

- 시스템의 **기능**을 정확하고 완벽하며 일관성 있게 작성한 것
- 소프트웨어에 포함될 기능과 제약 조건들을 나열
- 기능에 대한 자세한 설명과 **예외처리** 기술
- 시스템 성능과 관련된 사항, 속도, 정확성, 사용 용이성 포함
- 과정
 - 명세서 작성
 - 명세서 검토

Exception examples:

- Erroneous inputs
- Faults in the computer(s), the software and communication lines

- 사용자와 개발자간의 이해를 돕기 위함
- Gilbert가 제안한 요구 분석서 작성시 주의사항
 - 요구 분석서는 사용자와 개발자 모두가 쉽게 이해할 수 있도록 써야 한다
 - 요구 분석서에 기술된 조건은 개발자와 사용자가 모두 동의한 것이어야 한다.
 - 요구 분석서는 목표 시스템에 의하여 수행될 모든 기능을 정확히 기술하여야 한다.
 - 요구 분석서는 목표 시스템에 영향을 주는 모든 제약 조건을 기술한다.
 - 요구 분석서는 시스템의 인수를 위한 테스트 기준을 제공하여야 한다.
 - 요구 분석서는 원하는 시스템의 품질과 상대적인 중요도 및 품질을 재는 방법이 기술 되어야 한다.

요구 분석서 목차

- 1. 개요
 - 1.1 시스템의 목적
 - 1.2 범위
 - 1.3 정의, 약어
 - 1.4 참조
- 2. 기능적 요구
 - 2.1 외부 인터페이스 요구
 - 2.1.1 사용자 인터페이스
 - 2.1.2 하드웨어 인터페이스
 - 2.1.3 소프트웨어 및 통신 인터페이스
 - 2.2 기능 요구
 - 2.2.1 기능 #1(사용 사례 #1)
 - 2.2.2 기능 #2(사용 사례 #2)
 - ...
- 3. 기타 요구 및 제약 사항
 - 3.1 성능 요구(반응 시간, 처리 소요 시간, 처리율)
 - 3.2 H/W 요구(기억 장치 규모, 통신수용도)
 - 3.3 예외 조건 및 이의 처리
 - 3.4 자원, 인력에 대한 제약 조건
- 4. 인수 조건
 - 4.1 기능 시험 및 성능 시험
- 5. 참고 자료

- 명세 스타일: 자연어 vs 구조적

자연어

3.2 매 10분마다 시스템은 혈당량을 측정하고 필요한 경우 인슐린을 전달해야 한다(shall). (혈당의 변화는 상대적으로 느리기 때문에, 이보다 더 자주 측정할 필요는 없다; 이보다 덜 측정하게 되면 불필요하게 높은 혈당 수준이 발생할 수 있다.)

3.6. 시스템은 테스트할 조건에 대해 매분마다 정기적인 자가진단을 실시해야 하며(shall) 이와 관련한 동작을 표 1에 정의하였다. (정기적 자가진단은 하드웨어와 소프트웨어의 문제점을 발견하고 사용자에게 정상적인 동작이 불가능하다는 사실을 알려줄 수 있다.)

기능	인슐린의 용량을 계산: 안전한 혈당 수준
설명	현재 측정된 혈당 수준이 3~7 단위 사이의 안전 범위에 속하는 경우에 전달할 인슐린의 용량을 계산한다.
입력	현재 혈당 수치(r_2), 이전에 측정된 수치 2건(r_0 , r_1)
출처	센서로부터 측정된 현재 혈당 수치. 다른 수치들은 메모리로부터 가져온다.
출력	CompDose—전달할 인슐린의 용량
목적지	메인 제어 루프
동작:	혈당 수준이 안정적이거나, 떨어지거나 또는 증가하더라도 증가율이 감소하는 경우에는 $\text{CompDose} = 0$. 혈당 수준이 증가하고, 증가율도 증가하는 경우에는 현재 혈당 수준과 이전 수준의 차이값을 4로 나눈 값을 반올림하여 compDose를 계산하고 만약 반올림한 값이 0이면 투여할 수 있는 최소 용량으로 CompDose를 설정한다(그림 4.14 참조).
필요사항	혈당 수준의 변화율을 계산하기 위해 필요한 이전 두 건의 수치
사전조건	인슐린 통에는 인슐린 1회 용량보다 많은 양을 포함하고 있어야 한다.
사후조건	r_0 를 r_1 로 대체한 후, r_1 을 r_2 로 대체한다.
부작용	없음

● 요구 분석 명세서 평가 기준

- **무결성**과 **완벽성** – 요구 분석서는 사용자의 요구를 오류없이 완벽하게 반영하고 있어야 한다.
- **일관성** – 요구 분석서 안에 서로 모순되는 부분이 없어야 한다.
- **명확성** – 요구 분석의 내용이 여러 의미로 해석되는 모호한 점이 없는지 살펴본다
- **기능적** – ‘어떻게’ 보다 ‘무엇을’에 관점을 두고 기술되어야 한다.
- **검증 가능성** – 요구 분석은 두 가지로 검증 가능해야 함
 - 사용자 요구 만족
 - 시스템이 요구 분석에 기술된 내용과 일치하는가
- **추적 가능성** 및 변경 용이성 – 내용은 체계적으로 정리되어야 한다

요구 관리 도구

- **ProR** : GUI part of eclipse RMF. ReIF(Requirements Interchange Format) 지원.



