



210804 요구사항분석실무_3

| | |
|---------|-----------------------------|
| 🕒 생성일 | @2021년 8월 4일 오전 9:02 |
| ▼ 성함 | 나한주 |
| 📎 속성 | BWG_요구사항 분석 실무_실습1_현동빈.xlsx |
| ▼ 수업 유형 | 이론 및 실습 |
| 🕒 수정일 | @2021년 8월 4일 오후 6:47 |
| 👤 작성자 | 현동빈 |

실습. 컨텍스트/프로세스 작성

금융IT 변천사

6교시. 테스트케이스 설계

실습. 컨텍스트/프로세스 작성

금융IT 변천사

▼ 역사의 연구(by. 아놀드 토인비)

운명

▼ 계층형DB

▼ 관계형DB

- 오라클

▼ 클라우드

- 통합DB 불가
- DB 쪼개기 → 중복이 없게

▼ 언어의 변화

▼ 컴퓨터적 사고

▼ 객체지향언어

- 관계형DB 성능이 좋아짐 → 객체지향언어 발생

▼ 메타데이터

▼ 빅데이터

▼ 모델링분석

- Waterfall방식 → 에자일

▼ 1세대(과목별 온라인)

- 비즈니스 모델 관점 : 전국 온라인 처리, 대량 거래 처리
- 시스템 개발 운영 관점 : 과목별로 카피해서 신규 개발, Assembler, 인력 부족, 프로그램 중복
- IT기술 관점 : 메인프레임, Assembler 사용

▼ 2세대(종합 온라인)

- 비즈니스 모델 관점 : 창구 역할 변화, 원스탑서비스, 온라인 고객 정보 활용, 경영 정보 요구
- 시스템 개발 운영 관점 : 프로그램 복잡성 가중, 신상품 수용 곤란
- IT기술 관점 : 메인프레임, COBOL, PLI, DB/DC, 계정계, 정보계 개념 등장, 어플리케이션 프레임워크 도입(CAP)

▼ 3세대(차세대)

- 비즈니스 모델 관점 : 상품출시 경쟁(Product Factory), 고객중심 정보-통합, 인터넷뱅킹, 24/365, 자본시장업무, 리스크 관리
- 시스템 개발 운영 관점 : 연평균 시스템 변경 건수가 1만건 상회
- IT기술 관점 : C/S(hetero 환경), HLL, C, web, Package 솔루션, CBD(Component Based Development)

▼ 3.5세대(포스트 차세대)

- 비즈니스 모델 관점 : 융복합상품서비스(비금융권과 제휴), 지주사 차원 고객-통합 서비스, 앱 금융서비스, 국외업무 확산, 실시간 데이터 분석 요구
- 시스템 개발 운영 관점 : UX의 차별화 중시, 보안의 중요성 심화, 기술 다양화 수용 곤란
- IT기술 관점 : Cloud Appliance, HLL, Java, SOA, MDM, CEP, BI 솔루션, Open SW, MDD(Model Driven Development), Maintenance and Evolution(보안/운영/

유지/진화 중시)

▼ 방법론 변화

- 구조적 방법론(복잡성 극복) → 정보공학 방법론(자동화) → 객체지향 방법론(모듈화) → CBD 방법론(재사용) → Agile 방법론(적시성)

6교시. 테스트케이스 설계

▼ 테스트

- 어플리케이션 또는 시스템(구성요소를 포함해서)의 동작과 성능, 안정성이 **요구사항**을 만족하는지 확인하는 과정
- **요구사항을 만족시키지 못함 = 결함(Defect)**
- 결함(Defect) ≠ 에러(Error)
- Error == Exception
- Error → **코드 상(개발자입장)에서** 의도한 방향으로 안됨

▼ 테스트의 원리

- 완벽한 테스트는 불가능
- 테스트를 개발 초기에 시작함 ⇒ **코딩을 한 후에 테스트케이스를 도출하는 것이 아님(요구사항을 도출할 때 같이 함)**
- 결함 집중 : 대다수의 결함들은 **소수의 특정 모듈에** 집중되어 발생
- 살충제 패러독스
 - 동일한 테스트케이스로 **동일한 테스트를** 반복적으로 수행한다면, **나중에는 더 이상 새로운 결함 x**
 - 테스트케이스는 **정기적으로 리뷰하고 추가해야 함** ⇒ 방법론적으로 정의 : V모델
- 유닛테스트(단위테스트) → 개발자 입장
 - 특정 클래스의 메소드 단위의 테스트
 - 의존관계를 끊고 테스트
- 거래단위테스트
 - 하나의 거래(서비스)를 완성하기 위해 다이어그램에서 표현되는 클래스 간의 의존관계를 포함한 테스트
- 통합테스트

- 프로세스 단위의 테스트
- 프로세스와 프로세스의 결합을 테스트
- 인수테스트
 - 인수인계 시 테스트
- 기능테스트
 - 기능 요구사항을 만족한지 확인하는 테스트
- 비기능테스트
 - 품질속성을 만족하는지 확인하는 테스트
- Regression 테스트(반복 테스트 → 자동화)
 - 과거에 정상적으로 작동하는 기능이 SW의 다른 기능을 수정한 이후에도 여전히 제대로 작동하는지 확인하는 테스트
- 신규비교테스트
 - Legacy(As-Is)시스템의 결과와 To-Be시스템의 결과가 동일한지 검증하는 테스트
- 싸이클테스트
 - 시간의 흐름에 따른 업무 프로세스 테스트
- 데이터이행테스트(DBA, DA, MO)
 - Legacy(As-Is)시스템으로부터 이행한 데이터의 정합성을 확인하는 테스트
- 테마집중테스트
 - 상품출시, 정보보안, 권한관리, 회계분개 처리 등등... ⇒ 주제별로 테스트
 - 다양한 케이스에 대해 테스트 가능
- 대외연계테스트
 - 대외기간과의 I/F 정합성을 확인하는 테스트

▼ 테스트 설계 기법

- White Box Test : 구조 기반 테스트케이스 설계 → 소스코드를 통해 찾아냄 (개발자 관점)
- Black Box Test : 명세 기반 테스트케이스 설계 → 실행하면서 찾아냄 (사용자 관점)
- 요구사항 & 소프트웨어

- 요구사항에 명시 but 구현 x
- 요구사항대로 구현 but 때에 따라 정상작동 x
- 요구사항에 명시 x but 구현

▼ 명세 = 예제 = 테스트케이스

- 명세
- 예제
- 요건이 기술되는 언어는 모호 ⇒ 복잡한 요건의 경우 예제로 구체화
- 일관된(통일된) 예제로 작성(공식예제)
- 공식예제 → 모호함 방지
- 예제 구체화 x → 요건이 바뀔 가능성이 큼
- 공식예제를 인수테스트케이스로 삼음
- 뭔가 잘못되었을 때 공식예제를 가지고 커뮤니케이션, 정제하여 더욱 분명하게 합의가능

▼ TDD(테스트 주도 개발) by. 켄트벡

- 외부설계가 끝나면, 개발스프린트가 시작하기 전에 테스트의 성공조건(=테스트케이스)에 대해서 합의
- 개발 이전에 테스트케이스 작성
- 테스트자동화
- 반복적인 Regression 테스트

▼ 테스트시나리오, 테스트케이스, 스텝

- 테스트시나리오
 - 애플리케이션의 테스트되어야 할 기능/특징을 기술한 일반적으로 한 줄짜리 문장
 - 보통 유스케이스나 업무프로세스로부터 도출됨
 - 테스트 시나리오가 테스트 케이스와 일대다 관계를 가짐
- 테스트케이스
 - 특정한 기능/특징이 비즈니스 요구사항을 준수하는지 여부를 평가하기 위한 조건들의 집합
 - 해당 테스트케이스를 확인하기 위한 상세스텝을 가짐
- 스텝

- 테스트케이스를 수행하기 위한 상세한 스텝

- 입력값(input values), 기댓값(expected values)를 가짐

▼ 테스트케이스 설계

- 1단계 : 테스트 시나리오 작성
- 2단계 : 테스트 케이스 도출
- 3단계 : 스텝설계