



### 1. 서론

미국 Standish 보고에 따르면, 소프트웨어 프로젝트의 성공률은 매우 낮은 수준으로, 실패 원인의 49.3%가 사용자와의 비효율적인 의사소통으로 인한 불완전한 요구사항의 정의와, 이로 인한 잦은 요구사항의 변경인 것으로 나타나고 있다[1]. 즉 프로젝트의 실패가 적용 기술이나 방법론의 부족에 기인하기보다는, 고객의 요구를 이해, 분석, 문서화하고, 변경을 관리 통제하는 기술과 방법의 문제인 것으로 나타났다(그림 1). 이러한 문제는 국제표준화 기구인 ISO/IEC JTC1/SC7 WG4에서 조사한 요구사항관리에 대한 현황 조사 보고서에도 잘 나타나 있다(그림 2).

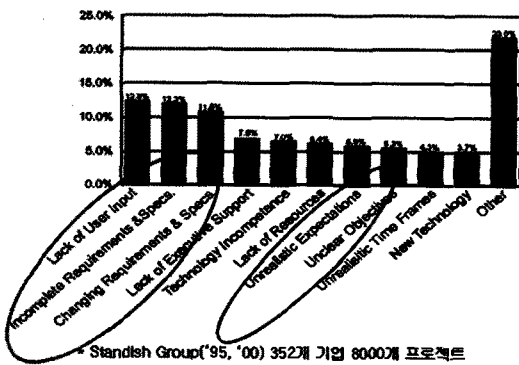


그림 1 소프트웨어사업 실패원인(Standish Group)

보고서에는 대부분의 요구사항 관리 활동들이 그 중요성에 비해 실제 수행 시의 만족도에 있어서는 매우 낮은 것으로 나타났는데, 그 중에서도 특히 요구사항 도출 및 정의 활동, 요구사항 문서화 및 보고 활동, 변경 관리 활동 등의 만족도가 타 요구사항 관리 활동들보다 상대적으로 낮음을 알 수 있다[2].

이러한 문제들의 근본 원인은 소프트웨어 프로젝트 초기에 다양한 이해당사자로부터 도출되는 여러 형태의

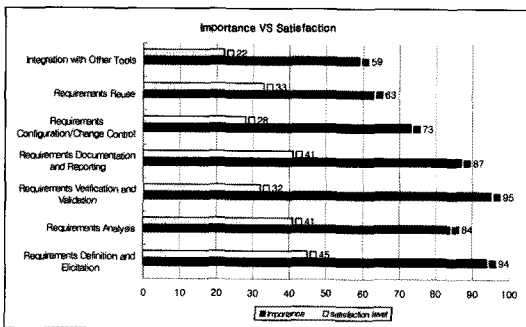


그림 2 요구사항 관리의 어려움

(시스템 요구사항, 소프트웨어 요구사항, 기능/품질 요구사항, 제약사항 등) 요구사항을 상세하고 명확히 명세하지 못하고, 이를 그대로 설계, 구현, 테스트, 유지보수 등의 활동으로까지 이어가기 때문이다. 불명확한 요구사항은 이해당사자간의 합의를 이끌어 내지 못하여 추후 잦은 요구사항 변경을 초래하게 되고, 이는 전체 소프트웨어 개발 및 관리 활동을 어렵게 하는 요인이 된다.

많은 연구에서 이러한 문제점을 적시하고 나름대로의 해결책을 제시하고는 있지만, 대부분의 경우, 고객으로부터의 잦은 요구사항의 변경을 막기 위한 더욱 엄격하고 제한된 명세와 변경만을 강조하고 있다. 이러한 방법은 요구사항의 변경 빈도를 최대한 억제하는 효과를 얻을 수 있지만 현실적이지는 못하다. 고객으로부터의 요구사항 변경 요청을 억제하게 되면 고객이 원하지 않는 제품이 인도될 수 밖에 없고 결국 개발 제품의 가치는 소멸하게 된다.

본 논문에서는 소프트웨어 개발 범위, 내용, 품질목표, 수·발주자간의 상호 기대 등을 사업초기부터 요구사항에 명확하게 명세하면서도 요구사항 변경에도 유연하게 대처할 수 있는 요구사항 명세 표준을 제시한다. 제시하는 명세표준은 사용자 중심의 효율적 의사소통 및 협력의 기반이 되는 요구사항 명세, 명세서 작성 노력을 줄이면서 측정 가능한 요구사항 명세, 그리고 변경 및 추적 가능한 요구사항 명세를 지향한다[3]. 이러한 명세 목표를 달성하기 위해, 명세서가 가져야 할 세부 특성으로는 ① 요구사항 명세관련 이해당사자의 명확한 정의, ② 요구공학 활동에서 산출되는 다양한 형태의 요구사항 반영, ③ 요구사항으로부터 사업의 규모, 기간 등의 대가 산정 가능, ④ 요구사항의 이해와 해석을 명확하게 하기 위한 명세 지침, ⑤ 요구사항 기준선 및 변경기준선의 정의 및 관리 등 5가지로 요약될 수 있다(그림 3).

본 논문은 다음과 같이 구성된다. 2절에서는 기존 국내외에서 활용되고 있는 요구사항 명세서를 비교 분석하며, 3절에서는 이를 바탕으로 본 논문에서 제시하는 표준 요구사항 명세서의 구성 요소 및 내용을 기술한다.

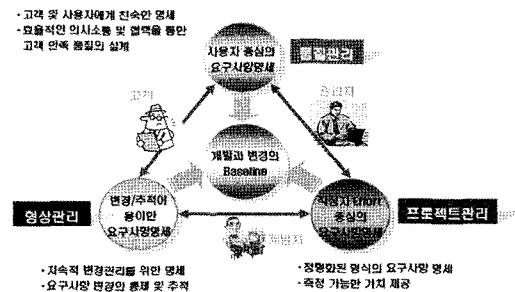


그림 3 요구사항 명세가 갖추어야 할 특성

4절에서는 명세서의 활용 방안 및 사례를 소개하고, 마지막 5절에서는 결론 및 향후 연구에 대해 기술한다.

## 2. 기존 요구사항 명세의 현황 및 진단

기존의 요구명세서가 갖는 특성들을 파악하기 위해 현재 국내외적으로 널리 활용되고 있는 NASA, DOD, IEEE, 그리고 국내 A, B사의 표준 요구명세서를 비교 분석하였다. 분석은 1절에서 기술한 5가지 특성과 [4]의 연구결과를 토대로 수행되었다.

### 2.1 NASA DID P200

NASA 프로젝트의 특성(임베디드 소프트웨어 개발)을 반영하여 품질 요구사항에 대해 정량화된 명세를 요구하고 있으며(1절의 기준 3), 반복 점증적 개발에 따른 요구사항의 변경/추가 발생 시, 이를 향후 개발 활동으로(설계, 구현, 테스트, 유지보수 등) 연결/추적하기 위한 추적 기능을 강조하고 있다(1절의 기준 5). 하지만 하나의 요구사항이 형태(시스템 요구사항, 기능 요구사항, 품질 요구사항, 제약사항 등)에 따라 여러 조각으로 분산되어 명세되기 때문에 동일 수준(계층)의 요구사항 간의 연결/추적 기능은 제공하지 못한다. 또한 요구사항의 명확한 이해와 해석을 위한 정형화된 지침이 부족하며(1절의 기준 4), 유즈케이스 명세와 같은 기본 설계에 대한 명세 수단이 없어(1절의 기준 1, 2) 설계 및 구현 활동에 유용한 정보를 제공하지 못한다[5-8].

### 2.2 DOD 498

미 국방성에서 오랫동안 사용해왔던 일반 소프트웨어 개발 표준인 7935A와 임베디드 소프트웨어 개발 표준 2167A를 최근 하나의 표준(498)으로 통합하여 통일성을 제고한 것이 특징이다. 이 표준은 프로젝트의 특징에 맞게 명세서를 테일러링할 수 있는 상세한 지침과 풍부한 예제를 제공하며(1절의 기준 4), 임베디드 소프트웨어 개발을 위한 품질 요구사항에 대한 정량화를 강조하고 있는 것이 장점이다(1절의 기준 3). 단점으로는 요구사항이 여러 구성 요소로 분산되지 않아 모듈화는 높아지지만, 대량의 요구사항이 발생할 경우, 이를 하위 계층으로 분해하여 명세할 수 있는 방법이 없으며, 따라서 상하위 요구사항들간의 연결/추적 기능이 강조되지 않는다(1절의 기준 5). 또한 NASA의 경우에서와 같이, 유즈케이스와 같은 기본 설계에 대한 명세 수단이 없어(1절의 기준 1, 2) 설계 및 구현 활동에 유용한 정보를 제공하지 못한다[9,10].

### 2.3 IEEE 830

일반적으로 가장 많이 활용되는 표준으로, 다양한 프로젝트 및 조직의 환경에 맞게 명세서를 테일러링하기 쉬우며, 풍부한 예제와 지침을 제공하고 있다(1절의 기준 4). 또한 유즈케이스 명세와 같은 다양한 명세 수단을 프로젝트의 특성에 맞게 선택적으로 적용할 수 있는

것이 장점이다(1절의 기준 1, 2). 하지만 요구사항의 계층적 분해가 지원되지 않으며, 요구사항들간의 연결/추적 기능이 없기 때문에 반복/점증적인 개발 시 필연적으로 발생하는 요구사항의 추가/변경을 적절히 지원하지 못한다(1절의 기준 5). 따라서 기본적으로는 폭포수 모형 기반의 소프트웨어 개발에 가장 적합한 구조를 가지고 있다[11]. 또한 품질 요구사항에 대한 정량화 기능도 지원하고 있지 못하다(1절의 기준 3).

### 2.4 국내 A, B 사

국내의 A사에서 사용하고 있는 명세서의 경우, 유즈케이스 명세는 요구하고 있지 않으나, 대신에 비즈니스 프로세스 분석을 통해 현재 시스템과 목표 시스템과의 차이 분석 결과를 명세하도록 하고 있으며, 이를 통해 기능 및 비기능 요구사항을 도출할 수 있도록 구성되어 있다(1절의 기준 1, 2). B사의 경우, 품질 요구사항에 대한 목표 기대치, 우선순위 등을 상세히 명세하도록 하고(1절의 기준 3), 품질 요구사항과 기능적 요구사항간의 연결 관계를 명세하도록 하고 있다. 하지만 두 경우 모두, 요구사항의 계층적 분해는 지원하지 않으며, 따라서 각 종 요구사항(기능 요구사항, 데이터 요구사항, 품질 요구사항, 인터페이스 요구사항, 운영 요구사항)들이 서로 혼재되어 있어 요구사항의 기준/변경선 관리가 용이하지 않다(1절의 기준 5). 또한 다양한 프로젝트 및 조직의 특성이나 환경에 맞도록 명세서를 테일러링하기

표 1 국내의 요구사항 명세의 특징

NASA-DID-P200	DoD MIL-STD-498	IEEE 830-1998	국내 A, B 사
<ul style="list-style-type: none"> <li>· 품질 요구사항의 정량화 강조</li> <li>· 요구사항의 계층별 추적은 지원되나 단일 요구사항이 여러 색선으로 분산되는 경우 추적성 결여</li> <li>· 테일러링 등의 적용 지침 부족</li> <li>· 유즈케이스 명세 수단이 없어 기본 설계에 대한 정보를 제공하지 못함</li> </ul>	<ul style="list-style-type: none"> <li>· 품질 요구사항의 정량화 강조</li> <li>· 풍부한 적용 가이드라인 제공</li> <li>· 요구사항의 계층적 분해 및 추적 관리 어려움</li> <li>· 유즈케이스 명세 수단이 없어 기본 설계에 대한 정보 제공하지 못함</li> </ul>	<ul style="list-style-type: none"> <li>· 작성 가이드라인 및 풍부한 예제 제공</li> <li>· 프로젝트/조직 환경 별로 SRS 구성요소의 커스텀마이징 용이</li> <li>· 요구사항의 계층적 분해 및 추적 관리가 어려워 폭포수 모델에 적합</li> <li>· 품질 요구사항에 대한 정량화를 지원 못함</li> </ul>	<ul style="list-style-type: none"> <li>· 요구사항 목록 정리 수준</li> <li>· 요구사항의 계층적 분해 및 추적 관리가 어려움 / 요구사항 목록 수준에서의 추적성은 제공</li> <li>· 유즈케이스 명세 수단은 지원하지 않으나 현재 시스템 현황 및 목표 시스템과의 GAP분석 지원</li> <li>· 비즈니스 업무 프로세스가 요구사항으로 연결</li> </ul>

표 2 국내의 요구사항 명세서의 구성요소 별 비교

구분	NASA-DID-P200	DoD MIL-STD-498	IEEE 830-1998	국내 A, B 사
Req. Req	X	X	X	O
Interface Req	외부 IF	내/외부 IF	외부 IF	X
Functional Req	O	O	O	O
Data Req	O	O	기능요구에 포함	O
Use Case	X	X	O	X
Quality Req	O	O	O	O
Operation Req	X	O	X	O
Constraints	O	O	O	O
Traceability	O	O	X	O

어려우며, 관련 지침서도 존재하지 않거나 상세하지 않다(1절의 기준 4). 분석결과에 대한 요약은 표 1, 2와 같다. 표 2는 기준 1과 2에서 제시하고 있는 다양한 이해당사자 및 요구사항을 분석 대상 명세서가 얼마나 지원하고 있는지를 보여준다.

### 3. 요구사항 명세화 표준지침 개발

#### 3.1 표준 명세서의 구성요소

본 논문에서 제안하는 표준 명세서의 요구사항 구성은 크게 비즈니스 요구사항, 기술 요구사항, 그리고 제약 요구사항으로 나뉘며, 각 범주의 세부 항목은 그림 4와 같다. 표준 명세서는 이들 항목들이 상호 유기적인 관계를 가지면서 이해당사자들간의 상호 의사소통의 기반이 될 수 있도록 요구사항들간의 연결/추적, 요구사항의 정량화, 명세의 정량화 수단을 지원한다(그림 5).

연결/추적 기능은 1절에서 요구사항 명세서의 특성으로 제시된 '요구사항의 기준선과 변경 기준선의 정의 및 관리'를 지원하며, 정량화 기능은 '요구사항으로부터 사업의 규모, 기간 등의 대가를 산정'하기 위한 수단으로, 정형화는 '다양한 이해당사자 및 요구사항의 반영과 반영된 요구사항의 이해와 해석을 명확'하게 하기 위한 수단으로 지원한다.

그림 6은 표준명세서의 목차 템플릿을 보여주고 있다. 그림에서 항목 3과 4는 비즈니스 목표, 전략, 현 시스템의 문제와 관련된 요구사항을, 항목 5, 6, 7, 8은 기술요구사항을, 항목 9는 제약 요구사항을 나타내고 있다.

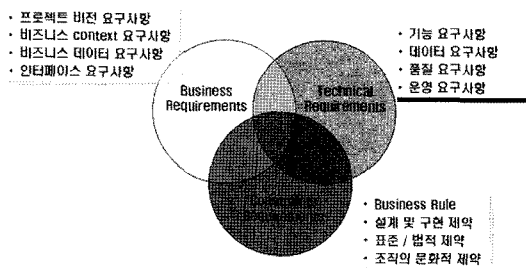


그림 4 표준 명세서의 요구사항 유형

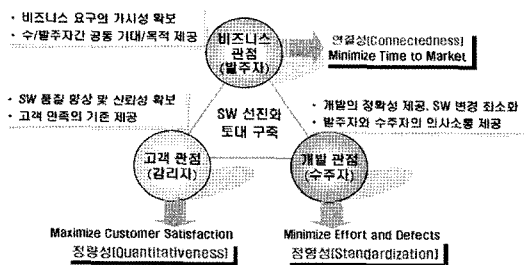


그림 5 요구사항의 연결/추적성, 정량성, 정형성

1. SRS 소개 (Introduction to SRS)
2. 시스템 개요 (Overall description of System)
3. 비즈니스 요구사항 (Business requirements)
4. 인터페이스 요구사항 (Interface requirements)
5. 기능 요구사항 (Functional requirements)
6. 데이터 요구사항 (Data requirements)
7. 품질 요구사항 (Quality requirements)
8. 운영 요구사항 (Operational requirements)
9. 제약사항 (Constraints)
10. Use Case 명세 (Use case Specification)
11. 요구사항 추적 매트릭스 (Traceability)

그림 6 요구사항 명세서의 구성 목차

며, 항목 10은 기본 설계 정보의 제공을 위한 유즈케이스 명세를 제공하고 있다. 또한 명세된 모든 요구사항에 대한 추적성을 제공하기 위해 추적 테이블을 지원한다. 요구사항의 정량화와 정형화는 각 항목 별로 템플릿과 지침을 통해 지원한다.

#### 3.2 연결/추적성

표준 명세서는 다양한 계층 및 범주의 요구사항을 상호 연결/추적/관리할 수 있는 수단을 지원한다. 먼저, 요구사항의 근거와 원인이 되는 비즈니스 프로세스 목표 및 전략, 현재 시스템과 목표시스템과의 차이, 현재 시스템의 문제점 등이 해당 요구사항으로 수직적으로 연결/추적될 수 있도록 지원한다(표 3). 또한 하나의 요구사항이 여러 요구사항 범주(기능요구사항, 데이터요구사항, 인터페이스 요구사항, 품질요구사항, 운영요구사항 등)에 분산되어 명세가 되는 경우, 이들 요구사항들간의 횡적 연결/추적도 가능하다(표 4-6). 비즈니스 프로세스 목표 및 전략은 다음과 같은 다양한 관점에서 기술될 수 있도록 구성되며, 각 관점 사항들은 해당 요구사항으로 연결/추적된다.

- ① 산출물 관점: 사업 비전/목표, 범위
- ② 사용자 관점: 그룹 식별, 영향도, 우선순위
- ③ 사업 관점: 비용, 일정, 노력, 품질

표 3 비즈니스 목적 및 프로세스와 해당 요구사항의 수직적 연결/추적

비즈니스 목적	비즈니스 프로세스	시스템 기능	소프트웨어기능
고객만족	회원관리	회원기업관리	회원정보 등록
			회원정보 테이블
			회원정보 삭제
			회원정보 수정
			회원정보 조회
			비밀번호관리
실명확인관리	비밀번호 등록		
		실명확인 인증	
	회원검색	...	...

표 4 기능요구사항들 간의 연결/추적

RID	관련 요구사항
SFR003	SFR003.01, SFR003-002, SMR-010

표 5 기능요구사항과 품질요구사항의 연결/추적

기능 RID	신뢰성	사용성	유지 보수성	효율성	이식성	보안성
SFR003비밀번호등록	QRR-1	QRU-4	-	QRE-3	-	QRS-2

표 6 기능요구사항과 소프트웨어 프로세스 산출물과의 연결/추적

요구사항 ID	요구사항 이름	출처	모델 문서	설계 문서	프로그램 모듈	테스트 케이스
SFR003	비밀번호 등록	회의록 0810	DFD-003	SC-003	FM-003	TC-003

④ 업무 관점: 업무 분할

⑤ 제약 관점: 제약사항, 운영환경, 외부 환경

**3.3 정형화**

표준명세서는 사용자가 명세서를 쉽게 작성하고, 작성된 내용의 명확한 이해와 해석을 돕기 위해 요구 명세서의 정형화(문장형식의 템플릿화)를 지원한다(표 7-9). 또한 프로젝트의 규모와 관련 이해당사자의 요구사항 명세에 대한 숙련 정도 등을 고려하여 요구 명세서의 구성을 테일러링할 수 있도록 지원한다(표 10).

**3.4 정량화**

표준명세서는 명세된 요구사항으로부터 기능접수(Function Point) 등 사업대가 산출에 사용되는 지표의 획득이 가능하도록 지원한다(표 11). 이를 위해 각 요구사항은 기능에 대한 명세뿐만 아니라 입출력 유형, 파일유형, 각 기능에 관련된 입출력 데이터를 함께 제공함으로써 소프트웨어 규모를 예측할 수 있다. 또한 각 요구사항에 대한 고객의 등급이나 우선순위, 적합성(선택/필수) 수준, 위험수준 등도 정량화함으로써 가치 기반의 요구사항을 실현한다.

표 7 요구사항 명세에 대한 기본정보 지원

정보 종류	정보 내용
식별 정보	<ul style="list-style-type: none"> <li>요구사항 번호, 요구사항 이름, 관련자</li> <li>요구사항 내용, I/O 정보, Fit Criteria, 지원 산출물</li> </ul>
속성 정보	<ul style="list-style-type: none"> <li>요구사항 종류 (비즈니스, 기능, 품질, 운영, 제약사항)</li> <li>Compliance level (mandatory, optional, desired)</li> <li>필리츠 정보, 우선순위, 중요도, 위험도, effort, 비용, 일정</li> </ul>
요구사항 품질 정보	<ul style="list-style-type: none"> <li>관련 품질 속성, 측정 기준, 방법</li> <li>요구사항 충돌 정보</li> </ul>
상태 정보	<ul style="list-style-type: none"> <li>요구사항 처리 상태</li> </ul>
요구사항 변경 정보	<ul style="list-style-type: none"> <li>요구사항의 변경 가능성 (global/static requirements)</li> <li>내역정보, 변경 처리 상태</li> </ul>
요구사항 추적 정보	<ul style="list-style-type: none"> <li>요구사항과 관계된 근거/소스 및 산출물과의 추적 정보</li> <li>다른 요구사항과의 종속성</li> </ul>

표 8 명세서의 정형화를 위한 템플릿 구조 및 관련지침 예

<b>명세 개요</b>
시스템이 제공하는 효율성을 특정 시간이나 수준에서 계량적으로 기술한다.
<b>명세 내용</b>
(1) 시간 효율성 : 태스크를 완성하기 위하여 가용한 시간의 양을 기술한다.
· 측정 대상 정보 :
- 평균 / 피크타임 / 부하상태에서의 처리시간(transaction time)
- 응답시간(response time) 또는 반응시간(turnaround time)
- 데이터 전송속도/시간
· 측정 방법 및 기준 : 측정 대상 정보의 측정 방법과 목표 값을 기술
(2) 자원 효율성 : RAM, 프로세서, 처리장치, 디스크 등의 사용 자원의 효율성을 기술한다.
· 측정 대상 정보 : 메인 메모리 점유율 제한 기준, CPU 부하율
· 측정 방법 및 기준 : 측정대상정보의 측정방법과 목표 값을 기술
(3) 처리 효율성 : 시스템이 처리가능한 최대부담을 기술한다.
· 측정 대상 정보 :
- 동시 접속자 수, 동시 트랜잭션 사용자 수
- 시스템이 예상되는 부하를 처리하는 능력(throughput), 태스크 수
· 측정 방법 및 기준 : 측정 대상 정보의 측정 방법과 목표 값을 기술
<b>명세 기준</b>
· 측정 단위 및 요구되는 값의 범위를 기술한다.
· 각 요구사항은 정량적으로 테스트가 가능해야 한다.
· 사용자 상호작용에 대한 허용오차를 포함해야 한다.
<b>명세 형식</b>
· 기능 요구사항 형식을 사용한다.

표 9 요구사항 문장형식의 정형화 지원

“시스템은 <운영 조건>에서 <대상 객체>에 대해 <성능/품질>을 만족하는 <기능>을 제공해야 한다.”
“<대상객체>는 <자극유발원>으로부터 받은 <자극>에 대해 <환경조건>에서 <응답조건>안에서 <응답>을 제공해야 한다.”

표 10 요구 명세의 테일러링 지원

요구사항 형태	개략 명세	상세 명세
1. SRS소개	○	○
2. 시스템 개요	○	○
3. 비즈니스요구사항	○	○
4. 인터페이스요구사항	외부 인터페이스 요구사항	상세 인터페이스 요구사항
5. 기능 요구사항	시스템 기능구조	상세기능 요구사항
6. 데이터 요구사항	도메인 데이터 모델	논리적 데이터베이스 요구사항
7. 품질 요구사항	시스템 품질목표	상세 품질 요구사항
8. 운영 요구사항	-	○
9. 제약사항	○	○
10. Use Case 명세	-	○
11. 요구사항 추적 매트릭스	요구사항 추적 매트릭스 정의	요구사항 추적 매트릭스 기술

표 11 요구사항 명세 정보의 정량화

요구사항 번호	SMR-003	비밀번호 등록	기능
작성/승인자	김요구	08-10-10	Ver 1.0
이해관계자	시스템관리자, 개발자, 품질관리자, 테스터		
내용	1. 사용자는 비밀번호 입력 창에서 신규회원의 비밀번호를 확인을 위하여 2번 입력해야 한다. 2. 시스템은 입력된 비밀번호를 회원정보 DB에 저장해야 한다. 3. 저장된 비밀번호는 암호화되어야 한다.		
입력데이터	비밀번호	출력데이터	암호화된 비밀번호
입력유형	외부입력	파일 유형	내부파일 (회원정보 DB)
평가방법	1. 비밀번호를 입력하여 화면에 입력 내용을 확인할 수 없도록 결과가 출력되는지를 확인한다. 2. 데이터베이스에 저장된 비밀번호가 암호화되었는지를 확인한다.		
평가기준	기능 구현의 완전성, 기능 구현의 정확성, 접근 통제 가능성, 인터페이스의 정확성		
중요수준	필수	처리상태	승인
우선순위	상	중요도	상
품질수준	보안성, 효율성, 신뢰성, 사용성		
변경가능성	낮음	변경내역	
관련 문서	변답서-0810	관련 소스	
관련 요구사항	SMR-010	관련 산출물	

4. 표준 요구사항 명세서의 활용 방안

기존의 요구사항 명세서는 주로 개발자 관점에서 작성되고 활용되는 측면이 강했으나, 본 표준명세서는 개발자뿐만 아니라 소프트웨어 관련 지식이 없는 발주자도 요구사항의 근거나 이유 등을 명세서를 통해 쉽게 이해하고 추적할 수 있도록 지원한다. 특히 명세된 요구사항을 통해 사업 규모, 비용, 기간, 품질 정도를 쉽게 예측할 수 있어 발주자가 소프트웨어 사업관리를 용이하게 수행할 수 있다.

그림 7은 소프트웨어 사업 단계에서 필요로 하는 다양한 형태의 요구사항 활용 예를 보여주고 있다. 발주자는 비즈니스 요구사항을 토대로 RFP 제안 요청서를 낼 수 있으며, 수발주자 모두 시스템 요구사항을 토대로 사업 규모, 일정, 품질, 비용 등에 상호 협상 및 합의를 할 수 있다. 수주자는 또한 사용자 요구사항을 토대로 사업 실행 전적을 제시할 수 있게 되며, 이는 추후 개발 요구사항으로 발전하게 된다. 이 모든 형태의 요구사항이 본 표준 명세서를 통해 효과적으로 명세될 수 있다.

제안하는 표준 명세서는 다음의 사업 프로젝트 환경에서 효율적으로 활용될 수 있다.

- 잦은 요구사항 변경으로 인한 변경기준선이 모호해지고 변경관리 절차가 복잡해지는 경우 본 표준 명세서

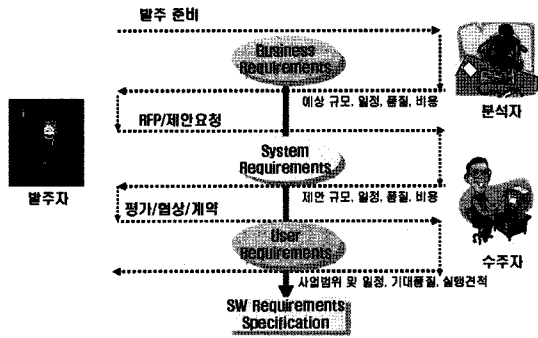


그림 7 이해당사자 별 요구사항 명세서의 활용

의 요구사항 추적 매트릭스를 활용함으로써 계층적으로 분해된 요구사항뿐만 아니라 타 요구사항(예, 품질 요구사항) 및 프로세스 산출물들과의 연결 추적이 용이해진다(표 3, 4, 5, 6 참조).

- 개발 범위의 불명확화로 인한 개발 비용/일정/품질/소요인력의 예측 및 통제가 어려운 경우 본 표준명세서의 시스템 요구사항 및 비즈니스 요구사항 명세를 활용할 수 있다. 시스템 및 비즈니스 요구사항 명세는 현재 시스템과 목표 시스템의 차이(Gap) 분석을 통해 소프트웨어 개발 범위를 조직의 비즈니스 목표와 전략에 맞추어 정의할 수 있게 한다. 표준명세서는 또한 각 요구사항의 개발 규모 및 비용을 산정할 수 있도록 각 요구사항 별로 기능 유형 및 파일 유형에 대한 데이터를 입력하게 하고, 이들의 합이 상위 수준에서의(비즈니스 요구사항, 시스템 요구사항) 개발 규모 및 비용과 일치될 수 있도록 일관성을 유지시켜 준다(표 11 참조).
- 어려운 명세 형식 및 용어로 인해 사용자가 요구사항을 명확하게 이해하고 해석하기 어려운 경우, 표준에서 제시하는 정형화된 템플릿을 활용함으로써 보다 명확한 이해와 해석이 가능하다. 특히 명세 문장의 형식화를 통해(표 9 참조) 모든 문장이 통일성과 일관성을 유지할 수 있게 하며, 나아가 작성된 문장으로부터 기능점수를 예측하기 위해 필요한 데이터 요소 및 레코드 요소를 쉽게 파악할 수 있게 해준다.
- 요구사항에 대한 검증 및 품질관리가 어려운 경우 표준의 템플릿 구성요소 들 중 명세되지 못하고 누락된 요소를 파악함으로써 명세서의 완전성과 충실성을 검증할 수 있으며, 제공되는 추적 매트릭스와 템플릿을 통해 정형성과 연결/추적성까지 검증할 수 있다.
- 비 효율적인 의사소통으로 인해 고객과의 요구사항 합의 및 외주 관리가 어려운 경우에도 본 표준명세서가 효과적으로 활용될 수 있다. 특히 최근 대두 되고 있는 소프트웨어 분할 발주 제도가 시행되면 본 표준

명세서를 이용하여 사용자 요구사항의 명세뿐만 아니라 유즈케이스 명세까지를 포함하여 구현 이전의 기본 설계가 가능하게 된다. 또한 기존의 턴키 방식의 소프트웨어 사업 프로세스를 개선하는 데에도 본 표준명세서가 유용하게 활용될 수 있다. 사용자 요구사항의 도출 및 명세는 사전에 용역사업을 통해 개발하고(사전요구사항), 이후 설계 및 구현은 별도의 사업으로 수행될 수 있다. 이 때 사전요구사항은 본 표준명세서를 통해 명세화 될 수 있으며, 사전요구사항만으로도 전체 사업의 규모와 일정에 대한 예측이 가능해짐으로 인해 소프트웨어 사업 관리가 더욱 용이해진다(그림 8). 도출된 사전요구사항명세서는 사업제안 요청서(Request of Proposal)의 기초 문서로 활용될 수 있다[12].

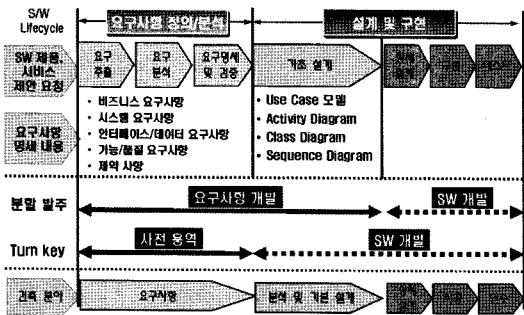


그림 8 소프트웨어사업 발주·관리절차상의 적용

### 5. 결론 및 향후 연구

소프트웨어 프로젝트 실패의 가장 큰 요인은, 다양한 이해당사자들의 복잡 다양한 요구사항을 이해, 분석, 문서화, 관리 및 통제하기가 어려운 데에 있다. 개발초기의 요구사항이 명확하게 정의, 명세 되지 못함으로 인해 프로젝트 발주(획득)자에게는 불명확한 사업 목적 및 범위, 잘못된 사업 예측, 낮은 품질 및 납기 지연의 문제를 야기할 수 있으며, 수주자(공급, 개발자)에게는 지속적인 요구사항의 변경 부담과 이에 따른 개발 비용의 증가 문제를 초래할 수 있다. 이러한 문제를 해결하기 위해서는 사업 초기부터 요구사항에 대한 명확한 명세가 이루어져야 하며, 요구사항의 잦은 변경에도 유연하게 대처할 수 있어야 한다. 요구사항의 명확한 명세와 유연한 변경추적을 지원하기 위해서는 정형화되고 통일된 표준 형식이 필요하다. 본 논문에서는 정량화, 정형화, 연결/추적이 용이한 표준 명세서를 제안하고, 이의 활용 방안을 소개하였다.

제안하는 표준명세서는 요구사항들의 연결/추적성, 정량화, 정형화를 달성하기 위해 많은 양의 정보를 사용자

로부터 요구하고 있는데, 이는 명세서의 사용성을 저하시키는 요인으로 작용할 수도 있다. 따라서 표준명세서를 조직 및 프로세스 환경이나 특성에 따라 효과적으로 테일러링할 수 있는 방안에 대한 추가 연구가 필요하다. 또한 다양한 산업 도메인(국방, 의료, 건설, 조선 등), 조직 및 프로젝트 규모 별로 특화된 템플릿을 개발함으로써 사용자의 테일러링에 대한 부담과 비용을 줄일 수 있는 방안도 함께 연구되어야 한다. 이 외에도 비기능적 요구사항으로부터 소프트웨어 사업 대가를 산정 예측할 수 있는 방안, 요구사항 리스트를 업무 도메인/시스템 별로 정리, 저장, 관리, 재사용할 수 있는 기술에 대한 연구도 필요하다.

### 참고 문헌

- [1] The Standish Group, Standish Group Report, 2005.
- [2] ISO/IEC JTC1/SC7 WG4, Study Period Report on Requirement Engineering Tool Capabilities, ISO/IEC JTC1/SC7 WG4, 2004.
- [3] Karl Wieggers, Software Requirements, Microsoft Press, 2003.
- [4] Giakoumakis and Xylomenos, "Evaluation and selection criteria for software requirements specification standards," *IEE/BCS Software Engineering Journal*, vol.11, no.5, pp.307-319, 1996.
- [5] ESA (European Space Agency), ESA PSS-05-0, Software Engineering Standards, ESA, 1987.
- [6] NASA, SMAP-DID-P200-SW, Software requirements DID, 1989.
- [7] NASA, SMAP-DID-P210, External interface requirements DID, 1989.
- [8] JPL(Jet Propulsion Laboratory), JPL D-4005-Software requirements analysis phase, JPL, 1989.
- [9] DoD, DoD-STD-7935A, System/subsystem specification, DoD, 1988.
- [10] DoD, DoD-STD-2167A, Defense system software development, DoD, 1988.
- [11] ANSI/IEEE, ANSI/IEEE STD. 830- Guide to software requirements specifications, ANSI/IEEE, 1984.
- [12] 한국소프트웨어진흥원, SW 사업관리 감독에 관한 일반 기준 - 공공부문 SW사업 발주 관리 표준 프로세스[교육교재], 지식경제부, vol.1.0, 2008.



이 병 길

1988년 University of Bridgeport 물리학과 학사. 1996년 Auburn University 컴퓨터학과 석사. 1998년 Auburn University 컴퓨터학과 박사. 1998년~현재 서울여자대학교 정보미디어대학 컴퓨터학 전공 교수. 2002년~현재 기술표준원 ISO/IEC JTC1/SC7 전문위원. 관심분야는 소프트웨어 향상관리, 소프트웨어 프로세스개선, 요구공학, 신뢰성공학



황 만 수

1984년 중앙대학교 전자계산학과(이학사). 1986년 중앙대학교 전자계산학과(이학석사). 2001년 숭실대학교 컴퓨터 공학과(공학박사). 1987년~1993년 LG 소프트웨어 연구원. 1993년~현재 신홍대학 컴퓨터정보계열 교수. 관심분야는 요구공학, 소프트웨어 품질관리, 소프트웨어 프로세스 개선



이 예 복

1989년~2001년 기획예산처 정보화담당관실. 2001년~현재 한국소프트웨어진흥원 소프트웨어공학단. 관심분야는 소프트웨어사업 발주관리, 소프트웨어사업 사업대가



이 혁 재

2001년 미국 CarnegieMellon University SE Master 연수과정 수료. 2008년 한국 전기안전공사 정보화자문위원. 2009년 SW 프로세스 품질인증(SP인증) 심사원. 2009년~현재 한국소프트웨어진흥원 SW공학기술센터 팀장. 관심분야는 소프트웨어 프로세스개선, 소프트웨어사업 발주관리

## 백 중 문

정보과학회논문지 : 소프트웨어 및 응용  
제 36 권 제 5 호 참조



이 창 근

2007년 호서대학교 컴퓨터응용기술학과 석사. 2006년 포스테이타 품질보증팀장  
2007년~현재 포스테이타 품질경영팀장  
관심분야는 품질경영, 소프트웨어 품질보증, 요구사항관리