<Chapter4. 요구분석>

요구분석 : 소프트웨어 개발의 실질적인 첫단계. 사용자의 요구에 대해 이해하고 정리하는 작업.간단한 모델 만들고 잘못된 점 찾아서 가능한 빨리 수정하는게중요.

요구분석의 3단계 : 요구추출=> 요구분석 및 정의=> 요구 확인

요구 : 시스템에 대한 고객의 요청을 확정한 것. 진정한 요구를 찾는 것은 프로젝트 성공의 필수조건이며 여러 이해 당사자의 이해 관계와 관련됨.

-**제약사항** : 특정 프로그래밍언어, 특정 제품 사용, 소프트웨어 시스템의 해결책을 제한.

-**분류** : **기능요구** = 고객이 요구하는 시스템이 처리할 기능. (동사로 표현되며 쉽게 파악된다, 제품기능, 사용사례로 정리) (업무절차, 기계동작을 표현)

**비기능요구** = 요청된 기능외에 시스템이 갖춰야 할 조건 특징 (형용사로 표현됨. 파악 어려움, 제품속성, 품질속성, 시나리오로정리) (응답속도, 고장에서 회복되는 기간, 보안 등)

기능요구

1. 시스템이 외형적으로 나타내는 기능과 동작. (Ex. 현금인출기 : 현금의 인출, 잔금조회, 계좌이체, 현금 서비스. )

2. 시스템과 외부 요소들간의 인터랙션(상호작용) (Ex. 어떤 상태일 때 외부의 데이터나 명령을 받아들여 어떤 반응을 하는지 기술한다.)

3. 입력 자료가 제공되어 이를 변환 처리하여 결과를 출력.

기능의 종류 : IEEE표준830에서 기능 요구는 입력의 검증, 정확한 작업순서, 비정상적상태에대한 응답 및 복구, 매개변수의 유효성, 입출력관계를 포함해야한다고 한다.

기능요구의 종류와 사례 4분류 : 1.기능 2.자료 3.입출력 4. 사용자 P.142참고

비기능요구 : 시스템이 제공하는 기능에 직접 관련되지 않은 요구. (ex. 항공권 예약 할 때 시승템이 제공하는 모든 화면이 5초안에 사용자에게 보여야한다) 시스템 구축에 대한 성능, 보안, 품질, 안전등에 대한 요구사항 (ex. 시스템이 질의에 대한 결과가 3초안에 떠야한다. 구매요청에 대한 회신이 4시간 이전에 이뤄져야한다.)

비기능요구의 종류 : 외부 인터페이스, 메모리 제약, 성능 요구, 사용자의 특성과 가정, 설치 환경의 적합성 => 이거 모두 다 대표적 소프트웨어에 대한 제약조건. P144 => 비기능요구의 종류와 사례

요구 대상에 의한 분류

1. 비즈니스 요구 : 소프트웨어 적용업무사례로부터 획득하는 요구사항
2. 아키텍처 및 설계요구 : 비즈니스 요구보다는 더 상세한 요구사항. 구현에 필요한 전체적인 설계와 관련된 요구사항
3. 시스템 및 통합요구 : 개발자가 코딩하는데 필요한 아주 상세한 요구사항

p.145 => 요구대상에 따른 분류사례

요구사항 추출 3단계

1. 응용에 대한 정보 출처파악
2. 응용에 대한 정보 취합
3. 요구와 제한 사항의 정의

요구사항 정보 추출방법 : 고객의 발표, 문헌/양식 조사, 인터뷰, 설문조사, 브레인스토밍 회의, 관찰, 작업분석, 프로토타이핑

1. **요구정보출처파악**

유형(p.147)

-고객 : 프로젝트를 유치시키고 재정적윽로 지원하며 운명을 결정하는 당사자

-도메인전문가 : 비즈니스도메인을 지원하는 시스템을 구축하기 위해 필요

-이해 당사자(stakeholder) : 시스템 운용으로 인해 영향을 받는 사람들

-사용자 : 시스템을 직접 사용하는 사람들

-역공학 : 레거시 시스템이나 현재 문서를 시간과 자원을 절약하기위해 철저히 조사하고 분석함

1. **정보취합방법 (응용에 대한 정보 취합)**
2. 고객의 발표 : 개발팀이 구축하는 시스템에 대해 초기에 개념 잡기 가능. 효과적인 가이드라인. 고객업무를 잘 알고있는 운영 책임자/ 관리자가 발표. 발표하기전 개발팀원이 필요한 정보가 있는지 검토. 의심가는 부분을 질문해서 명확히 한다. 구현과 관련된 토의는 배제. 발표 내용의 복사본을 팀원과 공유. 2시간 이상의 발표회는 지양.
3. 문헌양식조사 : 유사 프로젝트조사. 현재 개발할 시스템에 대한 통찰 제공. 업무 문서나 양식 조사. 현재의 업무나 시스템 정보에 대해 깊은 이해 가능. 산업 및 기업 표준 조사. 관련 정부 정책/규제 조사
4. 인터뷰 : 가능하면 많은 당사자와 인터뷰하는게좋음. 여유로운 인터뷰 일정. 인터뷰 약속시간을 넘기더라도 여유롭게. 중요한 관련자와는 여러차례인터뷰. **정형적인터뷰** : 미리 정해진 질문을 가지고 묻는다(정보가 제한적) **비정형인터뷰** : 주제만 정하고 자유롭게 대화(진행자가 노련해야함) **부분정형인터뷰** : 정형적과 비정형의 장점을 결합함. 사전에 준비한 조사표에따라 질문하다가 필요하면 추가질문. 효과적인 인터뷰는 인터뷰준비, 1:1인터뷰, 개발형 질문, 후속질문하기, 대답듣고 분석하기로 이루어진다. 인터뷰의 절차 : p.151
5. 설문조사 : 관리자나 사용자와 같은 이해 당사자를 대상으로 함. 이해 당사자들이 의사결정 과정에 포함. **무기명설문** = 이해 당사자들의 관심과 내부정보, 개선의견 도출. 감추어진 정보를 끌어내기 쉬움. 신분(이름) 쓰지않음. **기명설문** = 신분(이름) 씀. **유의사항** : 질문은 간단하고 중요한 이슈에 집중 적절하고 잘 기술된 질문으로 해야함
6. 브레인스토밍 : 아이디어를 낼 목적으로 여러명으로부터 정보를 얻기위한 회의. 훈련된 요원이 주재. 토론보다는 아이디어를 쏟아놓는 회의, 익명성 보장. 서로 자극이 되어서 열정을 가지고 아이디어 창안가능. JAD(Joint Application Development) 집중 브레인스토밍 세션.
7. 프로토타이핑

**프로토타입** : 최종 시스템의 예상기능중 일부를 빠르게 구현한 프로그램.

**페이퍼 프로토타입(그림)** : 가장 단순한 형태, 무엇이 일어날지 설명한 그림을 순서대로 그린 것. 병행하여 만들기 적합함.

**모의 사용자 인터페이스(화면)** : 프로토타이핑 언어로 작성. 가장 흔함. 컴퓨팅,DB접근, 다른 시스템과의 상호작용 불가능, 시스템의 특별한 측면을 프로토타이핑 하기도 함.

요구분석 : 요구 후보를 분석하고 결정하여 요구로 확정

**요구 품질**(요구사항이 다음과 같은 성질 가지는지 분석) :

원자적 : 요구사항이 복합된 목적이 아니라 단일 목적을 가지는 성질

완전성 : 요구 사항 안에 정의된 것이 정보의 모든 것을 포함한 것 비모호성과 통일성 : 명확하지 않은 것을 기술하거나 같은 내용을 다르게 언급하지 않음

추적성 : 요구 사항을 쉽게 추적할 수 있도록 고유번호매김

우선순위화 : 요구 사항의 중요도를 파악할 수 있는 성질

테스트 가능성 : 요구 사항이 검증 가능하도록 기술되어있는 성질

도메인 : 요구의 배경

도메인 분석 : 설계 모델링에 필요한 여러 개념과 비즈니스 룰을 파악. 응용 분야에 존재하는 개념을 잘 정의하고 분석하여 시스템에 존재하는 개념으로 정립하는 단계.

도메인 분석 방법 : 도메인 개념찾기, 도메인 사전 작성, 비즈니스 규칙 정리. 사례분석 : p158~159

시나리오 기반 분석 : 다양한 사람들이 참여하여 다양한 용어와 개념을 전달하여 요구를 도출한다. 시스템 사용에 대해 시나리오기반(5W1H) when where who what why how) , 사용자 스토리(애자일 방법, 스토리 카드)을 작성하여 커뮤니케이션의 장벽을 해소. ( <사용자/역할(who)>는 <목표/혜택/이익(why)>를 얻기 위해 <행위/작업(what)>을 원한다.)

유스케이스 : 시스템의 기능을 나타내기 위해 사용자의 요구를 추출하고 분석하는데 사용하는 그림. 유스케이스와 액터로 구성. 도메인 분석의 결과를 액터, 사용사례, 관계들로 구성된 시스템 명세로 매핑하는 작업. 유스케이스는 여러 개별 시나리오를 묶은 것. 시나리오로부터 유스케이스를 형성한다. 유스케이스는 시나리오의 집합을 정의하며 시스템이 실행하는 일련의 동작으로 특정 액터에게 관찰 가능한 결과를 제공함

유스케이스 요소 : 액터, 시스템 범위, 유스케이스, 관계

유스케이스 분석 과정 : 액터 찾기 => 유스케이스 찾기 => 유스케이스 사이의 관계 찾기

유스케이스 명세 : 시스템이 제공해야 할 서비스를 시간이 경과되는 순서로 정렬하여 기술한 것

액터를 찾기 위한 질문 : 어떤 사용자 그룹이 작업을 수행하기 위해 시스템의 지원을 받는가? 어떤 사용자 그룹이 시스템의 주요기능을 사용하는가? 어떤 사용자 그룹이 유지 보수와 관리 등의 부수적 기능을 사용하는가? 시스템이 다른 외부하드웨어나 소프트웨어 시스템과 동작하는가?

유스케이스를 찾아낼 때 유용한 질문 : p.163

유스케이스 명세 : p.164

유스케이스 관계 : 관계를 이용하여 모형의 복잡도를 줄이고 이해도를 높인다.

유스케이스 관계 종류

1. **대안흐름** : 기본 유스케이스에서 이벤트의 흐름이 약간 변형되거나 선택, 예외인 경우
2. **포함 관계** : 다른 유스케이스에서 재사용 할 수 있도록 캡슐화하려는 경우(유스케이스 사이의 중복을 제거한다. 어떤 유스케이스가 다른 유스케이스를 포함하는 관계. 공통된 동작을 떼어 낼 수 있다.)
3. **확장 관계** : 기본 유스케이스의 기본 이벤트 흐름이 특정 조건에 만족되었을 때 분리 확장된 것 (유스케이스가 일정한 조건 아래 확장된 동작을 포함한다면 다른 유스케이스를 확장하는 관계에 있다. Ex)결제과정에 멤버십이 있는 경우 할인 적용. 기본 유스케이스인 ‘결제’는 자체로 완전한 유스케이스이고 멤버쉽에 가입되어있다는 조건에 만족하면 ‘멤버쉽할인’ 유스케이스가 삽입.)

사용자 스토리 vs 유스케이스 : p.168

요구명세 : 사용자와 개발자간의 이해를 돕기위함. p.169

요구명세 작성법 (Gilbert가 제안한 주의사항)

1. 요구 분석서는 사용자와 개발자 모두가 쉽게 이해할 수 있도록 써야한다.
2. 요구 분석서에 기술된 조건은 개발자와 사용자가 모두 동의한 것이어야한다
3. 요구 분석서는 목표 시스템에 의해 수행될 모든 기능을 정확히 기술해야한다
4. 요구 분석서는 목표 시스템에 영향을 주는 모든 제약 조건을 기술한다
5. 요구 분석서는 시스템의 인수를 위한 테스트 기준을 제공해야한다
6. 요구 분석서는 원하는 시스템의 품질과 상대적인 중요도 및 품질을 재는 방법이 기술되어야한다.

요구 검증사항

1. 이해용이성 : 요구 명세서를 읽을 때 요구의 의미를 잘 이해할 수 있는가
2. 중복 : 필요없이 중복된 부분이 없는가
3. 완전성 : 빠진 요구가 없는지 요구를 기술하는데 빠진 정보가 없는가
4. 일관성 : 요구사항이 서로 모순되지 않는가
5. 모호성 : 요구분석의 내용이 모호함 없이 모든 참여자들에 의해 명확하게 이해될 수 있는가
6. 검증 가능성 : 요구분석 명세서에 기술된 내용이 사용자의 요구를 만족하는가? 개발된 시스템이 요구사항 분석 내용과 일치하는지를 검증 할 수 있는가?
7. 추적 가능성 : 시스템 요구사항과 시스템 설계문서 및 구현과 매핑되어 추적할 수 있는가?