2019년 7월 22일

소프트웨어 주요 암기 사항 1

@ 소프트웨어 생명 주기 모형(SDLC; Software Development Life Cycle Model)

종류 및 순서

1. 폭포수 모형(Waterfall Model) : 소프트웨어 개발 각 단계는 검토 및 승인을 통해 다음 단계 진행하는 선형 순차적 모형, 개발 순서 : 타당성 검토 -> 계획 -> 요구분석 -> 설계 -> 구현(코딩) -> 시험(검사) -> 유지보수

2. 프로토타입 모형(Prototye Model) : 사용자 요구분석 단계에서 사항의 정확한 파악 위해 견본(시제)품(Prototype)을 만들어 평가 및 승인 완료되면 개발 진행, 개발순서 : 요구 수집 -> 빠른 설계 -> 프로토타입 구축 -> 고객평가 -> 프로토타입 조정 -> 구현

3.나선형 모형(Spiral Model) : 폭포수 모형과 프로타입 모형 장점에 위험 분석 기능 추가, 점진적으로 프로토타입을 발전시켜 완벽한 소프트웨어 개발, 개발순서 : 계획 및 정의(Planning) -> 위험 분석(Risk Analysis) -> 공학적 개발(Engineering) -> 고객 평가(Customer Evaluation)

소프트웨어공학 주요 암기사항 2

@ 비용산정기법 - LOC(Line Of Code) 기법 : 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법

- 예측치 = (a + 4m + b) / 6 (a : 낙관치, b : 비관치, m : 중간치(기대치))

- 노력(인월) : 개발기간(월) x 투입인원, LOC / 1인당 월평균 생산 코드 라인 수

- 개발비용 : 노력(인월) x 단위 비용(1일단 월평균 인건비)

- 개발 기간 : 노력(인월) / 투입인원

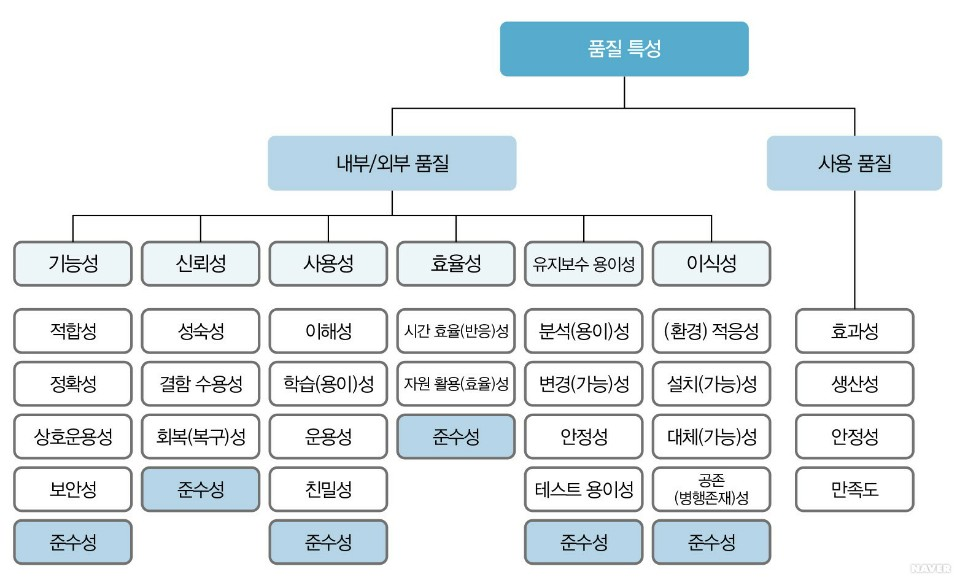
- 생산성 : LOC / 노력(인월)

소프트웨어공학 주요 암기사항3

@ 소프트웨어 제품 품질표준(ISO9126)

기,신,사,효,유,이

기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성



소프트웨어공학 주요 암기사항4

@ 결합도(Coupling) : 모듈 간에 상호 의존하는 정도, 독립적인 모듈이 되기 위해서 각 모듈 간의 결합도가 약해야 하며 의존하는 모듈이 적어야 함.

- 결합도의 종류(자, 스, 제, 외, 공, 내) : 약함(좋음)-자료(Data) < 스탬프(Stamp) < 제어(Control) < 외부(External) < 공통(Common) < 내용(Content)-강함(나쁨)

@ 응집도(Cohesion) : 모듈 안의 요소들이 서로 관련되어 있는 정도, 모듈이 독립적인 기능으로 정의되어 있는 정도, 독립적인 모듈이 되기 위해서는 각 모듈의 응집도가 강해야 함.

- 응집도의 종류(우, 논, 시, 절, 통, 순, 기) : 약함(나쁨)-우연적(Coincidental) < 논리적(Logical) < 시간적(Temporal) < 절차적(Procedural) < 교환(통신)적(Communicational) < 순차적(Sequential) < 기능적(Functional) - 강함(좋음)

--------------------------------------------------------------------------------------------------------------

소프트웨어공학 주요 암기사항 5

@ 소프트웨어 검사(test) 단계(단통시인설)

- 단위 검사(Unit Test) : 모듈 시험, White Box Test(기초 경로, 조건, 루프, 데이터 흐름 검사)

- 통합 검사(Integration Test) : 모듈간 인터페이스 검사, Black Box Test(동치 분할, 경계값 분석, 원인효과 그래프, 오류 예측, 비교 검사), 하향식/상향식 통합

- 시스템 검사(System Test) : 전체 시스템의 기능 테스트, 복구/보안/강도/성능 검사

- 인수(확인), 검증 검사(Validation Test) : 사용자 만족(요구사항 만족도 평가), 형상검사(구성 검토, 감사), 알파 검사, 베타 검사

- 설치 검사 : 사용자 환경의 배치/설치 검사

소프트웨어공학 주요 암기사항 6

@ 객체지향 기법 주요 원칙(캡추다정상)

- 캡슐화(Encapsulation) : 데이터와 함수를 클래스로 묶음, 객체의 세부 내용이 외부에 은폐, 변경 발생 오류 파급 효과 적음, 재사용 용이, 인터페이스 단순, 객체 간 결함도 낮아짐

- 추상화(Abstraction) : 불필요한 부분 생략, 객체 속성 중 가장 중요한 것만 개략화/모델화

- 다형성(Polymorphism) : 한 메시지가 객체에 따라 다른 방법으로 응답할 수 있도록 설계, 상속받은 여러 개의 하위 객체들이 다른 형태 특성을 갖는 객체로 이용 가능한 성질

- 정보은닉(Information Hiding) : 다른 객체에게 자신의 정보를 숨기고 연산만을 통하여 접근 허용, 객체의 수정이 다른 객체에게 주는 영향 최소화, 유지보수와 소프트웨어 확장 시 오류 최소화

- 상속성(Inheritance) : 이미 정의된 상위 클래스(슈퍼 클래스나 부모 클래스)의 모든 성과 연산을 하위 클래스가 물려 받는 것, 하위 클래스는 상위 클래스의 모든 속성과 연산을 자신의 클래스 내에서 다시 정의하지 않고서도 즉시 자신의 속성으로 사용 가능

소프트웨어공학 주요 암기사항 7

@ 소프트웨어 개발 유형

- 조직형(Organic Mode) : 기관 내부에서 개발된 5만(50KDSI)라인 이하의 중소규모 소프트웨어, 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합

- 반분리형(Semi-Detached Mode) : 조직형과 내장형의 중간형, 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등 30만(300KDSI) 라인 이하의 소프트웨어 개발 유형, 컴파일러/인터프리터와 같은 유틸리티 개발에 적합

- 내장형(Embedded Mode) : 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI) 라인 이상의 소프트웨어 개발 유형, 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발 적합

소프트웨어공학 주요 암기사항 8

@ 브룩스(Brooks)의 법칙 : 프로젝트 진행 중에 새로운 인력을 투입할 경우 작업 적응 기간과 기존 인력과의 의사소통 수 증가 등 부작용으로 인해 일정 더욱 지연시키고 프로젝트 혼란을 가져온다는 법칙

소프트웨어공학 주요 암기사항 9

@ 정형 기술 검토(FTR; Formal Technical Review) : 소프트웨어 기술자들에 의해 수행되는 소프트웨어 품질 보증 활동

# 유형 : 검토 회의(Walkthrough), 검열(Inspections) 등 모두 회의 형태

# 목적 : 검토 중인 소프트웨어가 해당 요구사항과 일치하는 지 검증, 미리 정해진 표준에 따라 표현 되고 있는지 확인, 기능과 로직의 오류 확인, 균일한 방식 개발 유도, 프로젝트의 용이한 관리 유도

# 지침 사항

- 제품의 검토에만 집중하라

- 의제를 제한하여 진행하라

- 논쟁과 반박을 제한하라

- 문제 영역을 명확히 표현하라

- 해결책이나 개선책에 대해서 논하지 말아라

- 참가자의 수를 제한, 사전 준비를 강요하라

- 검토될 확율이 있는 각 제품에 대한 체크 리스트를 개발하라

- 자원과 시간 일정을 할당하라

- 모든 검토자들을 위해 의미 있는 훈련을 행하라

데이터베이스 주요 암기사항1

@ 데이터베이스의 정의(통,저,운,공/I,S,O,S)

- 통합된 데이터(Integrated Data) : 자료의 중복 배제 데이터 모임

- 저장된 데이터(Stored Data) : 컴퓨터 저장 매체에 저장된 자료

- 운영 데이터(Operataional Data) : 조직의 업무 수행에 반드시 필요하고 운영가능한 데이터

- 공용 데이터(Shared Data) : 여러 응용 시스템들의 공동 소유 및 유지하는 자료

데이터베이스 주요 암기사항2

@ 데이터베이스의 특징(실,계,동,네)

- 실시간 접근성(Real Time Accessiblity) : 수시적 비정형적인 질의(조회) 실시간 처리(Real-Time Processing) 응답 가능

- 계속적인 변환(Continuous Evolution) : 새로운 데이터의 삽입(Insertion), 삭제(Deletion), 갱신(Update), 최신의 데이터 유지

- 동시 공유(동시 공용)(Concurrent Sharing) : 여러 사용자 동시에 원하는 데이터 이용 가능

- 내용에 의한 참조(Content Reference) : 데이터 참조시 주소나 위치가 아닌 사용자가 요구하는 데이터 내용으로 참조

데이터베이스 주요 암기사항3

@ 스키마의 3계층(외부-개념-내부)

- 외부 스키마(External Schema) = 서브 스키마 = 사용자 뷰(View) : 사용자나 응용 프로그래머가 필요로 하는 데이터베이스의 논리적 구조 정의, 하나의 데이터베이스 시스템에 여러 개의 외부 스키마 존재, 하나의 외부 스키마를 여러 개의 응용 프로그램이나 사용자 공용 가능, 질의어(SQL) 사용하여 DB 사용

- 개념 스키마(Conceptual Schema) = 전체적인 뷰(View) : 데이터베이스의 전체적인 논리적인 구조, 데이터를 통합한 조직 전체의 데이터베이스로 하나만 존재, 개체 간의 관계와 제약조건/접근 권한/보안 및 무결성 규칙 명세 정의, 단순히 스키마라하면 개념 스키마 의미, 데이터베이스 관리자에 의해 구성

- 내부 스키마(Internal Schema) = 물리 스키마 : 데이터베이스의 물리적 구조 정의, 데이터의 실제 저장방법 기술, 물리적 저장장치의 입장에서 본 데이터베이스 구조, 물리적인 저장장치와 밀접한 계층, 데이터베이스에 저장될 레코드의 물리적인 구조 정의, 저장 데이터 항목의 표현 방법, 내부 레코드의 물리적 순서 표현, 시스템 프로그래머나 시스템 설계자 관점의 스키마