



5과목-정보시스템 구축 관리

(Part 1. 소프트웨어 개발 방법론 활용-1)

정보시스템 구축 관리 총 파트

정보시스템 구축 관리 5과목은 총 4Part로 이루어져 있다.

1장 소프트웨어 개발 방법론 활용(30.39%)

2장 IT프로젝트 정보 시스템 구축 관리 (36.46%)

3장 소프트웨어 개발 보안 구축(19.34%)

4장 시스템 보안 구축(13.81%)

소프트웨어 개발 방법론 활용

소프트웨어 개발 방법론 활용 Part는 9개의 섹션으로 구성되어 있다.

001 소프트웨어 개발 방법론

002 S/W 공학의 발전적 추세

003 비용 산정 기법 - 상향식

004 수학적 산정 기법

005 프로젝트 일정 계획

006 소프트웨어 개발 방법론 결정

007 소프트웨어 개발 표준

008 소프트웨어 개발 방법론 테일러링

009 소프트웨어 개발 프레임워크

5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

1) 소프트웨어 개발 방법론의 개요

; 소프트웨어 개발 방법론은 소프트웨어 개발, 유지보수 등에 필요한 여러 가지 일들의 수행 방법과 이러한 일들을 효율적으로 수행하려는 과정에서 필요한 각종 기법 및 도구를 체계적으로 정리하여 표준화한 것이다.

- 소프트웨어 개발 방법론의 목적은 소프트웨어의 생산성과 품질 향상이다.
- 소프트웨어 개발 방법론의 종류에는 구조적 방법론, 정보공학 방법론, 객체지향 방법론, 컴포넌트 기반(CBD; Component Based Design) 방법론, 애자일(Agile) 방법론, 제품 계열 방법론 등이 있다.

5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

2) 구조적 방법론

; 구조적 방법론은 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화 하는 처리(Process) 중심의 방법론이다.

- 1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법론이다.
- 쉬운 이해 및 검증이 가능한 프로그램 코드를 생성하는 것이 목적이다.
- 복잡한 문제를 다루기 위해 분할과 정복(Divide and Conquer) 원리를 적용한다.
- 구조적 방법론의 절차

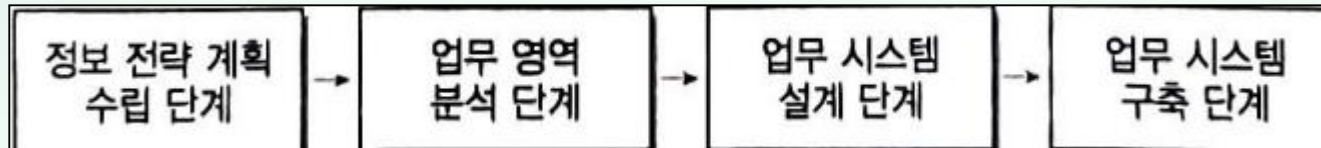


5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

3) 정보공학 방법론

; 정보공학 방법론은 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론이다.

- 정보 시스템 개발 주기를 이용하여 대규모 정보 시스템을 구축하는데 적합하다.
- 데이터베이스 설계를 위한 데이터 모델링으로 개체 관계도(ERD; Entity Relationship Diagram)를 사용한다.
- 정보공학 방법론의 절차

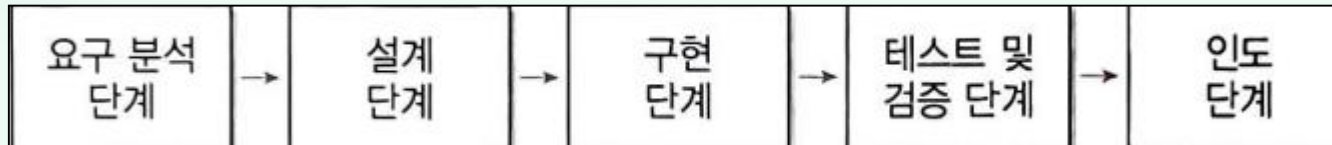


5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

4) 객체지향 방법론

; 객체지향 방법론은 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 소프트웨어를 개발할 때 기계의 부품을 조립하듯이 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론이다.

- 객체지향 방법론은 구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 채택되었다.
- 설계 과정에서 주로 사용되는 모델링 언어에는 패키지 다이어그램(Package Diagram), 배치 다이어그램(Deployment Diagram), 상태 전이도(State Transition Diagram) 등이 있다.
- 객체지향 방법론의 절차



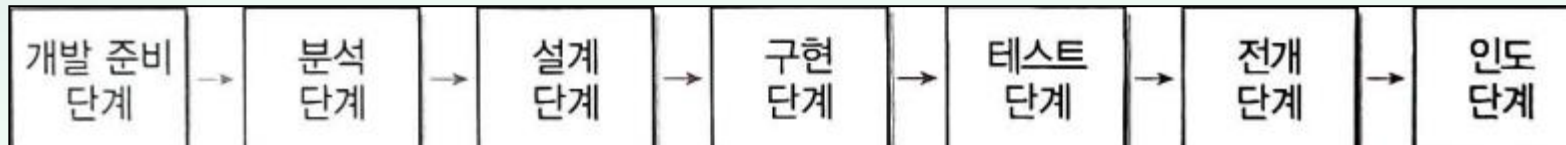
현실 세계의 개체 : 사람, 자동차, 컴퓨터, 고양이 등과 같이 우리 주위에서 사용되는 물질적이거나 개념적인 것으로 명사로 사용된다.

5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

5) 컴포넌트 기반(CBD; Component Based Design) 방법론

; 컴포넌트 기반 방법론은 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.

- 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.
- 새로운 기능을 추가하는 것이 간단하여 확장성이 보장된다.
- 유지 보수 비용을 최소화하고 생산성 및 품질을 향상시킬 수 있다.
- 컴포넌트 기반 방법론의 절차



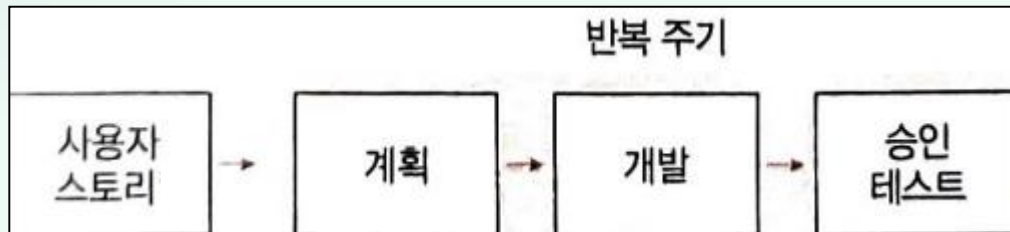
컴포넌트(Component) : 문서, 소스 코드 파일, 라이브러리 등과 같은 모듈화된 자원으로 재사용이 가능하다.

5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

6) 애자일(Agile) 방법론

; 애자일은 '민첩한', '기민한'이라는 의미로, 애자일 방법론은 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론이다.

- 소규모 프로젝트, 고도로 숙달된 개발자, 급변하는 요구사항에 적합하다.
- 애자일 방법론의 대표적인 종류에는 **익스트림 프로그래밍(XP; eXtreme Programing)**, **스크럼(Scrum)**, **칸반(Kanban)**, **크리스탈(Crystal)** 등이 있다.
- 애자일 방법론의 절차



XP(eXtreme Programming) : 수시로 발생하는 고객의 요구 사항에 유연하게 대응하기 위해 고객의 참여와 개발 과정의 반복을 극대화하여 개발 생산성을 향상하는 방법을 의미한다.

스크럼(Scrum) : 럭비에서 반칙으로 경기가 중단된 경우 양 팀의 선수들이 럭비공을 가운데 두고 상대팀을 밀치기 위해 서로 대치해 있는 대형을 말한다. 스크럼은 이처럼 팀이 중심이 되어 개발의 효율성을 높인다는 의미가 내포된 용어이다.

칸반(Kanban) : 연속적 흐름 처리 방식이다. 이슈는 큐에 입력되고, 개발 프로세스의 단계에 따라 당겨진다. 칸반은 칸반 보드로 시각화되고 각각 단계는 열로 표시한다.

크리스탈(Crystal) : 환경의 변화와 프로젝트의 특정한 특성에 맞추어 해결하기 위해 개발되었다.

사용자 스토리(User Story) : 사용자의 요구사항을 의미한다.

5. 소프트웨어 개발 방법론 활용-SEC_01(소프트웨어 개발 방법론)

7) 제품 계열 방법론

; 제품 계열 방법론은 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론이다.

- 임베디드 소프트웨어를 만드는데 적합하다.
- 제품 계열 방법론은 영역 공학과 응용 공학으로 구분된다.
 - 영역 공학 : 영역 분석, 영역 설계, 핵심 자산을 구현하는 영역이다.
 - 응용 공학 : 제품 요구 분석, 제품 설계, 제품을 구현하는 영역이다.
- 영역 공학과 응용 공학의 연계를 위해 제품의 요구사항, 아키텍처, 조립 생산이 필요하다.

임베디드 소프트웨어(Embedded Software) : 임베디드 소프트웨어란 디지털 TV, 전기밥솥, 냉장고 등 해당 제품의 특정 기능에 맞게 특화 되어서 제품 자체에 있는 하드웨어(칩 등) 포함된 소프트웨어를 말한다.

소프트웨어 개발 방법론 활용 - SEC_01(소프트웨어 개발 방법론) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(소프트웨어 개발 방법론)

1. CBD(Component Based Design)에 대한 설명으로 틀린 것은?

- ① 개발 기간 단축으로 인한 생산성 향상
- ② 새로운 기능 추가가 쉬운 확장성
- ③ 소프트웨어 재사용이 가능
- ④ 1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법이다.

1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법은 구조적 방법론이다.

컴포넌트 기반(CBD; Component Based Design) 방법론

컴포넌트 기반 방법론은 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.

- ▶ 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.
- ▶ 새로운 기능을 추가하는 것이 간단하여 확장성이 보장된다.
- ▶ 유지 보수 비용을 최소화 하고 생산성 및 품질을 향상시킬 수 있다.
- ▶ 컴포넌트 기반 방법론의 절차

3. 소프트웨어 개발 방법론 중 CBD(Component Based Design)에 대한 설명으로 틀린 것은?

- ① 생산성과 품질을 높이고, 유지보수 비용을 최소화할 수 있다.
- ② 컴포넌트 제작 기법을 통해 재사용성을 향상시킨다.
- ③ 모듈의 분할과 정복에 의한 하향식 설계 방식이다.
- ④ 독립적인 컴포넌트 단위의 관리로 복잡성을 최소화할 수 있다.

분할과 정복(Divide and Conquer)은 구조적 방법론의 대표적인 특징이다.

4. 정보공학 방법론에서 데이터베이스 설계의 표현으로 사용하는 모델링 언어는?

- ① Package Diagram
- ② State Transition Diagram
- ③ Deployment Diagram
- ④ Entity-Relationship Diagram

패키지 다이어그램(Package Diagram), 배치 다이어그램(Deployment Diagram), 상태 전이도(State Transition Diagram)는 객체 지향 방법론의 모델링 언어이다.

정보공학 방법론

정보공학 방법론은 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축

소프트웨어 개발 방법론 활용 - SEC_01(소프트웨어 개발 방법론) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(소프트웨어 개발 방법론)

5. 개체를 기계의 부품처럼 하나의 객체로 만들어, 기계적인 부품들을 조립하여 제품을 만들 듯이 소프트웨어를 개발할 때에도 객체들을 조립해서 작성할 수 있도록 하는 소프트웨어 개발 방법론은?

- ① 컴포넌트 기반(CBD) 방법론
- ② 애자일(Agile) 방법론
- ③ 제품 계열 방법론
- ④ 객체지향 방법론

객체지향 방법론

객체지향 방법론은 **현실 세계의 개체(Entity)**를 기계의 부품처럼 하나의 **객체(Object)**로 만들어, 소프트웨어를 개발할 때도 기계의 부품을 조립하듯이 여러 개의 클래스를 만들어 필요한 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론이다.

▶ 객체지향 방법론은 구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 채택되었다.

▶ 설계 과정에서 주로 사용되는 모델링 언어에는 패키지 다이어그램(Package Diagram), 배치 다이어그램(Deployment Diagram), 상태 전이도(State Transition Diagram) 등이 있다.

7. 고객의 요구사항을 바로 바로 반영하고 상황에 따라 주어지는 문제를 풀어나가는 소프트웨어 개발 방법론은?

- ① 애자일(Agile) 방법론
- ② 컴포넌트 기반(CBD) 방법론
- ③ 객체지향 방법론
- ④ 구조적 방법론

애자일(Agile) 방법론

애자일은 사전적 의미로 '민첩한', '기민한'이라는 의미로, 애자일 방법론은 **고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론**이다.

▶ 소규모 프로젝트, 고도로 숙달된 개발자, 급변하는 요구사항에 적합하다.

▶ 애자일 방법론의 대표적인 종류에는 **익스트림 프로그래밍(XP; eXtreme Programming)**, 스크럼(Scrum), 칸반(Kanban), 크리스탈(Crystal) 등이 있다.

▶ 애자일 방법론의 절차

사용자 스토리(요구사항) -> 계획 -> 개발(반복) -> 승인 테스트

익스트림 프로그래밍(XP; eXtreme Programming) : 수시로 발생하는

소프트웨어 개발 방법론 활용 - SEC_01(소프트웨어 개발 방법론) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(소프트웨어 개발 방법론)

9. 다음 중 애자일 방법론의 종류에 해당하지 않는 것은?

- ① 익스트림 프로그래밍(eXtreme Programming)
- ② 스크럼(Scrum)
- ③ 크리스탈(Crystal)
- ④ 짝 프로그래밍(Pair Programming)

짝 프로그램(Pair Programming)은 하나의 컴퓨터로 두 사람이 함께 프로그래밍을 하는 것을 의미한다.

익스트림 프로그래밍(eXtreme Programming), 스크럼(Scrum), 크리스탈(Crystal), 칸반 방법론은 애자일 방법론의 종류이다.

10. CBD(Component Based Development) SW 개발 표준 산출물 중 분석 단계에 해당하는 것은?

- ① 클래스 설계서
- ② 통합시험 결과서
- ③ 프로그램 코드
- ④ 사용자 요구사항 정의서

CBD(컴포넌트 기반) 방법론에서의 분석 단계에서는 사용자 요구 사항 정의가 산출되고, 설계 단계의 산출물은 클래스 설계서이며,

11. 임베디드 소프트웨어를 만드는데 적합한 소프트웨어 방법론은?

- ① CBD(컴포넌트 기반) 방법론
- ② 객체지향 방법론
- ③ 제품 계열 방법론
- ④ 구조적 방법론

제품 계열 방법론

제품 계열 방법론은 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론이다.

▶ 임베디드 소프트웨어를 만드는데 적합하다.

▶ 제품 계열 방법론에는 영역 공학과 응용 공학으로 구분된다.

- 영역 공학 : 영역 분석, 영역 설계, 핵심 자산을 구현하는 영역이다.

- 응용 공학 : 제품 요구 분석, 제품 설계, 제품을 구현하는 영역이다.

▶ 영역 공학과 응용 공학의 연계를 위해 제품의 요구사항, 아키텍처, 조립 생산이 필요하다.

5. 소프트웨어 개발 방법론 활용-SEC_02(S/W 공학의 발전적 추세)

1) 소프트웨어 재사용의 개요

; 소프트웨어 재사용(Software Reuse)은 이미 개발되어 인정받은 소프트웨어의 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 사용하는 것이다.

- 소프트웨어 개발의 품질과 생산성을 높이기 위한 방법으로, 기존에 개발된 소프트웨어와 경험, 지식 등을 새로운 소프트웨어에 적용한다.
- 재사용의 이점
 - 개발 시간과 비용을 단축시킨다.
 - 소프트웨어 품질을 향상시킨다.
 - 소프트웨어 개발의 생산성을 향상시킨다.
 - 프로젝트 실패의 위험을 감소시킨다.
 - 시스템 구축 방법에 대한 지식을 공유하게 된다.
 - 시스템 명세, 설계, 코드 등 문서를 공유하게 된다.

5. 소프트웨어 개발 방법론 활용-SEC_02(S/W 공학의 발전적 추세)

2) 소프트웨어 재사용 방법

; 소프트웨어 재사용 방법에는 합성 중심 방법과 생성 중심 방법이 있다.

합성 중심 (Composition-Based)	전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법으로 블록 구성 방법이라고도 한다.
생성 중심 (Generation-Based)	추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법으로 패턴 구성 방법이라고도 한다.

5. 소프트웨어 개발 방법론 활용-SEC_02(S/W 공학의 발전적 추세)

3) 소프트웨어 재공학의 개요

; 소프트웨어 재공학(Software Reengineering)은 새로운 요구에 맞도록 기존 시스템을 이용하여 보다 나은 시스템을 구축하고, 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것이다.

- 유지보수 비용이 소프트웨어 개발 비용의 대부분을 차지하는 문제를 염두에 두어 기존 소프트웨어의 데이터와 기능들의 개조 및 개선을 통해 유지 보수성과 품질을 향상시키려는 기술이다.
- 기존 소프트웨어의 기능을 개조하거나 개선하므로, 예방(Preventive) 유지보수 측면에서 소프트웨어 위기를 해결하는 방법이라고 할 수 있다.
- 소프트웨어의 수명이 연장되고, 소프트웨어 기술이 향상될 뿐만 아니라 소프트웨어의 개발 기간도 단축된다.
- 소프트웨어에서 발생할 수 있는 오류가 줄어들고, 비용이 절감된다.

5. 소프트웨어 개발 방법론 활용-SEC_02(S/W 공학의 발전적 추세)

3) 소프트웨어 재공학의 개요

● 주요 활동

분석(Analysis)	기존 소프트웨어의 명세서를 확인하여 소프트웨어의 동작을 이해하고, 재공학할 대상을 선정하는 활동이다.
재구성 (Restructuring)	<ul style="list-style-type: none">•기존 소프트웨어의 구조를 향상시키기 위하여 코드를 재구성하는 활동이다.•소프트웨어의 기능과 외적인 동작은 바뀌지 않는다.
역공학(Reverse Engineering)	<ul style="list-style-type: none">•기존 소프트웨어를 분석하여 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 활동이다.•일반적인 개발 단계와는 반대 방향으로 기존 코드를 복구하거나, 기존 소프트웨어의 구성 요소와 그 관계를 파악하여 설계도를 추출한다.
이식(Migration)	기존 소프트웨어를 다른 운영체제 하드웨어 환경에서 사용할 수 있도록 변환하는 활동이다.

5. 소프트웨어 개발 방법론 활용-SEC_02(S/W 공학의 발전적 추세)

4) CASE의 개요

; CASE(Computer Aided Software Engineering)는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

- 객체지향 시스템, 구조적 시스템 등 다양한 시스템에서 활용되는 자동화 도구(CASE Tool)이다.
- CASE 도구를 통해 관리되는 공통 모듈을 사용할 수 있어 재사용성을 향상시킬 수 있다.
- CASE 도구가 모듈 관리를 자동으로 수행하므로 유지보수가 간편해진다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 및 품질 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구들을 지원하고, 개발자들은 이 도구를 사용하여 **소프트웨어 개발의 표준화**를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- CASE 도구는 **요구 분석, 설계** 과정을 지원하는 상위 CASE 도구와 **구현, 테스트** 과정을 지원하는 하위 CASE 도구로 구분할 수 있다.
- CASE의 주요 기능 : 소프트웨어 생명 주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원, 모델들의 모순 검사 및 오류검증, 자료흐름도 작성 등
- CASE의 원천 기술: 구조적 기법, 프로토타이핑, 자동 프로그래밍, 정보 저장소, 분산처리

자동화 도구(CASE Tool) : 자동화 도구는 소프트웨어 공학과 관련된 작업 중에서 하나의 작업을 자동화하는 패키지를 의미한다.

소프트웨어 개발 방법론 활용 - SEC_02(S/W 공학의 발전적 추세) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(S/W 공학의 발전적 추세)

1. 전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추는 방법으로 소프트웨어를 완성시키는 재사용 방법은?

- ① 합성 중심 ② 생성 중심
- ③ 분리 중심 ④ 구조 중심

소프트웨어 재사용 방법에는 합성 중심 방법과 생성 중심 방법 2가지가 존재한다.

1. 합성 중심(Composition-Based)

전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법으로 블록 구성 방법이라고도 한다.

2. 생성 중심(Generation-Based)

추상화 형태로 쓰여진 명세서를 구체화시켜서 프로그램을 만드는 방법으로 패턴 구성 방법이라고도 한다.

2. 소프트웨어 재공학이 소프트웨어의 재개발에 비해 갖는 장점으로 거리가 먼 것은?

- ① 위험 부담 감소 ② 비용 절감
- ③ 시스템 명세의 오류 억제 ④ 개발 시간의 증가

3. 소프트웨어 재공학의 주요 활동 중 기존 소프트웨어 시스템을 새로운 기술 또는 하드웨어 환경에서 사용할 수 있도록 변환하는 작업을 의미하는 것은?

- ① Analysis ② Migration
- ③ Restructuring ④ Reverse Engineering

소프트웨어 재공학의 주요 활동

분석(Analysis)

기존 소프트웨어의 명세서를 확인하여 소프트웨어의 동작을 이해하고, 재공학할 대상을 선정하는 활동을 분석 활동이라 한다.

재구성(Restructuring)

▶ 기존 소프트웨어의 구조를 향상시키기 위하여 코드를 재구성하는 활동이다.

▶ 소프트웨어의 기능과 외적인 동작은 바뀌지 않는다.

역공학(Reverse Engineering)

▶ 기존 소프트웨어를 분석하여 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 활동이다.

▶ 일반적인 개발 단계와는 반대 방향으로 기존 코드를 복구하거나,

소프트웨어 개발 방법론 활용 - SEC_02(S/W 공학의 발전적 추세) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(S/W 공학의 발전적 추세)

5. 소프트웨어를 재사용함으로써 얻을 수 있는 이점으로 가장 거리가 먼 것은?

- ① 생산성 증가
- ② 프로젝트 문서 공유
- ③ 소프트웨어 품질 향상
- ④ 새로운 개발 방법론 도입 용이

소프트웨어 재사용은 새로운 것이 대상이 아니고 기존의 검증을 받은 소프트웨어를 재사용해서 이점을 얻는다.

6. CASE(Computer-Aided Software Engineering)의 원천 기술이 아닌 것은?

- ① 구조적 기법
- ② 프로토타이핑 기술
- ③ 정보 저장소 기술
- ④ 일괄처리 기술

일괄 처리 기술은 CASE의 원천 기술과는 관계가 없다.

▶ CASE의 원천 기술 : 구조적 기법, 프로토타이핑, 자동 프로그래밍, 정보 저장소, 분산처리

7. 다음 중 상위 CASE 도구가 지원하는 주요 기능으로 볼 수 없는 것은?

- ① 모델들 사이의 모순검사 기능
- ② 전체 소스 코드 생성 기능
- ③ 모델의 오류검증 기능
- ④ 자료 흐름도 작성 기능

소스코드 작성을 지원하는 기능은 하위 CASE 도구이며, 하위 CASE 도구도 전체 소스 코드를 생성하지는 못한다.

▶ CASE 도구는 요구 분석, 설계 과정, 모델들 사이의 모순 검사 기능, 모델의 오류 검증 기능, 자료 흐름도 작성 기능을 지원하는 상위 CASE 도구와 구현, 테스트, 일부 소스 코드 작성을 지원하는 하위 CASE 도구로 구분할 수 있으며, 하위 CASE 도구도 전체 소스 코드를 생성하지는 못한다.

8. CASE(Computer Aided Software Engineering)에 대한 설명으로 틀린 것은?

- ① 소프트웨어 모듈의 재사용성이 향상된다.
- ② 자동화된 기법을 통해 소프트웨어 품질이 향상된다.
- ③ 소프트웨어 사용자들에게 사용 방법을 신속히 숙지시키기 위해 사용된다.

기출 및 출제 예상 문제(S/W 공학의 발전적 추세)

9. CASE(Computer-Aided Software Engineering) 도구에 대한 설명으로 거리가 먼 것은?

- ① 소프트웨어 개발 과정의 일부 또는 전체를 자동화하기 위한 도구이다.
 - ② 표준화된 개발 환경 구축 및 문서 자동화 기능을 제공한다.
 - ③ 작업 과정 및 데이터 공유를 통해 작업자간 커뮤니케이션을 증대한다.
 - ④ 2000년대 이후 소개되었으며, 객체지향 시스템에 한해 효과적으로 활용된다.
- CASE는 객체지향 시스템 뿐만 아니라 구조적 시스템 등 다양한 시스템에서 활용되는 자동화 도구이다.

5. 소프트웨어 개발 방법론 활용-SEC_03(비용 산정 기법 - 상향식)

1) 상향식 비용 산정 기법의 개요

- ; 상향식 비용 산정 기법은 프로젝트의 세부적인 작업 단위별로 비용을 산정한 후 집계하여 전체 비용을 산정하는 방법이다.
- 상향식 비용 산정 기법에는 LOC(원시 코드 라인 수, source Line Of Code) 기법, 개발 단계별 인월수 기법, 수학적 산정 기법 등이 있다.

인월(人月) : 한 사람의 1개월 간 작업량, 다른 말로 Man/Month, M/M, MM 로 쓴다.
대가를 계산하는 방식은 한 사람(man)이 하루 8시간 한 달(month) 근무하는 것을 기준으로 1M/M이면 1명을 한달 간 투입한 것이고, 2M/M이면 2명을 한달 또는 1명을 두 달간 투입한 것이 된다.

5. 소프트웨어 개발 방법론 활용-SEC_03(비용 산정 기법 - 상향식)

2) LOC(원시 코드 라인 수, source Line Of Code) 기법

; LOC 기법은 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.

- 측정이 용이하고 이해하기 쉬워 가장 많이 사용된다.
- 예측치를 이용하여 생산성, 노력, 개발 기간 등의 비용을 산정한다.

$$\text{예측치} = \frac{a+4m+b}{6} \quad \text{단, } a : \text{낙관치, } b : \text{비관치, } m : \text{기대치(중간치)}$$

● 산정 공식

- 노력(인월) = 개발 기간 X 투입 인원
= LOC / 1인당 월평균 생산 코드 라인 수
- 개발 비용 = 노력(인월) × 단위 비용(1인당 월평균 인건비)
- 개발 기간 = 노력(인월) / 투입 인원
- 생산성 = LOC / 노력(인월)

- 비관치 : 가장 많이 측정된 코드 라인 수
- 낙관치 : 가장 적게 측정된 코드 라인 수
- 기대치 : 측정된 모든 코드 라인 수의 평균

5. 소프트웨어 개발 방법론 활용-SEC_03(비용 산정 기법 - 상향식)

2) LOC(원시 코드 라인 수, source Line Of Code) 기법

예제1) LOC 기법에 의하여 예측된 총 라인 수가 30,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간은?

- 노력(인월) = $LOC / 1인당\ 월\ 평균\ 생산코드\ 라인\ 수 = 30,000/300 = 100명$
- 개발 기간 = $노력(인월) / 투입\ 인원 = 100 / 5 = 20개월$

예제2) 어떤 소프트웨어 개발을 위해 10명의 개발자가 20개월 동안 참여되었다. 그런데 그 중 7명은 20개월 동안 계속 참여했지만 3명은 3개월 동안만 부분적으로 참여했다. 이 소프트웨어 개발을 위한 인월(Man Month)은 얼마인가?

- 노력(인월) = $7명 * 20 = 140, 3명 * 3 = 9$
- 전체 노력(인월) = $140 + 9 = 149$

5. 소프트웨어 개발 방법론 활용-SEC_03(비용 산정 기법 - 상향식)

3) 개발 단계별 인원수(Effort Per Task) 기법

; 개발 단계별 인원수 기법은 LOC 기법을 보완하기 위한 기법으로, 각 기능을 구현시키는 데 필요한 노력을 생명 주기의 각 단계별로 산정한다.

- LOC 기법보다 더 정확하다.

소프트웨어 개발 방법론 활용 - SEC_03(비용 산정 기법 - 상향식) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(비용 산정 기법 - 상향식)

1. 두 명의 개발자가 5개월에 걸쳐 10,000 라인의 코드를 개발 하였을 때, 월별(Person Month) 생산성 측정을 위한 계산 방식 으로 가장 적합한 것은?

- ① $10,000 / 2$
- ② $10,000 / 5$
- ③ $10,000 / (5 * 2)$
- ④ $(2 * 10,000) / 5$

생산성 = 원시 코드 라인 수 / 노력(인월) 이다. 노력은 소프트웨어를 한달 간 개발하는데 소요되는 총 인원 또는 한 사람을 기준으로 몇 개월에 걸쳐 개발했느냐를 나타낸다. 이를 계산하는 방법은 투입 인원 * 개발 기간이 된다. 하여 $10,000 / (5 * 2)$ 가 된다.

2. 상향식 비용 산정 기법 중 LOC(원시 코드 라인 수) 기법에서 예측치를 구하기 위해 사용하는 항목이 아닌 것은?

- ① 낙관치 ② 기대치
- ③ 비관치 ④ 모형치

LOC 기법은 소프트웨어 각 기능의 원시 코드 라인 수의 비관치,

3. LOC 기법에 의하여 예측된 총 라인수가 50,000라인, 프로그래머의 월 평균 생산성이 200라인, 개발에 참여할 프로그래머가 10인일 때, 개발 소요 기간은?

- ① 25개월 ② 50개월
- ③ 200개월 ④ 2000개월

노력(인월) = 개발 기간 * 투입 인원,

LOC / 1인당 월 평균 생산 코드 라인 수

개발 비용 = 노력(인월) * 단위 비용(1인당 월 평균 인건비)

개발 기간 = 노력(인월) / 투입 인원

50,000라인을 10명이 개발하는데 한 사람이 한 달에 200라인을 생산 한다면 $50,000 / (10 * 200) = 25$ 즉, 개발 기간은 25개월이다.

4. 비용 산정 기법 중 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법은?

- ① Effort Per Task 기법
- ② 전문가 감정 기법
- ③ LOC 기법
- ④ 델파이 기법

소프트웨어 개발 방법론 활용 - SEC_03(비용 산정 기법 - 상향식) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(비용 산정 기법 - 상향식)

5. 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)은 얼마인가?

- ① 100 ② 70
- ③ 79 ④ 60

노력(인월) = 개발 기간 * 투입 인원,

LOC / 1인당 월 평균 생산 코드 라인 수

$$10 * 7 = 70, 3 * 3 = 9$$

위의 공식에 의거하여 노력(인월)을 구하면 전체 노력은 $70 + 9 = 79$ 가 된다.

6. COCOMO의 비용 산정에 의해 개발에 소요되는 노력이 40PM (Programmer-Month)으로 계산되었다. 개발에 소요되는 기간이 5개월이고, 1인당 인건비가 100만 원이라면 이 프로젝트에 소요되는 총 인건비는 얼마인가?

- ① 2억 원 ② 1억 원
- ③ 4천만 원 ④ 2천만 원

7. 원시 코드 라인 수(LOC) 기법에 의하여 예측된 총 라인 수가 30,000라인, 개발에 참여할 프로그래머가 5명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간은?

- ① 10개월 ② 15개월
- ③ 20개월 ④ 30개월

노력(인월) = LOC / 1인당 월 평균 생산 코드 라인 수 = $30,000 / 300 = 100$ 이 된다.

개발 기간 = 노력 / 투입 인원, $100 / 5 = 20$, 즉 개발 기간은 20개월이 소요된다.

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

1) 수학적 산정 기법의 개요

; 수학적 산정 기법은 상향식 비용 산정 기법으로, 경험적 추정 모형, 실험적 추정 모형 이라고도 하며, 개발 비용 산정의 자동화를 목표로 한다.

- 비용을 자동으로 산정하기 위해 사용되는 공식은 과거 유사한 프로젝트를 기반으로 하여 경험적으로 유도된 것이다.
- 수학적 산정 기법에는 COCOMO 모형, Putnam(푸트남) 모형, 기능 점수(FP) 모형 등이 있으며 각 모형에서는 지정된 공식을 사용하여 비용을 산정한다.

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

2) COCOMO 모형 개요

; COCOMO(CONstructive COst Model) 모형은 보헴(Boehm)이 제안한 것으로, 원시 프로그램의 규모인 LOC (원시 코드 라인 수에 의한 비용 산정 기법이다.

- 개발할 소프트웨어의 규모(LOC)를 예측한 후 이를 소프트웨어 종류에 따라 다르게 책정되는 비용 산정 방정식에 대입하여 비용을 산정한다.
- 비교적 작은 규모의 프로젝트들을 통계 분석한 결과를 반영한 모델이므로 중소 규모 소프트웨어 프로젝트 비용 추정에 적합하다.
- 같은 규모의 프로그램이라도 그 성격에 따라 비용이 다르게 산정된다.
- 비용산정 결과는 프로젝트를 완성하는 데 필요한 노력(Man-Month)으로 나타난다.

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

3) COCOMO의 소프트웨어 개발 유형

; 소프트웨어 개발 유형은 소프트웨어의 복잡도 혹은 원시 프로그램의 규모에 따라 조직형(Organic Mode), 반분리형(Semi-Detached Mode), 내장형(Embedded Mode)으로 분류할 수 있다.

● 조직형(Organic Mode)

조직형은 기관 내부에서 개발된 중·소 규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.

- 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합하다.
- 비용을 산정하는 공식은 다음과 같다.

$$\text{노력(MM)} = 2.4 \times (\text{KDSI})^{1.05}$$

$$\text{개발 기간(TDEV)} = 2.5 \times (\text{MM})^{0.38}$$

KDSI(Kilo Delivered Source Instruction) : 전체 라인 수를 1,000라인 단위로 묶은 것으로 KLOC(Kilo LOC)와 같은 의미이다.

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

3) COCOMO의 소프트웨어 개발 유형

● 반분리형(Semi-Detached Mode)

반분리형은 조직형과 내장형의 중간형으로 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.

- 컴파일러, 인터프리터와 같은 유틸리티 개발에 적합하다.
- 비용을 산정하는 공식은 다음과 같다.

$$\text{노력(MM)} = 3.0 \times (\text{KDSI})^{1.12}$$

$$\text{개발 기간(TDEV)} = 2.5 \times (\text{MM})^{0.35}$$

● 내장형(Embedded Mode)

내장형은 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI) 라인 이상의 소프트웨어를 개발하는 유형이다.

- 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합하다.
- 비용을 산정하는 공식은 다음과 같다.

$$\text{노력(MM)} = 3.6 \times (\text{KDSI})^{1.20}$$

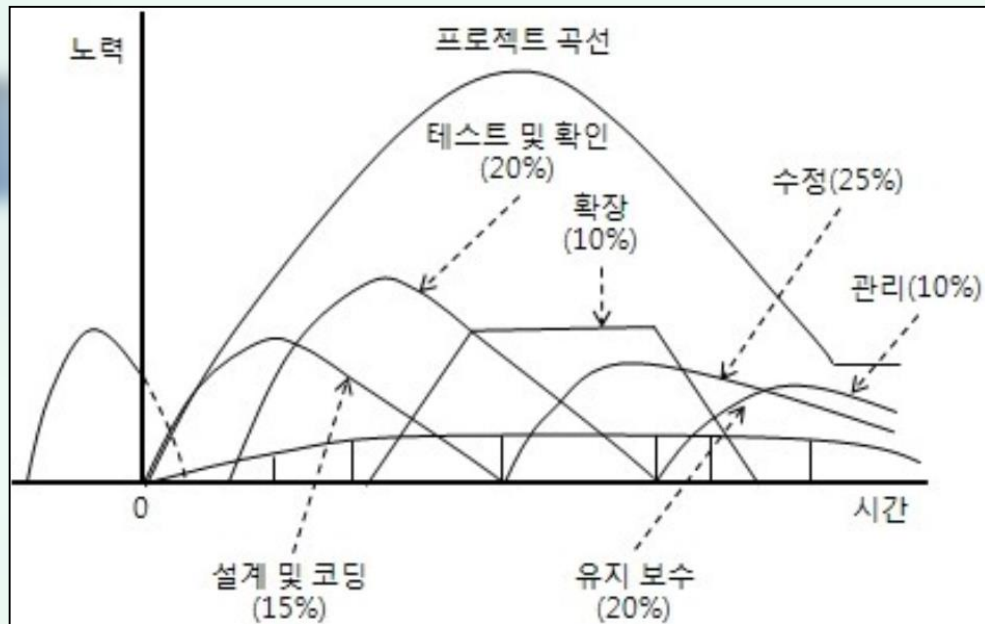
$$\text{개발 기간(TDEV)} = 2.5 \times (\text{MM})^{0.32}$$

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

4) Putnam 모형

; Putnam 모형은 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.

- 푸트남(Putnam)이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.



- 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

Rayleigh-Norden 곡선 : 노든(Norden)이 소프트웨어 개발에 관한 경험적 자료를 수집하여 이를 근거로 그린 곡선이다.

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

4) Putnam 모형

● 산정 공식

$$\text{개발 노력(MM)} = \frac{L^3}{C_k^3 \cdot Td^4}$$

- L : 원시 코드 라인 수
- Td : 개발 기간
- C_k : 환경 상수(빈약 환경 = 2,000, 좋은 환경 = 8,000, 최적 환경 = 12,000)

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

5) 기능 점수(FP) 모형

; 기능 점수(Function Point) 모형은 알브레히트(Albrecht)가 제안한 것으로, 소프트웨어의 기능을 증대시키는 요인 별로 가중치를 부여하고, 요인 별 가중치를 합산하여 총 기능 점수를 산출하며 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법이다.

$$\text{기능 점수(FP)} = \text{총 기능 점수} \times [0.65 + (0.1 \times \text{총 영향도})]$$

- 발표 초기에는 관심을 받지 못하였으나 최근에는 그 유용성과 간편성으로 비용 산정 기법 가운데 최선의 평가를 받고 있다.
- 기능별 가중치

소프트웨어 기능 증대 요인	가중치		
	단순	보통	복잡
자료 입력(입력 양식)	3	4	6
정보 출력(출력 보고서)	4	5	7
명령어(사용자 질의수)	3	4	5
데이터 파일	7	10	15
필요한 외부 루틴과의 인터페이스	5	7	10

5. 소프트웨어 개발 방법론 활용-SEC_04(수학적 산정 기법)

6) 자동화 추정 도구

; 비용 산정의 자동화를 위해 개발된 도구로는 SLIM과 ESTIMACS가 있습니다.

- SLIM(Putnam 모형) : Rayleigh-Norden 곡선과 Putnam 예측 모델을 기초로 하여 개발된 자동화 추정 도구이다.
- ESTIMACS(기능 점수 모형) : 다양한 프로젝트와 개인별 요소를 수용하도록 FP 모형을 기초로 하여 개발된 자동화 추정 도구이다.

소프트웨어 개발 방법론 활용 - SEC_04(수학적 산정 기법) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(수학적 산정 기법)

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

- ① COCOMO ② Putnam
- ③ Function-Point ④ PERT

수학적 산정 기법의 개요

수학적 산정 기법은 상향식 비용 산정 기법으로, 경험적 추정 모형, 실험적 추정 모형이라고 하며, 개발 비용 산정의 자동화를 목표로 한다.

▶ 비용을 자동으로 산정하기 위해 사용되는 공식은 과거 유사한 프로젝트를 기반으로 하여 경험적으로 유도된 것이다.

▶ 수학적 산정 기법에는 COCOMO 모형, Putnam 모형, 기능 점수(FP; Function Point) 모형 등이 있으며 각 모형에서는 지정된 공식을 사용하여 비용을 산정한다.

PERT는 프로젝트에 필요한 전체 작업의 상호 관계를 표시하는 네트워크로 일정 계획을 위한 도구이다.

2. COCOMO 모형의 프로젝트 유형으로 거리가 먼 것은?

- ① Organic ② Semi-Detached
- ③ Embedded ④ Sequential

3. COCOMO 모형에 의한 비용 산정에 대한 설명으로 옳지 않은 것은?

- ① 보헴이 제안한 원시 프로그램의 규모에 의한 비용 예측 모형이다.
- ② 같은 규모의 소프트웨어라도 그 유형에 따라 비용이 다르게 산정 된다.
- ③ 비용 산정 유형으로 Organic Mode, Embedded Mode, Semi-Detached Mode가 있다.
- ④ UFP(Unadjusted Function Point)를 계산한다.

미조정 기능 점수(UFP : Unadjusted Function Point) 계산

$$UFP = data\ FP + transaction\ FP$$

단순히 기능적인 요구 사항에 대해서만 계산을 하며, 여러 가지 특성을 고려를 전혀 하지 아니한다.

기능 점수(FP) 모형

기능 점수(Function Point) 모형은 알브레히트가 제안한 것으로, 소프트웨어의 기증을 증대시키는 요인 별로 가중치를 부여하고, 요인 별 가중치를 합산하여 총 기능 점수를 산출하며 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용하여 비용을 산정하는 기법이다.

▶ 발표 초기에는 관심을 받지 못하였으나 최근에는 그 유용성과 간편성으로 비용 산정 기법 가운데 최선의 평가를 받고 있다.

소프트웨어 개발 방법론 활용 - SEC_04(수학적 산정 기법) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(수학적 산정 기법)

5. Rayleigh-Norden 곡선의 노력 분포도를 이용한 프로젝트 비용 산정 기법은?

- ① Putnam 모형 ② 델파이 모형
- ③ COCOMO 모형 ④ 기능점수 모형

Rayleigh-Norden 곡선 : 노든(Norden)이 소프트웨어 개발에 관한 경험적 자료를 수집하여 이를 근거로 그린 곡선이다.

Putnam 모형

Putnam 모형은 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.

- ▶ 푸트남이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- ▶ 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.
- ▶ 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- ▶ 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

6. 기능 점수(Functional Point) 모형에서 비용 산정에 이용되는 요소가 아닌 것은?

- ① 클래스 인터페이스 ② 명령어(사용자 질의수)

7. Putnam 모형을 기초로 해서 만든 자동화 추정 도구는?

- ① SQLR/30 ② SLIM
- ③ MESH ④ NFV

자동화 추정 도구의 종류

비용 산정의 자동화를 위해 개발된 도구로는 SLIM과 ESTIMACS가 있다.

▶ **SLIM(Putnam 모형)** : Rayleigh-Norden 곡선과 Putnam 예측 모델을 기초로 하여 개발된 자동화 추정 도구이다.

▶ **ESTIMACS(기능 점수 모형)** : 다양한 프로젝트와 개인별 요소를 수용 하도록 FP 모형을 기초로 하여 개발된 자동화 추정 도구이다.

8. 소프트웨어 비용 산정 기법 중 개발 유형으로 Organic, Semi-Detached, Embedded로 구분되는 것은?

- ① PUTNAM ② COCOMO
- ③ FP ④ SLIM

개발 유형이나 규모에 따라 여러 프로젝트 유형이 제시되는 모델은 COCOMO 이다.

소프트웨어 개발 방법론 활용 - SEC_04(수학적 산정 기법) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(수학적 산정 기법)

9. 다음 중 소프트웨어 비용 산정에 이용되는 기법이 아닌 것은?

- ① COCOMO(CONstructive COst MOdel) 방정식
- ② 기능 점수(Function Point) 모형
- ③ 홀스테드 노력 방정식(Halstead Effort Equation)
- ④ 전문가의 감정과 델파이 기법

전문가의 감정과 델파이 기법은 소프트웨어 비용 산정 방법 중 하향식 방법이다.

홀스테드 노력 방정식은 코드 생성 후 피연산자 연산자 수를 더하여 코드의 복잡도를 측정하는데 사용되는 평가 척도 중의 하나의 기법이다.

10. 비용 예측을 위한 기능 점수 방법에 대한 설명 중 가장 옳지 않은 것은?

- ① 입력, 출력, 질의, 파일, 인터페이스의 개수로 소프트웨어의 규모를 표현한다.
- ② 기능 점수는 원시코드의 구현에 이용되는 프로그래밍 언어에 종속적이다.
- ③ 경험을 바탕으로 단순, 보통, 복잡한 정도에 따라 가중치를

11. COCOMO 모델에 의한 비용(Cost) 산정 과정에 해당하지 않는 것은?

- ① KDSI(or KLOC)를 측정한다.
- ② UFP(Unadjusted Function Point)를 계산한다.
- ③ 개발 노력 승수(Development Effort Multipliers)를 결정한다.
- ④ 비용 산정 유형으로 조직형, 반분리형, 내장형이 있다.

UFP(Unadjusted Function Point)를 계산하는 것은 기능 점수(Function Point) 모델의 기능이다.

5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)

1) 개요

; 프로젝트 일정(Scheduling) 계획은 프로젝트의 프로세스를 이루는 소작업을 파악하고 예측된 노력을 각 소작업에 분배하며, 소작업의 순서와 일정을 정하는 것이다.

- 소프트웨어 개발 기간의 지연을 방지하고 프로젝트가 계획대로 진행되도록 일정을 계획한다.
- **계획된 일정은 프로젝트의 진행을 관리하는 데 기초 자료가 된다.**
- 계획된 일정과 프로젝트의 진행도를 비교하여 차질이 있을 경우 여러 조치를 통해 조정할 수 있다.
- **프로젝트 일정 계획을 위해서는 WBS, PERT/CPM, 간트 차트 등이 사용된다.**

WBS(Work Breakdown Structure, 업무 분류 구조) : WBS는 개발 프로젝트를 여러 개의 작은 관리 단위(소작업)로 분할하여 계층적으로 기술한 업무 구조이다.

PERT/CPM : PERT/CPM 네트워크는 프로젝트의 지연을 방지하고 계획대로 진행되게 하기 위한 일정을 계획하는 것으로 대단위 계획의 조직적인 추진을 위해 자원의 제약하에 비용을 적게 사용하면서 초단시간 내 계획 완성을 위한 프로젝트 일정 방법이다.

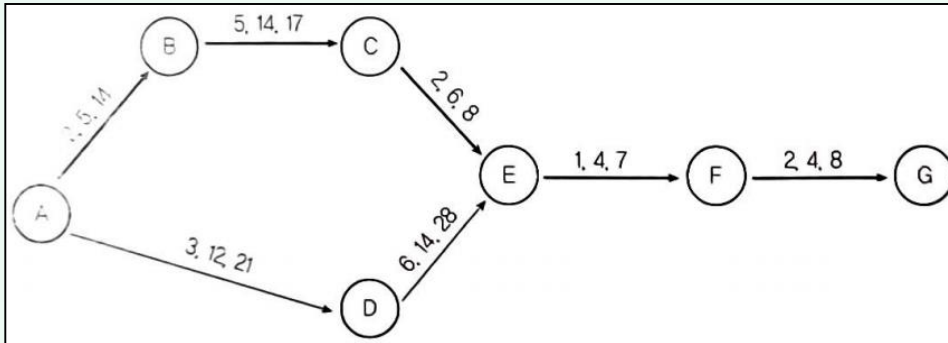
5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)

2) PERT(Program Evaluation and Review Technique, 프로그램 평가 및 검토 기술)

; PERT는 프로젝트에 필요한 전체 작업의 상호 관계를 표시하는 네트워크로 각 작업 별로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료 시기를 결정하는 방법이다.

- 과거에 경험이 없어서 소요 기간 예측이 어려운 소프트웨어에서 사용한다.
- 노드와 간선으로 구성되며 원 노드에는 작업을, 간선(화살표)에는 낙관치, 기대치, 비관치를 표시한다.
- 결정 경로, 작업에 대한 경계 시간, 작업 간의 상호 관련성 등을 알 수 있다.
- 다음과 같은 PERT 공식을 이용하여 작업 예측치를 계산한다.

$$\text{작업 예측치} = \frac{\text{비관치} + 4 \times \text{기대치} + \text{낙관치}}{6} \quad \text{평방 편차} = \left[\frac{(\text{비관치} - \text{낙관치})}{6} \right]^2$$



결정 경로 : A-B-C-E-F-G 순서로 진행된다.

작업에 대한 경계 시간 : 각 작업간의 경계 시간을 알 수 있다.

상호 관련성 : C는 B 작업이 진행된 후에 수행된다는 것을 알 수 있다.

•**낙관치 :** 모든 상황이 좋아서 최대한 빨리 진행될 때 걸리는 시간

•**기대치 :** 모든 상황이 정상적으로 진행될 때 걸리는 시간

•**비관치 :** 모든 상황에 많은 장애가 생겨서 가장 늦게 진행될 때 걸리는 시간

5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)

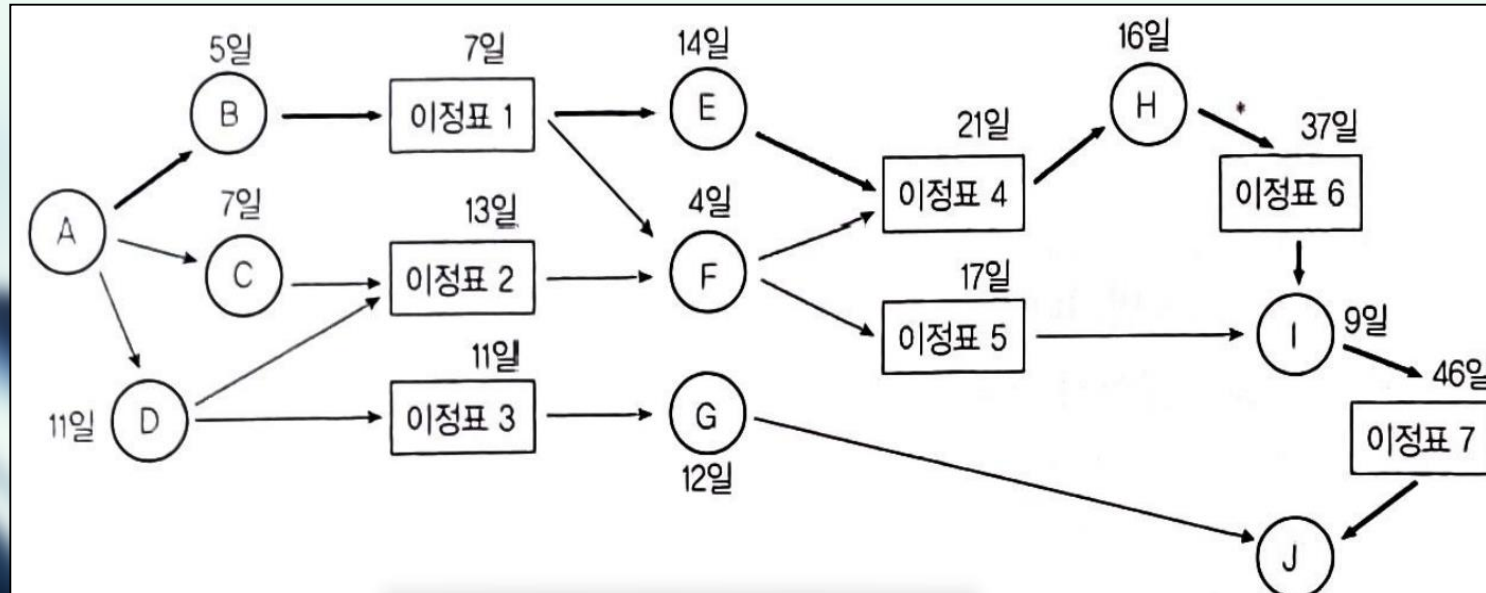
3) CPM(Critical Path Method, 임계 경로 기법)

; CPM은 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는데 사용하는 기법이다.

- CPM은 노드와 간선으로 구성된 네트워크로 노드는 작업을, 간선은 작업 사이의 전후 의존 관계를 나타낸다.
- 원형 노드는 각 작업을 의미하며 각 작업 이름과 소요 기간을 표시하고, 박스 노드는 이정표를 의미하며 박스 노드 위에는 예상 완료 시간을 표시한다.
- 간선을 나타내는 화살표의 흐름에 따라 각 작업이 진행되며, 전 작업이 완료된 후 다음 작업을 진행할 수 있다
- 각 작업의 순서와 의존 관계, 어느 작업이 동시에 수행될 수 있는지를 한눈에 볼 수 있다.
- 경영층의 과학적인 의사 결정을 지원하며, 효과적인 프로젝트의 통제를 가능하게 해 준다.
- 병행 작업이 가능하도록 계획할 수 있으며, 이를 위한 자원 할당도 가능하다.
- 임계 경로는 최장 경로를 의미한다.

5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)

3) CPM(Critical Path Method, 임계 경로 기법)



굵은 선은 임계 경로, 즉 최장 경로를 의미한다.

일정 계획의 순서

CPM 네트워크를 사용한 일정 계획 순서는 다음과 같다.

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

5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)











4) 간트 차트(Gantt Chart)

; 간트 차트는 프로젝트의 각 작업들이 언제 시작하고 언제 종료되는지에 대한 작업 일정을 막대 도표를 이용하여 표시하는 프로젝트 일정표로, 시간선(Time-Line) 차트 라고도 한다.

- 중간 목표 미달성 시 그 이유와 기간을 예측할 수 있게 한다.
- 사용자와의 문제점이나 예산의 초과 지출 등도 관리할 수 있게 한다.
- 자원 배치와 인원 계획에 유용하게 사용된다.
- CPM 네트워크의 데이터를 바탕으로 간트 차트를 제작할 수 있다.
- 작업 경로는 표시할 수 없으며, 계획의 변화에 대한 적응성이 약하다.
- 계획 수립 또는 수정 때 주관적 수치에 기울어지기 쉽다.
- 간트 차트는 일정표, 작업 일정, 작업 기간, 산출물로 구성되어 있다.
- 수평 막대의 길이는 각 작업(Task)의 기간을 나타낸다.

5. 소프트웨어 개발 방법론 활용-SEC_05(프로젝트 일정 계획)

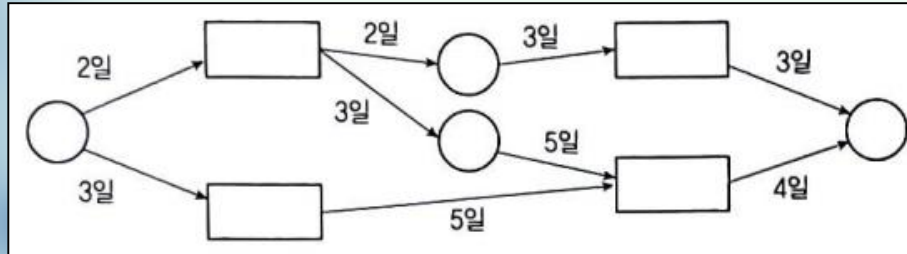
4) 간트 차트(Gantt Chart)

작업 일정		이정표												산출물
작업 단계		1	2	3	4	5	6	7	8	9	10	11	12	
계획														시스템 계획서 프로젝트 정의서
분석														요구 분석 명세서
기본 설계														기본 설계서
상세 설계														상세 설계서
사용자 지침서														사용자 지침서
시험 계획														시험 계획서
구현														원시 코드
통합 테스트														시스템 통합 계획서
시스템 테스트														시스템
인수 테스트														개발 완료 보고서

소프트웨어 개발 방법론 활용 - SEC_05(프로젝트 일정 계획) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(프로젝트 일정 계획)

1. CPM 네트워크가 다음과 같을 때 임계 경로의 소요 기일은?



- ① 10일 ② 12일
- ③ 14일 ④ 16일

임계 경로는 최장 경로를 의미한다.

경로 1 : $2 + 2 + 3 + 3 = 10$ 일

경로 2 : $2 + 3 + 5 + 4 = 14$ 일

경로 3 : $3 + 5 + 4 = 12$ 일

경로 2가 임계 경로가 되며 소요 기일은 14일이 된다.

2. 프로젝트 일정 관리 시 사용하는 PERT 차트에 대한 설명에 해당하는 것은?

- ① 각 작업들이 언제 시작하고 언제 종료되는지에 대한 일정을 막대 도표를 이용하여 표시한다.(간트 차트)
- ② 시간선(Time-Line) 차트라고도 한다.(간트 차트)

3. 간트 차트(Gantt Chart)에 대한 설명으로 틀린 것은?

- ① 프로젝트를 이루는 소작업 별로 언제 시작되고 언제 끝나야 하는지를 한 눈에 볼 수 있도록 도와준다.
- ② 자원 배치 계획에 유용하게 사용된다.
- ③ CPM 네트워크로부터 만드는 것이 가능하다.
- ④ 수평 막대의 길이는 각 작업(Task)에 필요한 인원수를 나타낸다.

수평 막대의 길이는 각 작업의 기간을 나타낸다.

간트 차트(Gantt Chart)

간트 차트는 프로젝트의 각 작업들이 언제 시작하고 언제 종료되는지에 대한 작업 일정을 막대 도표를 이용하여 표시하는 프로젝트 일정표로, 시간선(Time-Line) 차트라고도 한다.

- ▶ 중간 목표 미달성 시 그 이유와 기간을 예측할 수 있게 한다.
- ▶ 사용자와의 문제점이나 예산의 초과 지출 등도 관리할 수 있게 한다.
- ▶ 자원 배치와 인원 계획에 유용하게 사용된다.
- ▶ CPM 네트워크의 데이터를 바탕으로 간트 차트를 제작할 수 있다.
- ▶ 작업 경로는 표시할 수 없으며, 계획의 변화에 대한 적응성이 약하다.
- ▶ 계획 수립 또는 수정 때 주관적 수치에 기울어지기가 쉽다.
- ▶ 간트 차트는 일정표, 작업 일정, 작업 기간, 산출물로 구성되어 있다.

기출 및 출제 예상 문제(프로젝트 일정 계획)

5. 프로젝트 일정(Scheduling) 계획에 대한 설명으로 틀린 것은? ①

노력을 각 소작업에 분배한다.

② 소프트웨어 개발 기간의 지연을 방지한다.

③ 계획된 일정은 프로젝트의 진행을 관리하는데 기초 자료가 된다.

④ 프로젝트 일정 계획을 위해서는 간트 차트 만이 사용된다.

프로젝트 일정(Scheduling) 계획 은 프로젝트의 프로세스를 이루는 소작업을 파악하고 예측된 노력을 각 소작업에 분배하며, 소작업의 순서와 일정을 정하는 것이다.

▶ 소프트웨어 개발 기간의 지연을 방지하고 프로젝트가 계획대로 진행되도록 일정을 계획한다.

▶ 계획된 일정은 프로젝트의 진행을 관리하는데 기초 자료가 된다.

▶ 계획된 일정과 프로젝트의 진행도를 비교하여 차질이 있을 경우 여러 조치를 통해서 조정할 수 있다.

▶ 프로젝트 일정 계획을 위해서는 **WBS, PERT, CPM, 간트 차트** 등이 사용된다.

WBS(Work Breakdown Structure, 업무 분류 구조) : WBS는 개발 프로젝트를 여러 개의 작은 관리 단위(소작업)로 분할하여 계층적



감사합니다.