객관식 정답

1. 2
2. 4
3. 1
4. 2
5. 2
6. 2
7. 4
8. 3
9. 2
10. 3
11. 2
12. 3
13. 3
14. 3
15. 1
16. 4
17. 1
18. 3
19. 4
20. 4
21. 3
22. 3
23. 3
24. 3

..~~~~..

\* 주관식 1번 정답

test case 1

input : 2000/01/01

output : 25살

옳은 입력.

test case 2

input : 3024/01/01

output : 0세

아직 오지 않은 날짜를 입력.

test case 3

input : abcd/ef/gh

output : error

입력 형식의 오류. 숫자가 아닌 영문을 입력.

\* 주관식 2번 정답

블랙 박스 테스트는 인풋과 아웃풋만이 있는 테스트이다. 즉 프로그램의 내부 구조는 신경쓰지 않는다. 그러나 화이트 박스 테스트는 코드 실행이 기준이다. 따라서 화이트 박스 테스트에서는 내부 절차가 중요하다.

이러한 특징으로 인해 블랙 박스 테스트에서는 발견되지 않지만 화이트 박스 테스트에서만 발견되는 오류는 오버플로우 문제가 있다. 결과물은 옳게 나오지만 시스템 내부에서는 보안 관련 취약점 문제가 있을 수도 있다.

또한 무한 루프 문제도 있다. 조건이 명확하지 않으면 결과물이 나와 프로그램은 계속 실행될 수도 있다.

즉, 단발성인 테스트의 결과는 잘 나올 수 있지만 프로그램 내부에 문제가 남아 있는 경우가 블랙 박스 테스트에서는 문제점을 찾을 수 있지만 화이트 박스 테스트에서는 문제점을 찾을 수 있는 유형이다. 논리 오류, 제어 흐름 오류, 변수 또는 데이터 유형 오류가 그 예시가 될 수 있다.

\* 주관식 3(A)번 정답

EI는 소프트웨어의 오류 상태를 나타내는 지표이다. 특히 위험 수준을 구분한 후 각 위험 수준에 해당하는 오류가 발생하는 빈도를 체크한다. 각 수준과 빈도를 곱한 후, 나온 결과들을 모두 더한 후에, 총 프로그램 크기로 나누는 것이 EI이다.

공식 EI = Σ(i \* PI(i)) / PS에서 i는 오류가 심각한 수준을 의미한다. PI(i)는 각 오류 수준 i가 발생한 빈도 수를 의미한다. PS는 전체 소프트웨어 크기로 일반적으로 코드 라인 수를 의미한다.

\* 주관식 3(B) 정답

PI는 위상 지수로 특정 단계 i에서 발견된 총 결함 수를 의미한다. PI를 구하면서 프로그램의 각 단계에서 발견된 결함을 추적하고 계산할 수 있다.

공식 PI = D(i)/ ΣD(j)에서 PI(i)는 단계 i에서의 위상 지수를 의미한다. D(i)는 단계 i에서 발견된 결함 수를 의미한다. ΣD(j)는 j=1부터 현재 단계인 n까지 누적된 결함 수를 의미한다.

PI를 통해서 어느 단계해서 결함이 많이 발생되었는지 파악할 수 있다.

\* 주관식 4(A) 정답

FP는 기능 점수 측정값으로 기능 점수 분석을 사용자 관점에서 소프트웨어 개발을 측정하기 위한 표준화된 방법이다.

공식 FP = Count\_Total [0.65+0.01 sum(Fi)], i=1 ~ 14에서 Count Total은 식별된 모든 사용자 기능의 총 개수로 입력, 출력, 조회 등이 있다. 복잡도에 따라 가중치가 있다. Fi는 일반 시스템 특성으로 소프트웨어의 복잡성에 영향을 미치는 요소이다. 각 Fi는 영향력에 따라 0~5 사이의 값이 있다. sum(Fi)는 이러한 14가지 시스템 특성의 점수를 더한 것이다. [0.65+0.01 sum(Fi)]는 값 조정 계수로 시스템의 복잡성에 따라 조정하지 못한 기능 포인트를 조정하는 데 사용한다. 0.65는 기본 값 조정 상수이고 0.01은 가중치 상수를 의미한다.

\* 주관식 4(B) 정답

장점 : FP 평가는 일단 표준화된 방법이라는 장점이 있다. 또한 사용자 관점에서 기능을 측정하므로 비전문가 관계자가 더 쉽게 이해를 할 수 있다는 장점이 있다. 따라서 사용자의 요구를 보다 쉽게 반영할 수 있다. 그리고 FP 평가는 프로그래밍 언어, 방법론, 기술과 독립적이므로 다양한 프로젝트에서 일관된 방식으로 소프트웨어 평가가 가능하다.

단점 : FP 평가는 분류 가중치가 주관적이라는 단점이 있다. 따라서 평가자 양성이 중요하며 이에 따른 비용과 시간이 요구된다. 그리고 FP 평가는 다른 방법에 비해 복잡하다는 단점도 존재한다.

\* 주관식 5 정답

Use Case Model과 Object Model, Dynamic Model 각 모델들이 서로 일관되게 유지되어야 시스템의 정확성과 신뢰성을 확보할 수 있다.

Use Case Model에서 언급된 객체는 Object Model에서 정의되어 있어야 한다. 또한 Use case Model에서 정의된 함수는 Object Model의 함수와 일치해야 한다.

Use Case Model과 Dynamic Model의 경우 Use Case Model에서의 시나리오와 Dynamic Model의 시퀀스 다이어그램이 일치해야 한다. 즉, Use Case Model의 사용자 동작 흐름이 Dynamic Model에서 정확하게 반영되어야 한다. 또한 Use Case Model에서 정의된 주요 사건은 Dynamic Model에서 상태 전이로 표현되어야 한다.

Object Model과 Dynamic Model 사이에서 Dynamic Model의 시퀀스 다이어그램에서의 객체 간의 메시지 교환이 Object Model에서 정의된 객체, 메서드와 일치해야 한다. 또한 Dynamic Model의 상태 변화가 Object Model의 속성과 일치해야 한다.

이러한 일관성을 만족시키면 시스템이 정확하게 정의되고 시스템 동작 이해에 도움이 된다.