OAT) or Field (Acceptance) Testing.
Acceptance Testing is performed by end users or customers who have domain knowledge and expertise in evaluating if the system meets their needs and expectations.
A, C, and D are incorrect answers. Component testing, maintenance testing, and system testing are not test types that call for involvement of an expert from the financial department for a new system that deals with all financial transactions in a company. Component testing is testing of individual software components in isolation from other components, usually done by developers. Maintenance testing is testing of a modified system or component after changes have been made to it, usually done by testers. System testing is testing of an integrated system as a whole to verify that it meets specified requirements, usually done by testers.

QUESTION NO: 32
테스트 디자인이란 무엇입니까?
A. 시스템이 지정된 요구 사항을 충족하는지 확인하기 위해 시스템을 테스트하는 전체 프로세스입니다.
B. 테스트된 시스템이 구현한 기능을 실행하기 위한 테스트 기법을 선택하는 과정
C. 일반적인 테스트 목표를 실질적인 테스트 조건 및 테스트 케이스로 변환하는 프로세스
D. 제품의 위험도가 높은 부분에 초점을 맞춰 제품 위험 수준을 낮추기 위한 테스트 접근 방식
Answer: C
Explanation:
Test design is the process of transforming general testing objectives into tangible test conditions and test cases that can be executed and verified against expected results. Test design involves identifying test techniques, coverage criteria, input data, expected outcomes and other test parameters.
Reference: A Study Guide to the ISTQB Foundation Level 2018 Syllabus1, Chapter 3, Section 3.2.1, page
53.

QUESTION NO: 33
농약 역설'이라는 용어는 무엇을 의미합니까?
A. 버그가 많은 코드 조각이 숨겨져 있을 가능성이 높지만 발견되지 않는 현상
B. 버그가 많은 코드에서 디버깅을 수행하면 효율성이 감소합니다.
C. 동일한 시나리오에 집중하고 반복되는 테스트 케이스의 효율성 감소
D. 블랙 박스와 화이트 박스 기법 모두에서 동일한 객체를 테스트하는 중복성
Answer: C
Explanation:
The term 'Pesticide paradox' refers to the phenomenon where the effectiveness of test cases that are repeated and focused on the same scenarios decreases over time because they tend to find the same defects or no defects at all. This is because the system under test

becomes more resistant or immune to the existing test cases, just like pests become more resistant or immune to pesticides over time. To overcome the pesticide paradox, test cases should be regularly reviewed and updated to cover new or changed requirements, scenarios, risks, or defects. Test cases should also be designed to cover different aspects and perspectives of the system under test, such as functionality, usability, performance, security, etc. You can find more information about the pesticide paradox in A Study Guide to the ISTQB Foundation Level 2018 Syllabus, Chapter 4, Section 4.11.

QUESTION NO: 34
다음 중 소프트웨어 제품을 테스트하는 데 필요한 테스트 노력에 가장 큰 영향을 미치는 요소는 무엇입니까? [K1]
A. 테스트를 실행할 수 있는 직원 수
B. 테스트 계획의 세부 수준
C. 제품의 신뢰성 및 보안에 대한 요구사항
D. 사용된 검정 추정 방법
Answer: C
Explanation:
The testing effort required to test a software product depends on various factors, such as the size and complexity of the product, the quality of the requirements and design documents, the testability of the product, the test strategy and scope, the test environment and tools, the skills and experience of the testers, and the quality expectations and standards of the stakeholders1. Among these factors, the requirements for reliability and security in the product will most affect the testing effort required to test a software product1. Reliability and security are quality attributes that measure how well a software product performs its intended functions under specified conditions and protects itself from unauthorized access or harm1. Testing for reliability and security requires more rigorous and thorough testing techniques, such as reliability testing, security testing, penetration testing, stress testing, etc1. These techniques may require more time, resources, tools, and skills to perform effectively1. Therefore, the requirements for reliability and security in the product will most affect the testing effort required to test a software product.

QUESTION NO: 35
테스터는 UAT 사양 검토에 언제 참여해야 합니까? [K1]
A. 프로젝트 시작 시
B. 요구 사항이 승인되는 즉시
C. UAT 사양 초안이 작성되자마자
D. UAT가 시작되기 전 언제든지
Answer: C
Explanation:
Testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted1. UAT stands for user acceptance testing, which is a type of testing that verifies that the software product meets the acceptance criteria and expectations of the end users or customers1. A UAT specification is a document that defines the scope, objectives, approach, and criteria for UAT1. Testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted, as this can help to ensure that the UAT

specification is clear, complete, consistent, testable, and aligned with the user requirements1. Testers can also provide feedback and suggestions to improve the UAT specification and avoid potential issues or conflicts during UAT execution1. Therefore, testers should be involved in reviewing a UAT specification as soon as the UAT specification has been drafted.