소프트웨어공학

문 1. 다음과 같은 소프트웨어 시스템을 개발할 때, 적용할 수 있는 적합한 개발모형은?

> 전자교환기 소프트웨어 시스템은 지난 수십 년 간 개발되어 사용해 왔으며, 새로운 기능이 추가되기보다는 다양한 하드웨어 플랫폼에 맞도록 최적화시키는 일이 빈번히 일어났다.

① 폭포수(waterfall) 모델
② 프로토타이핑(prototyping) 모델
③ 나선형(spiral) 모델
④ UP(Unified Process) 모델

문 2. 기능점수에 대한 설명으로 옳지 않은 것은?
① 기능점수는 소프트웨어 시스템이 가지는 기능을 정량화한 것이다.
② 기능점수의 산출 시 적용되는 가중치는 시스템의 특성에 따라 달라질 수 있다.
③ 기능점수는 구현언어와 밀접한 관련이 있는 메트릭(metric)이다.
④ 기능점수는 원시코드가 작성되기 전이라도 계산할 수 있다.

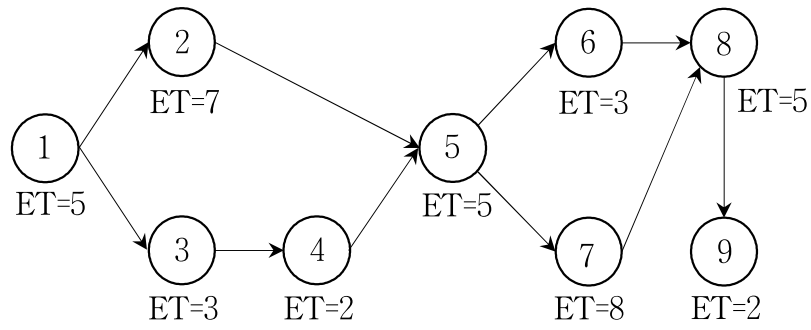
문 3. 소프트웨어 형상 관리(software configuration management)에 대한 설명으로 옳지 않은 것은?
① 소프트웨어 형상 항목에는 시스템 명세서, 소프트웨어 프로젝트 계획서, 소프트웨어 요구사항 명세서 등이 포함된다.
② 소프트웨어 개발 과정에서 소프트웨어에 대한 변경 사항을 관리하기 위해 수행되는 일련의 활동들을 의미하며, 성공적인 형상 관리를 위해서는 유지보수 단계에서 계획하여야 한다.
③ 소프트웨어 형상 관리 활동에 관련된 사람들을 형상 통제 위원회라고 부르며, 구성원에는 프로젝트 관리자, 품질담당자, 기술담당자와 고객 측 담당자 등이 포함된다.
④ 형상 관리 도구로는 Clearcase, CVS 등이 있다.

문 4. 비기능적 요구사항을 고려한 아키텍처의 설계로 가장 적합하지 않은 것은?
① 가용성이 중요한 요구사항일 경우 아키텍처에 여분의 구성 요소들이 포함되도록 설계하여 시스템을 중단 없이 구성 요소를 대치하고 갱신할 수 있게 한다.
② 보안성이 중요한 요구사항일 경우 계층구조의 아키텍처를 사용하여 가장 중요한 자산들을 가장 내부의 계층에서 보호하고 이 계층에 높은 수준의 보안 인증을 적용한다.
③ 안전성이 중요한 요구사항일 경우 아키텍처는 안전에 관련된 오퍼레이션 모두를 하나의 서브시스템이나 소수의 서브시스템 내부로 국한시키도록 설계한다.
④ 성능이 중요한 요구사항일 경우 아키텍처는 중요 오퍼레이션을 가능한 많은 서브시스템으로 분할시킨다.

문 5. A 회사에서 ERP 시스템 개발을 위해 추정된 규모는 33,600 LOC (Line of Code)이다. 이 회사의 과거 평균 생산성이 600 LOC/PM (Person Month)이고, 직원의 평균 인건비가 480만원이라고 할 때, 이 프로젝트의 개발에 필요한 인원 및 개발비용으로 옳은 것은?

	추정인원	총 프로젝트 비용
①	55명	268,800,000원
②	55명	269,000,000원
③	56명	268,800,000원
④	56명	269,000,000원

문 6. 다음 PERT Chart에 대한 설명으로 옳지 않은 것은?
(단, ET : 예상작업시간, 단위 : 일)



① 프로젝트가 종료되는 데에 소요되는 최소 시간은 32일이다.
② 프로젝트 일정 중 임계 경로(critical path)는 1-2-5-7-8-9 이다.
③ 여유시간(slack time)은 4번 노드가 가장 많으며 2일의 여유 시간이 있다.
④ 최대한 빠르게 끝날 수 있는 시간(earliest finish time)과 최대로 늦추어 끝날 수 있는 시간(latest finish time)이 같은 노드는 임계 경로(critical path)에 있다.

문 7. 아키텍처의 유형에 관한 설명 중 옳지 않은 것은?
① CORBA는 파이프 필터 유형의 아키텍처를 지원한다.
② 3-계층(tier) 구조는 클라이언트 서버 유형의 변형이라 할 수 있다.
③ peer to peer 아키텍처 유형에서 각 서브시스템은 서비스를 요청하고 제공하는 능력이 있다.
④ 계층구조 유형에서는 하위 계층이 제공하는 서비스를 상위 계층의 서브시스템이 사용하도록 구성된다.

문 8. 정형적 요구 명세 방법에 대한 설명으로 옳은 것은?
① 요구 분석과 명세화에 드는 시간을 많이 절약할 수 있다.
② 명세의 오류와 모호성을 쉽게 찾을 수 있다.
③ Z나 VDM은 대수학적(algebraic) 접근 방법에 바탕을 두고 만든 정형 명세 언어이다.
④ 수학적 기반의 명세로 되어 있어 누구라도 쉽게 사용할 수 있다.

문 9. MVC(Model-View-Controller) 아키텍처에서 Model의 역할로 옳은 것은?

① 이벤트 형태로 사용자 입력을 처리한다.
② 처리 결과 및 콘텐츠를 사용자에게 보여주는 기능을 수행한다.
③ 어플리케이션과 관련된 데이터 및 데이터 처리에 대한 로직을 가지고 있다.
④ 최신 데이터를 가져와 표시된 정보를 갱신한다.

문 10. 자동화 테스팅 도구의 정적분석 도구에 대한 설명으로 옳은 것은?

① 프로그램의 실행 상태를 순간 포착하여 감시하는 프로그램 모니터링 도구
② 스텁(stub)과 드라이버
③ 프로그램에서 오류 가능성이 있는 부분을 지적하는 코드분석 도구
④ 테스트케이스에 의해 프로그램 각 문장이 실행된 횟수를 측정하는 도구

문 11. 비기능적 요구사항에 대한 명세로 적절하게 표현되지 않은 것은?

① 시스템은 초당 10개 이상의 트랜잭션을 처리할 수 있어야 한다.
② 시스템은 20세 이상의 사업장의 작업자에게 사용편리성을 제공해야 한다.
③ 시스템에 설치되는 모듈 A의 크기는 500 Kbyte 이하이어야 한다.
④ 시스템의 평균 고장시간은 시간당 0.1초 이하를 유지해야 한다.

문 12. 통합 테스팅(integration testing)에 대한 설명으로 옳은 것은?

① 통합 테스팅 동안 발생하는 주요 어려움은 오류들을 지역화(localization)하는 것이다.
② 상향식 통합은 시스템의 계층구조와 상위층의 중요한 인터페이스를 조기에 테스팅할 수 있다.
③ 하향식 통합은 최하위 모듈을 먼저 통합하여 테스팅하는 방식이다.
④ 단위 모듈 테스팅을 철저하게 하면 통합테스팅을 수행할 필요가 없다.

문 13. 유스케이스 다이어그램에 대한 설명으로 옳지 않은 것은?

① 유스케이스 다이어그램을 통하여 시스템의 범위를 파악할 수 있어야 한다.
② 시스템과 상호작용을 하는 외부시스템은 액터로 파악해서는 안 된다.
③ 액터가 인식할 수 없는 시스템 내부의 기능을 하나의 유스케이스로 파악해서는 안 된다.
④ 유스케이스 이름으로부터 해당 유스케이스가 나타내는 시스템의 기능을 명확하게 표현할 수 있어야 한다.

문 14. UML은 시스템의 정적인 부분과 동적인 부분을 표현하기 위하여 여러 다이어그램을 제공한다. 정적인 부분을 표현하는 다이어그램들은 시스템의 구조를 나타내기 위해서 사용되며, 동적인 부분을 표현하는 다이어그램들은 시스템의 행위를 나타내기 위해 사용된다. 다음 중 성격이 다른 다이어그램은?

① 클래스 다이어그램(class diagram)
② 협동 다이어그램(collaboration diagram)
③ 상태 다이어그램(state diagram)
④ 활동 다이어그램(activity diagram)

문 15. 모듈 결합도(coupling)에 대한 설명으로 옳은 것은?

① 모듈 간에 의존도가 클수록 결합도는 낮아진다.
② 모듈간의 제어 요소(function code, flag 등)를 전달하는 것은 전역변수를 사용하는 것보다 바람직하지 않다.
③ 전역변수를 많이 사용할 경우 프로그램 코드가 간단해지므로 유지보수가 편리해진다.
④ 모듈간의 goto문 사용은 내용결합(content coupling)을 유발할 수 있다.

문 16. 리스코프 대체(Liskov's substitution) 원칙으로 옳은 것은?

① 모듈의 설계는 확장(extension)에 대해서는 열려 있어야 하고 수정(modification)에 대해서는 닫혀 있어야 한다.
② B가 클래스 A의 자식(child) 클래스라면 A 클래스의 인스턴스가 요구되는 곳에서 B 클래스의 인스턴스가 사용될 수 있도록 설계해야 한다.
③ 구체적인 클래스보다 인터페이스나 추상클래스에 대해 의존되도록 설계해야 한다.
④ 범용적 목적의 단일한 인터페이스보다는 클라이언트별 인터페이스를 다수 갖도록 설계하는 것이 좋다.

문 17. 애자일(agile) 소프트웨어 개발과 가장 관련이 적은 내용은?

① 적응적 소프트웨어 개발(adaptive software development)
② 익스트림 프로그래밍(extreme programming)
③ 테스트 주도 개발(test-driven development)
④ 철저한 계획 및 문서화

문 18. 좋은 소프트웨어가 가져야 할 특성과 그 설명의 연결이 옳지 않은 것은?

① 확실성(dependability) – 신뢰성, 보안성, 안전성을 포함하는 포괄적인 특성이다.
② 결함 내성(fault tolerance) – 소프트웨어는 고객의 변경 요구를 수용할 수 있는 방법으로 작성되어야 한다.
③ 사용편리성(usability) – 사용자가 소프트웨어를 편리하게 사용할 수 있어야 한다.
④ 효율성(efficiency) – 소프트웨어는 메모리, 프로세서와 같은 자원을 낭비하지 않아야 한다.

문 19. 다음 중 성격이 다른 설계 패턴은?

① 브리지(bridge) 패턴

② 팩토리 메소드(factory method) 패턴

③ 프로토타입(prototype) 패턴

④ 싱글톤(singleton) 패턴

문 20. 소프트웨어 테스팅 기법에 관한 설명으로 옳은 것은?

① 빅뱅 통합 테스팅은 모듈을 한꺼번에 통합하여 테스팅하는 방법이며 오류가 발생하였을 경우 어느 부분에서 오류가 났는지를 쉽게 찾을 수 있다.

② 인스펙션은 동적 테스팅 방법이다.

③ 블랙박스 테스팅은 프로그램의 제어구조를 기반으로 테스트 케이스를 설계하는 방법이다.

④ 화이트박스 테스팅으로 프로그램에 존재하는 모든 경로를 테스팅하여도 오류가 발견되지 않는 경우가 있다.