

프로젝트를 처음 시작할 때 아무리 시간이 촉박해도 작성해야 하는 문서가 딱 두가지가 있다. 이것은 애자일 지향하던, RUP에 맞추던 어떠한 방법론을 사용하더라도 작성하게 된다. 문서의 양은 많아야 30-50 페이지를 넘지는 않는다. 그것은 Sofeware Requirments Specification(이하 SRS), Software Architecture Document(이하 SAD) 이다. 너무나 시간이 촉박할 경우엔 SAD를 생략하기도 하지만 개발이 끝난 후 소프트웨어로 반드시 SAD를 작성하여 회고해야 한다. 우리를 실패를 통해 배우기 때문이다.

오늘은 그 중 첫번째인 SRS에 대해서 이야기 하고자 한다. 특히 SRS 가 중요한 것은 스펙을 정함으로써 이후 모든 개발의 모든 분야 일정 예측, 설계, 구현, 테스트의 기준이 되면서, 개발 진행 상황 파악, 의사소통을 할 수 있기 때문이다. 십여년전에 작성을 처음 시작할때는 적절한 템플릿도 없이 생각 나는대로 주욱 작성하였다가 이미 표준화된 양식이 있음을 안 뒤부터는 템플릿에 따라서 작성하였다. 하지만 기계적으로 작성하면서 문서를 위한 문서가 되는 경우도 있었다. 그러한 멍청한 과정을 거치면서 수십차례의 SRS 문서를 작성하고나니 나름대로의 통찰력을 얻었고 과연 이 템플릿의 각 항목에서 무엇을 말해야 하는가를 깨닫게 되었다. 각 목차에 대해서 그것이 무엇을 의미하는가에 대해 설명하려고 한다.

아래 내용중 목차와 이탤릭체 설명은 정보 문화사에서 출판한 "소프트웨어 요구사항 패턴 " 에 나온 것을 인용한 것이다. 그 아래에 내가 가진 생각들과 예문들을 시간 날때마다 추가하도록 하겠다.

SoftwareRequirements Specification

1 소개(Introduction)

1.1 목적(Purpose)

이 문서에요구사항이 명시되어있는 제품또는 애플리케이션을설명한다. 이 SRS가전체 시스템중 일부에만관련된 것이라면그 부분또는 하위시스템을 설명한다.

1.2 문서규칙(DocumentConvention)

텍스트 스타일, 하이라이트 또는 주석과 같은 모든 표준 또는 표기규칙을 설명한다.

1.3 독자대상과 읽는 방법(Intend Audience and Reading Suggestion)

SRS가대상으로 하고있는 다양한독자계층을 나열한다. SRS의 나머지 부분과, SRS가 조직되어 있는 방법을 설명하고, 각각의 독자 계층에 대해 가장 적합한 읽기 순서를 설명한다.

1.4 프로젝트 범위(Project Scope)

설명되고 있는소프트웨어와 그목적에 대해간단하게 설명한다. 소프트웨어를 사용자 또는 기업의 목표, 비즈니스 목표, 전략과연계시킨다. 별도의 비전과 범위 문서를 사용할 수 있다면 그 내용을 중복시키지 말고 그것을 참조하게 한다. 진화하는 제품의 특정 버전을 설명하는 SRS는 장기적인 전략적 제품 버전의 하위집합으로 자신만의 범위 선언을 가지고 있어야 한다.

1.5 참조(Reference)

이 SRS가 참조하고 있는 모든 문서 또는 다른 리소스를 나열하며, 가능한경우에는 하이퍼링크도포함시킨다. 여기에는 사용자 인터페이스 스타일 가이드, 계약, 표준, 시스템 요구사항 명세, 유스케이스 문서, 인터페이스명세, 운영개념문서, 또는 관련 제품의 SRS도 포함된다. 읽는 사람이 제목, 저자, 버전 번호, 날짜, 소스 또는 위치 등의 참조를 이용할 수 있도록 충분한 정보를 제공한다.

2 전체 설명(Overall Description)

2.1 제품조망(ProductPerspective)

제품의 구성과유래를 설명한다. 제품이 확장되는 제품군의 다음 구성제품인지, 완성된 시스템의 다음 버전인지, 기존애플리케이션을 대체하는것인지, 완전히 새로운 제품인지를 설명한다. 이 SRS가 대규모 시스템의 컴포넌트를 정의하는 것이라면, 이소프트웨어가 전체시스템과 어떻게연계되는 지를설명하고 둘간의 주요인터페이스를 설명한다.

2.2 제품기능(ProductFeature)

제품이 가지고있는 주요기능 또는제품이 수행하는중요한 기능을나열한다. 상세한 내용은 SRS의 3번 절에 설명되어 있기 때문에, 여기에서는 추상적인 요약만 하면 된다. 요구사항의 주요 그룹과 그 그룹이 연결되어 있는 방법을 설명하는 최상위 데이터 플로우 다이어그램, 유스케이스 다이어그램, 클래스 다이어그램 등이 도움이 된다.

2.3 사용자 계층과 특징(User Classes and Characteristic)

이 제품을사용할 것으로예상되는 사용자계층을 파악하고그들의 특징을설명한다. 일부 요구사항은 특정 사용자 계층에만 해당될 수 있기 때문에, 선호하는사용자 계층을파악한다. 사용자 계층은 비전과 범위 문서에 설명되니 관련자들의 하위집합을 의미한다.

2.4 운영환경(OperationEnvironment)

하드웨어 플랫폼, 운영체계와 버전, 사용자, 서버와 데이터베이스의 지리적 위치 등과 같은 소프트웨어가 동작되는 환경을 설명한다. 시스템이 아무런 문제없이연동해야 하는다른 소프트웨어컴포넌트 또는애플리케이션을 나열한다. 비전과 범위 문서는 추상적인 수준에서 이 정보를 포함하고 있을 수 있다.

2.5 설계 및 구현 제약사항(Design and Implementation constraint)

개발자가 선택할수 있는사항을 제약하는모든 요소와각 제약조건의이유를 설명한다. 제약 조건은 다음과 같다.

- 반드시 사용하거나 피해야 하는 특정 기술, 툴, 프로그래밍 언어와 데이터 베이스
 - 사용될 웹 브라우저의 유형과 버전과 같이 제품의 운영환경으로 인한 제약
 - 필요한 개발 규칙 또는 표준(예를들면 고객의조직이 소프트웨어를유지보수 할예정이라면, 그 조직은 하청업체가 따라야 하는 설계 표기법과 코딩 표준을 명시할 수 있다.)
 - 이전 제품과의 호환성
 - 비즈니스 규칙에 따른 제약
 - 메모리 또는 프로세스의 제약, 크기, 무게, 비용과 같은 하드웨어의 제약
 - 기존 제품을 개선하는 경우에 따라야 하는 기존 사용자 인터페이스 규칙
 - XML과 같은 표준 데이터 교환 형식
- 2.6 사용자 문서(User Documentation)
- 소프트웨어와 함께제공할 사용자문서를 나열한다. 사용자 문서로는 사용자 매뉴얼, 온라인도움말, 교재 등이 있으며 따라야 하는 문서 전달 형식, 표준 또는 툴이 있다면 그것들을 설명한다.
- 2.7 가정과 종속관계(Assumptions andDependencies)
- 가정은 물증또는 확실한지식이 없는경우에 사실이라고믿는 선언으로, 이 가정이 잘못되거나 이것을 공유하지 않는다면 문제가 발생할 수 있기 때문에 어떤 가정은 프로젝트 위험으로 간주된다. SRS를 읽은 사람은 그 제품이 특정 사용자 인터페이스 규칙을 따르고 있다고 가정할 수 있는 반면에, 다른사람은 다른생각을 할수 있다. 개발자는 특정 기능들이 사용자 정의로 작성된다고 가정하지만, 분석가는 이것들이 이전의 프로젝트에서 재사용된다고 믿고 있으며 프로젝트 관리자는 상업용 함수 라이브러리를 구매해야 한다고 생각할 수 있다.
- 운영체계의 다음버전 발표되는 산업표준발표와 같이프로젝트가 통제할수 없는외부 요소에어느정도 종속되는지를설명한다. 다른 프로젝트가 개발하고 있는 어떤 컴포넌트를 시스템에 통합하려고 한다면, 그프로젝트가 해당컴포넌트를 제시간에제공하는 것을기다려야 한다. 이런 종속관계가 프로젝트 계획과 같은 다른 문서에 이미 정리되어 있다면 그 문서들을 참조하도록 한다.

3 유스케이스(Usecase)

3.1 Usecase 이름

가 요약시나리오

나 이벤트 흐름

① 기본 흐름

② 예외흐름

다 사전조건

라 사후조건

마 쟁점

바 현실화

4 시스템 특징(System Feature)

4.1 시스템특징 X

3.X 철자확인과 같은몇 단어만으로특징을 설명한다. 각각의 시스템 특징에 대해서는 3.x.1 에서 3.x.3까지의 하위 절을 반복한다.

가 설명과 우선순위(Description and Priority)

기능에 대해간단하게 설명하고그것이 높은우선순위인지 낮은우선순위인지를 나타낸다. 우선순위는 프로젝트 중에 변할 수 있는 동적인 것으로, 요구사항관리 툴을사용한다면 요구사항특성의 우선순위를정의한다.

나 자극/응답 순서(Stimulus/Response Sequence)

입력 자극(사용자 행동, 외부 장비의 신호 또는 다른 자극)의순서와 이기능에 대한동작을 정의하는시스템 반응을나열한다. 이 자극들은 유스케이스의 초기 대화단계 또는 외부 시스템 이벤트에 해당한다.

다 기능요구사항(Functional requirement)

이 기능과관련된 상세한기능 요구사항을함목으로 나열한다. 이것들은 사용자가 기능의 서비스를 수행하기 위해 또는 유스케이스를 수행하기 위해 사용하는 소프트웨어의 기능들이다. 제품이 예상되는 예러 상황, 무효한입력과 동작에대해 어떻게응답해야 하는지를 설명한다. 각각의 기능 요구사항에 유일한 레이블을 붙인다.

5 외부 인터페이스 요구사항(External InterfaceRequirement)

5.1 사용자 인터페이스

- 시스템이 요구하는 각각의 사용자 인터페이스의 논리적인 특징을 설명한다. 따라야 할 GUI표준또는 제품스타일 가이드에대한 참조
- 폰트, 아이콘, 버튼 레이블, 이미지, 색상 체계, 필드명 순서, 공통으로 사용되는 컨트롤 등에 대한 표준
- 화면 레이아웃 또는 해상도 제약 조건
- 도움말 버튼과 같이 모든 화면에 나타나는 표준 버튼, 기능 또는 탐색 링크
- 단축키
- 메시지 표시 규칙
- 소프트웨어 번역을 원활하게 하는 레이아웃 표준
- 시각장애자를 위한 기능

사용자 인터페이스의설계 상세내용은 SRS가 아닌 별도의 사용자 인터페이스 명세에 문서로 정리한다.

5.2 하드웨어 인터페이스(Hardware Interface)

시스템의 소프트웨어와하드웨어 컴포넌트간의 모든인터페이스의 특징을설명한다. 설명에는 지원되는 장비 유형, 소프트웨어와 하드웨어간의 데이터와 컨트롤 연동, 사용될 통신 프로토콜 등이 포함된다.

5.3 소프트웨어 인터페이스(Software Interface)

이 제품과다른 소프트웨어컴포넌트(데이터베이스, 운영체제, 툴, 라이브러리, 통합 상업용 컴포넌트)간의 연결을설명한다. 소프트웨어 컴포넌트 간에 교환되는 메시지, 데이터와컨트롤 항목을설명한다. 외부 소프트웨어 컴포넌트가 요구하는 서비스와 컴포넌트 간의 통신 성격을 설명하고 소프트웨어 컴포넌트들이 공유할 데이터를 파악한다.

5.4 통신 인터페이스(CommunicationsInterface)

이메일, 웹 브라우저, 네트워크 통신 프로토콜, 전자 문서와 같이 제품이 사용할 모든 통신 기능에 대한 요구사항을 설명한다. 관련된 모든 메시지 형태를 정의하고 통신 보안 도는 암호화 문제, 데이터전송률과 동기화메커니즘을 명시한다.

6 기능 이외의 다른 요구사항(Other Nonfunctional Requirement)

6.1 성능 요구사항(PerformanceRequirement)

다양한 시스템운영에 대한특정 성능요구사항을 설명한다. 개발자들이 적합한 설계를 선택할 수 있게 만든 논리를 설명한다. 예를 들면 엄격한 데이터 베이스 응답시간 때문에 설계자들은 여러 위치에 데이터베이스를 미러링 하거나 더 빠른 질의응답을 위해 데이터베이스 테이블의 정규화를 해제시킬 수 있다. 지원되어야 하는 초당 트랜잭션 수, 응답 시간, 연산의정확도와 실시간시스템의 속도조절관계를 명시한다. 또한 메모리와 디스크 공간 요구사항, 동시 사용자 부하 또는 데이터베이스 테이블에 저장되는 최대 row 수를 명시한다.

성능 요구사항은가능한 분명하게계량적으로 표현한다. 예를 들면 “MS XP에서 1GHz P4환경에서 메모리가 60%의 여유가 있는 상태에서 데이터 베이스의 질의 중 95%가 3초 내에 완료된다” 는 식으로 표현한다.

6.2 안전 요구사항(Safety Requirement)

반드시 방지해야하는 잠재적으로위험한 행동뿐만아니라 반드시취해야 할모든 안전장치또는 행동을정의한다. 제품이 따라야 하는 보안인증, 정책 또는 규제를 정의한다.

6.3 보안 요구사항(Security Requirement)

제품에 대한접속과 제품사용에 영향을미치는 보안, 무결성 또는 사생활 문제, 제품이사용하거나 만드는데이터 보호를모두 명시한다. 보안 요구사항은 일반적으로 비즈니스 규칙에서 만들어 지기 때문에, 제품이준수해야 하는모든 보안, 사생활 정책 또는 규제를 모두 명시한다. 이것대신에, 무결성이라고 부르는 품질 특성을 통해 이 요구사항들을 해결할 수 있다. 다음은 보안 요구사항의 예들이다.

SE-1 모든사용자는 첫번째로그인 에성공한 후에처음에 할당한로그인 암호를즉시 변경해야한다. 초기 암호는 절대로 재사용되지 않는다.

6.4 소프트웨어 품질 특성(Software Quality Attribute)

고객 또는개발자에게 중요하모든 별도의품질 특성을설명한다. 이런 특성들은 명확하고 계량적이며 확인이 가능해야 한다.

7 다른 요구사항(Other Requirement)