

2019311801_이균서_QA_sheet_5

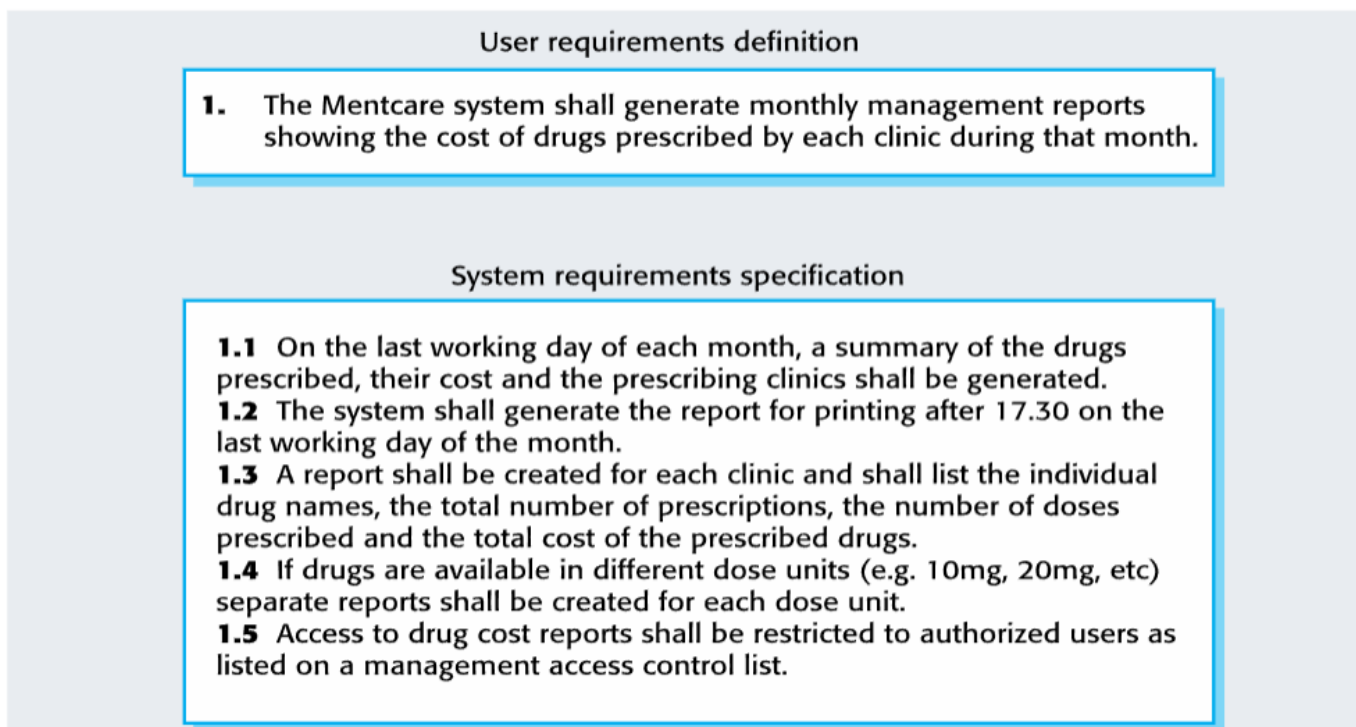
2019311801 이균서

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

- 요구사항(requirement)은 requirement engineering process 중에 생성되는 시스템 서비스 및 제약조건에 대한 설명이다.
- requirements engineering은 고객이 시스템에서 요구하는 서비스와 시스템이 운영되고 개발되는 제약 조건을 확립하는 프로세스이다.

2. Compare user requirement and system requirement.

- user requirement: 자연어로 된 설명과 시스템이 제공하는 서비스 및 운영 제약 조건에 대한 다이어그램이 포함되어 있다. customer를 위해 작성된다.
- system requirement: 시스템의 기능, 서비스 및 운영 제약 조건에 대한 자세한 설명이 명시된 구조화된 문서이다. 구현해야 할 사항을 정의하여 customer와 contractor 간의 계약의 일부가 될 수 있다.



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

- functional requirement:
 - 시스템이 제공해야 하는 서비스, 시스템이 특정 입력에 어떻게 반응해야 하는지, 특정 상황에서 시스템이 어떻게 작동해야 하는지에 대한 설명이다.
 - 시스템이 수행해서는 안 되는 작업을 명시할 수 있다.
- nonfunctional requirement:
 - 시기 제약, 개발 프로세스 제약, 표준 등 시스템에서 제공하는 서비스 또는 기능에 대한 제약이 있다.
 - 개별 기능이나 서비스보다는 시스템 전체에 적용되는 경우가 많다. 안전, 성능 등과 같은 시스템 속성.
- domain requirement:
 - 운영 영역에서 system에 대한 제약 조건

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.

높은 성능 요구 사항의 경우, waterfall이 더 적합하다.

waterfall을 이용하여, 아키텍처가 well-designed돼야 high-performance를 낼 수 있다고 생각한다.

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

- 시스템의 operational domain은 시스템에 requirements를 부과한다. 도메인 요구사항은 새로운 기능 요구 사항, 기존 요구 사항에 대한 제약 조건 또는 특정 계산을 정의할 수 있다. 그래서 도메인 요구 사항이 충족되지 않으면, 시스템이 작동하지 않을 수 있다. 그래서 domain requirements는 중요하다.
- software engineer가 domain을 이해하는 데에 어려움을 겪는 경우거나, 도메인 전문가가 해당 영역을 너무 잘 이해하고 아는 나머지, 도메인 요구 사항을 명시적으로 만들 생각을 하지 못하는 경우가 있기 때문에, domain requirements를 명확히 articulate하는 것이 어렵다.

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

agile method에서 requirement engineering은 지속적이고 반복적이다.

requirement은 종종 사용자 스토리 또는 사용 사례로 표현되며, 제품 백로그에서 우선 순위가 지정된다.

개발이 스프린트에서 진행됨에 따라 피드백과 변화하는 요구 사항을 기반으로, 요구 사항을 세밀화하거나 추가하거나 변경할 수 있다.

agile method안에서 requirement engineering이 없는 게 아니다. 여전히 중요하다.

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

IEEE는 요구 사항 문서의 표준 형식을 제안하여 명확성, 일관성 및 포괄성을 보장한다.

각 섹션은 특정 세부 정보를 제공하며 시스템의 모든 요구 사항 측면이 포함되도록 보장한다.

- **Preface:** 여기에는 문서의 예상 독자층을 정의하고 다음을 설명해야 한다. 새 버전 생성의 근거와 각 버전의 변경 사항 요약에 포함된 버전 이력을 확인할 수 있다.
- **Introduction:** 시스템의 필요성을 설명해야 한다. 시스템의 기능을 간략하게 설명하고 다른 시스템과 어떻게 작동하는지 설명해야 한다. 또한 시스템이 소프트웨어를 의뢰 하는 조직의 전체 비즈니스 또는 전략적 목표에 어떻게 부합하는지도 설명해야 한다.
- **Glossary:** 문서에 사용된 기술 용어를 정의해야 한다. 독자의 경험이나 전문 지식에 대해 가정해서는 안 된다.
- **User requirements definition:** 여기에서는 사용자에게 제공되는 서비스를 설명한다. 비기능적 시스템 요구 사항도 이 섹션에 설명해야 한다. 이 설명에는 고객이 이해할 수 있는 자연어, 도 표 또는 기타 표기법을 사용할 수 있다. 준수해야 하는 제품 및 프로세스 표준을 명시해야 한다
- **System architecture:** 이 장에서는 예상되는 시스템 아키텍처에 대한 개략적인 개요를 제시하여, 시스템 모듈 간의 기능 분포를 보여 주어야 한다. 재사용되는 아키텍처 구성 요소를 강조 표시해야 한다.
- **System requirements specification:** 여기에는 기능적 요구 사항과 비기능적 요구 사항이 더 자세하게 설명되어 있어야 한다. 필요한 경우 비기능 요구사항에 추가 세부 사항을 추가할 수도 있다. 다른 시스템에 대한 인터페이스가 정의될 수 있다.
- **System models:** 여기에는 시스템 구성 요소와 시스템 및 환경 간의 관계를 보여주는 그래픽 시스템 모델이 포함될 수 있다. 가능한 모델의 예로는 객체 모델, 데이터 흐름 모델 또는 시맨틱 데이터 모델이 있다.
- **System evolution:** 여기에는 시스템의 기반이 되는 기본 가정과 하드웨어의 발전, 사용자 요구 사항의 변화 등으로 인해 예상되는 변경 사항을 설명해야 한다. 이 섹션은 시스템 설계자가 향후 시스템 변경 가능성을 제한하는 설계 결정을 피하는 데 도움이 될 수 있으므로 시스템 설계자에게 유용하다.
- **Appendices:** 여기에는 개발 중인 애플리케이션과 관련된 상세하고 구체적인 정보(예: 하드웨어 및 데이터베이스 설명)가 제공되어야 한다. 하드웨어 요구 사항은 시스템의 최소 및 최적 구성을 정의한다. 데이터베이스 요구 사항은 시스템에서 사용하는 데이터의 논리적 구성과 데이터 간의 관계를 정의한다.
- **Index:** 문서에 대한 여러 색인이 포함될 수 있다. 일반적인 알파벳 색인뿐만 아니라 다이어그램 색인, 함수 색인 등이 있을 수 있다.

8. Describe the overlapping features between requirement and design.

원칙적으로 요구 사항에는 시스템이 수행해야 하는 작업이 명시되어야 하며, 설계에는 이를 수행하는 방법이 설명되어야 한다.

그러나, 실제로 요구 사항과 디자인은 분리할 수 없는 관계.

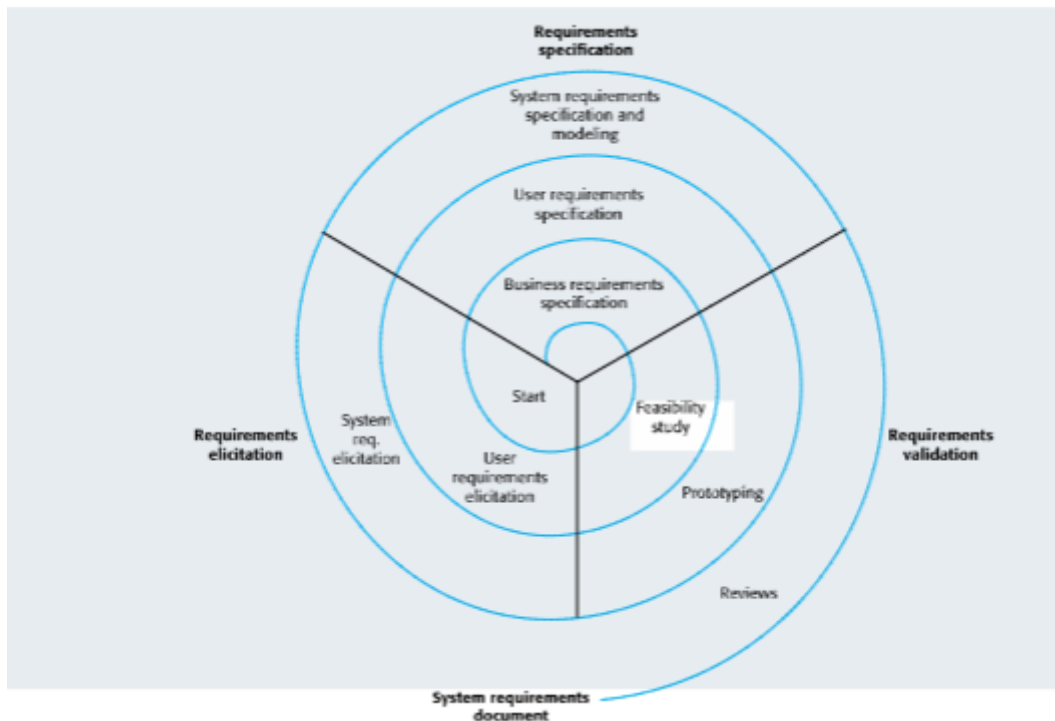
- 시스템 아키텍처는 요구 사항을 구조화하도록 설계될 수 있다.
- 이 시스템은 design requirements를 생성하는 다른 시스템과 상호 운용될 수 있다.
- 비기능적 요구 사항을 충족하기 위해 특정 아키텍처를 사용 하는 것은 도메인 요구 사항일 수 있다.
 - 이는 규제 요건에 따른 결과일 수 있다.

질의에 대한 교수님의 답변:

design작업 중에 architecture design, abstract design, interface design 등은 design작업임에도 불구하고 requirement engineering과정에서 사전에 어느 정도 이루어져야 하는 작업이기 때문에 두 과정에서의 중복되는 업무로 볼 수 있다.

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

- RE(requirements engineering)에 사용되는 프로세스는 애플리케이션 도메인, 관련된 사람 및 요구 사항을 개발하는 조직에 따라 매우 다양하다.
- 그러나 모든 프로세스에 공통적으로 적용되는 몇 가지 일반적인 활동이 있다.
 - 요구 사항 도출 (start (proposal document),user requirements elicitation, system req. elicitation)
 - 요구 사항 명세 (business requirements specification, user requirements specification, system requirements specification and modeling)
 - 요구 사항 유효성 검사 (feasibility study, prototyping, reviews)실제로 RE는 이러한 프로세스가 반복되는 반복적인 활동.

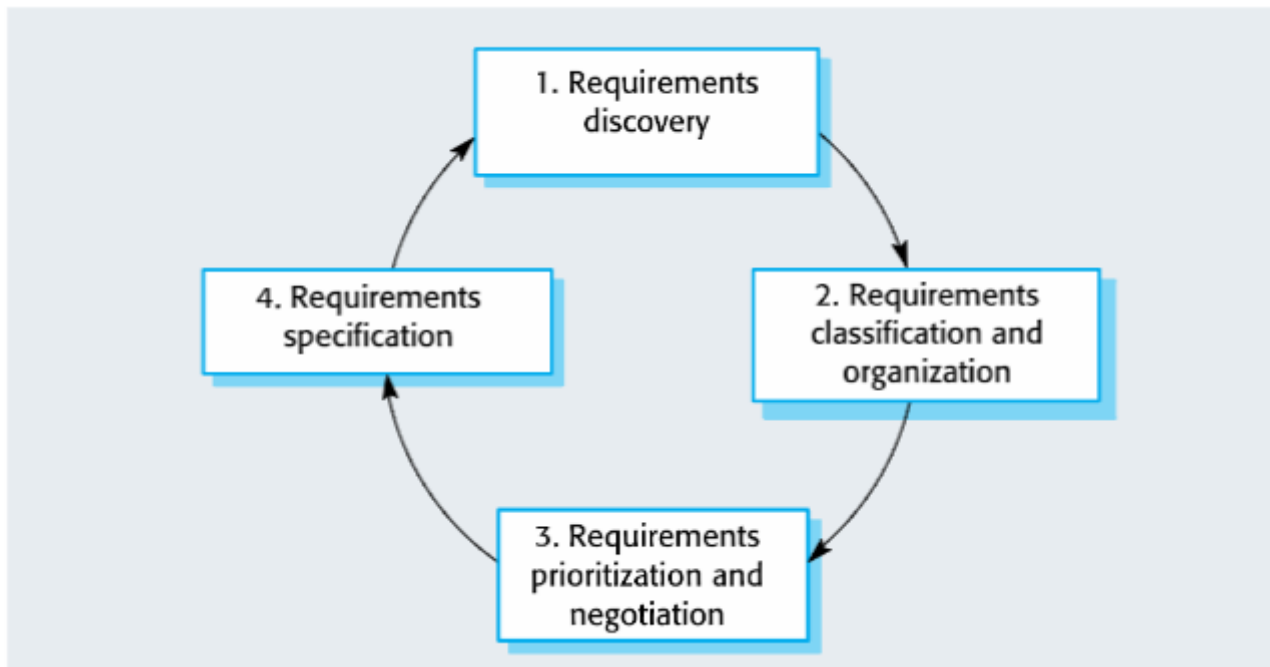


10. Summarize the process of requirements elicitation and the problems that occur during the process.

- 요구 사항 검색
 - 이해관계자와 소통하여 요구 사항을 파악한다.
 - 도메인 요구 사항도 이 단계에서 발견됩니다.
- 요구 사항 분류 및 구성
 - 관련 요구 사항을 그룹화하여 일관된 클러스터로 구성한다.
- 우선순위 지정 및 협상
 - 요구 사항의 우선 순위를 정하고 요구 사항 충돌을 해결한다.
- 요구 사항 명세
 - 요구사항은 문서화되어 나선형의 다음 라운드에 입력된다

requirements elicitation의 문제:

- 이해관계자는 자신이 진정으로 원하는 것이 무엇인지 모른다.
- 이해관계자는 각자의 용어로 요구 사항을 표현한다.
- 이해관계자마다 상충되는 요구 사항이 있을 수 있다.
- 조직 및 정치적 요인이 시스템 요구 사항에 영향을 미칠 수 있다.
- 분석 프로세스 중에 요구 사항이 변경된다.
- 새로운 stakeholder가 등장하고 비즈니스 환경이 변화할 수 있다.



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

인터뷰는 이해 관계자가 하는 일과 시스템과 상호작용하는 방식에 대한 전반적인 이해를 얻는 데 유용하다.

- 그러나 인터뷰는 도메인 요구 사항을 이해하는 데에 좋지 않다.
 - 요구사항 엔지니어는 특정 도메인 용어를 이해할 수 없다.
 - 일부 도메인 지식이 너무 익숙해서, 사람들이 표현하기 어렵다고 느끼지 않고, 표현할 가치가 없다고 생각하는 경우가 있다.

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.

start : 코드의 탄소 발자국을 계산하는 웹사이트에 유저가 접속한다.

normal : input으로 코드를 입력한다. server에서 코드를 분석한다. 이 때 필요한 상수값을 서버 내부에서 가져온다. 이를 바탕으로 탄소 발자국을 계산하여 알려준다.

what can go wrong: 코드의 execution runtime이 길거나, compile 오류가 있을 수 있다. 이 때는 에러를 user에게 알린다. 그 이후 코드를 수정하여 다시 입력하라는 알림을 띄운다.

other activities : server에서 처리 중인 코드가 있다면, user가 코드를 수정하거나, 코드를 다시 제출하려 해도 이는 적용되지 않는다.

system state on completion : user의 코드와 탄소 발자국이 web browser에 보여진다.

actor 유저 : 탄소발생량 확인.

서버 매니저 : 코드와 탄소 발자국 확인.

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

- 장점: 현존하는 process들을 이해하는 데에 효과적이다.
- 단점: 새로운 system에 추가돼야 할 새로운 feature를 확인하기가 어렵다.

질의에 대한 교수님의 응답:

여기서 unanswered question이라 함은 미해결된 질문이라는 의미로, ethnography를 통해 세밀한 관찰을 하면서 클라이언트의 숨겨진 의도나 작업상황 등에 대한 이해가 높아지고, 이는 prototyping을 하는 목적 중의 하나도 그것이기 때문에 prototyping의 작업 부하를 상당히 줄여 줄 수 있음을 의미합니다.

effective use situation:

- requirements를 직접 인터뷰를 하거나, 문헌을 조사하거나, 관련 시스템을 분석하여 얻기 어려운 경우.

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

요구사항 검증은 개발을 위해 정의된 요구사항을 확인하고, 고객이 정말로 원하는 시스템을 정의하는 프로세스이다.

요구사항과 관련된 문제를 확인하기 위해 요구사항 검증을 수행한다.

오류가 개발 프로세스 후반에 감지되면 과도한 rework가 증가할 수 있으므로 일반적으로 요구 사항 검증을 사용하여 개발 초기 단계에서 오류를 확인한다.

왜 중요한가?

구현이 다 되어 가는 상태에서 requirements가 invalid하다는 것을 알게 되면, cost가 더 많이 들게 된다.

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

1. Requirements reviews(요구 사항 검토): 요구 사항을 체계적으로 수동 분석. 검증 가능성, 이해하기 쉬운지, 추적성 등을 identify
2. Prototyping(프로토타이핑): 요구 사항을 확인하기 위해 시스템의 실행 가능한 모델 사용.
3. Test-case generation(테스트 케이스 생성) - 테스트 가능성을 확인하기 위한 요구 사항에 대한 테스트 개발.

16. Describe the process of requirement change management.

요구 사항 변경 관리에는 다음이 포함된다:

1. Problem analysis and change specification (문제 분석 및 변경사항 명세): 이 단계에서는 문제 또는 변경 제안이 유효한지 check하기 위해 분석한다. 이 분석 결과는 변경 요청자에게 피드백되며, 변경 요청자는 보다 구체적인 요구 사항 변경 제안으로 응답하거나, 요청을 철회할 수 있다. (**a validity of change request**).
2. Change analysis and costing (변경 분석 및 비용 산정): 제안된 변경의 효과는 추적성 정보와 시스템 요구 사항에 대한 일반적인 지식을 사용하여 평가된다. 이 분석이 완료되면, 요구 사항 변경을 진행할지에 대한 여부가 결정된다. (**a validity of change execution**)
3. Change implementation (변경 구현): 요구 사항 문서와 필요한 경우 시스템 설계 및 구현이 수정된다. 이상적으로는 변경 사항을 쉽게 구현할 수 있도록 문서를 구성해야 한다.

Q1 from myself

요구 사항이 충돌하는 경우에 해소를 어떻게 해야 하는가?

요구 사항에 우선 순위가 주어지면 좋을 것 같다.

Q2 from myself

요구 사항이 어떤 형식으로 주어졌을 때, agile이 어울리는 경우가 어떤 경우가 있을까?

사용자가 점점 늘어나면서, 요구 사항이 추가되거나 변경되는 시스템.