

1. 프로젝트 관리 : 계획대로 완료될 수 있도록 관리 ★★★★★☆

- 목적 : 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템 개발

1) 프로젝트 관리 대상 : 계획, 개발팀 관리, 비용 관리, 일정 관리, 위험 관리, 형상 관리, 품질 관리

=> **고객 관리 (X)**

2) **효과적인 프로젝트 관리를 위한 3대 요소 (3P)**

① 사람(people) : 인적 자원

② 문제(problem) : 문제 인식

③ 프로세스(process) : 작업 계획

1. 프로젝트 계획 : 수행 전 예측하는 작업 ★★☆☆☆☆

- 프로젝트가 수행되기 전에 소프트웨어 개발 영역(범위) 결정, 필요한 자원, 비용, 일정 등을 예측

1) 프로젝트 계획 수립 시 예측 대상

- 범위, 비용, 일정, 성능, **유지보수비용 (X)**, **위험성 (X)** : 돌발 상황

2) 프로젝트 계획 수립 시 소프트웨어 영역 결정 사항

- 기능, 성능, 제한조건, 신뢰도, **위험성 (X)**

1. 개발팀 관리 ★★☆☆☆☆

① **중앙 집중형** (책임 프로그래머 팀) : 한 사람에 의하여 통제 -> 소규모 프로젝트 적합

- 책임 프로그래머 : 분석, 설계, 작업 지시 등 모든 기술적 **판단**

- 보조 프로그래머 : 책임 프로그래머 업무 지원

- 프로그래머 : 코딩, 검사, 디버깅, 문서 작성 등

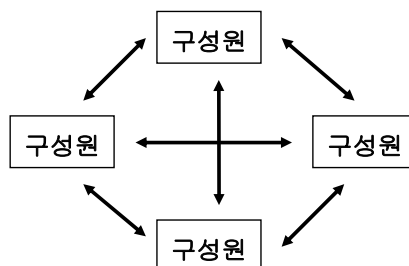
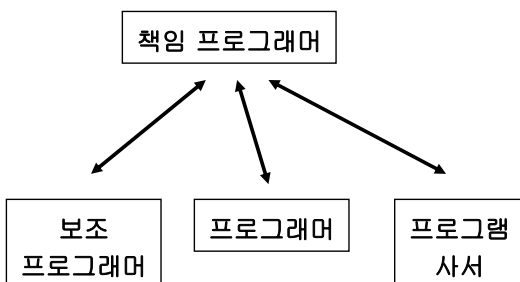
- 프로그램 사서 : 프로그램 리스트, 설계 문서, 검사 계획 등

② **분산형** (민주주의식) : 링 모양 구조

- 모든 팀 구성원이 동등한 위치에서 의사 결정 -> 장기 프로젝트 적합

- 서로의 일을 검토하고 결과에 대해 같은 그룹의 일원으로 책임짐

- 의사 교류를 활성화 -> 구성원의 작업 만족도 증대



1. 비용 관리 ★★☆☆☆

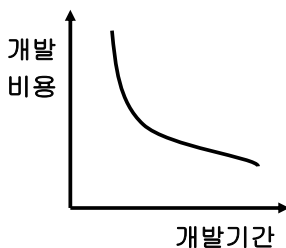
1) 비용 결정 요소

- 개발자의 능력, 요구되는 신뢰도, 개발 제품의 복잡도
- 하드웨어 성능 (X)

2) 비용을 정확하게 예측하기 위한 방법

- 예측을 가능한 한 뒤로 미룸 (현실성 X)
- 이미 수행된 유사 프로젝트 참고
- 프로젝트를 상대적으로 잘게 분리하여 예측
- 경험적 예측 모델을 활용 : 실험에 의한 결과 활용

3) 개발비용과 개발기간 상관 관계



-> 개발완료기간을 앞당기면 비용은 더 증가

2. 비용 예측 기법 > LOC 기법 ★☆☆☆☆

1) 용어 정리

- ① LOC : Line Of Code (원시코드 라인 수)
- ② 인월 (PM) : 개발에 소요되는 기간을 1개월로 고정할 경우 필요한 총 인원수
 - ex) 12인월 : 12명이 1개월 동안 작업해야 하는 작업의 양

2) 기출문제

두 명의 개발자가 5개월에 걸쳐 10,000 라인의 코드를 개발하였을 때, 월별(Person Month) 생산성 ?
=> $10,000 / 5 * 2 = 1,000$

어떤 소프트웨어 개발을 위해 10명의 개발자가 10개월 동안 참여되었다. 그런데 그 중 7명은 10개월 동안 계속 참여했지만 3명은 3개월 동안만 부분적으로 참여했다. 이 소프트웨어 개발을 위한 인월(Man Month) ?
=> $7 * 10 + 3 * 3 = 79$

LOC 기법에 의하여 예측된 총 라인수가 25,000 라인일 경우 개발에 투입될 프로그래머의 수가 5명이고, 프로그래머들의 평균 생산성이 월당 500 라인일 때, 개발에 소요되는 기간 ?
=> $25,000 / 5 * 500 = 10$

개발에 소요되는 노력이 40PM(Programmer Month)으로 계산되었다. 개발에 소요되는 기간이 5개월이고, 1인당 인건비가 100만원이라면 이 프로젝트에 소요되는 총 인건비 ?
=> $40 * 100 = 4,000$

3. 비용 예측 모형 > COCOMO 모형 ★★★★★

1) 특징

- Boehm이 제안한 원시 프로그램의 규모에 의한 비용예측 모형 (인월 : man-month)
- 소프트웨어의 종류에 따라 다르게 책정되는 비용신장 방정식을 이용
- 같은 규모의 프로그램이라도 그 성격에 따라 비용이 다르게 생성
- 비용 견적의 강도 분석 및 비용견적의 유연성이 높아 소프트웨어 개발비 견적에 널리 통용

2) COCOMO 모형

- ① Basic COCOMO (기본형)
- ② Intermediate COCOMO (중간형)
- ③ Detailed COCOMO (진보형)

3) COCOMO 유형(모드) : 기본(Basic) 모형은 단순히 소프트웨어의 크기와 개발 모드에 의해서 구해진다

- ① Organic 프로젝트 (유기형) : 5만 라인 이하 규모 (일괄처리, 과학 기술 계산용 등)
- ② Semi-Detached 프로젝트 (반분리형) : 30만 라인 이하 규모 (운영체제 등)
- ③ Embedded 프로젝트 (내장형) : 30만 라인 이상의 최대형 규모 (운영체제 등)

* 그 외 비용 예측(추정) 모형

- Putnam 모형, Function-Point 모형

1. 일정 관리 ★★★★★☆

1) 프로젝트 일정 계획 기법 : WBS(작업 분해), PERT/CPM, Gantt Chart

2) 브룩스(Brooks) 법칙

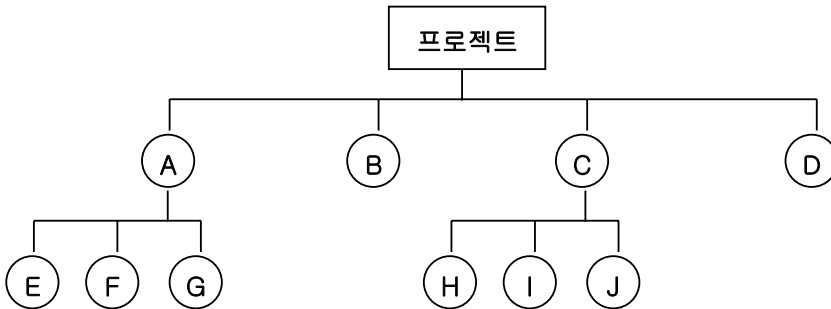
- 새로운 개발 인력이 진행 중인 프로젝트에 투입될 경우 작업 적응 기간과 부작용으로 인해 빠른시간 내에 프로젝트는 완료될 수 없다.

3) 일정 계획의 순서

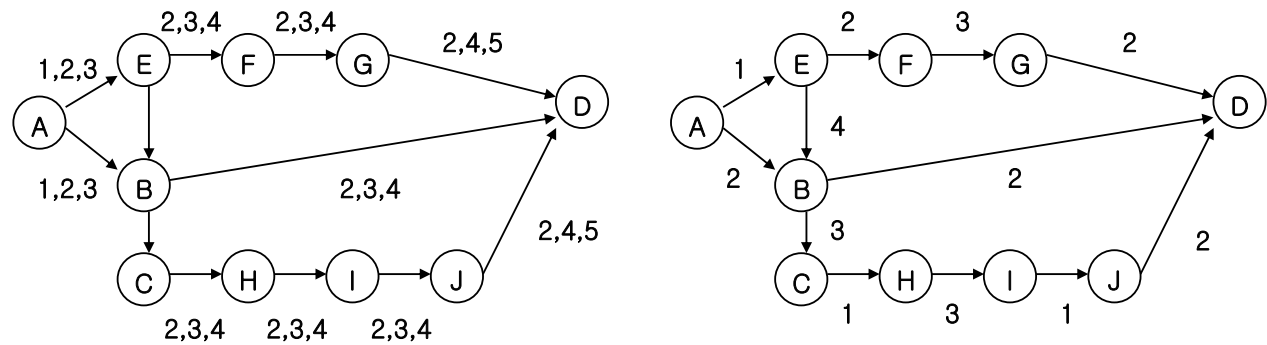
- ① 프로젝트 규모 추정
- ② 소단위 작업 분해
- ③ 각 작업의 상호관계를 CPM 네트워크로 표현 (PERT, CPM)
- ④ Gantt Chart로 표현

2. 작업 분해

- 프로젝트를 여러 개의 작은 소단위로 분해하여 계층 구조로 표현



3. PERT/CPM 개념 이해하기



4. PERT/CPM 특징 ★★★★★

- CPM (Critical Path Method, 임계 경로 기법)
- 프로젝트의 지연을 방지하고 계획대로 진행되게 하기 위한 일정 계획 방법
- 대단위 계획의 조직적인 추진을 위해 자원의 제약 하에 비용을 적게 사용하면서 초단기간 내 계획 완성을 위한 프로젝트 일정 방법
- 병행작업이 가능하도록 계획할 수 있음
- 노드에서 작업을 표시하고 간선은 작업 사이의 전후 의존 관계를 나타냄
- 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는 데 사용
- 박스 노드는 프로젝트 중간 점검을 뜻하는 이정표로 이 노드 위에는 예상 완료 시간을 표시
- 프로젝트 작업 사이의 관계를 나타내며 최장경로(임계 경로)를 파악할 수 있음
- 정확한 예측, 노력과 비용 (X)

5. Gantt Chart ★★★★★

작업 일정 작업 단계	이정표												산출물
	1	2	3	4	5	6	7	8	9	10	11	12	
A													AA
B													BB

- 포함되는 내용 : 이정표, 작업 일정, 작업 기간, 산출물, 작업 경로 (X)

1. 품질 관리 ★★★★★

1) 품질 보증 : 어떤 항목이나 제품이 설정된 기술적 요구사항과 일치하는가를 적절하게 확인하는데 필요한 체계적이고도 계획적인 유형의 활동

2) 품질 목표의 항목

- 정확성 : 사용자 요구 기능 충족 정도
- 신뢰성 : 옳고 일관된 결과를 얻기 위해 요구되는 기능
- 이식성 : 다른 H/W 환경에서 운용 가능
- 상호 운용성 : 다른 S/W와 정보를 교환할 수 있는 기능
- 유지보수성 : 변경 시 수정에 대한 노력의 최소화 정도
- 효율성 : 기능 수행 시 필요한 자원의 소요 정도
- 무결성 : 허용되지 않는 사용이나 자료의 변경을 제어
- 사용 용이성 : 사용하기 쉬운 정도
- 유연성 : 쉽게 수정할 수 있는 정도
- 시험 용이성 : 평가를 쉽게 해 주는 정도
- 재사용성 : 전체나 일부 소프트웨어가 다른 응용 목적으로 사용 가능
- S/W 종속성, 중복성, 복잡성, 최적화 (X)

2. 품질관리 위원회 ★★☆☆☆

: 소프트웨어 품질 향상을 목적으로 구성

1) 정형 기술 검토 (FTR : Formal Technical Review)

- 가장 일반적인 검토 방법으로 소프트웨어 품질 보증 활동
- 목적 : 기능과 로직의 오류 발견, 사용자 요구사항의 확인, 프로젝트 관리의 편리성 등
- 지침 사항 : 의제 제한성, 논쟁과 반박의 제한성, 제품 검토의 집중성, 참가 인원의 제한성 등

2) 워크스루 (Walkthrough)

- 각 단계가 끝나면 검토 회의
- 오류 검출에 초점을 두고 해결책은 나중에 미룬다.
- 발견된 오류는 문서화
- 검토를 위한 자료를 사전에 배포하여 검토

3. 신뢰성(가용성) 측정 ★☆☆☆☆

A-가동	B-고장	C-가동	D-고장	E-가동	F-고장
------	------	------	------	------	------

8

2

9

4

4

1

기출) 어떤 시스템의 운용 기간이 다음과 같을 때 신뢰도를 계산하면 얼마인가

$$\Rightarrow (A+C+E)/(A+B+C+D+E+F) = (8+9+4)/(8+2+9+4+4+1) = 0.75$$

[SE 2강]-프로젝트 관리

1. 위험 관리 ★★★★★☆

- 프로젝트 추진 과정에서 예상되는 각종 돌발 상황을 미리 예상하고 이에 대한 적절한 대책을 수립하는 활동

1) 위험 관리 절차 : 위험 식별 -> 위험 분석 및 평가 (위험표 작성) -> 위험 관리 계획 -> 위험 감시 및 조치

2) 위험표에 포함될 사항 : 위험 내용, 위험 종류, 위험 발생 확률, 위험에 따른 영향력, 위험 감시 및 조치

- 위험 발생 시간 (X)

3) 위험 요소 : 사용자 요구 사항 변경(가장 대표적), 인력부족, 예산부족

4) 위험 모니터링(monitring) : 위험 요소 징후들에 대하여 계속적으로 인지하는 것

1. 형상 관리 ★★★★★☆

1) 형상 정의 : 소프트웨어 개발 단계의 각 과정에서 만들어지는 프로그램, 문서, 데이터 등을 통칭

2) 형상 관리 : 소프트웨어의 생산물을 확인하고 소프트웨어 통제, 변경 상태를 기록하고 보관하는 일련의 작업
- 유지보수 단계에서 행해진다

3) 형상 관리 항목 : 정의 단계의 문서, 개발 단계의 문서와 프로그램, 유지보수 단계의 변경 사항, 비용 (X)

[SE 2강]-프로젝트 관리

1. 프로젝트 관리의 대상으로 거리가 먼 것은?

- 가. 비용 관리 나. 일정 관리
다. 고객 관리 라. 품질 관리

2. 효과적인 소프트웨어 프로젝트 관리를 위한 3P에 해당되지 않는 것은?

- 가. People(사람) : 인적 자원
나. Product(생산물) : 생산 일정
다. Problem(문제) : 문제 인식
라. Process(프로세스) : 작업 계획

3. 분산형 팀 구성에 관한 설명 중 옳지 않은 것은?

- 가. 의사 결정을 민주주의 식으로 하며 팀 구성원의 작업 만족도를 높이고 이직률을 낮게 한다.
나. 팀 구성원 각자가 서로의 일을 검토하고 다른 구성원이 일한 결과에 대하여 같은 그룹의 일원으로 책임을 진다.
다. 팀 구성원 사이의 의사 교류를 활성화시키므로 복잡한 장기 프로젝트에 적합하지 않다.
라. 링 모양의 구조는 계층 없이 전체 팀 구성원이 동등한 레벨에 있다는 것을 나타낸다.

4. 중앙 집중형 팀 구성에서 역할 분담에 관한 설명 중 옳지 않은 것은?

- 가. 책임 프로그래머 : 분석 및 설계, 기술적 판단, 작업 지시와 배분을 담당
나. 보조 프로그래머 : 책임 프로그래머의 업무 지원
다. 프로그래머 : 원시 코드 작성, 검사, 디버깅, 문서 작성 담당
라. 프로그램 사서 : 컴파일, 디버깅, 목적 프로그램 작성

5. 프로젝트 계획 수립을 시작할 때 제일 먼저 해야 하는 작업은?

- 가. 개발 완료 날짜 파악
나. 과거의 데이터를 분석하는 일
다. 개발비용 산정
라. 프로젝트의 규모 파악

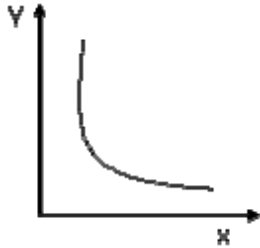
6. S/W 프로젝트를 신뢰성 있게 예측하는 방법 중 현실성이 부족한 것은?

- 가. 예측을 가능한 뒤로 미룬다.
나. 이미 수행된 유사 프로젝트를 참고 한다.
다. 프로젝트를 상대적으로 잘게 분리하여 예측한다.
라. 경험적 예측 모델을 활용한다.

7. 프로젝트의 개발비용 산정 시 결정에 영향을 주는 요소로서 거리가 먼 것은?

- 가. 비용 산정 기법 나. 시스템의 크기
다. 시스템의 신뢰도 라. 제품의 복잡도

8. 소프트웨어 개발비용은 다른 여러 가지 요소들과 일정한 상관관계가 있다. 다음 그래프의 y축을 개발비용이라고 했을 때, x축은 어떤 요소라고 보는 것이 가장 타당한가?



- 가. 시스템 크기 나. 개발 기간
다. 신뢰도 라. 투입 인력

9. 소프트웨어 추정 모형(Estimation Model)이 아닌 것은?

- 가. COCOMO 나. Putnam
다. Function-Point 라. PERT

10. COCOMO의 프로젝트 모드가 아닌 것은?

- 가. Organic Mode 나. Semi-detached Mode
다. Medium Mode 라. Embedded Mode

11. COCOMO 모델에 대한 설명으로 옳지 않은 것은?

- 가. Boehm이 제시한 비용 추정 모델이다.
나. 비용 추정 단계 및 적용 변수의 구체화 정도에 따라 기본(Basic), 중간(Intermediate), 진보(Advanced)형 모델로 구분할 수 있다.
다. 비용 건적의 강도 분석 및 비용 건적의 유연성이 높아 소프트웨어 개발비 건적에 널리 통용되고 있다.
라. 기본(Basic) 모형은 단순히 소프트웨어의 크기와 개발 모드에 의해서 구해진다.

[정답] 7.가 8.나 9.라 10.다 11.나

12. COCOMO (COConstructive COst MOdel) 모형에 대한 설명으로 옳지 않은 것은?

- 가. 산정 결과는 프로젝트를 완성하는데 필요한 Man-Month로 나타난다.
나. Boehm이 고안한 개발비 산정 모델로 프로젝트의 예상되는 크기와 유형에 관한 정보가 주로 사용된다.
다. 프로젝트 특성을 15개로 나누고 각각에 대한 승수 값을 제시 하였다.
라. 각 모델 별로 개발되어지는 프로젝트 개발 유형에 따라 Object Mode, Dynamic Mode, Function Mode의 3가지 모드로 구분한다.

13. 소프트웨어 프로젝트 관리를 효과적으로 수행 하는데 필요한 3P와 거리가 먼 것은?

- 가. People 나. Power
다. Problem 라. Process

[정답] 12.라 13.나

14. S/W Project 일정이 지연된다고 해서 Project 말기에 새로운 인원을 추가 투입하면 Project는 더욱 지연되게 된다고 주장하는 법칙은?

- 가. Putnam의 법칙 나. Mayer의 법칙
다. Brooks의 법칙 라. Boehm의 법칙

15. Gantt Chart에 포함되지 않는 사항은?

- 가. 이정표 나. 작업 일정
다. 작업 기간 라. 주요 작업 경로

16. 브룩스(Brooks) 법칙의 의미로 가장 적절한 것은?

- 가. 프로젝트 개발에는 많은 개발자가 필요하지 않다.
나. 새로운 개발 인력이 진행 중인 프로젝트에 투입될 경우 작업 적응 기간과 부작용으로 인해 빠른 시간 내에 프로젝트는 완료 될 수 없다.
다. 프로젝트에는 많은 비용이 투입되어야 한다.
라. 프로젝트에 개발자가 많이 참여 할수록 프로젝트의 준공 기간은 지연된다.

[정답] 14.다 15.라 16.나 17.다 18.가 19.라

17. 일정 계획과 관계가 먼 것은?

- 가. 작업 분해 나. CPM 네트워크
다. 프로그램 명세서 라. 간트 차트(Gantt Chart)

18. 프로젝트 관리 중 일정 계획에 필요한 작업들을 기술한 것이다. 순서대로 나열 된 것은?

- ① 각 작업의 상호 의존 관계를 CPM 네트워크로 나타낸다.
② 일정 계획을 간트 차트로 나타낸다.
③ 프로젝트의 규모를 추정한다.
④ 각 단계에 필요한 작업들을 분리한다.

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

19. CPM(Critical Path Method) 네트워크에 대한 설명으로 옳지 않은 것은?

- 가. 노드에서 작업을 표시하고 간선은 작업 사이의 전후 의존 관계를 나타낸다.
나. 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는 데 사용한다.
다. 박스 노드는 프로젝트의 중간 점점을 뜻하는 이정표로 이 노드 위에는 예상 완료 시간을 표시한다.
라. 한 이정표에서 다른 이정표에 도달하기 전의 작업은 모두 완료되지 않아도 다른 작업을 진행할 수 있다.

20. 소프트웨어 품질 관리 기술에서 품질 목표의 항목과 거리가 먼 것은?

- 가. 정확성 나. 유지보수성
다. 무결성 라. S/W 종속성

21. 소프트웨어 품질 목표에 대한 설명으로 옳지 않은 것은?

- 가. 신뢰성(Reliability) : 정확하고 일관된 결과를 얻기 위해 요구된 기능을 수행하는 정도
나. 이식성(Portability) : 다양한 하드웨어 환경에서도 운용 가능하도록 쉽게 수정 될 수 있는 정도
다. 상호 운용성(Interoperability) : 다른 소프트웨어와 정보를 교환할 수 있는 정도
라. 사용 용이성(Usability) : 전체나 일부 소프트웨어가 다른 응용 목적으로 사용 될 수 있는 정도

[정답] 20.라 21.라 22.라 23.가 24.가

22. 워크스루(Walkthrough)의 특징으로 거리가 먼 것은?

- 가. 발견된 오류는 문서화 한다.
나. 오류 검출에 초점을 두고 해결책은 나중에 미룬다.
다. 검토를 위한 자료를 사전에 배포하여 검토하도록 한다.
라. 정해진 시간과 비용으로 프로젝트를 완성시키기 위한 대책을 수립한다.

23. 소프트웨어 품질 보증 활동 중 정형 검토 (Formal Review)의 목적이라고 할 수 없는 것은?

- 가. 적정 인력의 투입 확인
나. 기능과 로직의 오류 발견
다. 사용자 요구사항의 확인
라. 프로젝트 관리의 편리성

24. 소프트웨어의 신뢰성과 가용성에 대한 설명 중 잘못된 것은?

- 가. 소프트웨어의 신뢰성은 과거와 개발상의 자료를 이용하여 측정과 예측이 불가능하다.
나. 소프트웨어의 간단한 신뢰성 측정은 MTBF로 가능하다.
다. 소프트웨어의 가용성은 프로그램이 요구사항에 따라 운영되는 확률이다.
라. 가용성은 $\{MTTF/(MTTF+MTTR)\} \times 100\%$ 로 정의 된다.

[SE 2강]-프로젝트 관리

25. 어떤 시스템의 운용 기간이 다음과 같을 때 신뢰도를 계산하면 얼마인가?

가동 중 (8)	고장 중 (2)	가동 중 (9)	고장 중 (4)	가동 중 (4)	고장 중 (1)
-------------	-------------	-------------	-------------	-------------	-------------

가. 0.75 나. 0.25 다. 9.3 라. 7

26. 신뢰도를 평가하는 MTBF(Mean Time Between Failure)는 가동된 평균 시간을 나타내며, MTTR(Mean Time To Repair)은 평균 수리 시간을 의미한다. 이 두 가지 척도를 사용하여 신뢰도를 구하는 식을 바르게 나타낸 것은?

- 가. $MTTR / (MTBF + MTTR)$
 나. $MTTR / MTBF$
 다. $MTBF / (MTBF + MTTR)$
 라. $MTBF / MTTR$

[정답] 25.가 26.다 27.가 28.가 29.다

27. 프로젝트 추진 과정에서 예상되는 각종 돌발 상황을 미리 예상하고 이에 대한 적절한 대책을 수립하는 일련의 활동을 무엇이라고 하는가?

- 가. 위험 관리 나. 일정 관리
 다. 코드 관리 라. 모형 관리

28. 위험 관리의 일반적인 절차로 적합한 것은?

- 가. 위험 식별→위험 분석 및 평가→위험 관리 계획→위험 감시 및 조치
 나. 위험 분석 및 평가→위험 식별→위험 관리 계획→위험 감시 및 조치
 다. 위험 관리 계획→위험 감시 및 조치→위험 식별→위험 분석 및 평가
 라. 위험 감시 및 조치→위험 식별→위험 분석 및 평가→위험 관리 계획

29. 소프트웨어 품질 보증을 위한 정형 기술 검토의 지침 사항으로 옳지 않은 것은?

- 가. 논쟁과 반박을 제한한다.
 나. 각 체크 리스트를 작성하고, 자원과 시간 일정을 할당한다.
 다. 의제와 참가자의 수를 제한하지 않는다.
 라. 검토의 과정과 결과를 재검토 한다.

[SE 2강]-프로젝트 관리

30. 소프트웨어 형상 관리(Configuration Management)란?

- 가. 소프트웨어 구성 항목을 관리하는 것
 나. 개발 과정의 변화되는 사항을 관리하는 것
 다. 테스트 과정에서 소프트웨어를 통합하는 것
 라. 개발 인력을 관리하는 것

31. 소프트웨어에 대한 변경을 관리하기 위해 개발된 일련의 활동을 나타내며 이런 변경에 의해 전체 비용이 최소화되고 최소한의 방해가 소프트웨어의 현 사용자에게 야기되도록 보증하는 것을 목적으로 하는 것은?

- 가. 위험 관리 나. 형상 관리
 다. 프로젝트 관리 라. 유지보수 관리

32. 형상 관리(Configuration Management)의 관리 항목으로 거리가 먼 것은?

- 가. 정의 단계의 문서
 나. 개발 단계의 문서와 프로그램
 다. 유지보수 단계의 변경 사항
 라. 소프트웨어 개발비용

[정답] 30.나 31.나 32.라 33.라 34.나

33. 소프트웨어 형상 관리(Software Configuration Management)의 설명으로 가장 적합한 것은?

- 가. 소프트웨어 개발 과정을 문서화하는 것이다.
 나. 하나의 작업 산출물을 정해진 시간 내에 작성하도록 하는 관리이다.
 다. 수행 결과의 완전성을 점검하고 프로젝트의 성과 평가척도를 준비하는 작업이다.
 라. 소프트웨어의 생산물을 확인하고 소프트웨어 통제, 변경 상태를 기록하고 보관하는 일련의 관리 작업이다.

34. 소프트웨어 형상 관리(Configuration management)의 의미로 가장 적절한 것은?

- 가. 비용에 관한 사항을 효율적으로 관리하는 것
 나. 개발 과정의 변경 사항을 관리하는 것
 다. 테스트 과정에서 소프트웨어를 통합하는 것
 라. 개발 인력을 관리하는 것

[SE 2강]-프로젝트 관리

[기-08년5월]

35. 프로젝트 일정을 관리하는 PERT 차트로 알 수 있는 사항이 아닌 것은?

가. 결정 경로

나. 태스크의 시작/종료 시간

다. 태스크에 대한 경계시간

라. 태스크간의 상호관련성

[기-08년9월]

36. 어떤 소프트웨어 개발을 위해 10명의 개발자가 20개월동안 참여되었다. 그 중 7명은 20개월 동안 계속 참여했고 3명은 5개월 동안만 참여했다. 이 소프트웨어 개발에 필요한 MM(Man-Month)은 얼마인가?

가. 5 나. 20 다. 79 라. 155

[기-08년3월]

37. 소프트웨어 프로젝트를 계획하려면 먼저 소프트웨어 범위를 결정해야 한다. 다음 사항과 관계가 되는 범위 결정 요소는 무엇인가?

- 소프트웨어에 의해 간접적으로 제어되는 장치와

소프트웨어를 실행하는 하드웨어

- 순서에 의거 절차적 운영상 소프트웨어 실행 절차

- 기존의 소프트웨어나 새로운 소프트웨어를 연결하는 소프트웨어

- 키보드나 기타 I/O 장치들을 통하여 소프트웨어를 사용하는 사람

가. 기능

나. 성능

다. 제약조건

라. 인터페이스

[정답] 35.나 36.라 37.라