**Software Engineering Q/A Sheet (#5)**

date:2020/09/11 number:2016310936 name: 우승민

**Questions from Prof.**

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

Requirement는 system이 제공하는 services의 descriptions, requirements engineering process 동안에 발생하는 제약사항이다.

Requirement engineering process는 system에서 customer가 요구하는 services를 개발과정 중의 제약사항을 고려하여 설립하는 과정이다.

1. Compare user requirement and system requirement.

User requirement는 customers가 system이 제공해야 하는 것과 제약사항을 natural language로 작성하는 것이다.

System requirement는 system의 function, service, operational constraints를 상세하게 묘사한 structured document이다.

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

Functional requirement : system이 해야 하는 것과 하지 말아야 하는 기능 요구사항

Non-functional requirement : system에서 제공하는 service나 function에서의 제약사항이다. Functional requirement보다 critical하다.

Domain requirement : system의 운영환경에서 주어지는 requirement로 개발자가 스스로 도출해 내야한다.

1. 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.

Non-functional requirement가 많아지면 그에 따라 functional requirement가 많아지므로 agile model보다는 waterfall model이 더 적합할 것이다.

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

Domain requirements가 만족되어야 그 system 환경에서 정상적으로 작동할 수 있기 때문에 중요하다. 그런데 개발자는 그 system에 대해 자세히 알지 못하고 customer는 이에 대해 충분한 설명을 못해주기 때문에 명확하게 하는 것이 어렵다.

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

Agile methods은 incremental requirements engineering 과정이므로 필요한 requirements도 매번 추가되거나 바뀔 수 있다. 따라서 requirement engineering process가 비효율적이다.

1. Read and learn the form of requirement documents proposed by IEEE.
2. Preface : 예상되는 독자에 설정과 개발하는 System의 version의 역사와 새로운 version에 대한 요약 및 각 version에서 존재했던 변경점을 기술한다.
3. Introduction : System이 필요한 이유를 기술한다. System의 functions에 대한 간략한 기술과, System이 다른 system들과 어떤 식으로 사용되어질 지를 기술한다. 또한 system이 해당 software를 사용하는 단체의 business 혹은 전략적 목표에 어떻게 부합하는가를 기술한다.
4. Glossary : 해당 document에서 사용하는 기술적인 용어들에 대한 설명을 기술한다. 독자에 대한 경험 혹은 능력치를 가정하여 기술해선 안된다.
5. User requirements definition : User에게 제공되는 services들을 기술한다. Non-functional system requirement들 또한 기술되어야 한다. 자연어, diagrams, 혹은 기타 notations들을 사용하여 customer가 이해할 수 있게 기술되어야 하며, 충족되어야 하는 product, process standards 또한 기술되어야 한다.
6. System architecture : 전체적인 system architecture에 대한 high-level overview가 기술되어야 한다. 이는 전체 system modules에서 functions들의 분포를 담고 있어야 한다. 재사용 되는 Architectural components들은 반드시 강조되어야 한다.
7. System requirements specification : Functional, Non-functional requirements을 보다 자세하게 기술하여야 한다. 필요하다면 nonfunctional requirement들에 더 자세한 내용이 들어가도 된다. 타 system과의 interfaces 또한 기술되어야 한다.
8. System models : Graphical system model를 사용하여 system components과 system, system environment들의 관계를 보여주어야 한다. Object model, data-flow model, semantic data models같은 예시가 있다.
9. System evolution : System이 근반으로 잡고 있는 fundamental assumptions들에 대한 기술이 필요하다. 또한 hardware evolution, changing user needs, 등으로 인해 예상되는 변경점들 또한 기술되어야 한다. 이는 system designers들에게 용이하며, 후에 수정이 필요할 수 있는 결정들을 피할 수 있는 근거가 된다.
10. Appendices : 개발되어지는 application에 연관된 자세한 정보를 기술해야 한다. 예시로 hardware와 database에 대한 설명, system에 필요한 hardware requirement에 대한 구성, system에 사용되어지는 data에 대한 Database requirement의 논리적 구성, data간의 relationship등이 있다.
11. Index : Document의 index를 기술한다. 일반적인 alphabetic index와 diagrams에 대한 index, functions에 대한 index또한 포함된다.
12. Describe the overlapping features between requirement and design.

Architecture design, abstract design, interface design의 경우들이 requirement에서 시작하여 design에서 이어서 진행한다.

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

Requirements elicitation, analysis, validation, management 의 4가지 과정을 spiral 형태로 반복하는 것이다.

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

다양한 stakeholders에게서 application domain, services, constraints 들을 모으는 과정이다. 문제는 stakeholders가 정확히 뭘 원하는 것인지 모른다는 것이고, 서로의 requirements들이 충돌할 수 있다. 또한 analysis 과정에서 requirements가 변할 수 있다.

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

Interview는 stakeholders가 뭘 하는지, 어떻게 그들이 system과 상호작용하는지 전면적으로 이해하는데 좋다는 것이 장점이다. 단점은 domain requirements를 이해하는데 좋지 않다는 것이다. 이유는 engineers들이 domain의 전문적인 지식이 부족하여 용어를 잘 모르기 때문이다.

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

Team project가 아닌 개인 project이고, 아직 시작하지 못하였습니다. 죄송합니다.

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

장점은 현재 processes들을 이해하는 것에 효율적인 것이고, 한계점은 새로운 features을 식별하지 못한다는 것이다. 본문에 있듯이 air traffic control process의 상황에서 효율적이다.

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

Customer가 실제로 원하는 것과 requirement를 비교하여 검증하는 것이다. 실제로 defects가 절반 이상이 requirements 부문에서 일어나고 requirements에서 발생한 defects를 수정하는 비용이 크기에 validation이 무척 중요하다.

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

Requirements review : verifiability, comprehensibility, traceability, adaptability

Prototyping : validity, realism

Test-case generation : completeness

1. Describe the process of requirement change management.

Problem analysis and change specification 과정에서 change request의 validity를 얻고 change analysis and costing 과정에서 비용, 기술적 측면에서 change execution의 validity를 얻고 마지막으로 change implementation과정에서 실제로 바꾼다.

Questions from your ownself

1. formal review와 informal review의 차이점

Formal review는 명확하게 document가 존재하고, 관계자들이 정해진 시간, 장소에서 formal하게 진행한다. 반면에 Informal review는 미완성된 단계에서 검토한다.

2. Traceability가 중요한 이유

차후에 변경작업이 있을 때 traceability가 좋아야 어디 부분을 바꾸어야 하는지 명확하고 빠르게 알 수 있다.