



[취업폭격기 Zeromini 위클리 개념 폭격 #19]

📖 과목 : 소프트웨어공학

🔥 참고문제 : 2023년 국가직 7급

😊 문제 수정 버전 : V 1.0



1. 애자일 방법론

- 문제: 애자일 방법론의 핵심 원칙 중 '작업 결과물'에 대한 중요성과 그것이 프로젝트 진행에 어떻게 영향을 미치는지 설명해주세요.
- 해설: 애자일 방법론은 변화에 유연하게 대응하는 것을 중요시하며, '작업 결과물'은 프로젝트의 진행 상황을 명확하게 파악하고 고객에게 지속적으로 가치를 제공하는 수단이다. 이를 통해 프로젝트의 방향성을 지속적으로 조정하며 효율적으로 진행할 수 있다.

2. 객체지향 설계 원칙

- 문제: 객체지향 설계 원칙 중 '개방-폐쇄 원칙'에 대해 설명하고, 이 원칙이 소프트웨어 개발에 어떤 이점을 가져다주는지 서술해주세요.
- 해설: '개방-폐쇄 원칙'은 소프트웨어의 구성 요소는 확장에는 열려 있어야 하고, 변경에는 닫혀 있어야 한다는 원칙이다. 이 원칙을 따르면 기존 코드를 변경하지 않고도 새로운 기능을 추가할 수 있어 유지보수와 확장성이 향상된다.

3.요구사항 수집 기법

- 문제: JAD(Joint Application Development) 방식의 요구사항 수집 기법의 특징과 그 방식이 프로젝트에 어떤 장점을 가져다주는지 설명해주세요.
- 해설: JAD는 프로젝트 관리자, 사용자, 개발자 등이 함께 모여 상호 토론을 통해 요구사항을 수집하는 기법이다. 이 방식은 다양한 관점에서의 피드백을 즉시 얻을 수 있어 요구사항의 정확도가 향상되며, 프로젝트의 방향성을 빠르게 조정할 수 있다.

4.모듈의 결합도와 응집도

- 문제: 소프트웨어 설계에서 '결합도'와 '응집도'의 중요성에 대해 설명하고, 이 두 개념이 소프트웨어 품질에 어떻게 영향을 미치는지 서술해주세요.
- 해설: '결합도'는 모듈 간의 연관 정도를, '응집도'는 모듈 내부의 기능적 일관성을 나타낸다. 낮은 결합도와 높은 응집도를 가진 소프트웨어는 유지보수와 확장성이 용이하며, 오류 발생 시 해당 오류의 범위가 제한적이다.

5.컴포넌트 기반 개발

- 문제: 컴포넌트 기반 개발 방법론의 핵심 원칙과 이 방법론이 소프트웨어 개발에 어떤 장점을 가져다주는지 설명해주세요.
- 해설: 컴포넌트 기반 개발은 재사용 가능한 컴포넌트를 중심으로 소프트웨어를 개발하는 방법론이다. 이를 통해 개발 시간을 단축시키고, 검증된 컴포넌트의 재사용으로 소프트웨어의 안정성을 향상시킬 수 있다.

6.V 모델

- 문제: V 모델의 특징과 그것이 전통적인 폭포수 모델과 어떻게 다른지, 그리고 이러한 차이가 프로젝트 진행에 어떤 장점을 가져다주는지 설명해주세요.
- 해설: V 모델은 폭포수 모델의 확장으로, 각 개발 단계와 테스트 단계가 V자 형태로 대응된다. 이는 개발과 테스트가 병렬적으로 진행되어 오류를 빠르게 발견하고 수정할 수 있게 한다. 이로 인해 프로젝트의 품질이 향상되며, 리스크를 줄일 수 있다.

7.CMMI의 단계

- 문제: CMMI의 "정량적 관리" 단계의 목적과 이 단계에서 수행되는 주요 활동, 그리고 이를 통해 얻을 수 있는 프로젝트 관리의 이점에 대해 설명해주세요.

- 해설: "정량적 관리" 단계는 프로세스의 성능을 정량적으로 관리하고 제어하는 것을 목적으로 한다. 이 단계에서는 프로세스와 제품의 품질에 대한 정량적 측정 및 분석이 이루어진다. 이를 통해 프로세스의 성능을 지속적으로 모니터링하고 개선할 수 있어 프로젝트의 예측 가능성과 품질이 향상된다.

8.아키텍처 패턴

- 문제: 브로커 아키텍처 패턴의 핵심 개념과 이 패턴이 적합한 시스템 유형, 그리고 이 패턴을 사용함으로써 얻을 수 있는 주요 이점에 대해 설명해주세요.
- 해설: 브로커 아키텍처는 분산 컴포넌트들이 원격 서비스를 호출하여 상호 작용하는 시스템을 위한 패턴이다. 이 패턴은 분산 환경에서의 시스템 개발에 적합하다. 브로커를 통해 컴포넌트 간의 통신 복잡성이 감소하며, 확장성과 유지보수성이 향상된다.

9.프로젝트 관리 기법

- 문제: 스크럼 방법론에서 사용되는 '데일리 스크럼'의 목적과 이를 통해 프로젝트 팀이 얻을 수 있는 주요 이점에 대해 설명해주세요.
- 해설: '데일리 스크럼'은 프로젝트 팀이 매일 진행 상황을 공유하고 문제점을 식별하는 짧은 미팅이다. 이를 통해 팀원 간의 커뮤니케이션 효율이 향상되며, 발생한 문제나 장애를 신속하게 인식하고 대응할 수 있다.

10.테스트 전략

- 문제: 화이트박스 테스트와 블랙박스 테스트의 주요 차이점과 각각의 테스트 방법이 적합한 시나리오에 대해 설명해주세요.
- 해설: 화이트박스 테스트는 소프트웨어의 내부 구조와 동작을 기반으로 테스트하는 반면, 블랙박스 테스트는 소프트웨어의 외부 기능에 초점을 맞춘다. 화이트박스는 코드의 논리적 오류를 찾는 데 적합하고, 블랙박스는 사용자 요구사항에 따른 기능적 오류를 찾는 데 적합하다.

11.소프트웨어 생명주기

- 문제: 소프트웨어 생명주기의 각 단계와 그 단계에서 수행되는 주요 활동에 대해 설명하십시오.
- 해설: 소프트웨어 생명주기는 요구사항 분석, 설계, 구현, 테스트, 유지보수의 단계로 구성된다. 요구사항 분석에서는 사용자의 필요를 파악하고, 설계에서는 시스템의 구조를 정의하며, 구현에서는 코드를 작성하고, 테스트에서는 오류를 찾아 수정하며, 유지보수에서는 시스템을 지속적으로 개선한다.

12.소프트웨어 품질

- 문제: 소프트웨어 품질을 평가하는 주요 지표 중 하나를 선택하고, 그 지표가 소프트웨어의 어떤 측면을 반영하는지 설명하십시오.

- 해설: 가용성은 소프트웨어 품질의 주요 지표 중 하나로, 시스템이 정상적으로 작동하는 시간의 비율을 나타낸다. 높은 가용성은 시스템의 안정성과 신뢰성을 반영한다.

13. 프로젝트 관리 도구

- 문제: Gantt 차트의 특징과 프로젝트 관리에서 이 도구를 사용함으로써 얻을 수 있는 주요 이점에 대해 설명하십시오.
- 해설: Gantt 차트는 프로젝트의 일정을 시각적으로 표현하는 도구로, 작업의 시작과 종료 시간, 중요한 마일스톤을 한눈에 파악할 수 있다. 이를 통해 프로젝트의 진행 상황을 명확하게 이해하고, 리소스 할당 및 일정 조정이 용이하다.

14. 소프트웨어 아키텍처 스타일

- 문제: 클라이언트-서버 아키텍처 스타일의 핵심 개념과 이 스타일이 적합한 애플리케이션 유형에 대해 설명하십시오.
- 해설: 클라이언트-서버 아키텍처는 중앙 서버와 여러 클라이언트로 구성되며, 클라이언트는 서버에 서비스를 요청하고 서버는 응답한다. 이 스타일은 중앙화된 데이터 관리가 필요한 시스템이나 네트워크 환경에서의 애플리케이션에 적합하다.

15. 소프트웨어 테스트 기법

- 문제: 동등 분할 테스트 기법의 핵심 원칙과 이 기법을 사용함으로써 얻을 수 있는 테스트의 효율성에 대해 설명하십시오.
- 해설: 동등 분할은 입력 데이터를 유사한 동작을 하는 그룹으로 나누는 테스트 기법이다. 이 기법을 사용하면 각 그룹에서 대표 값을 선택하여 테스트할 수 있어 테스트 케이스의 수를 줄이면서도 효과적인 테스트를 수행할 수 있다.

16. 소프트웨어 유지보수

- 문제: 소프트웨어 유지보수의 네 가지 유형 중 '적응 유지보수'에 대해 설명하고, 이 유형의 유지보수가 필요한 상황 예를 들어 설명하십시오.
- 해설: '적응 유지보수'는 환경의 변화에 대응하기 위한 유지보수 유형이다. 예를 들어, 운영 체제의 업그레이드나 하드웨어의 변경으로 인해 소프트웨어를 수정해야 할 때 이 유형의 유지보수가 필요하다.

17. 소프트웨어 위기

- 문제: 1960년대에 발생한 '소프트웨어 위기'의 주요 원인과 이를 해결하기 위해 제안된 소프트웨어 공학의 핵심 원칙에 대해 설명하십시오.
- 해설: '소프트웨어 위기'는 소프트웨어의 복잡성 증가와 품질 저하, 비용 초과 등의 문제로 인해 발생했다. 이를 해결하기 위해 소프트웨어 공학에서는 체계적이고 표준화된 개발 프로세스와 품질 관리, 문서화 등의 원칙을 제안하였다.

18. 소프트웨어 요구사항

- 문제: 기능적 요구사항과 비기능적 요구사항의 차이점을 설명하고, 각각의 예를 들어 구체적으로 설명하시오.
- 해설: 기능적 요구사항은 소프트웨어가 수행해야 할 기본 기능을 설명하는 반면, 비기능적 요구사항은 시스템의 성능, 보안, 사용성 등의 특성을 설명한다. 예를 들면, '사용자는 로그인을 해야한다'는 기능적 요구사항이며, '시스템은 1초 내에 응답해야한다'는 비기능적 요구사항이다.

19. 소프트웨어 설계 원칙

- 문제: '단일 책임 원칙'에 대해 설명하고, 이 원칙을 준수함으로써 얻을 수 있는 주요 이점에 대해 서술하시오.
- 해설: '단일 책임 원칙'은 한 클래스는 하나의 책임만 가져야 한다는 원칙이다. 이 원칙을 준수하면 코드의 가독성이 향상되고, 유지보수가 용이해지며, 오류의 범위가 제한적으로 발생한다.

20. 소프트웨어 개발 방법론

- 문제: 폭포수 모델의 단계별 특징과 이 모델의 주요 장단점에 대해 설명하시오.
- 해설: 폭포수 모델은 순차적으로 요구사항 분석, 설계, 구현, 테스트, 유지보수의 단계를 거친다. 이 모델의 장점은 단계별 명확한 출력물이 있어 관리가 용이하다는 것이고, 단점은 중간에 변경이 어렵다는 점이다.

21. 소프트웨어 테스트

- 문제: 단위 테스트의 목적과 단위 테스트를 수행함으로써 얻을 수 있는 주요 이점에 대해 설명하시오.
- 해설: 단위 테스트는 개별 모듈이나 함수의 정확성을 검증하는 테스트이다. 이를 통해 초기 단계에서 오류를 발견하고 수정할 수 있어 전체 테스트 시간의 단축과 품질 향상에 기여한다.

22. 소프트웨어 유지보수

- 문제: 소프트웨어의 '예방 유지보수'에 대해 설명하고, 이 유형의 유지보수가 시스템에 어떤 장점을 가져다주는지 서술하시오.
- 해설: '예방 유지보수'는 미래의 문제를 예방하기 위해 미리 수행하는 유지보수이다. 이를 통해 장기적인 관점에서 시스템의 안정성을 유지하고, 큰 문제나 장애의 발생을 미리 방지할 수 있다.

23. 프로젝트 관리

- 문제: 위험 관리의 중요성과 프로젝트에서 위험 관리를 제대로 수행함으로써 얻을 수 있는 주요 이점에 대해 설명하십시오.
- 해설: 위험 관리는 프로젝트의 잠재적 문제를 미리 인식하고 대응하는 과정이다. 위험 관리를 통해 프로젝트의 예측 가능성이 향상되며, 큰 문제나 지연을 미리 방지하여 프로젝트의 성공 확률을 높일 수 있다.

24.소프트웨어 아키텍처

- 문제: MVC 아키텍처 패턴의 각 구성 요소와 그 사이의 상호 작용에 대해 설명하십시오.
- 해설: MVC는 Model, View, Controller로 구성된다. Model은 데이터와 비즈니스 로직을, View는 사용자 인터페이스를, Controller는 사용자의 입력을 처리한다. 사용자의 요청은 Controller를 통해 Model로 전달되고, Model의 변경 사항은 View를 통해 사용자에게 표시된다.

25.소프트웨어 품질 보증

- 문제: 소프트웨어 품질 보증(QA)의 주요 목적과 QA 활동을 통해 프로젝트에 가져다주는 주요 이점에 대해 설명하십시오.
- 해설: QA의 주요 목적은 소프트웨어의 품질을 보장하는 것이다. QA 활동을 통해 오류를 미리 발견하고 수정할 수 있으며, 사용자의 요구사항과 일치하는 품질 높은 소프트웨어를 제공할 수 있다.

26.소프트웨어 라이선스

- 문제: 오픈 소스 라이선스와 프로프리터리 라이선스의 주요 차이점과 각각의 장단점에 대해 설명하십시오.
- 해설: 오픈 소스 라이선스는 소스 코드가 공개되어 있어 자유롭게 수정 및 재배포가 가능하다. 이로 인해 커뮤니티의 지원을 받을 수 있지만, 상업적 이용에 제약이 있을 수 있다. 프로프리터리 라이선스는 소스 코드가 비공개이며, 라이선스 구매가 필요하다. 상업적 이용이 용이하지만, 수정이 제한적이다.

27.소프트웨어 개발 프로세스

- 문제: 반복적 개발 프로세스의 핵심 원칙과 이러한 접근 방식이 전통적인 개발 방법론에 비해 어떤 장점을 가지는지 설명하십시오.
- 해설: 반복적 개발은 프로젝트를 여러 단계로 나누어 각 단계마다 개발과 테스트를 반복하는 방식이다. 이 접근 방식은 변화에 유연하게 대응할 수 있으며, 초기 단계에서의 피드백을 통해 품질을 지속적으로 향상시킬 수 있다.

28.소프트웨어 테스트 자동화

- 문제: 테스트 자동화의 주요 목적과 자동화 테스트 도구를 사용함으로써 얻을 수 있는 주요 이점에 대해 설명하시오.
- 해설: 테스트 자동화의 주요 목적은 반복적인 테스트 작업을 자동화하여 테스트의 효율성과 정확성을 향상시키는 것이다. 자동화 도구를 사용하면 테스트 시간을 단축시키고, 일관된 테스트 환경을 제공하여 오류 발견률을 높일 수 있다.

29.소프트웨어 보안

- 문제: SQL 인젝션 공격의 원리와 이러한 공격을 방지하기 위한 주요 방법에 대해 설명하시오.
- 해설: SQL 인젝션은 악의적인 SQL 쿼리를 웹 애플리케이션의 입력값으로 주입하여 데이터베이스를 공격하는 기법이다. 이를 방지하기 위해서는 사용자 입력값을 검증하고, 파라미터화된 쿼리를 사용하는 것이 중요하다.

30.소프트웨어 프로젝트 관리

- 문제: 프로젝트의 범위 관리의 중요성과 범위 변동이 프로젝트에 미치는 영향에 대해 설명하시오.
- 해설: 범위 관리는 프로젝트의 목표와 결과물을 명확하게 정의하고 관리하는 것을 의미한다. 범위의 변동 없이 관리하지 않으면 프로젝트의 시간과 비용이 초과될 수 있으며, 원하는 결과물을 얻기 어려울 수 있다.

31.소프트웨어 아키텍처

- 문제: 마이크로서비스 아키텍처의 핵심 개념과 이 아키텍처 스타일의 주요 장점에 대해 설명하시오.
- 해설: 마이크로서비스는 각 기능을 독립적인 서비스로 분리하여 개발하고 운영하는 아키텍처 스타일이다. 이 스타일은 서비스 간의 독립성이 높아져 확장성과 유지보수성이 향상되며, 빠른 배포와 혁신이 가능하다.

32.소프트웨어 개발 도구

- 문제: 지속적 통합(CI)의 핵심 원칙과 CI 도구를 사용함으로써 얻을 수 있는 주요 이점에 대해 설명하시오.
- 해설: 지속적 통합은 코드 변경을 지속적으로 통합하고 자동화된 테스트를 수행하는 것을 의미한다. CI 도구를 사용하면 오류를 빠르게 발견하고 수정할 수 있으며, 코드 품질의 지속적인 향상을 도모할 수 있다.

33.소프트웨어 유저 인터페이스

- 문제: 사용자 경험(UX)의 중요성과 좋은 UX 디자인이 소프트웨어 제품에 미치는 영향에 대해 설명하시오.

- 해설: 사용자 경험은 사용자가 소프트웨어 제품과 상호 작용할 때의 느낌과 반응을 의미한다. 좋은 UX 디자인은 사용자의 만족도를 높이고, 제품의 사용성을 향상시켜 사용자의 재방문률과 추천률을 높일 수 있다.

34. 소프트웨어 품질 기준

- 문제: ISO/IEC 9126 표준에서 제시하는 소프트웨어 품질의 주요 특성 중 하나를 선택하고, 그 특성의 중요성에 대해 설명하시오.
- 해설: ISO/IEC 9126 표준에서는 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성 등의 특성을 제시한다. 예를 들어, 사용성은 사용자가 제품을 쉽게 이해하고 사용할 수 있는 정도를 나타내며, 사용자의 만족도와 제품의 성공 가능성에 큰 영향을 미친다.

35. 소프트웨어 메트릭

- 문제: 코드 복잡도를 측정하는 메트릭 중 하나를 선택하고, 그 메트릭이 코드 품질에 어떤 영향을 미치는지 설명하시오.
- 해설: 사이클로매틱 복잡도는 코드 내의 선형적으로 독립된 경로의 수를 측정하는 메트릭이다. 높은 사이클로매틱 복잡도는 코드의 이해와 유지보수가 어려울 수 있으며, 테스트의 어려움을 야기할 수 있다.

36. 소프트웨어 프로젝트 추정

- 문제: 소프트웨어 프로젝트의 비용과 일정을 추정하는 주요 방법 중 하나를 선택하고, 그 방법의 특징과 장단점에 대해 설명하시오.
- 해설: COCOMO(Constructive COst MOdel)는 소프트웨어 크기를 기반으로 비용과 일정을 추정하는 모델이다. 이 방법은 다양한 프로젝트 유형에 적용 가능하나, 초기 정보가 부정확할 경우 추정의 정확도가 떨어질 수 있다.

37. 소프트웨어 검증과 확인

- 문제: 소프트웨어의 검증(Validation)과 확인(Verification)의 차이점과 각각의 목적에 대해 설명하시오.
- 해설: 검증은 개발된 소프트웨어가 사용자의 실제 요구사항을 만족하는지 확인하는 과정이며, '우리가 올바른 제품을 만들었는가?'에 대한 질문을 해결한다. 확인은 소프트웨어가 정확하게 개발되었는지 확인하는 과정이며, '우리가 제품을 올바르게 만들었는가?'에 대한 질문을 해결한다.

38. 소프트웨어 재사용

- 문제: 소프트웨어 컴포넌트의 재사용의 중요성과 컴포넌트 기반 개발에서 재사용을 적용함으로써 얻을 수 있는 주요 이점에 대해 설명하시오.

- 해설: 소프트웨어 컴포넌트의 재사용은 개발 시간과 비용을 절약하며, 검증된 컴포넌트를 사용함으로써 품질을 향상시킬 수 있다. 컴포넌트 기반 개발에서 재사용을 적용하면 시스템의 모듈성과 유연성이 향상되며, 빠르게 시장에 출시할 수 있는 경쟁력을 얻을 수 있다.

39.소프트웨어 문서화

- 문제: 소프트웨어 문서화의 중요성과 효과적인 문서화를 위한 주요 원칙에 대해 설명하십시오.
- 해설: 소프트웨어 문서화는 개발 및 유지보수 과정에서의 이해와 커뮤니케이션을 돕는다. 효과적인 문서화를 위해서는 명확하고 간결한 내용, 일관된 포맷, 그리고 필요한 정보만을 포함하는 것이 중요하다.

40.소프트웨어 팀 구성

- 문제: 크로스-펄서널 팀의 특징과 이러한 팀 구성이 소프트웨어 프로젝트에 미치는 영향에 대해 설명하십시오.
- 해설: 크로스-펄서널 팀은 다양한 전문성을 가진 구성원들로 이루어져 있어, 팀 내에서 다양한 역할과 기능을 수행할 수 있다. 이러한 팀 구성은 문제 해결의 다양한 관점을 제공하며, 빠른 의사결정과 효율적인 프로젝트 진행을 도모한다.