1단원

> SDLC = 소프트웨어 생명 주기

폭포수

- 결과물 명확 산출

- 두 개 이상 병행 수행 x

검토 > 계획 > 요구 분석 > 설계 > 구현 > 시험 > 유지 보수

프로토 타입

- 견본 제품을 만들어 최종 결과물 예측 모형

- 인터페이스 중점

나선형 모형 (spiral model)

- 보헴 = 폭포수 모형 , 프로토타입 모형 장점에 위험 분석 기능 추가

- 점진적 모형

- 요구사항 추가 가능

- 정밀, 유지보수 과정 필요 x

애자일 모형 (agile model)

- 민첩, 기민함 의미

- 변화에 유연함, 일정 주기 반복 및 개발.

- 고객 소통에 초점 맞춤

> 스크럼(scrum)

- 팀원 스스로 스크럼 팀 구성

- 개발 작업등 모든 것 스스로 해결

제품 책임자 (po)

- 백로그, 우선순위 지정

스크럼 마스터(sm)

- 일일 스크럼 , 공론화

개발팀(dt)

- sm을 제외 모든 팀원

- 최대 인원 7~8명

스프린트 계획 회의 > 스프린트 > 일일 스크럼 > 스크럼 검토 회의 > 스프린트 회고

> XP 5가지 핵심

- 의사소통 (coummnication)

- 단순성 (simplicity)

- 용기 (courage)

- 존중 (respect)

- 피드백 (feedback)

> 현행 시스템 파악

아키텍처 구성 파악

- 최상위 수준에서 계층별 표현 아키텍처

시스템 인터페이스 파악

- 데이터 종류, 형식, 프로토콜, 연계 유형, 주기 명시

> 개발 기술 환경 파악

미들웨어

- OS와 응용 프로그램 사이에서 활동

DBMS

- DB 구성, 접근 방법, 유지관리 책임

- JDBC = 자바

- ODBC = 응용 프로그램

웹 어플리케이션 서버(WAS)

- 사용자 요구에 따라 변하는 동적 컨텐츠 미들웨어

- 데이터 접근, 세션 관리, 트랜잭션 관리, 라이브러리 제공

> 요구사항

기능 요구사항

- 사항, 기능

비기능 요구사항

- 요구사항, 제약사항

ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

요구사항 개발 프로세스

도출 > 분석 > 명세 > 확인

elicitation > analysis > specification > validation

정형 분석 (foraml analysis)

구문, 의미, 수학적 표현

다이어 그램

- 구조적

클, 객, 컴, 배, 복, 패

- 행위

유, 시, 커, 상, 활, 상, 타

UI 세가지 분야

- 물리적 제어

- 전체적 구성

- 기능

UI 특성

- 편리성, 가독성

- 최소 노력

- 매게 역할

UI 구분

- CLI , GUI, NUI, VUI, OUI

텍스트 마우스 말,행동 음성 모든 사물 및 사용자

웹 3요소

- 표준, 접근성, 호환성

standards, accessilbility, browsing

와이어 프레임

- 뼈대

- 화면 단위 설계

목업

- 실제화면과 유사

- 정적 형태

스토리 보드

- 페이지간 이동 흐름

프로토타입

- 인터랙션 적용

- 실제 구현된 것 처럼 테스트

- 동적인 형태

유스 케이스

- 문서화

ISO/IEC 9126

- 기 신 사 효 유 이

UI prototype 개요

- 실제 동작 하는 것 처럼 만든 동적 형태

- 테스트 가능

- 페이퍼 프로토타입, 디지털 프로토타입

UI 설계서 작성

개요 > 표지 작성 > 개정 이력 작성 > 정의서 작성 > 구조 작성 > site map > 프로세스 > 화면

- 개정 이력 = 초안 작성 버전

- site map = 사이트 표시할 콘텐츠 한 눈에 보기 쉽게 설계

Usability 평가

- 실행 차 : 실행 기능

- 평가 차 : 실행 결과

UI 상시 설계

- 인터랙션 흐름 정의

- 순서, 분기, 조건, 루프

sequence, branch, condition, loop

UI 시나리오 문서 요건

완전성 (complete) - 최대 상세 기술

일관성 (consistent) - 일관성 유지

이해성 (understandable) - 쉽게 표현

가독성 (readable) - 표준 템플릿, 문서 쉽게 읽을 수 있어야 함

수정 용이성 (modifiable) - 수정 개선이 쉬워야 함

추적 용이성 (traceable) - 쉽게 추적 해야 함

HCI - 사람이 시스템을 편리, 안전하게 사용 연구

UX - 사용자가 시스템을 사용하고 느낀 경험

감성 공학 - 감성적

S/W 아키텍처

- 사용자 비기능적 요구사항으로 제약 반영

- 기능적 요구사항을 구현 방법 해결 과정

모듈화

- 성능 향상, 수정 및 재사용, 유지관리

- 모듈 단위

추상화

- 포괄적 개념을 설계 후 차례로 세분화 구체화 시킴

단계적 분해

- Niklaus Wirth 제안 하향식 설계 전략

- 추상화 반복 세분화 함

- 상세 내역은 뒤로 미루어 진행

정보 은닉

- 필요한 정보만 인터페이스 통해 주고 받음

- 모듈 독립적 수행 , 모듈 변경 시에도 영항 주지 않음

S/W 아키텍처 설계 과정

- 목표 설정 > 타입 결정 > 패턴 적용 > 구체화 > 검토

아키텍처 패턴

- 레이어 패턴 = 계층 구분, OSI 참조 모델

- 클라이언트 - 서버 패턴 = 하나의 서버 컴포넌트와 다수 클라이언트 컴포넌트로 구성 되는 패턴

\* 컴포넌트 = 독립적 업무 수행 실행 코드 기반 작성 모듈

- 파이프 - 필터 패턴 = 데이터 스트림 절차를 필터 컴포넌트로 캡슐화 해 파이프로 전송 패턴

- 모델 - 뷰 - 컨트롤 = 3개의 부분으로 구조화

- 서로 영향 받지 않음

- 한개 모델 여러개 뷰 필요 하는 대화형 APP 적합

ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

객체지향

-객체(object) = 객체가 반응할 수 있는 메시지(Message)의 집합 행위

-클래스(class) = 객체 일반적 타입, 인스턴스, 인스턴스화, 공통된 속성 및 행위

-캡슐화(encapsulation)

-상속(inheritance) = S/W의 재사용성(Reuse) 높이는 중요 개념

-다형성(polymorphism) = 각각 객체가 가지고 있는 고유 방법으로 응답 가능 능력

모듈

> 결합도(Coupling)

-내용 결합도 :(content) 어떤 모듈에 선언된 데이터를 다른 모듈에 직접 참조 및 수정

-공유 결합도 :(common) 공통적인 데이터를 다른 모듈에서 사용(전역 변수)

-외부 결합도 :(external) 어떤 모듈의 데이터가 외부의 모듈로부터 참조될 때

-제어 결합도 :(control) 어떤 내부 모듈을 논리적 제어 신호

-스탬프 결합도 :(stamp) 모듈간 인터페이스가 관련 자료들을 전달

-자료 결합도 : (data) 모듈 간 인터페이스가 관련 자료로만 구성

> 응집도(Cohesion)

-기능적(func) 단일문제

-순차적(sequn) 모듈간 출력된걸 다음 모듈의 입력

-교환적(comm) 모듈이 입'출력

-절차적(Proc) : 구성요소들이 순차적 수행

-시간적(Temp) : 같은 시간에 시행

-논리적(logic) : 유사성격 및 특정 형태 분류

-우연적(coin) : 연관없는거

공통 모듈의 개요

-정확성(correctness) = 정확 작성

-명확성(clarity) = 명확 작성

-완전성(complenteness) = 시스템 구현 위해 모든 기술 작성

-일관성(consistency) = 공통 기능 충돌 발생 방지

-추적성(traceability) = 시스템 관계 파악 작성

코드

-순차코드 = 순차적으로

-블록코드 = 공통성 있는 것 끼리 블록화

-10진코드 = 0~9까지 10분할

-그룹 분류 코드 = 대,중,소 분류 및 일련번호 부여

-연상 코드 = 숫자, 문자, 기호 이용 코드 부여

-표의 숫자 코드 = 길이, 넓이, 부피, 지름, 높이 물리적 수치

-합성코드 = 2개 이상 코드 조합

디자인 패턴

>생성 패턴

-추, 빌, 팩, 프, 싱

> 구조 패턴

-어, 브, 컴, 데, 퍼, 플, 프

> 행위 패턴

-책, 커, 인, 반, 중, 메, 옵, 상, 전, 템, 방

요구사항 검증 방법

- 요구사항 검토(Requirements Review)

= 동료검토, 워크스루, 인스펙션

인터페이스 방법 명세화

- 내'외부 시스템 연계 작동할 때 인터페이스별 송'수신 방법, 데이터, 오류 식별 및 처리 방안 문서

시스템 연계 기술

- DB link, API/Open API, 연계 솔루션, 소켓, 웹 서비스

Interface 통신 유형

- 단방향 = 거래 요청 후 응답 없음

- 동기 = 거래 요청 후 응답 대기(Request - Reply)

- 비동기 = 거래 요청하고 다른 업무 보다가 응답 오면 처리

Interface 처리 유형

- 실시간 = 요청 내용 바로 처리

- 지연 처리 = 데이터를 매건 단위 처리 경우 비용 많이 발생 사용 방식

- 배치 = 대량 데이터 처리 사용 방식

오류코드

ex] ES 100 01

구분자 그룹번호 일련번호

미들웨어 솔루션 명세

- OS와 OS 사이 제공하는 서비스

기능

-DB = 클라이언트에서 원격 db와 연결 / 2-Tier 아키텍처

-RPC = 프로시저를 사용해 원격 프로시저를 로컬 프로시저처럼 호출 방식

-MOM = 메시지 기반 비동기형 전달 방식 / mq, message-q, jms

-TP-Monitor = 항공기, 철도 예약 업무 온라인 업무 처리 감시

-ORB = 미들웨어 코바 표준 스펙을 구현

\*코바 = Network 분산 프로그램 객체 생성, 배포, 관리

-WAS = 사용자 요구 따라 변하는 동적 콘텐츠 처리 사용