

[정보처리기사 기출문제은행]



5과목 - 소프트웨어공학

★ 주요 키워드 ★

- (1) 소프트웨어 공학, 소프트웨어 위기
- (2) 소프트웨어 생명 주기 > 폭포수 모델
- (3) 소프트웨어 생명 주기 > 프로토타입 모형
- (4) 소프트웨어 생명 주기 > 나선형 모델
- (5) 2008년 기출문제(중복제거)
- (6) 2009년 기출문제(중복제거)

(1) 소프트웨어 공학, 소프트웨어 위기

[기-08년9월][기-00년3월][기-04년3월]

1. 소프트웨어 공학의 기본 원칙이라고 볼 수 없는 것은?

- 가. 현대적인 프로그래밍 기술 적용
- 나. 지속적인 검증 시행
- 다. 결과에 대한 명확한 기록 유지
- 라. 충분한 인력 투입

[기-04년9월][기-07년9월]

2. 소프트웨어 개발의 생산성에 영향을 미치는 요소가 아닌 것은?

- 가. 프로그래머의 능력
- 나. 팀 의사 전달
- 다. 제품의 복잡도
- 라. 소프트웨어 사용자의 능력

[기-06년9월][기-05년3월]

3. 소프트웨어 공학의 발전을 위한 소프트웨어 사용자 (Software User)로서의 자세로 옳지 않은 것은?

- 가. 프로그래밍 언어와 알고리즘의 최근 동향을 주기적으로 파악한다.
- 나. 컴퓨터의 이용 효율이나 워크스테이션에 관한 정보들을 체계적으로 데이터베이스화 한다.
- 다. 타 기업의 시스템에 몰래 접속하여 새로운 소프트웨어 개발에 관한 정보를 획득한다.
- 라. 바이러스에 대한 예방에 만전을 기하여 시스템의 안전을 확보한다.

[기-08년3월][기-07년5월][기-02년5월][기-99년8월]

4. 소프트웨어의 위기현상과 거리가 먼 것은?

- 가. 유지보수의 어려움
- 나. 개발인력의 급증
- 다. 성능 및 신뢰성의 부족
- 라. 개발기간의 지연 및 개발비용의 증가

[기-06년5월][기-02년9월]

5. 컴퓨터의 발달 과정에서 소프트웨어의 개발 속도가 하드웨어의 개발 속도를 따라가지 못해 사용자들의 요구 사항을 감당할 수 없는 문제가 발생함을 의미하는 것은?

- 가. 소프트웨어 위기(Crisis)
- 나. 소프트웨어 오류(Error)
- 다. 소프트웨어 버그(Bug)
- 라. 소프트웨어 유지보수(Maintenance)

[기-09년3월][기-06년5월][기-05년3월]

6. 소프트웨어 공학에 대한 가장 적절한 설명은?

- 가. 소프트웨어 위기(software crisis)를 완전히 해결한 공학적 원리의 체계이다.

나. 신뢰성 있는 소프트웨어를 만들기 위한 도구만을 연구하는 학문이다.

다. 가장 경제적으로 신뢰도 높은 소프트웨어를 만들기 위한 방법, 도구와 절차들의 체계이다.

라. 점차 많은 비용이 소요되는 소프트웨어 개발에서 가장 경제적인 방법을 찾고자 하는 것이다.

[기-06년3월]

7. 소프트웨어공학의 공학(Engineering)이 가지는 의미와 어울리지 않는 것은?

- 가. 예술성
- 나. 경제성
- 다. 보편타당성
- 라. 적시성

[기-05년5월]

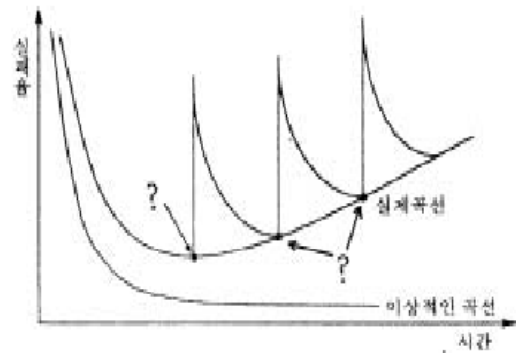
8. 소프트웨어공학이 나타나게 된 배경과 관계가 먼 것은?

- 가. S/W 비용의 증가
- 나. 유지보수 비용의 감소
- 다. S/W 품질과 생산성의 재고
- 라. 특정 개인에 의존한 시스템 개발

[기-05년3월]

9. 다음은 소프트웨어 실패율(오류율)을 나타내는 그림이다.

"?"이 지적하는 것은 무엇을 의미하는가?



- 가. 소프트웨어의 변경
- 나. 소프트웨어의 인도
- 다. 소프트웨어의 실행
- 라. 소프트웨어 수명주기 각 단계

[기-04년5월]

10. 좋은 소프트웨어의 조건이라고 할 수 없는 항목은?

- 가. 남이 알아보기 쉬워야 한다.
- 나. 경제적이어야 한다.
- 다. 문서화가 잘 되어 있어야 한다.
- 라. 프로그램이 독창적이어야 한다.

[기-09년8월][기-99년4월]

11. 공학적으로 잘 작성된 소프트웨어의 특성이 아닌 것은?

- 가. 소프트웨어는 신뢰성이 높아야 하며 효율적이어야 한다.
- 나. 소프트웨어는 사용자가 원하는 대로 동작해야 한다.
- 다. 소프트웨어는 편리성이나 유지보수성에 점차 비중을 적게 두는 경향이 있다.
- 라. 소프트웨어는 잠재적인 에러가 가능한 적어야 하며 유지보수가 용이해야 한다.

[기-00년7월]

12. 소프트웨어 위기(crisis)의 현상과 관계가 먼 것은?

- 가. 개발 기간의 지연 및 하드웨어 비용을 초과하는 개발비용의 증가

- 나. 성능 및 신뢰성 부족
다. 소프트웨어의 개발도구 부족
라. 유지 보수의 어려움에 따른 엄청난 비용

[기-07년9월]

13. 공학적으로 좋은 소프트웨어에 대한 설명으로 적절하지 않은 것은?

- 가. 사용법, 구조의 설명, 성능, 기능이 이해하기 쉬워야 한다.
나. 유지보수가 용이해야 한다.
다. 실행 속도가 빠르고, 소요 기억 용량을 많이 차지할수록 좋다.
라. 사용자 수준에 따른 적당한 사용자 인터페이스를 제공해야 한다.

[기-07년9월]

14. 다음 중 공학적으로 잘 작성된 소프트웨어가 갖는 특성으로 옳은 것은?

- 가. 원하는 요구사항 중에 중요한 사항만 반영한다.
나. 유지보수 비용이 많이 들어간다.
다. 신뢰성이 떨어지더라도 효율성이 높다.
라. 사용자가 손쉽게 사용할 수 있다.

(2) 소프트웨어 생명 주기 > 폭포수 모델

[기-06년9월][기-01년9월]

15. 다음 중 전통적인 소프트웨어 개발 방법론인 폭포수형(waterfall) 모델에서 개발 순서가 옳은 것은?

- 가. 타당성 검토 → 계획 → 분석 → 구현 → 설계
나. 타당성 검토 → 분석 → 계획 → 설계 → 구현
다. 타당성 검토 → 계획 → 분석 → 설계 → 구현
라. 타당성 검토 → 분석 → 계획 → 구현 → 설계

[기-06년3월]

16. 소프트웨어의 전통적 개발 단계 중 요구 분석 단계에 대한 설명으로 옳지 않은 것은?

- 가. 프로젝트를 이해할 수 있는 개발의 실질적인 첫 단계이다.
나. 현재의 상태를 파악하고 문제를 정의한 후, 문제 해결과 목표를 명확히 도출하는 단계이다.
다. 소프트웨어가 가져야할 기능을 기술하는 단계이다.
라. 고 품질의 소프트웨어를 개발하기 위해 소프트웨어의 내부 구조를 기술하는 단계이다.

[기-05년3월]

17. 소프트웨어 공학의 전통적인 개발 방법인 선형 순차 모형의 순서를 옳게 설명한 것은?

- 가. 구현 - 분석 - 설계 - 테스트 - 유지보수
나. 유지보수 - 테스트 - 분석 - 설계 - 구현
다. 분석 - 설계 - 구현 - 테스트 - 유지보수
라. 테스트 - 설계 - 유지보수 - 구현 - 분석

[기-03년8월][기-99년4월]

18. 소프트웨어 수명주기 모형 중 폭포수 모형에 대한 설명으로 옳지 않은 것은?

- 가. 적용사례가 많다.
나. 단계별 정의가 분명하다.
다. 단계별 산출물이 명확하다.

라. 요구사항의 변경이 용이하다.

[기-03년8월]

19. 폭포수 모형(waterfall model)의 진행 단계로 옳은 것은?

- | | | |
|---------|---------|-------|
| 1. 요구분석 | 2. 유지보수 | 3. 검사 |
| 4. 구현 | 5. 설계 | 6. 운용 |

가. 1-5-3-4-6-2

나. 5-1-4-3-2-6

다. 5-1-3-4-6-2

라. 1-5-4-3-6-2

[기-03년5월]

20. 폭포수 모형(Waterfall model)에 대한 설명으로 옳지 않은 것은?

- 가. 산출물이 명확하여 개발 공정의 기준점을 잘 제시한다.
나. 모델의 적용 경험과 성공사례가 많다.
다. 단계적 정의가 분명하고 전체 공조의 이해가 용이하다.
라. 각 단계의 병렬 수행이 가능하다.

[기-03년3월][기-01년3월]

21. 폭포수 모형(waterfall model)의 진행 단계로 옳은 것은?

- | | | |
|------|--------|------|
| ① 시험 | ② 분석 | ③ 계획 |
| ④ 코딩 | ⑤ 유지보수 | ⑥ 설계 |

가. ①-②-③-④-⑤-⑥

나. ②-⑥-④-⑤-①-③

다. ③-②-⑥-④-①-⑤

라. ④-①-②-⑥-⑤-③

[기-02년9월][기-99년8월]

22. 폭포수 모델(Waterfall model)에 대한 설명으로 옳지 않은 것은?

- 가. 앞 단계가 끝나야만 다음 단계로 넘어갈 수 있다.
나. 요구분석 단계에서 프로토타입을 사용하는 것이 특징이다.
다. 제품의 일부가 될 매뉴얼을 작성해야 한다.
라. 각 단계가 끝난 후 결과물이 명확히 나와야 한다.

[기-02년5월]

23. 소프트웨어 수명주기 모형 중 폭포수 모형의 단계별 개발 과정 순서는?

- 가. 요구사항→분석→설계→코딩→테스트→유지보수
나. 요구사항→설계→분석→코딩→테스트→유지보수
다. 설계→요구사항→분석→테스트→코딩→유지보수
라. 분석→요구사항→설계→테스트→코딩→유지보수

[기-99년10월]

24. 소프트웨어 생명주기의 역할로 거리가 먼 것은?

- 가. 프로젝트의 비용 산정과 개발 계획을 수립할 수 있는 기본 골격이 된다.
나. 단계별 종료 시점을 명확하게 한다.
다. 용어의 표준화를 가능하게 한다.
라. 문서화가 충실한 프로젝트 관리를 가능하게 한다.

[기-00년10월]

25. 폭포수 모델에 대한 설명으로 옳지 않은 것은?

- 가. 소프트웨어 개발 과정의 각 단계가 순차적으로 진행된다.
나. 앞 단계에서 발견하지 못한 오류를 다음 단계에서 발견했을 때 오류 수정이 용이하다.
다. 두 개 이상의 과정이 병행 수행되거나 이전 단계로 넘어 가는 경우가 없다.
라. 개발 과정 중에 발생하는 새로운 요구나 경험을 설계에 반영하기 힘들다.

[기-07년3월][기-05년9월]

26. 사용자의 요구사항 분석 작업이 어려운 이유와 거리가 먼 것은?

- 가. 개발자와 사용자 간의 지식이나 표현의 차이가 커서 상호 이해가 쉽지 않다.
- 나. 사용자의 요구는 예외가 거의 없어 열거와 구조화가 어렵지 않다.
- 다. 사용자의 요구사항이 모호하고 부정확하며, 불완전하다.
- 라. 개발하고자 하는 시스템 자체가 복잡하다.

[기-09년5월][기-07년9월]

27. 소프트웨어 생명 주기 모형 중 Boehm 이 제시한 고전적 생명주기 모형으로 선형 순차적 모델이라고도 하며, 타당성 검토, 계획, 요구사항 분석, 설계, 구현, 테스트, 유지보수의 단계를 통해 소프트웨어를 개발하는 모형은?

- 가. 폭포수 모형
- 나. 프로토타입 모형
- 다. 나선형 모형
- 라. RAD 모형

(3) 소프트웨어 생명 주기 > 프로토타입 모형

[기-07년3월]

28. 소프트웨어 생명 주기 모형 중 다음 설명에 해당하는 것은?

- 시스템 기능을 사용자에게 미리 보여줌으로써 개발자와 사용자 간의 오해 요소를 줄인다.
- 사용자와 개발자 간의 커뮤니케이션이 원활하지 못할 때 서로의 이해에 도움을 준다.
- 실제 개발된 시스템 견본을 미리 만들어 최종 결과물을 예측하는 모형이다.

- 가. 폭포수 모형
- 나. 나선형 모형
- 다. 프로토타입 모형
- 라. 4GT 모형

[기-06년9월][기-04년9월][기-99년8월][기-01년6월]

29. 프로토타입 모형(Prototyping Model)에 대한 설명으로 옳지 않은 것은?

- 가. 최종 결과물의 만들어지기 전에 의뢰자가 최종 결과물의 일부 또는 모형을 볼 수 있다.
- 나. 개발 단계에서 오류 수정이 불가능하므로 유지보수 비용이 많이 발생한다.
- 다. 프로토타입은 발주자나 개발자 모두에게 공동의 참조 모델을 제공한다.
- 라. 프로토타입은 구현 단계의 구현 골격이 될 수 있다.

[기-06년5월][기-06년3월][기-05년3월][기-03년3월][기-00년7월][기-07년9월]

30. 프로토타입 모형의 장점으로 가장 적절한 것은?

- 가. 비용과 시간의 절감
- 나. 책임 한계의 명백한 구분
- 다. 요구사항의 충실 반영
- 라. 프로젝트 관리의 용이

[기-06년3월]

31. 프로토타입(Prototyping) 접근 방법을 채용할 때의 이익은 주로 정보 문제의 본질에 대한 불확실성과 그 정보 문제를 해결하기 위해 사용자가 제시하는 요구의 불확실성을 줄이는 데 있다. 다음 중 불확실성 결정 요인에 해당하지 않는 것은?

- 가. 자원이 필요한 일로부터의 요구연역(要求演繹)
- 나. 사용자와 분석자의 지식과 경험의 수준
- 다. 커뮤니케이션 문제가 일어날 가능성
- 라. 프로토타이핑시 소요되는 비용문제

[기-05년9월][기-00년3월]

32. 시스템의 일부 혹은 시스템의 모형을 만드는 과정으로서 요구된 소프트웨어의 일부를 구현하며, 추후 구현단계에 사용될 골격코드가 되는 모형은?

- 가. 폭포수 모형
- 나. 점증적 모형
- 다. 프로토타이핑 모형
- 라. 계획수립 모형

[기-08년3월][기-05년3월][기-01년9월]

33. 소프트웨어 수명주기 모형 중 프로토타이핑 모형(prototyping model)의 가장 큰 장점은?

- 가. 위험요소가 쉽게 발견된다.
- 나. 유지보수가 쉬워진다.
- 다. 사용자 요구사항을 정확하게 파악할 수 있다.
- 라. 소프트웨어 개발 일정을 정확하게 수립할 수 있다.

[기-04년3월][기-01년3월]

34. 실제 상황이 나오기 전에 가상으로 시뮬레이션을 통해 최종 결과물에 대한 예측을 할 수 있는 소프트웨어 수명 주기 모형은?

- 가. 점증적 모형(spiral model)
- 나. 프로토타이핑 모형(prototyping model)
- 다. 코코모 모형(cocomo model)
- 라. 폭포수 모형(waterfall model)

[기-99년10월]

35. 프로토타입 모델 개발 방법이 가장 적절하게 적용될 수 있는 경우는?

- 가. 테스트 작업이 중요하지 않을 경우
- 나. 고객이 빠른 시간 내에 개발의 완료를 요구할 경우
- 다. 구축하고자 하는 시스템의 요구사항이 불명확한 경우
- 라. 고객이 개발 과정에는 참여하지 않고자 하는 경우

[기-00년10월]

36. 다음은 프로토타이핑(원형) 모형의 개발에 필요한 작업을 기술한 것이다. 작업 순서대로 옳게 나열한 것은?

- | | |
|------------|------------|
| ① 빠른 설계 | ② 프로토타입 구축 |
| ③ 프로토타입 조정 | ④ 요구 수집 |
| ⑤ 구현 | ⑥ 고객평가 |

- 가. ④-②-①-③-⑥-⑤
- 나. ④-①-②-⑤-⑥-③
- 다. ④-①-②-③-⑥-⑤
- 라. ④-①-②-⑥-③-⑤

(4) 소프트웨어 생명 주기 > 나선형 모델

[기-04년9월][기-05년5월][기-99년8월]

37. 소프트웨어 수명주기 모형 중 나선형(spiral) 모델의 처리 절차에 해당되지 않는 것은?

- 가. 계획수립
- 나. 고객 평가
- 다. 위험 분석
- 라. 시스템 유지보수

[기-04년5월]

38. 소프트웨어 생명주기(life cycle) 모델 중 아래 보기

가 설명하는 모형은?

- a. 고객과의 의사소통(Communication)을 통해 계획수립과 위험분석, 구축, 고객평가의 과정을 거쳐 소프트웨어를 개발한다.
- b. 가장 큰 장점인 위험분석 단계에서 기술과 관리의 위험요소 들을 하나씩 제거해 나감으로서 완성도 높은 소프트웨어를 만들 수 있다.
- c. 반복적인 작업을 수행하는 점증적 생명주기 모델이다.
- d. 비용이 많이 들거나 시간이 많이 소요되는 대규모 프로젝트나 큰 시스템을 구축할 때 유리하다.

- 가. 프로토타입(prototype) 모델
- 나. 폭포수(waterfall) 모델
- 다. 나선형(spiral) 모델
- 라. RAD 모델

[기-03년5월][기-02년3월]

39. Boehm 이 제안한 나선형 모델의 태스크(task)에 해당되지 않는 것은?

- 가. 계획 수립(Planning)
- 나. 위험 분석(Risk Analysis)
- 다. 객체 구현(Object Implementation)
- 라. 고객 평가(Customer Evaluation)

[기-09년5월][기-02년9월]

40. 소프트웨어 수명주기 모형 중 나선형(spiral) 모형의 단계와 순서가 올바르게 구성된 것은?

- 가. Planning-Requirement Analysis-Development-Maintenance
- 나. Planning-Risk Analysis-Engineering-Customer Evaluation
- 다. Requirement Analysis-Planning-Design-Maintenance
- 라. Requirement Analysis-Risk Analysis-Development-Maintenance

(5) 2008년 기출문제(중복제거)

[기-08년3월]

41. 소프트웨어 공학에 대한 설명으로 가장 적합한 것은?

- 가. 소프트웨어의 제작부터 운영까지 생산성을 높이기 위해 기술적, 인간적인 요소에 대한 방법론을 제공한다.
- 나. 소프트웨어의 설계, 제작, 운영에 있어서 인간적인 요소를 배제한 프로그래밍 자체에 대한 공학적 연구를 의미한다.
- 다. 소프트웨어의 공학적이고 기술적인 영향을 사회 경제적인 시각에서만 설명한다.
- 라. 소프트웨어의 위기를 해결하기 위해서 현재 이미 해결된 문제들에 대해서 역사적 관점을 설명한다.

[기-08년5월]

42. 프로토타이핑 모형에 대한 설명으로 옳지 않은 것은?

- 가. 프로토타이핑 모형은 발주자나 개발자 모두에게 공동의 참조 모델을 제공한다.
- 나. 사용자의 요구사항을 충실히 반영할 수 있다.
- 다. 프로토타이핑 모형은 소프트웨어 생명주기에서 유지보수가 없어지고 개발 단계 안에서 유지보수가 이루어지는 것으로 볼

수 있다.

- 라. 최종 결과물이 만들어지는 소프트웨어 개발 완료 시점에서 최초로 오류 발견이 가능하다.

[기-08년9월]

43. 소프트웨어 공학에 대한 적절한 설명이 아닌 것은?

- 가. 소프트웨어의 개발, 운영, 유지보수, 그리고 폐기에 대한 체계적인 접근이다.
- 나. 소프트웨어 제품을 체계적으로 생산하고 유지보수와 관련된 기술과 경영에 관한 학문이다.
- 다. 과학적인 지식을 컴퓨터 프로그램 설계와 제작에 실제 응용하는 것이며, 이를 개발하고 운영하고 유지보수하는데 필요한 문서화 작성 과정이다.
- 라. 소프트웨어의 위기를 이미 해결한 학문으로, 소프트웨어의 개발만을 위한 체계적인 접근이다.

[기-08년9월]

44. 소프트웨어 생명주기 모형에 대한 설명으로 옳은 것은?

- 가. 폭포수 모형을 점진적 모형이라고도 한다.
- 나. 나선형 모형은 반복적으로 개발이 진행되므로 소프트웨어의 강인성을 높일 수 있다.
- 다. 프로토타입 모형은 개발 단계에서 요구사항 변경이 불가능하므로 유지보수 비용이 많이 발생한다.
- 라. 폭포수 모형은 최종 결과물이 만들어지기 전에 의뢰자가 최종결과물의 모형을 볼 수 있다.

(6) 2009년 기출문제(중복제거)

[기-09년3월]

45. 다음 중 소프트웨어 개발 모형이 가장 적절하게 선택된 경우는?

- 가. 구축하고자 하는 시스템의 요구사항이 불분명하여 프로토타입 모형을 선택하였다.
- 나. 개발 중에도 고객의 요구사항에 맞게 수정 작업을 할 수 있도록 폭포수 모형을 선택하였다.
- 다. 위험 분석을 통해 점증적으로 시스템을 개발할 수 있도록 폭포수 모형을 선택하였다.
- 라. 응용분야가 단순하고 설치 시점에 제품 설명서가 요구됨에 따라 나선형 모형을 선택하였다.

[기-09년8월]

46. 나선형 (Spiral) 모형에 대한 설명으로 옳지 않은 것은?

- 가. 위험성 평가에 크게 의존하기 때문에 이를 발견하지 않으면 문제가 발생 할 수 있다.
- 나. 대규모 시스템의 소프트웨어 개발에 적합하다.
- 다. 여러 번의 개발 과정을 거쳐 점진적으로 완벽한 소프트웨어를 개발한다.
- 라. 작업 순서는 타당성 검토, 계획, 요구분석, 설계, 구현, 시험, 유지보수의 단계로 이루어진다.

[기-09년8월]

47. 소프트웨어의 특성이 아닌 것은?

- 가. 물리적인 마모에 의해서 사용할 수 없게 된다.

- 나. 유형의 매체에 저장되지만 개념적이고 무형적이다.
 다. 수학이나 물리학에서 볼 수 있는 규칙적이고 정형적인 구조가 없다.
 라. 요구나 환경의 변화에 따라 적절히 변형시킬 수 있다.

[기-09년8월]

48. 프로젝트에 내재된 위험 요소를 인식하고 그 영향을 분석하여 이를 관리하는 활동으로서, 프로젝트를 성공시키기 위하여 위험 요소를 사전에 예측하여 대비하는 모든 기술과 활동을 포함하는 것은?

- 가. Critical Path Method
 나. Risk Analysis
 다. Work Breakdown Structure
 라. Waterfall Model

[SE01-소프트웨어공학,생명주기]

1	2	3	4	5	6	7	8	9	10
라	라	다	나	가	다	가	나	가	라
11	12	13	14	15	16	17	18	19	20
다	다	다	라	다	라	다	라	라	라
21	22	23	24	25	26	27	28	29	30
다	나	가	나	나	나	가	다	나	다
31	32	33	34	35	36	37	38	39	40
라	다	다	나	다	라	라	다	다	나
41	42	43	44	45	46	47	48	49	50
가	라	라	나	가	라	가	나		