

시스템 인터페이스 요구사항 분석

시스템 인터페이스는 독립적으로 떨어져 있는 시스템끼리 서로 연동하여 상호작용하기 위한 접속 방법이나 규칙을 의미한다.

목표로 하는 시스템과 외부시스템을 연동하는데 필요한 시스템 인터페이스에 대한 요구사항을 기술한 것이다.

시스템 인터페이스 요구사항 명세서에는 인터페이스 이름, 연계 대상 시스템, 연계 범위 및 내용, 연계 방식, 송신 데이터, 인터페이스 주기, 기타 고려 사항 등이 포함되어야 한다.

시스템 인터페이스 요구사항 분석

시스템 인터페이스 요구사항 분석은 요구사항 명세서에서 요구사항을 기능적 요구사항, 비기능적 요구사항으로 분류하고 조직화하여 요구사항 명세를 구체화하고 이를 이해관계자에게 전달하는 일련의 과정이다.

요구사항 분석은 소프트웨어 요구사항 분석 기법을 적절히 이용한다.

요구사항의 분해가 필요한 경우 적절한 수준으로 세분화 한다.

요구사항 분석 시 누락된 요구사항이나 제한조건을 추가한다.

요구사항에 대한 상대적 중요도를 평가하여 우선순위 부여

시스템 요구사항 분석 절차

1. 시스템 인터페이스 요구사항 목록에서 시스템 인터페이스 관련 요구사항을 선별하여 별도로 시스템 인터페이스 요구사항 목록을 만든다.
2. 시스템 인터페이스와 관련된 요구사항 및 아키텍처 정의서, 현행 시스템의 대 내외 연계시스템 현황 자료 등 시스템 인터페이스 요구사항과 관련된 자료를 준비한다.
3. 시스템 인터페이스에 대한 요구사항 명세서를 확인하여 기능적인 요구사항과 비기능적인 요구사항으로 분류한다.

요구사항 검증(Requirement Verification)

인터페이스의 설계 및 구현 전에 사용자들의 요구사항이 요구사항 명세서에 정확하고 완전하게 기술되어 있는지 검토하고 개발 범위의 기준인 베이스라인을 설정하는 것이다.

인터페이스 요구사항 검토 계획 수립

검토 기준 및 방법 : 프로젝트의 규모와 참여 인력, 검토 기간 등을 고려하여 검토 기준 및 방법을 정한다.

참여자: 프로젝트 규모에 따라 이해관계자들을 파악하여 프로젝트 관리자, 품질 관리자, 인터페이스 분석가, 소프트웨어 아키텍트, 시스템 사용자, 테스트 관리자 등 요구사항 검토 참여자를 선정한다.

체크리스트 : 완전성, 일관성, 명확성 등의 항목을 점검할 수 있는 요구사항 검토 체크리스트를 작성한다.

관련자료 : 인터페이스 요구사항 목록, 인터페이스 요구사항 명세서, 현행 및 표준 시스템 구성도 등 인터페이스 요구사항 검토에 필요한 자료들을 준비한다.

일정 : 인터페이스 요구사항 검토 일정을 정한다.

검토계획이 수립되면 인터페이스 요구사항 검토 참여자들에게 검토 관련 자료와 일정 등을 전달한다.

인터페이스 요구사항 검토 및 오류 수정

- 체크리스트의 항목에 따라 인터페이스 요구사항을 검토하는 것
- 오류가 발견되면 오류를 수정할 수 있도록 오류 목록과 시정 조치서를 작성한다.
- 요구사항 검토 결과를 검토 관련자들에게 전달한다.
- 시정 조치서를 작성한 경우 시정 조치가 완료되었는지 확인하여 시정 조치가 되면 인터페이스 요구사항 검토 작업을 완료한다.

인터페이스 요구사항 베이스 라인 설정

인터페이스 요구사항 검토를 통해 인터페이스 요구사항은 프로젝트 관리자와 주요 의사 결정자 에게 공식적으로 승인 받는다.

-소프트웨어 설계 및 구현을 위해 요구사항 명세서의 베이스라인을 설정한다.

요구사항 검증 방법

인터페이스 요구사항 검증은 다음과 같은 검증 방법을 적절히 이용한다.

요구사항 검토(Requirements Review): 요구사항 명세서의 오류 확인 및 표준준수 여부등의 결함 여부를 검토 담당자들이 수작업으로 분석하는 방법으로 **동료검토**, **워크스루**, **인스펙션**이 있다.

동료검토(Peer Review)	요구사항 명세서 작성자가 명세서 내용을 직접 설명하고 동료들이 이를 들면서 결함을 발견하는 형태의 검토 방법이다.
워크스루(Walk Through)	검토 회의 전에 요구사항 명세서를 미리 배포하여 사전 검토한 후에 짧은 검토회의를 통해 결함을 발견하는 형태의 검토 방법이다.
인스펙션(Inspection)	요구사항 명세서 작성자를 제외한 다른 검토 전문가들이 요구사항 명세서를 확인하면서 결함을 발견하는 형태의 검토 방법이다.

프로토타이핑(**Prototyping**) 사용자의 요구사항을 정확히 파악하기 위해 실제 개발될 소프트웨어에 대한 건본품(**Prototype**)을 만들어 최종 결과물을 예측한다.

테스트 설계 : 요구사항은 테스트할 수 있도록 작성되어야 하며, 이를 위해 케이스(**Test Case**)를 생성하여 이후에 요구사항이 현실적으로 테스트 가능한지를 검토한다.

CASE(Computer Aided Software Engineering) 도구 활용 : 일관성 분석(Consistency Analysis)을 통해 요구사항 변경사항의 추적 및 분석, 관리하고, 표준 준수여부를 확인한다.

인터페이스 요구사항 검증의 주요항목

완전성(Completeness): 사용자의 모든 요구사항이 누락되지 않고 완전하게 되어 있는가?

일관성(Consistency) : 요구사항이 모순되거나 충돌되는 점 없이 일관성을 유지하고 있는가?

명확성(Unambiguity) : 모든 참여자가 요구사항을 명확히 이해할 수 있는가?

기능성(Functionality) : 요구사항이 어떻게 보다 무엇을 에 중점을 두고 있는가?

검증가능성(Verifiability) : 요구사항이 사용자의 요구를 모두 만족하고, 개발된 소프트웨어가 사용자의 요구내용과 일치하는지를 검증할 수 있는가

추적가능성(Traceability) : 요구사항 명세서와 설계서를 추적할 수 있는가?

변경 용이성(Easily Changeable) : 요구사항 명세서의 변경이 쉽도록 작성되었는가?

개발 시스템 식별

인터페이스 관련 자료들을 기반으로 개발하고자 하는 시스템의 상세 식별 정보를 정의하고 목록을 작성하는 것이다.

* 인터페이스 관련자료

개발할 시스템에 대한 업무 정의서, 시스템 아키텍처 정의서, 유스케이스 정의서, 현행 시스템에 대한 인터페이스 요구사항 명세서 및 목록, 대내외 연계 시스템 목록, 연계대상 시스템에 대한 정의서 및 인터페이스 목록 등이있습니다.

- **시스템 아키텍처(System Architecture)** : 시스템 내부에서 각각의 하위 시스템들이 어떠한 관계로 상호작용을 하는지 파악할 수 있는 구성이나 동작 원리를 나타내는 것입니다.
- **유스케이스(Use Case)**: 사용자의 요구사항을 기능 단위로 표현하는 것

내 . 외부 시스템 식별

인터페이스 관련 자료들을 기반으로 개발할 시스템과 연계할 내. 외부 시스템들의 상세 식별 정보를 정의하고 목록을 작성하는 것이다.

내 . 외부 시스템 환경 및 관리 주체 식별

연계할 시스템 접속에 필요한 IP또는 URL, Port 정보 등 시스템의 실제 운용 환경을 의미한다.

관리 주체는 하드웨어를 실제적으로 관리하는 담당자를 의미한다.

내. 외부 시스템 네트워크 연결 정보 식별

시스템 로그인 및 DB정보를 의미한다.

인터페이스 식별

인터페이스 요구사항 명세서와 인터페이스 요구사항목록을 기반으로 개발할 시스템과 이와 연계할 내 . 외부 시스템 사이의 인터페이스를 식별하고 인터페이스 목록을 작성하는 것이다.

인터페이스 시스템 식별

인터페이스 별로 인터페이스에 참여하는 시스템들을 송신 시스템과 수신 시스템으로 구분하여 작성하는 것이다.

식별 대상 데이터

송 - 수신 시스템 사이에서 교환되는 데이터로 규격화된 표준 형식에 따라 전송한다.
교환되는 데이터의 종류에는 인터페이스 표준 항목, 송 수신 데이터 항목 공통코드가 있다.

인터페이스 표준항목

- 송 - 수신 시스템을 연계하는데 표준적으로 필요한 데이터를 말한다.
- 시스템 공통부와 거래 공통부로 나뉜다.

시스템 공통부	시스템 간 연동 시 필요한 공통 정보이다. 구성 정보에는 인터페이스 ID, 전송 시스템 정보, 서비스 코드 정보, 응답결과 정보, 장애 정보 등이 있다.
거래 공통부	시스템들이 연동된 후 송 수신 되는 데이터를 처리할 때 필요한 정보이다. 구상정보에는 직원 정보, 승인자 정보, 기기 정보, 매체 정보 등이 있다.

송수신 데이터 항목

- 업무를 수행하는데 사용하는 데이터이다.
- 전송되는 데이터 항목과 순서는 인터페이스 별로 다르다.

공통코드

- 시스템들에서 공통적으로 사용하는 코드이다.
- 연계 시스템이나 연계 소프트웨어에서 사용하는 상태 및 오류 코드등과 같은 항목에
대해 코드값과 코드 명, 코드 설명 등을 공통 코드로 관리한다.

정보 흐름 식별

정보 흐름을 식별하는 것은 개발할 시스템과 내 외부 시스템 사이에서 전송되는 정보들의
방향성을 식별하는 것이다.

- 개발할 시스템과 내 외부 시스템에 대한 각각의 인터페이스 목록을 확인하여 정보
흐름을 식별한다.
- 식별한 정보 흐름을 기반으로 송 수신 시스템 사이에서 교환되는 주요
데이터항목이나 정보 그룹을 도출한다.

송수신 데이터 식별

개발할 시스템과 연계할 내외부 시스템 사이의 정보 흐름과 데이터 베이스 산출물을
기반으로 송 수신 데이터를 식별한다.

- 송수신 데이터의 종류에는 인터페이스 표준 항목에 대한 데이터 항목과 코드성
데이터 항목이 있다.
- 인터페이스 표준항목과 송수신 데이터 항목 식별 : 송수신 시스템 사이의 교환
범위를 확인하고 인터페이스 표준 항목에 대해 송수신 데이터 항목을 식별한다.
- 코드성 데이터 항목 식별 : 코드성 데이터 항목에 대해 코드, 코드명, 코드 설명 등의
코드 정보를 식별한다.

인터페이스 방법 명세화의 개념

내 외부 시스템이 연계하여 작동할 때 인터페이스 별 송수신 방법, 송수신 데이터, 오류 식별 및 처리 방안에 대한 애용을 문서로 명확하게 정리하는 것이다.

인터페이스별로 송수신 방법을 명세화하기 위해서는 시스템 연계 기술, 인터페이스 통신 유형, 처리유형, 발생 주기 등에 대한 정보가 필요하다.

시스템 연계기술

개발할 시스템과 내 외부 시스템을 연계할 때 사용되는 기술을 의미한다.

주요 시스템 연계 기술에는 DB Link, Api/Open Api, 연계 솔루션, Socket, Web service등이 있다.

- **DB Link** : DB에서 제공하는 DB Link를 객체를 이용하는 방식이다.
- **API/Open API** 송신 시스템의 데이터베이스 (DB)에서 데이터를 읽어 와 제공하는 애플리케이션 프로그래밍 인터페이스 프로그램이다.
- 연계 솔루션 : EAI 서버와 송수신 시스템에 설치되는 클라이언트를 이용하는 방식이다.
- **Socket** : 서버는 통신을 위한 소켓을 생성하여 포트를 할당하고 클라이언트의 통신 요청시 클라이언트와 연결하여 통신하는 네트워크 기술이다.
- **Web Service** 웹 서비스에서 WSDL, UDDI, SOAP 프로토콜을 이용하여 연계하는 시스템이다.

* API (Application Programming Interface)/Open API

운영체제나 프로그래밍 언어 등에 있는 라이브러리를 응용 프로그램 개발 시 이용할 수 있도록 규칙 등에 대해 정의해 놓은 인터페이스를 말하고, OpenAPI는 이러한 기능을 누구나 무료로 사용하게끔 한 공개된 API를 말한다.

* WSDL(Web Services Description Language)

웹 서비스와 관련된 서식이나 프로토콜 등을 표준적인 방법으로 기술하고 게시하기 위한 언어

* UDDI(Universal Description Discovery and Integration)

인터넷에서 전세계의 비즈니스 업체 목록에 자신의 목록을 등록하기 위한 확장성 생성 언어(XML) 기반의 언어 규격

* SOAP(Simple Object Access Protocol)

웹 서비스를 실제로 이용하기 위한 객체 간의 통신 규약

인터페이스 통신 유형

개발할 시스템과 내외부 시스템 간 데이터를 송수신 하는 형태를 의미한다.

인터페이스 통신 유형에는 단방향,동기, 비동기 방식 등이 있다.

- **단방향** : 시스템에서 거래를 요청만 하고 응답이 없는 방식이다.
- **동기** : 시스템에서 거래를 요청하고 응답이 올 때까지 대기(Request-Reply)하는 방식이다.

- **비동기** : 시스템에서 거래를 요청하고 다른 작업을 수행하다 응답이 오면 처리하는 방식(Send-Receive, Send-Receive-Acknowledge, Publish-Subscribe)

인터페이스 처리 유형

인터페이스 처리 유형은 송수신 데이터를 어떤 형태로 처리할 것인지에 대한 방식을 의미한다.

실시간 방식 : 사용자가 요청한 내용을 바로 처리해야 할 때 사용하는 방식이다.

지연 처리방식 : 데이터를 매건 단위로 처리할 경우 비용이 많이 발생할 때 사용하는 방식이다.

배치방식 : 대량의 데이터를 처리할 때 사용하는 방식이다.

인터페이스 발생주기

개발할 시스템과 내외부 시스템 간 송수신 데이터가 전송되어 인터페이스가 사용되는 주기를 의미한다.

=====나머지 부분은 책을 참고하세요~ 다 똑같은 이야기 =====

미들웨어(Middleware) 개념 및 종류

미들(Middle)과 소프트웨어(Software)의 합성어로 운영체제와 응용프로그램, 또는 서버와 클라이언트 사이에서 다양한 서비스를 제공하는 소프트웨어이다.

미들웨어는 표준화된 인터페이스를 제공함으로써 시스템 간의 데이터 교환의 관성을 보장한다.

통신 제공 방법이나 기능에 따라 **DB, RPC, MOM, TP-Monitor, ORB, WAS** 등으로

구분한다.

DB(Data Base)

DB는 **데이터베이스 벤더**에서 제공하는 클라이언트에서 원격의 데이터 베이스와 연결하기 위한 미들웨어이다.

- **DB**를 사용하여 시스템을 구축하는 경우 보통 **2-Tier** 아키텍처라고 한다.
- 대표적인 **DB**의 종류에는 마이크로소프트의 **ODBC**, 볼랜드의 **IDAPI**, 오라클의 **Glue**등이 있다.

RPC(Remote Procedure Call)

원격 프로시저 호출은 응용프로그램의 프로시저를 사용하여 원격 프로시저를 마치 로컬 프로시저처럼 호출하는 방식의 미들웨어이다.

- 대표적인 **RPC**의 종류에는 이큐브시스템스의 **Entera**, OSF의 **ONC/RPC** 등이 있다.

MOM(Message Oriented Middleware)

MOM(메시지 지향 미들웨어)은 메시지 기반의 비동기형 메시지를 전달하는 미들웨어 이다.

- 온라인 업무보다는 이기종 분산 데이터 시스템의 데이터 동기를 위해 많이 된다.

- 대표적인 MOM의 종류에는 IBM의 MQ, 오라클의 Message Q ,JCP의 JMS등이 있다.

TP-Monitor

TP-Monitor(트랜잭션 처리 모니터) 는 항공기나 철도 예약 업무 등과 같은 온라인 트랜잭션 업무에서 트랜잭션을 처리 및 감시하는 미들웨어이다.

- 사용자 수가 증가해도 빠른 응답 속도를 유지해야 하는 업무에 주로 사용된다.
- 대표적인 TP-Monitor의 종류에는 오라클의 tuxedo, 티맥스소프트의 tmax 등이 있다.

ORB(Object Request Broker)

ORB(객체 요청 브로커)는 객체 지향 미들웨어로 코바 표준 스펙을 구현한 미들웨어이다.

WAS(Web Application Server)

정적인 콘텐츠를 처리하는 웹서버와 달리 사용자의 요구에 따라 변하는 동적인 콘텐츠를 처리하기 위해 사용되는 미들웨어이다.

- 클라이언트 서버 환경보다는 웹 환경을 구현하기 위한 미들웨어이다.
- HTTP 세션 처리를 위한 웹 서버 기능 뿐만 아니라 미션-크리티컬한 기업 업무까지 JAVA,EJB 컴포넌트 기반으로 구현이 가능하다.

미들웨어 솔루션 식별

개발 및 운영 환경에 사용될 미들웨어 솔루션을 확인하고 목록을 작성하는 것이다.

- 소프트웨어 아키텍처에서 정의한 아키텍처 구성 정보와 프로젝트에서 구매가 진행중이거나 구매 예정인 소프트웨어 내역을 확인하여 개발 및 운영 환경에서 사용될 미들웨어 솔루션을 식별한다.
- 식별한 미들웨어 솔루션들에 대해 솔루션의 시스템, 구분, 솔루션명, 버전,제조사 등의 정보를 정리한 미들웨어 솔루션 목록을 작성한다.
- 작성된 미들웨어 솔루션 목록은 이해관계자 등에게 전달하여 오류 및 누락을 확인하고 수정한다.