

1. 소프트웨어 재사용



1) 특징

- 이미 개발된 소프트웨어 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 이용
- 이미 만들어진 프로그램을 사용
- 재사용 부품의 크기가 작을수록 재사용률이 높음
- 클래스, 객체 등의 소프트웨어 요소는 소프트웨어 재사용성을 크게 향상시킴
- 소프트웨어 재사용에 가장 많이 이용되는 것 : 소스 코드
- 재사용 컴포넌트(Component)들이 많이 있어도 그들을 찾아내는 것이 어려운 경우 : 분류의 문제

2) 이점

- 개발시간과 비용의 단축
- 소프트웨어 품질 향상
- 생산성 향상
- 시스템 구축방법에 대한 지식 공유
- 시스템 명세, 설계, 코드 등의 문서 공유
- 소프트웨어 프로그래밍 언어의 종속 (X)
- 고급 프로그래머 배출이 용이 (X)
- 새로운 개발 방법론의 도입이 쉽다. (X)
- 사용자의 책임과 권한부여 (X)
- 표준화의 원칙을 무시할 수 있다. (X)
- 응용소프트웨어의 보안 유지 (X)

2. 소프트웨어 재공학 (Re-engineering) ★★★★★

1) 특징

- 기존 시스템을 이용하여 보다 나은 시스템을 구축하고 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것
- 기존에 있던 소프트웨어를 파기하지 않고 변경된 사용자의 요구사항이나 수정된 환경으로 기존 소프트웨어를 수정 보완하여 재구축
- 소프트웨어의 위기를 해결하기 위해 개발의 생산성이 아닌 유지보수의 생산성으로 해결하려는 방법
- 목적 : 유지보수성 용이성 향상 (주된 목적), 생산성, 품질의 향상
- 재공학 활동 : 분석, 재구성, 역공학, 이식 등이 있음
- 재공학 대상 : 자사 소프트웨어뿐만 아니라 모든 소프트웨어
- CASE(자동화 도구)의 정보 저장소와는 무관함 (X)

3. 소프트웨어 역공학 (reverse engineering) ★☆☆☆☆

- 소프트웨어를 분석하여 소프트웨어 개발 과정과 데이터 처리 과정을 설명하는 분석 및 설계 정보를 재발견하거나 다시 만들어 내는 작업
- 현재 프로그램으로부터 데이터, 아키텍처, 그리고 절차에 관한 분석 및 설계 정보를 추출하는 과정

4. 클라이언트/서버 시스템 ★☆☆☆☆

- : 분산 시스템의 가장 대표적인 모델 (클라이언트:서비스 요구, 서버:서비스 제공)
- 사용자의 요구사항은 서버의 데이터베이스 시스템에 영향을 미친다
 - 병목현상을 없애기 위해 비즈니스 로직을 분리하여 관리할 수 있다
 - 대부분 네트워크로 연결되어 있고 인증 작업을 필요로 한다.
 - 소프트웨어 요소 : 애플리케이션, 데이터베이스, 프리젠테이션/상호작용
 - **미들웨어의 사용은 서버와 클라이언트의 작업량을 증가시켰다. (X)**

* 미들웨어 : 분산 컴퓨팅 환경에서 서로 다른 기종의 하드웨어나 프로토콜, 통신환경 등을 연결하여, 응용프로그램과 그 프로그램이 운영되는 환경 간에 원만한 통신이 이루어질 수 있게 하는 소프트웨어를 말한다.

5. CASE (Computer Aided Software Engineering) ★

: 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정을 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 작업

1) 특징

- 소프트웨어 생명 주기의 전체 단계를 연결시켜 주고 자동화시켜 주는 통합된 도구를 제공해 주는 기술
- 개발 비용 절약, 품질 향상, 개발 기간 감소, 유지보수 간편
- 모듈의 **재사용성이 향상**
- 도구와 방법론의 결합
- 개발 단계의 표준화
- **CASE는 아직 표준화 되지 않아서 CASE 툴들은 호환이 안됨**
- **CASE의 정보 저장소** : 개발 과정 동안에 모아진 정보를 보관하여 관리하는 곳 -> 유지보수 용이
- > 시스템 표준화와 소프트웨어 재사용성의 기본
- **개발 방법론 생성 (X)**
- **설계지식이 없을 때 CASE 를 사용하면 효과적이다 (X)**

2) 분류

- 상위 CASE : 요구분석과 설계 단계 지원
- 하위 CASE : 코드 작성하고 테스트하며 문서화하는 과정 지원
- 통합 CASE : 개발 주기 전체 과정 지원

[SE 5강]-발전적 주제

1. 소프트웨어 재사용에 대한 설명으로 옳은 것은?
 - 가. 기계 중심 언어는 주기억장치를 매우 효과적으로 사용할 수 있고, 실행 속도를 최적화시킬 수 있다는 점에서 재사용성이 가장 뛰어난 프로그래밍 언어이다.
 - 나. 1990년대의 클래스, 객체 등의 소프트웨어 요소는 소프트웨어 재사용성을 크게 향상시켰다.
 - 다. 노호된 시스템에 대한 재분석, 문서화 작업을 통해 공학적으로 우수한 시스템을 만드는 것을 의미한다.
 - 라. 소프트웨어 재사용은 경제성을 고려하여 사용자의 요구가 있을 때만 적용하는 것이 바람직하다.
2. 소프트웨어 재사용의 이점에 속하지 않는 것은?
 - 가. 소프트웨어 품질 향상
 - 나. 소프트웨어의 개발 시간과 비용감소
 - 다. 소프트웨어 생산성 증가
 - 라. 소프트웨어 프로그래밍 언어의 종속
3. 소프트웨어 재사용을 통한 장점이라고 볼 수 없는 것은?
 - 가. 개발 시간과 비용을 감소시킨다.
 - 나. 소프트웨어 품질을 향상시킨다.
 - 다. 생산성을 증가시킨다.
 - 라. 고급 프로그래머 배출이 용이하다.
4. 소프트웨어 재공학에 관한 설명으로 옳지 않은 것은?
 - 가. 소프트웨어 재공학은 CASE의 정보 저장소와는 무관하다.
 - 나. 재공학 활동은 분석, 재구성, 역공학, 이식 활동 등으로 구분할 수 있다.
 - 다. 소프트웨어 재공학도 자동화된 도구를 사용하여 소프트웨어를 분석하고 수정하는 과정을 포함한다.
 - 라. 소프트웨어 재공학의 일반적인 개념은 데이터와 기능들의 개조 및 개선을 가해 유지보수 용이성을 향상시키자는 것이다.

[정답] 1.나 2.라 3.라 4.가



[SE 5강]-발전적 주제

5. 소프트웨어의 위기를 개발의 생산성이 아닌 유지보수의 생산성으로 해결 하려는 방법을 의미하는 것은?
 - 가. 소프트웨어 재사용
 - 나. 소프트웨어 재공학
 - 다. 클라이언트 / 서버 소프트웨어 공학
 - 라. 전통적 소프트웨어공학
6. 소프트웨어 리엔지니어링의 목표 중 거리가 먼 것은?
 - 가. 복잡한 시스템을 다루는 방법 구현
 - 나. 다른 뷰의 생성
 - 다. 기존 시스템의 해킹
 - 라. 잃어버린 정보의 복구 및 제거
7. 현재 프로그램으로부터 데이터, 아키텍처, 그리고 절차에 관한 분석 및 설계 정보를 추출하는 과정은?
 - 가. 재공학(Reengineering)
 - 나. 역공학(Reverse Engineering)
 - 다. 순공학(Forward Engineering)
 - 라. 재사용(Reuse)
8. 다음 중 클라이언트 / 서버 시스템을 위한 소프트웨어 요소에 해당되지 않는 것은?
 - 가. 객체지향 요소
 - 나. 애플리케이션 요소
 - 다. 데이터베이스 요소
 - 라. 프리젠테이션 / 상호작용 요소
9. 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정을 컴퓨터와 전용의 소프트웨어 도구를 사용하여 자동화하는 작업을 무엇이라고 하는가?
 - 가. CAT(Computer Aided Testing)
 - 나. CAD/CAM(Computer Aided Design and Manufacturing)
 - 다. CASE(Computer Aided Software Engineering)
 - 라. CAI(Computer Aided Instruction)
10. CASE에 관한 설명으로 옳지 않은 것은?
 - 가. 소프트웨어 개발비용을 절감할 수 있다.
 - 나. 자동화된 검사를 통해 소프트웨어 품질을 향상시킨다.
 - 다. 모듈의 수가 증가하므로 개발 기간이 늘어난다.
 - 라. 프로그램의 유지보수를 간편하게 한다.

[정답] 5.나 6.다 7.나 8.가 9.다 10.다



[SE 5강]-발전적 주제

11. CASE(Computer Aided Software Engineering) 툴(Tool)의 장점으로 거리가 먼 것은?

- 가. 자동화 기법을 통해 소프트웨어 품질이 향상된다.
- 나. 개발 기간이 짧다.
- 다. CASE표준이 있어서 대부분의 CASE툴들은 호환된다.
- 라. 개발비용이 절감된다.

12. CASE 사용에 관한 설명으로 옳지 않은 것은?

- 가. 소프트웨어 부품의 재사용성이 낮아진다.
- 나. 소프트웨어 개발 기간을 단축할 수 있다.
- 다. 유지보수를 간단하고 간편하게 수행할 수 있다.
- 라. 자동화 기법을 통하여 소프트웨어 품질이 향상된다.

13. 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 정보를 체계적으로 데이터베이스화한다.
- 다. 타 기업의 시스템에 몰래 접속하여 새로운 소프트웨어 개발에 관한 정보를 획득한다.
- 라. 바이러스에 대한 예방에 만전을 기하여 시스템의 안전을 확보한다.

[정답] 11.다 12.가 13.다