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

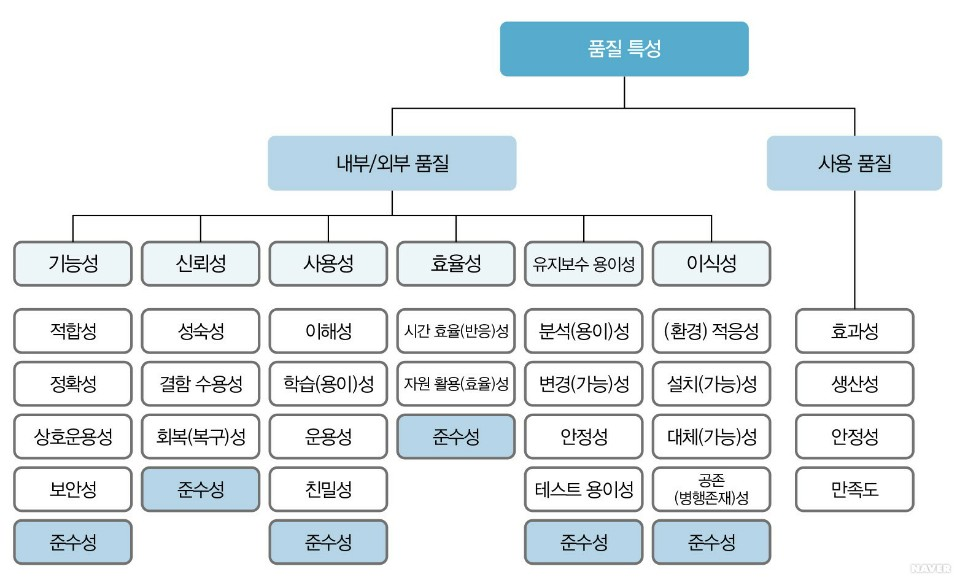
- 생산성 : LOC / 노력(인월)

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