

## QA Sheet 5

2017310695 이주민

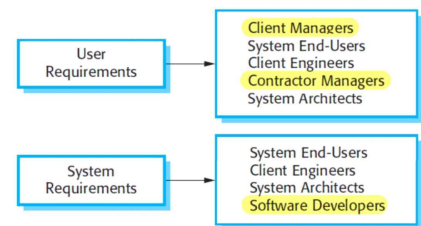
1. Define the notion of requirement and requirements engineering process.

- requirement : system service와 constraint를 담은 description으로 requirements engineering process 동안에 생성됨
- requirement engineering process : customer가 시스템에 원하는 service가 무엇이고, 시스템을 운영하고 개발하는 동안에 있는 constraints가 무엇인지 establish하는 과정

2. Compare user requirement and system requirement.

: 추상적인 정도에 따라 나뉨

- user requirements : system이 제공하는 services와 operational constraints에 대해 (자연어 + diagram + table)로 표현하며 보통 customer들을 위해 작성됨
- system requirement : system's functions, services, operational constraint를 담은 체계적이고 구조화된 description으로 client와 contractor 사이의 contract을 위해 작성됨



3. Compare functional requirement, nonfunctional requirement and domain requirement.

- functional requirement : system이 제공해야하는 services들의 statements로 ① 특정 input에 대해서 어떻게 반응하는지 ② 특정 situation에서 어떻게 시스템에 동작하는지 ③ 시스템이 해서는 안되는 것/해야 하는 것 기술
- nonfunctional requirement : service나 function의 constraint (ex. timing constraints, development process에서의 constraint, standard )으로 system 전반에 적용되는 system properties(emergent property인 safety, performance ..)에 대해 기술
- domain requirement : system's operational domain에서 오는 requirement로 functional/nonfunctional requirement들이 다 녹아들어가 있음.

4. Suppose that your client has a specific request for non-functional requirement such as high performance, and you have to determine the process model of the waterfall or agile model. Select a process model and explain why.

: system의 성능을 좋게 하기 위해서는 안정적인 개발이 가능한 waterfall model을 선택

- 고전적인 방법론으로써 적용 사례가 풍부
- 초기에 시스템 개발에 대한 계획을 모두 이루어지고 계획대로 진행이 되기 때문에 안정적
- non-functional requirement는 system 전반에 대해서 충족되어야 하므로 전체적으로 design하는 waterfall이 더 적합함

5. Explain why domain requirements are crucial for system development and difficult to articulate.

: domain requirement는 domain에서 사용되는 용어의 전문성 때문에 전문가들과 소통이 어려워지는 문제와 user가 developer가 이미 알고 있다고 생각해서 명확하게 표현하지 않는 문제 때문에 개발자가 개발하는 시스템의 domain을 학습해서 requirement를 도출해야 되기 때문에 어려움이 있다. Domain requirements가 충족되지 않으면 시스템이 제대로 작동하지 않기 때문에, 근본적으로 중요한 사항이 된다.

6. Discuss the requirement engineering process in the agile methods.

: Agile method를 지지하는 개발자들은 requirement이 너무나 빨리 바뀌고 있기 때문에 requirement document와 만들어진 system 사이에 gap이 발생할 수 밖에 없어서 프로듀싱한다는 자체가 시간낭비라고 생각함. 그럼에도 불구하고 requirement가 정리되어야 하므로 incremental requirements를 'user story' 형태로 정리함

(1) Requirements elicitation : 고객의 대표로서 PO(Product Owner)는 시스템의 기능, 개발 및 운영 상의 제약사항 등 Requirement를 제시함. 각 Increment에 대한 Requirement를 'User stories'로 표현함. 이를 Product backlog로 관리함. 최종적으로 프로젝트 도메인과 서비스, 제약사항들을 도출해냄.

(2) Requirements(User story) analysis and Plan sprint : User story를 분석하여 Task 단위로 분해함. 각 Task는 누가, 언제까지 수행할지, 그 결과에 대한 통합은 어떤 방식으로 진행할지 계획함. 이를 Sprint backlog로 관리함.

(3) Requirements(User story 혹은 Product backlog) management : Sprint Review를 통해 해당 Sprint 결과 User story가 제대로 반영되었는지 점검하며, 새롭게 추가되거나 변경된 User story는 Product backlog를 통해 관리하고 다음 Cycle에 반영하여 변화하는 요구사항들을 처리함.

7. Read and learn the form of requirement documents proposed by IEEE.

- Preface: 문서를 읽을 것으로 예상되는 사람들, 버전 히스토리에 대한 내용
- Introduction: 시스템의 필요성에 대한 설명
- Glossary: 앞으로 문서에서 사용될 용어들의 의미를 정의
- User requirements definition: 사용자를 위해 제공되는 서비스들에 대해 설명
- System architecture: 시스템의 아키텍처가 어떻게 구성될지를 개략적으로 설명( 구조설계 시에 함 )
- System requirements specification: User requirements를 상세화하여 나타냄
- System models: 시스템 요소들, 시스템, 시스템의 환경간의 관계를 나타내는 그래프 시스템 모델
- System evolution: 미래에 예측 가능한 변화에 대비한 시스템의 변경을 위한 가이드라인(방안)을 기술
- Appendices: 개발될 애플리케이션에 관한 세부적이고 구체적인 정보
- Index: 문서에 관한 몇 개의 인덱스를 나타냄

8. Describe the overlapping features between requirement and design.

: Architecture design, abstract design, interface design은 requirement engineering 단계에서 시작함

9. Explain the spiral view of the requirement engineering process.

: requirements elicitation, requirements analysis, requirements validation, requirements management와 같은 단계들이 반복적으로 일어남. 이러한 단계들이 일정한 주기를 가지고 반복적으로 일어나는 모양이 나선형의 구조를 가짐

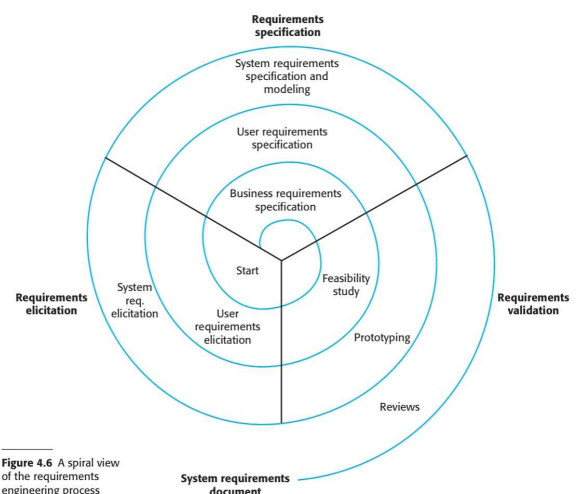


Figure 4.6 A spiral view of the requirements engineering process

10. Summarize the process of requirements elicitation and the problems that occur during the process.

- Requirements elicitation : 시스템과 관련성이 있는 이해당사자들을 통해 요구사항을 획득하는 것으로 요구사항에는 application domain을 찾아내고, 시스템이 제공해야 할 서비스들을 찾아내고, 요구되는 시스템의 성능을 찾아내고, 하드웨어 제약사항은 어떠한지와 같은 정보들이 포함됨
- Requirements elicitation은 Requirements discovery, Requirements classification and organization, Requirements prioritization and negotiation, Requirements specification의 단계들로 이루어짐
- 문제점
  - ① 이해당사자들은 그들이 정확히 무엇을 원하는지 알지 못함
  - ② 이해당사자들은 그들만의 언어로 요구사항을 표현하기 때문에 개발자가 이해하기 어려울 수 있음
  - ③ 이해당사자들이 여러 사람 있으면, 각자의 요구사항이 충돌하는 문제가 발생할 수 있음
  - ④ 조직적이고 정치적인 요소들이 시스템 요구사항에 영향을 줄 수 있음
  - ⑤ 분석 과정에서 요구사항이 계속 변화할 수 있음
  - ⑥ 새로운 이해당사자들이 등장할 수 있고, 비즈니스 환경이 변화할 수 있음

11. Interviewing is a popular and effective way to gather user requirements. But there are also strengths and weaknesses. Explain them.

- strengths : 이해당사자들이 어떤 일을 하고, 그들이 시스템과 어떤 방식으로 상호작용할지 overall understanding 하기에 좋음.
- weaknesses : domain requirement를 파악하기에 좋지 않고, requirements engineer들이 domain terminology를 이해하기 어려울 뿐만 아니라, 이해당사자들의 입장에서 어떠한 전문 지식은 구체화시키기 어려움

12. Describe a scenario about a specific service in your team project according to the template. And extract actors and use cases for each actor.

- ① Starting situation: 어떠한 목적지로 안전하게 이동하는 경로를 알기 위해 어플을 실행
- ② Normal flow of events: 자신이 가려는 곳의 목적지를 입력하고, 자신의 이동 수행 능력에 따라 필요한 기능을 요청
- ③ What can go wrong: 업데이트가 되지 않아 정확한 안내가 이루어지지 않았을 경우
- ④ Concurrent activities: 음성 안내 서비스, 큰 글씨의 화면을 제공함
- ⑤ When the scenario finishes: 사용자가 목적지에 무사히 도착함
- ⑥ Actors - Use Cases(구분되는 use case로만 작성)
  - 장애인 - 승강기 정보 이용, 공사중 도로 정보 이용
  - 임산부 - 경사로 안내 이용
  - 노약자 - 큰 글씨 기능 이용
  - 어린이 동반 보행자 및 어린이 - 공사중 도로 정보 이용, 내 아이 경로 찾기 이용

13. Describe the advantages and limitations of ethnography. And suppose its effective use situation.

- advantages : 이미 존재하는 과정을 이해하기에 좋음
- limitations : 시스템에 추가되어야 할 새로운 특징을 찾기에는 어려움
- Ethnography는 항공 교통 통제 과정에서 사용될 수 있고, prototyping과 결합되어 사용될 때 효율적이다.

14. Define the process of requirement validation and explain its importance.

: 요구사항이 시스템에서 사용자들이 원하는 요소와 일치한다는 것을 검증하는 과정임

- 요구사항을 잘못 정의했을 경우 발생하는 비용이 너무 크고, 통계적으로 시스템에서 대부분의 오류가 잘못된 요구사항에서 발생하기 때문에 요구사항 검증 과정은 매우 중요함

15. List the requirements validation techniques and explain what you can identify with each technique.

- Requirements reviews : 요구사항이 적힌 문서를 검토하는 과정으로, 요구사항의 시스템적인 매뉴얼 분석  
다른 요구사항에 영향을 주지 않는지, 검증가능한지

- Prototyping : 실제로 만든 모델을 시험해보면서, 어떠한 요구사항이 부족하거나 필요한지를 파악함  
고객의 요구를 충족할만한 실현 가능한 요구사항인지 확인

- Test-case generation : 요구사항에 직접 테스트 케이스를 대입해보면서,  
어떤 부분에서 오류가 발생하거나 보완이 필요한지를 파악함

16. Describe the process of requirement change management.

: 요구사항 변경이 수용되어야 할지 결정하는 단계

- Problem and change specification analysis : 현재 요청한 변경 사항이 시스템을 위해 타당한 변경인지를 분석함

- Change analysis and costing : 변경을 진행하는 것이 타당한지를 분석 man month로 판단

- Change implementation : 요구사항 문서, 시스템 디자인이 변경됨

## My Question

1. Ethnography는 왜 새로운 특징을 찾기 어려운지?

: 시스템을 사용하고 있는 현재의 stakeholder들을 바라보는 것으로 현재에 집중하므로 새로운 특징을 찾기 어려움 따라서, 현재의 상황과 테크닉을 가지고 미래를 디자인하는 과정이 필요

2. Test-case generation이 어떻게 requirements validation techniques로 사용될 수 있는지?

input에 대한 expected output을 넣으므로써 user가 그 output이 내가 원하는 것인지 아닌지를 판단할 수 있는 기준이므로 validation에 사용될 수 있음