

# 요구사항 확인

2021년 12월 03일

윤성혁

# 1. 현행 시스템 파악

- 1단계
  - 시스템 **구성** 파악
  - 시스템 **기능** 파악
  - 시스템 **인터페이스** 파악
- 2단계
  - **아키텍처** 구성 파악
  - **소프트웨어** 구성 파악
- 3단계
  - **하드웨어** 구성 파악
  - **네트워크** 구성 파악

# 시스템 구성 파악

- 현행 시스템의 구성은 조직의 주요 업무를 담당하는 **기간 업무**와 이를 지원하는 **지원 업무**로 구분하여 기술한다.
- 기간 업무
- 지원 업무
- 시스템(System)은 각 구성요소들이 **상호작용**하거나 **상호의존**하여 복잡하게 얽힌 **통일된 하나의 집합체(unified whole)**다. 또  
는 이 용어는 복잡한 사회적 체계의 맥락에서 구조와 행동을 통제하는 규칙들의 집합체를 일컫기도 한다.

# 시스템 기능 파악

- 현행 시스템의 기능은 단위 업무 시스템이 현재 제공하는 기능들을 주요 기능과 하부 기능, 세부 기능을 구분하여 계층형으로 표시한다.

# 시스템 인터페이스 파악

- 현행 시스템의 인터페이스에는 단위 업무 시스템 간에 주고받는 **데이터의 종류, 형식, 프로토콜, 연계 유형, 주기** 등을 명시한다.
- 데이터의 종류
- 형식 : HTML, XML, JSON
- 프로토콜 : 통신규약, TCP/IP
- 연계 유형: EAI, FEP
- 주기

# 아키텍처 구성 파악

- 현행 시스템의 아키텍처 구성은 **기간 업무 수행에** 어떠한 **기술 요소들이** 사용되는지 **최상위 수준에서 계층별로** 표현한 아키텍처 구성도로 작성한다.
- **아키텍처**는 시스템 내부에서 각각의 하위 시스템들이 어떠한 관계로 **상호 작용**하는지 파악 할 수 있도록 **구성이나 동작 원리**를 표현한 것

# 소프트웨어 구성 파악

- 소프트웨어 구성에는 단위 업무 시스템별로 업무 처리를 위해 설치되어 있는 소프트웨어들의 **제품명, 용도, 라이선스 적용 방식, 라이선스 수** 등을 명시한다.
- 상용 소프트웨어
- 오픈소스 소프트웨어
- 라이선스 적용 방식
- 코어(Core)

# 하드웨어 구성 파악

- 하드웨어 구성에는 단위 업무 시스템들이 운용되는 **서버의 주요 사양과 수량** 그리고 **이중화의 적용 여부**를 명시한다.
- 서버의 주요 사양:
  - CPU의 처리 속도
  - 메모리 크기
  - 하드디스크 용량
- **서버의 이중화란?** 운용 서버의 장애 발생시 대기 서버로 서비스를 계속 유지할 수 있도록 **운용 서버의** 자료 변경이 **예비 서버**에도 동일하게 복제되도록 관리하는 것



# 네트워크 구성 파악

- 네트워크 구성은 업무 시스템들의 네트워크 구성을 파악할 수 있도록 **서버의 위치, 서버 간의 네트워크 연결 방식을 네트워크 구성도로** 작성한다.
- 네트워크 토폴로지(Topology)
- 각 부서나 팀별 네트워크 구성도
- 서버 중심으로 표현

## 2. 개발 기술 환경 파악

- 개발 기술 환경의 정의
- 운영체제
- 운영체제 관련 요구사항 식별 시 고려사항
- 데이터베이스 관리 시스템
- DBMS 관련 요구사항 식별 시 고려사항
- 웹 애플리케이션 서버(WAS)
- 웹 애플리케이션 서버(WAS) 관련 요구사항 식별 시 고려사항
- 오픈 소스 사용에 따른 고려사항

# 운영체제 관련 요구사항 식별 시 고려사항

- 가용성
- 성능
- 기술 지원
- 주변 기기
- 구축 비용

# DBMS 관련 요구사항 식별 시 고려사항

- 가용성
- 성능
- 기술 지원
- **상호 호환성**: 설치가능 운영체제의 종류, JDBC, ODBC 호환여부
- 구축 비용

# WAS 관련 요구사항 식별 시 고려사항

- 가용성 : 재가동성, 트랜잭션 처리, 이중화
- 성능: 다양한 설정 옵션, 가비지 컬렉션(GC)
- 기술 지원
- 구축 비용: 총 소유 비용(TCO), 라이선스 비용, 유지관리 비용

### 3. 요구사항 정의

- 요구사항의 개념 및 특징
- 요구사항의 유형
- 요구사항 개발 프로세스
- 요구사항 도출(요구사항 수집)
- 요구사항 분석
- 요구사항 명세
- 요구사항 확인(검증)

# 요구사항의 개념 및 특징

- 요구사항은 소프트웨어가 어떤 문제를 해결하기 위해 제공하는 서비스에 대한 설명과 정상적으로 운영되는데 필요한 제약조건 등을 나타낸다.

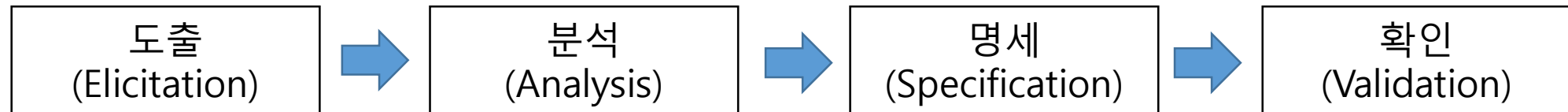
# 요구사항의 유형

- 기능 요구사항(Functional Requirement) : 핵심요구사항
- 비기능 요구사항(Non-functional Requirement)
- 사용자 요구사항(User Requirement) : 사용자 관점, **사용자 경험**
- 시스템 요구사항(System Requirement)



# 요구사항 개발 프로세스

- 요구사항 개발 프로세스는 개발 대상에 대한 요구사항을 체계적으로 **도출**하고 이를 **분석**한 후 분석 결과를 **명세서**에 정리한 다음 마지막으로 이를 **확인** 및 **검증**하는 일련의 **구조화된 활동**이다.



# 요구사항 도출

- 요구사항 도출은 시스템, 사용자 그리고 시스템 개발에 관련된 사람들이 서로 의견을 교환하여 요구사항이 **어디에** 있는지, **어떻게** 수집할 것인지를 **식별하고 이해하는 과정**
- 이해관계자(Stakeholder)
- 인터뷰, 설문, 브레인스토밍, 워크샵, 프로토타이핑, 유즈케이스

# 요구사항 분석

- 요구사항 분석은 개발 대상에 대한 사용자의 요구사항 중 **명확하지 않거나 모호하여** 이해되지 않는 부분을 **발견하고 이를 걸러내기 위한 과정**
- 중복 제거, 상충되는 요구사항 해결
- 개발 범위 파악
- 주변 환경이 상호 작용하는 방법을 이해

# 요구사항 명세

- 요구사항 명세는 요구사항을 체계적으로 분석한 후 승인될 수 있도록 문서화
- 기능의 누락이 있는가?
- 명확한가?
- 중복이 없는가?
- 개발을 위한 설계의 기초
- 기능 추적

# 소프트웨어 요구사항 명세서(SRS)

- Software Requirement Specification
- 소프트웨어가 반드시 제공해야 하는 기능, 특징, 제약조건 등을 명시
- 시스템의 동작, 성능, 보안, 사용성(품질) 기술
- 프로젝트 유형에 맞게 양식을 만듦
- 시스템의 기능, 데이터, 외부 인터페이스, 품질 요구사항은 **요구사항 단위별로 개별 요구사항 명세서를 작성**

# 요구사항 확인(검증)

- 요구사항 확인은 개발 자원을 요구사항에 할당하기 전에 **요구사항 명세서가 정확하고 완전하게 작성되었는지를 검토하는 활동**
- 내용이 이해가 쉬운가?
- 일관성이 있는가?
- 회사의 기준(표준)에 부합?
- 누락된 기능이 없는가?
- 요구사항은 이해관계들이 검토
- 형상 관리 수행(이력, 버전, 작성자, 추적, 설명)

## 4. 요구사항 분석 기법

- 요구사항 분석 기법은 개발 대상에 대한 사용자의 요구사항 중 **명확하지 않거나 모호한 부분을 걸러내기 위한 방법**
- 요구사항 분류(Requirement Classification)
- 개념 모델링(Conceptual Modeling)
- 요구사항 할당(Requirement Allocation)
- 요구사항 협상(Requirement Negotiation)
- 정형 분석(Formal Analysis)

# 요구사항 분류(Requirement Classification)

- 기능 요구사항
- 비기능 요구사항
- 요구사항의 원천:
  - 상위 요구사항에서 유도?
  - 이해관계자?
- 우선순위에 따라 분류
- 영향의 범위
- 생명 주기



# 개념 모델링(Conceptual Modeling)

- 요구사항을 보다 쉽게 이해할 수 있도록 현실 세계의 상황을 단순화하여 개념적으로 표현
- 문제의 주체인 개체(Entity)들과 그들 간의 관계 및 종속성을 반영
- 이해관계자별로 관점이 다양하므로 다양한 관점으로 모델링
- UML(Unified Modeling Language)로 작성
  - 유스케이스 다이어그램(Use Case Diagram)
  - 데이터 흐름 모델(Data Flow Model)
  - 상태 모델(State Model)
  - 목표 기반 모델(Goal-Based Model)
  - 사용자 인터랙션(User Interaction)
  - 객체 모델(Object Model)
  - 데이터 모델(Data Model)

# 요구사항 할당(Requirement Allocation)

- 요구사항 할당은 요구사항을 만족시키기 위한 **구성 요소를 식별**
- 식별된 구성 요소들 간에 어떻게 작용하는지 **분석하는 과정에서 추가적인 요구사항이 발견될** 수 있다.

# 요구사항 협상

- 요구사항 협상은 요구사항이 서로 **충돌될 경우** 이를 적절히 **해결하는 과정**
- 이해관계자가 요구하는 요구사항이 서로 충돌
- 자원이 충돌
- 기능 요구사항과 비기능 요구사항의 충돌
- **해결방안**
  - 우선순위
  - 중요도

# 정형 분석(Formal Analysis)

- 정형분석은 **구문(Syntax)**과 **의미(Semantics)**를 갖는 정형화된 언어를 이용해 요구사항을 **수학적 기호로 표현한 후** 이를 분석하는 과정

## 4. 요구사항 확인 기법

- 요구사항 확인 기법은 요구사항 개발 과정을 거쳐 문서화된 요구사항 관련 내용을 **확인하고 검증하는** 방법
- 요구사항이 자원에 배정되기 전에 **문제 파악**을 위한 **검증**을 수행
- 요구사항 검토(Requirement Reviews)
- 프로토타이핑(Prototyping)
- 모델 검증(Model Verification)
- 인수 테스트(Acceptance Tests)

# 요구사항 검토(Requirement Reviews)

- 이해당사자, 고객 대표자 검토
- 시스템 정의서
- 시스템 사양서
- 소프트웨어 요구 사항 명세서

# 프로토타이핑(Prototyping)

- 개발 대상 시스템 또는 그 일부분을 개략적으로 만든 원형
- 프로토타이핑을 수행하면서 지속적으로 재작업
- 새로운 요구사항을 도출
- 소프트웨어 **요구사항에** 대한 소프트웨어 엔지니어의 **해석이 맞는지 확인하기** 위한 수단
- **장점:** 추가/변경에 대한 피드백이 가능, 원활한 의사소통, 프로토타이핑이 개선될수록 변동 가능한 요구 사항들이 감소
- **단점:** 사용자의 관심이 **핵심에서 벗어나** 프로토타입 제작에만 집중될 수 있다. 대상 범위를 잘못 이해, **비용의 부담**

# 모델 검증(Model Verification)

- 모델 검증이란 **요구사항 분석 단계에서** 개발된 모델이 요구사항을 충족시키는지 검증
- 객체 모델을 통해서 **정적 분석**을 수행
- **정적 분석(Static Analysis)**: 실행을 통해서 확인하는 것이 아니라 **명세서의** 정확성이나 일관성 등을 확인하거나 **분석 도구**를 사용해 확인하는 방법



# 인수 테스트(Acceptance Tests)

- 인수 테스트는 사용자가 실제로 사용될 환경에서 요구사항들이 모두 충족되는지 **사용자 입장에서 확인**하는 과정
- 사용자 인수 테스트
- 운영상의 인수 테스트
- 계약 인수 테스트
- 규정 인수 테스트
- 알파 검사
- 베타 검사