Chapter 7. 정리

1. 프로세스를 가이드하는 원칙

2. 실무를 가이드하는 원칙

3. 각 프레임을 가이드하는 원칙

4. 프로젝트 계획 보고서

개방적 사고와 실천

◆정직과 성실 , 열정을 지키려는 노력

★ 올바른 자부심에 건설적이고 , 긍정적 안목과 태도

★ 능력을 파악하고 최선을 다하는 자세

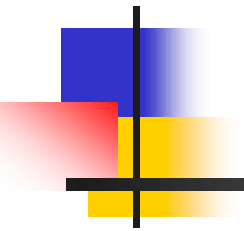
★ 인격체로서 인정 , 배려하는 리더십

★ 타인의 입장에서 생각하고 처리

★ 한단계 높고 넓은 시각으로 업무 처리

★ 건설적인 아이디어 창출

“ 10~20 년후 어디서 무엇을 하고 있을 것인가



제 8장. 요구사항 이해

April. 2018
Young-gon, Kim
ykkim@kpu.ac.kr
Department of Computer Engineering
Korea Polytechnic University

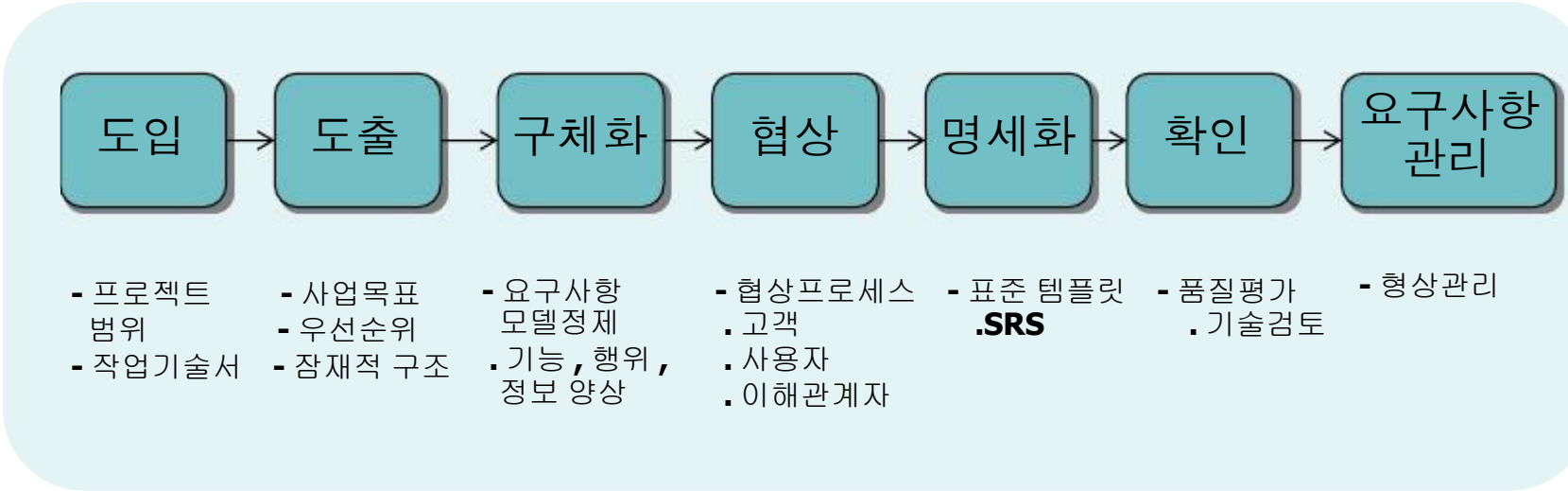
Topics covered

- 요구 공학
- 초기 구축
- 요구사항 추출
- 유스케이스 개발
- 분석 모델 개발
- 협상 요구사항
- 요구사항 모니터링
- 요구사항 검증
- 요구사항 문서화.

1. Requirement Engineering

◆ 요구공학 프로세스

- 요구사항 분석, 시스템 분석가
- 최종산출물 : SRS (software Requirement Specification)



1. Requirement Engineering

◆ 1. 도입

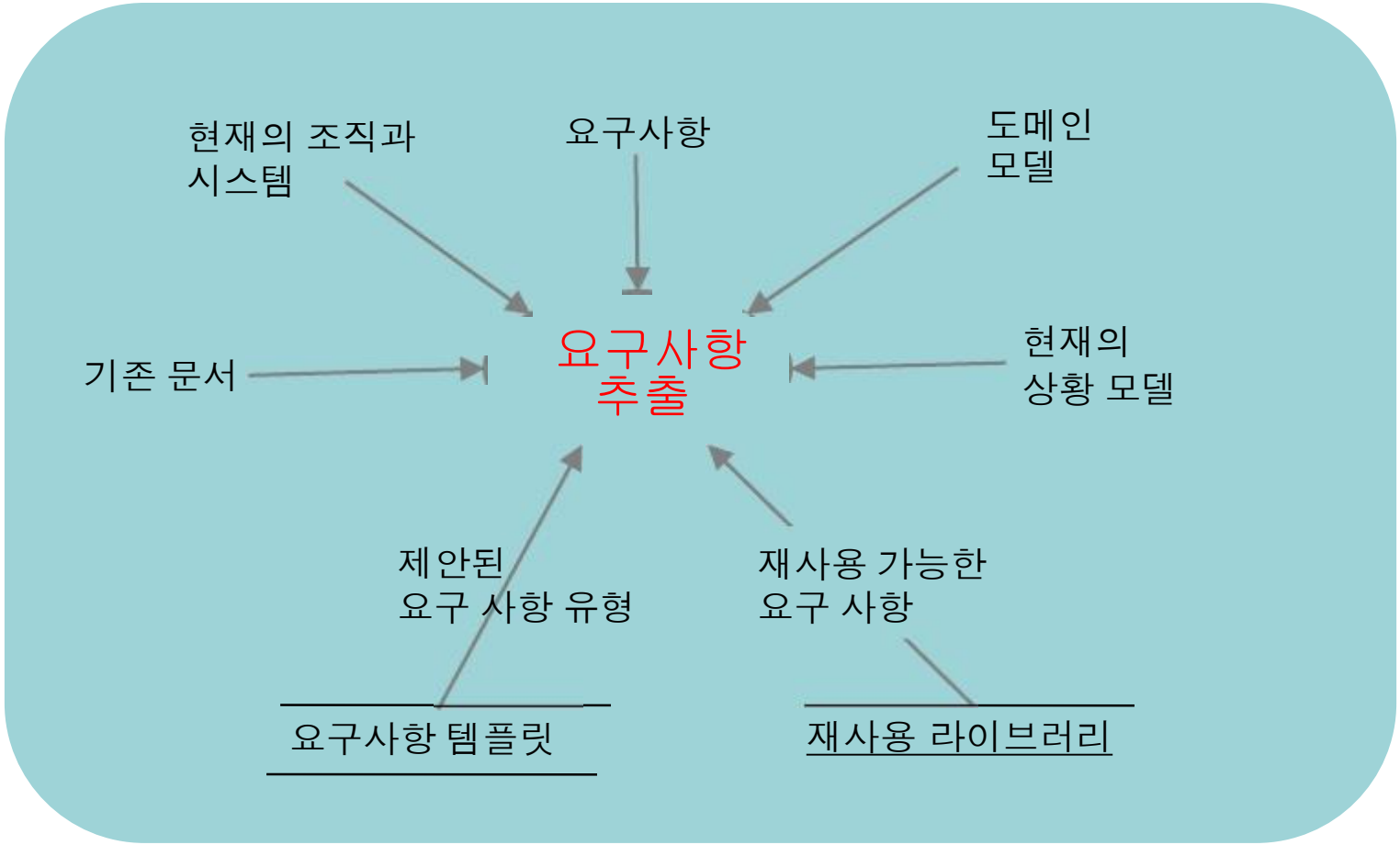
- 문제의 기본 이해 확립
  - 이해관계자와 소프트웨어 팀간의 사전의사소통과 협력의 효율성
- 솔루션을 원하는 사람
  - 해결책을 원하는 솔루션의 본질

◆ 2. 도출

- 사업 목표 수립
- 우선순위 결정 메커니즘 수립
- 의사소통과 조정을 위하여 메커니즘을 수립
- 잠재적인 구조에 대한 근거적인 설계
  - 이해관계자 목표 충족
- 도출을 위한 질문 내용
  - 시스템 / 제품의 목표 무엇인지 ?
  - 무엇이 달성되려 하는지 ?
  - 시스템 / 제품의 비즈니스 요구에 어떻게 적합 한지 ?
  - 시스템 / 제품이 어떻게 매일 사용될 수 있을지 ?

1. Requirement Engineering

◆ 가능한 요구사항 자원



1. Requirement Engineering

◆ 3. 구체화

- 요구사항 **모델 정제**
  - 소프트웨어 기능, 행위, 정보의 **다양한 양상** 정의
- **구체화**
  - 최종 사용자 (액터) 가 시스템 **상호작용** 기술
  - 사용자 **시나리오 생성과 정제**
    - 사용자 시나리오 : 분석 클래스 (도메인 개체들)로 분해 -> 속성 정의
  - **클래스간 협력** 정의 -> 보조 다이어그램 생성

◆ 4. 협상

- **충돌** 일반적
  - 고객과 사용자들 : 주어진 자원보다 달성하는 것 이상 요구
  - 다른 고객이나 사용자 : **상충하는 요구사항 제안**
  - **버전**이 "우리의 **특정 요구에 필수적** " 이라고 주장
- 협상 프로세스 : 충돌 절충
  - **우선순위 정의** : 우선순위에 따라 충돌 논의
  - 요구사항의 **위험도와 비용을 평가**
  - 요구사항 제거 / 통합 / 수정 -> 당사자간의 **만족 유도.**

1. Requirement Engineering

◆ 5. 명세화

- 명세서
  - 조합 { 문서 , 그래픽 모델 , 정형화된 수학모델 , 사용시나리오집합 , 프로토타입 }
  - **표준 템플릿** : 일관성 있고, 이해하기 용이한 방법, 유연함
- SRS(Software Requirement Specification) 템플릿

목차
변경이력
1. 소개
   1.1 목적
   1.2 문서규약
   1.3 의도된 청중과 읽을거리
   1.4 프로젝트 범위
   1.5 참고문헌
2. 전반적인 기술
   2.1 프로덕트 관점
   2.2 프로덕트 특징들
   2.3 사용자 클래스들과 특성들
   2.4 운영환경
   2.5 설계및 구현 제약사항
   2.6 사용자 문서
   2.7 가정과 의존관계
3. 시스템 특징
   3.1 시스템특징 1
   3.2 시스템 특징 2
4. 외부 인터페이스 요구사항
   4.1 사용자 인터페이스
   4.2 하드웨어 인터페이스
   4.3 소프트웨어 인터페이스
   4.4 의사소통 인터페이스
5. 다른 비기능적 요구사항
   5.1 성능 요구사항
   5.2 안전 요구사항
   5.3 보안 요구사항
   5.4 소프트웨어 품질 속성
6. 기타 요구사항
부록 A: 용어정의
부록 B: 분석모델
부록 C: 이슈 목록

1. Requirement Enginnering

6. 확인

- 요구사항 확인
  - 소프트웨어 **요구사항이 모호하지 않게** 표현 보장 -> 품질 평가
    - **불일치 , 생략** , 감지하고 수정될 수 있는 **오류**
    - 프로세스 , 프로젝트 , 제품에 대해 **구축된 표준에 대해 일치** 하는 결과물
- 요구사항 확인 메커니즘 : **기술 검토** (오류를 검색목적 명세서 조사)
  - 내용이나 **해석 , 명확화**가 요구될 수 있는 영역
  - **정보 누락**
  - **비일관성**
  - **상충** 하는 요구사항 또는 **비현실적** 요구사항.

1. Requirement Engineering

◆ 요구사항 확인 체크리스트

1. 요구사항이 **명확하게 표현**되어 있는가 ? **틀린 해석이 될 가능성**이 있는가 ?
2. 요구사항이 **소스 (예 : 사람 , 규정 , 문서)가 정의** 되어 있는가 ?
   요구사항의 최종 기술이 **원본소스에 의하거나 반하여** 조사되었는가 ?
3. 요구사항이 **정량적인 조건에 한계**가 있는가 ?
4. 다른 요구사항이 이요구사항과 **어떠한 연관** 되어 있는가 ?
5. 요구사항이 **시스템 도메인 제약을 위반**하는가 ? 요구사항이 **시험 가능** 한가 ?
   만약 그렇다면,우리는 요구사항을 수행하기 위해 **시험(확인 기준)을 기술**할 수 있는가?
6. 요구사항이 만들어진 어떠한 **시스템 모델을 추적**할 수 있는가 ?
   요구사항이 전체적인 시스템/프로덕트 **목적에 대해 추적**할 수 있는가 ?
7. 명세서가 보다 기술적인 작업 산출물로의 **쉽게 이해**되고, **쉽게 참조** 되고 ,
   쉽게 번역 되는 쪽으로 이끌 수 있도록 구조적인가 ?
8. **명세서를 위한 색인**이 만들어졌는가 ?
9. **성능, 행위, 그리고 운영 가능한 특성**과 연관되어 있는 요구사항이 명확하게
   기술되었는가 ? 어떤 요구사항이 **명시적**으로 나타나는가 ?

2. 초기 구축[요구공학을 이해하기위한 기초작업 수립 단계]

◆1. 이해관계자 식별

- 이해관계자 : 개발하려는 시스템의 **직/간접적인 방법으로 혜택을 받는자**
  - 사업운용 관리자, 제품 관리자, 마케팅 관계자, 내 / 외부 고객, 최종사용자, 컨설턴트, 제품공학자, 소프트웨어 공학자, 지원 및 유지보수공학자
- 요구사항이 도출 -> **공헌해 줄 사람들의 목록** 작성

◆2. 다중 관점 인식

- **마케팅그룹** : 잠재시장 흥분시킬 **기능 특징** 관심->시스템 팔기 쉽게 마케팅
- **사업관리자** : 제한된 **예산**에서 구축, 시장 출시
- **최종사용자** : 친숙하고 **배우기 쉽고 사용하기 쉬운** 특징
- **소프트웨어공학자** : 기능및 특징을 지원하는 **기반 구조**를 지원하는 기능
- **지원 공학자** : 소프트웨어 **유지보수성**

◆3. 협업을 위한 작업

- 요구사항 공학자 업무 : **영역 식별** -> 협력하여 최종 결정
  - **공통영역** : 이해관계자 **동의** 요구사항
  - **충돌 / 비일관적인 영역** : 이해관계자간 **마찰을 하는** 요구사항.

2. 초기 구축

◆ 4. 프로젝트 도입에서 요구되는 질문

- **첫번째 질문 : 총체적인 프로젝트 목적과 혜택**에 초점 – > “**자유로운** 문맥”
  - 이 작업을 위한 **요청 배후**는 누구인가 ?
  - **솔루션을 사용할 사람**은 누구인가 ?
  - 성공적인 **솔루션의 경제적인 혜택**은 무엇인가 ?
  - 당신이 원하는 **해결책에 대해서 다른 소스가** 있는가 ?
- **다음 질문 : 문제를 쉽게 이해하고 , 고객이 해결책 관여**
  - 당신은 성공적인 해결책에 의해 만들어진 “**좋은”결과물**을 어떻게 **특정** 짓겠는가 ?
  - 이 해결책이 **다루는 문제**는 무엇인가 ?
  - 당신은 나에게 솔루션이 사용될 **사업 환경을 보여 (설명해)**줄 수 있는가 ?
  - **특정 성능 이슈와 제한조건 해결책**이 접근하는 방법에 영향 을 주겠는가 ?
- **질문들 최종 조합** : 의사소통 활동 그 자체의 효과 에 초점
  - 당신은 이러한 질문에 **대답할 수 있는 올바른 사람**인가 ? 당신의 대답은 “ **공식적** ” 인가 ?
  - 나의 질문들이 당신이 가진 **문제와 관련성**이 있는가 ?
  - 내가 너무 **많은 질문**들을 하는가 ?
  - 다른 누군가가 **부가적인 정보를 제공**할 수 있는가 ?
  - 나는 당신에게 **다른 무엇을 요구**해야만 하는가 ?

2. 초기 구축

◆ 5. 비기능적 요구사항 : NFR[NonFuntional Requirement]

- **품질, 성능, 보안**, 시스템 일반적인 **제약조건**
- **비기능 요구사항** : 소프트웨어 요구사항 명세와 **분리**
  - 유용성, 시험성, 보안성, 유지보수성

◆ 6. 추적성

- 작업제품 (**요구사항과 시험사례**) 사이를 연결하는 문서
- **추적 행렬**
  - 소프트웨어공학 작업 결과물 사이에 **관계**를 표현
  - **진화를 추적**하기 위한 수단.

2. 초기 구축

◆ 비기능적 요구사항

Product requirements	Usability	
	Efficiency requirements	Performance requirements
		Space requirements
	Reliability requirements	
	Portability requirements	
Organizational requirements	Delivery requirements	
	Implementation requirements	
	Standard requirements	
External requirements	Interoperability requirements	
	Ethical requirements	
	Legislative requirements	Privacy requirements
		Safety requirements

3. 요구사항추출(수집):문제해결+구체화+협상+명세서요소

1. 협업 요구사항 수집

- 목표 : 문제식별 , 해결책 구성요소 제안 , 다른접근법 협상 , 요구사항 사전정의
- 기본 가이드 라인
  - 회의 (현실 , 가상)는 소프트웨어공학자와 다른 이해관계자들이 참석 하여 실시
  - 참여와 준비를 위한 규칙 수립
  - 의제는 공식적으로 충분한 내용 이 제안 , 비정형적으로 아이디어의 자유로운 흐름 권장
  - 조정자 는 회의를 제어
  - 정의 메커니즘 (작업 시트 , 플립차트 , 벽 스티커 / 전자게시판 , 대화방 / 가상포럼) 사용

2. 품질 기능 전개 : QFD[Qualify Function Deployment]

- SW 에 대한 고객의 요구를 기술적인 요구사항으로 변화하는 품질관리기술
- 요구사항 정의 : **고객 만족을 극대화**에 집중하는 방식
- 요구사항 수집 활동을 위한 원시 데이터 사용
  - 고객 인터뷰, 관찰, 설문조사, 이력 데이터 (문제보고서)의 시험

3. 사용 시나리오

- 시나리오 : 유스케이스 (어떻게 시스템이 사용될 수 있는지에 대해 기술서)
  - 기능들과 특징 : 최종 사용자들의 다른 클래스를 어떻게 사용하는지 이해.

3. 요구사항 추출

4. 제품 추출 작업

- 요구사항 도출 작업 결과물
  - 요구와 실행 가능성의 기술
  - 시스템 / 제품을 위한 범위와 제한된 기술
  - 요구사항 추출에 참여하는 고객, 사용자, 이해관계자 목록
  - 시스템의 기술적인 환경의 기술
  - 요구사항 목록과 각각에 적용되는 도메인 제약조건의 목록
  - 다른 운용조건하에서 시스템 / 제품의 사용의 통찰성을 제공하는 일련의 사용 시나리오
  - 요구사항을 좀 더 좋게 정의하기 위해 개발되는 어떤 프로토타입

5. 요구사항 추출

- 사용자 관점에서 작성된 단순한 시스템 요구사항 기술
- 사용자 이야기 -> 노트카드에 기록
- 제안자 : 개발자가 자신 의제보다 요구사항 선택하는 이해관계자의 의사소통 초점

6. 서비스 지향

- 시스템 : sets{서비스} , 서비스 -> 하나의 기능을 가능한 간단하게 제공
- 응용 소프트웨어에 의해 생성된 서비스의 정의 초점
- 요구사항 추출
  - 민속학적 연구 , 혁신적인 워크샵 , 초기의 저정밀도의 프로토타입.

4.유스케이스 개발

◆ 유스케이스

- 시스템이 이해관계자 중 한 명의 요청에 응답하는 것처럼 다양한 조건 하에 시스템 행위 를 기술
- 최종 사용자가 어떻게 일련의 특정 상황에서 시스템과 상호작용 하는 대한 형식화된 이야기
  - 이야기 : 서술적인 텍스트 또는 작업이나 상호작용 의 개요 , 형식 기반 의 기술서 , 도식화 된 표현
- 최종 사용자의 관점 에서 소프트웨어나 시스템을 묘사
- 첫단계 : 액터의 집합 정의
  - 액터 : 시스템 자체의 외부적인 시스템 / 제품과 의사소통
- 유스케이스 내용을 더 완벽한 관점을 제공하기 위한 질문
  - 주요 액터와 부수적 액터는 누구인가 ?
  - 액터의 목적은 무엇인가 ?
  - 이야기를 시작하기전 있어야 하는 사전조건은 무엇인가 ?
  - 액터에 의해 수행되어야 할 주요 작업이나 기능은 무엇인가 ?
  - 이야기가 기술되는 것에 의해 고려해야 하는 예외사항은 무엇인가 ?
  - 액터의 상호작용이 가능함에 따라 어떠한 변화가 있는가 ?
  - 액터가 획득하고 생산하고 , 변경하는 시스템 정보는 무엇인가 ?
  - 액터가 외부환경의 변경 에 대해서 시스템에게 알려줄 것인가 ?
  - 액터가 시스템으로 부터 바라는 정보는 무엇인가 ?
  - 액터는 예상치 못한 변경 에 대해 알려 줄 것인가 ?

5.분석 모델 개발

◆ 분석 모델

- 목적
  - 컴퓨터 기반 시스템에서 요구되는 기술적 기반 시스템을 위한 요구되는 정보 , 기능 , 행위 도메인을 기술서 제공
  - 어떤 주어진 시간에서의 요구사항 스냅 샷
- 분석 모델 요소
  - 시나리오 기반 요소
    - 모델링 요소의 생성을 위한 **입력**처럼 제공
    - 유스케이스를 사용 : 요구사항 도출과 표현을 위한 UML 활동 다이어그램
  - 클래스 기반 요소
    - 클래스 : 유사한 속성들과 **공통적인 행위**를 가진 것들의 집합
    - 액터가 시스템에 상호작용하는 것처럼 조작되는 **일련의 객체** -> 클래스
  - 행위적 요소
    - 행위를 묘사하는 모델링 요소 -> 요구사항 모델
    - 상태다이어그램 : 시스템의 상태 변화를 야기 시키는 **이벤트**와 그자체의 **상태**를 묘사에 의해 행위 표현.

6.협상 요구사항

◆ 협상 활동

- 시스템 또는 하위시스템의 핵심 이해 관계자들 식별
- 이해 관계자들의 " 이기는 조건들 "의 결정
  - 모든 문제에 대해서 일련의 win-win 조건으로 이해관계자들을 조화시키기 위해 이해관계자들의 이기는 조건의 협상

◆ 협상의 예술

- 협상이 경쟁이 아님을 인식하라 : 양쪽 타협
- 전략을 준비하라 : 본인과 상대 모두 달성 목표 / 방법
- 적극적으로 경청하라 : 잘 협상을 하기 위한 지식 습득
- 상대의 관심사에 초점을 맞춰라 : 갈등 회피를 위한 강한 입장 고수 포기
- 개인적으로 되지 마라 : 해결해야 할 문제에 집중
- 창조적으로 되어라 : 교착상태에서는 **상자 밖**에서 생각
- 상당한 노력을 기울일 준비를 해라 : 동의된 내용은 쓸데없는 말 금지하고 , 노력/진행.

7.요구사항 모니터링

◆ 요구사항 모니터링 작업

- 오류를 밝히고 오류의 원인을 결정하는 분산 디버깅
- 소프트웨어가 명세서와 일치하는지를 결정하는 실행시간 확인
- 진화하는 소프트웨어가 사용자의 목표를 충족하는지를 평가하는 실행시간 검사
- 시스템이 사업 목표를 만족하는지 평가하는 사업 활동 모니터링
- 시스템이 진화됨에 따라 이해관계자들에게 정보를 제공하기 위한 진화 그리고 통합설계.

7.요구사항 검증

◆ 요구사항 모델의 검토를 위한 질문

- 각 요구사항이 시스템 또는 제품을 위하여 전반적인 목표에 일치하는가?
- 모든 요구사항이 적절한 추상화 레벨에서 명세되었는가? (부적절한 상세화 레벨)
- 요구사항이 진정으로 필요 하거나 또는 시스템의 목적에 필수적이지 않을 수도 있는 추가적인 특징을 표현하는가?
- 각 요구사항이 경계가 정해져 있고 명확한가?
- 각 요구사항은 속성 을 갖고 있는가? 즉 소스 (특정 개인) 는 각 요구사항 에 대해서 잘 알려져 있는가?
- 요구사항 중 어떤 요구사항이 다른 요구사항과 충돌되는가?
- 각 요구사항이 시스템 또는 제품을 저장할 기술적 환경에서 달성 가능한가?
- 각 요구사항은 일단 구현되면, 시험 가능한가?
- 요구사항 모델은 구축될 시스템의 정보, 기능, 그리고 행위를 적절히 반영하는가?
- 요구사항 모델이 시스템에 대한 보다 상세한 정보를 점진적으로 노출하는 방식으로 “ 분할 ” 되었는가?
- 요구사항 패턴이 요구사항 모델을 단순화 하는 데에 사용되었는가?
- 모든 패턴이 적절히 검증되었는가?
- 모든 패턴이 고객 요구사항과 일치하는가?

9.요구사항 문서화

◆ 사용자 요구사항

- 시스템에 대한 사용자 요구사항
  - 기능적 요구사항과 비 기능적 요구사항을 기술
  - 자세한 **기술적 지식이 없는 시스템, 사용자도 이해** 할 수 있도록 해야 함
  - 자제 : 시스템 외부동작 지정, 시스템 설계 특성, 특수한 소프트웨어 용어, 구조적 표현, 형식적 표현, 구현 기술요구사항
- 사용자 요구사항 : **간단한 언어, 표, 직관적인 다이어그램**
- **자연어**로 사용하는 경우의 문제점
  - 명확성의 결여 : 읽기 어렵고 , 말이 많은 문서 -> 정확 (애매하지 않게)
  - 요구사항의 **혼돈** : 기능적 / 비기능적 , 시스템목표 / 설계정보 -> 분명히구분
  - 요구사항의 **혼합** : 여러가지 요구사항이 단일 요구사항으로 함께 표현
- **요구사항 문서 작성의 좋은 전략**
  - **사용자 요구사항 과 시스템 요구사항을 구분** : 비기술자 위축
  - **단순히 주요 기능만을 다룸** : 자율적인 개발 보장
  - **근거를 제시하려는 노력** : 변경시 유용성 확인.

9.요구사항 문서화

◆ 사용자 요구사항 작성시 오해 최소화를 위한 지침

1. **표준양식** 작성 : 누락 방지 , 쉽게 점검
   - 굵은 글씨체 (초기 사양 , 근거 , 참조), 제안자 성명 (변경시 문의 편리)
2. **언어를 일관성**있게 사용 : 필수적인 사항 / 바람직한 사항 구분
   - 필수적인 사항 (shall: 반드시 지원), 바람직한 사항 (should: 필수사항 아님)
3. 요구사항 **중요부분 강조**
   - 굵은 글 씨 , 이탤릭체 , 색글씨
4. 가능한 컴퓨터 **은어 사용 회피**
   - 기술적 용어 사용

* 초기 사용자 요구사용 작성 권고 : **하나의 요구사항을 하나의 카드에 작성**
  - 근거 , 다른 요구사항과의 의존관계 , 출처 , 지원 자료

2.6.1 눈금 기능

편집기는 수평과 수직선의 행렬이 편집기 윈도우의 배경으로 제공되는 눈금 기능을 제공한다 . 이 눈금은 개체의 정렬이 사용자의 책임인 수동 눈금 이다

근거 : 눈금은 사용자에게 잘 처리된 개체를 이용하여 작은 다이어그램을 만들 수 있도록 도와준다 .
능동적 눈금일 경우에 개체의 위치가 부정확하다 . 사용자가 개체의 위치잡기를 결정하는 최적의 사람이다 .

명세 : ECLISE 의 WS 의 Tolls 의 DE 의 FS Section 5.6

출처 : Ray Wilson, Glasgow Office

9.요구사항 문서화

◆ 표기 방법

- 1. **양식 기반**

  - 견본이나 표준양식 필요, 많은 정보 필요
  - 자연어 명세의 문제점 해결, 명세의 가변성 감소되고, 요구사항 효과적 구성
  - 복잡한 계산은 어려움

- 2. **표를 이용**

  - 계산 수행 표현
  - 여러가지 경우의 수가 많고
  - 각 경우에 행동 동작 표현

- 3. **그래픽 모델**

  - 시스템 상태 변화
  - 사용자와 시스템의 대화, 일련의 행동 수행
  - **Sequence diagram**

9.요구사항 문서화

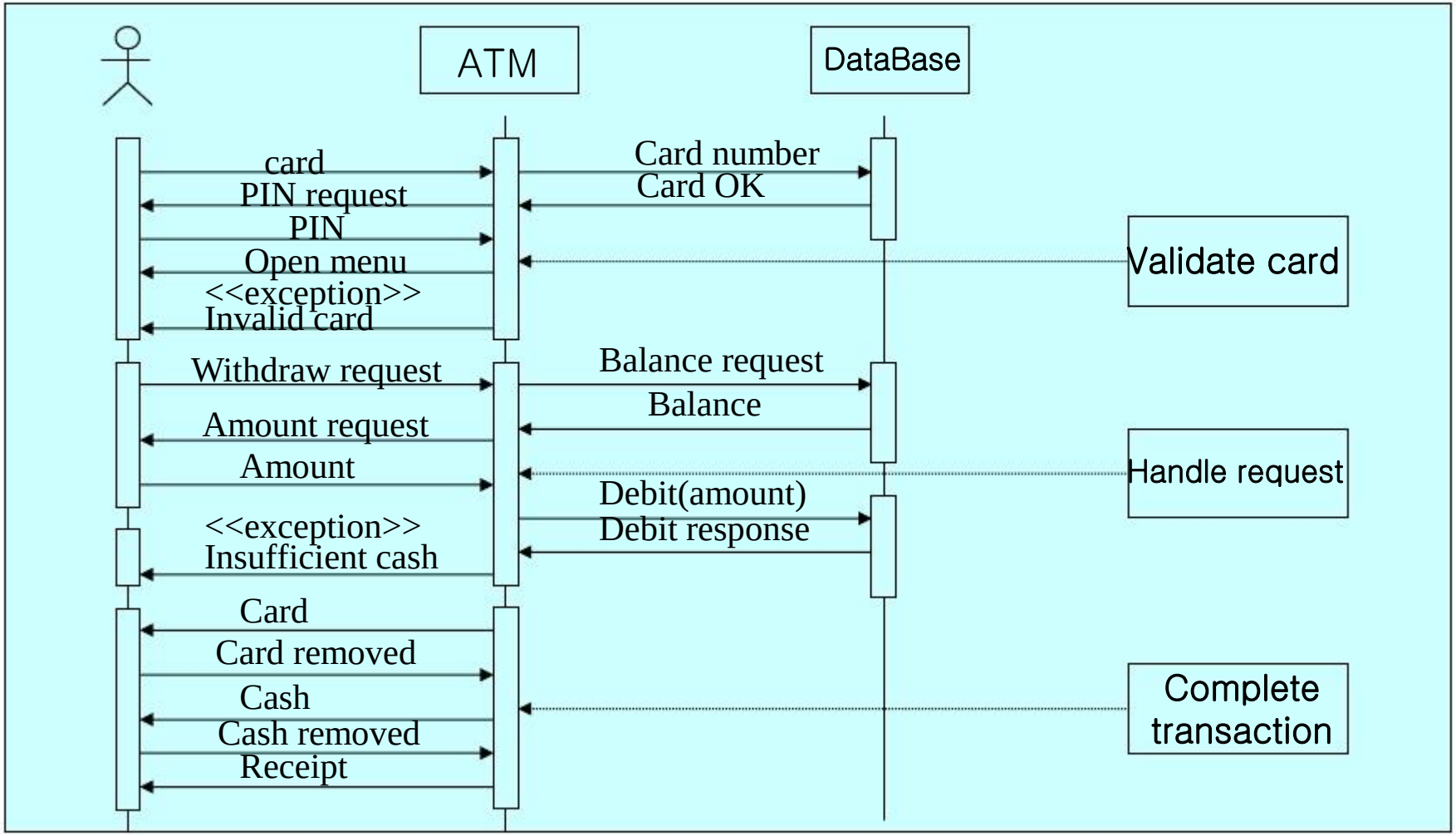
◆ 표준 양식을 사용한 시스템요구사항 명세

인슐린 펌프/제어 소프트웨어

- 기능 : 인슐린의 양을 계산한다 : 안전한 당 수치
- 설명 : 현재측정된 당수치가 3 과 7 사이의 안전지역에 있으면 투여될 인슐린의양계산
- 입력 : 현재 측정된 당 수치 (r2), 앞에서 읽은 두 번의 수치 (r0, r1)
- 출처 : 센서로부터 읽은 당 수치 , 메모리로부터 읽은 수치
- 출력 : CompDose- 투여될 인슐린의 양
- 목적지 : 주제어 루프
- 액션 : 당 수치가 안전적이거나 증가하였지만 , 증가 비율이 적으면 CompDose 의 값은 0, 만약 수치가 증가하였고 , 증가 비율도 증가하였으면 , CompDos 값은 현재 측정된 당 수치와 앞의 수치와 차이값을 4 로 나누어 반올림한다 . 만약 결과가 0 이면 CompDose 는 투여될 수 있는 최소의 양이 된다 .
- 필요사항 : 앞의 두 값을 읽어서 당 수치의 변경 비율을 계속할 수 있도록 한다 .
- 사전조건 : 인슐린 통은 단일 주사액보다 많은 인슐린의 양을 포함하고 있어야 한다 .
- 사후 조건 : r0 가 r1 으로 대체되고 , r1 이 r2 로 대체된다 .
- 부작용 : 없음

9.요구사항 문서화

◆ 그래픽 모델(순차 다이어그램)



Homework

◆ 요구사항 이해

8.1 요구공학 프로세스

8.2 요구사항을 추출하기 위한 가능한 자원

8.3 유스케이스 내용을 더 완벽한 관점을 제공하는 방법

8.4 요구사항 검증 방법

8.5 좋은 요구사항 문서 작성 전략

Project

1장. 프로젝트 개요
- 1.1 프로젝트 제목
- 1.2 선정 이유
- 1.3 팀 운영 방법

2장 시스템 정의
- 2.1 시스템 간략한 설명
- 2.2 유사 사례 간략한 설명

3장 프로세스 모델
- 3.1 규범적인 프로세스 모델 선정 및 이유
- 3.2 특수한 프로세스 모델 선정 및 이유

4장. 실무 가이드 원칙
- 4.1 각 프레임워크 원칙에서 중요한 3 개 정의
- 4.2 프로젝트 계획 보고서

5장. 요구사항 획득
- 5.1 기능 요구사항과 비기능 요구사항 정의
- 5.2 표준 양식을 사용한 시스템 요구사항 명세 3개 작성