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

date: 2023-10-10 number:2018312280 name: 이상수

**Questions from Prof.**

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

Requirement는requirements engineering process 과정 중에 도출해내는 개발해야할 시스템의 기능에 대한 설명과 제약사항들을 뜻합니다.

Requirement engineering은 customer가 원하는 서비스에 대한 requirements와 시스템이 작동될 배경의 제약사항을 명확히 하는 과정입니다.

1. Compare user requirement and system requirement.

User requirements : 시스템이 제공할 기능과 운영에 있어서 제약사항을 natural language와 diagram등으로 나타내어 customer가 읽을 수 있게 만든 document입니다.

System requirements : 시스템의 기능과 서비스, 그리고 운영에 있어서 제약사항들 대한 세부사항을 적은 체계화된 document입니다. 어떠한 기능이 구현되어야 하는지를 나타내며 client와 contractor간의 계약에 필요한 문서가 될 수 있습니다.

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

Functional requirement : 시스템이 제공해야할 서비스에 대한 설명으로, 시스템이 특정한 input에 대해 어떻게 반응해야하는지, 시스템이 특정한 상황에 어떻게 작동해야하는지에 대한 requirements입니다. 때로는 시스템이 해서는 안될 동작도 정의합니다.

Nonfunctional requirement : 시스템의 서비스나 기능에 있어서 제약사항들이 포함됩니다. 여기에는 개발과정과 기준에 대한 제약 등이 포함됩니다.

또한 이 nonfunctional requirements는 대개 시스템에 전체적으로 영향을 주는 시스템의 보안, 전체적인 기능 등과 같은 시스템 특성들을 포함합니다.

Domain requirement : domain에 관련된 시스템에 대한 제약사항들입니다. 이는 시스템이 작동할 domain에 대한 지식이 있어야 이해가 가능하며, customer가 제공을 잘 해주지 않기에 이에 대해 지식을 얻기위한 개발진의 노력이 필요합니다.

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.

Waterfall 모델이 좋아 보입니다.

Non-functional requirements는 시스템의 전체적 구조에 영향을 주는 requirements이기에 이를 최대한 명확하고 자세하게 정의하여 디자인에도 많은 투자를 해야합니다.

또한 non-functional requirements가 필요하다는 것은 그만큼 시스템 제작에 제약이 되기에 agile method를 적용하기 부적합합니다.

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

domain requirements는 시스템이 작동할 domain에서 요구하는 요구사항으로 이를 지키지 못하면 시스템 자체가 동작을 못하거나 사용이 불가능할 수 있습니다. 그렇기에 시스템 개발에 매우 중요한 역할을 하지만 이는 customer가 제공해주지 않고, 이 domain에 대해 개발진은 전문가가 아니기에 잘 알기 어려울 확률이 높습니다. 그렇기에 명확하게 파악하기 어려운 내용입니다.

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

Aigle method는 requirement documentation을 만드는 데에 많은 투자를 하지 않는 방법입니다. 그렇기에 document가 더 이상 유효하지 않을 경우도 있습니다.

그리고 XP(extreme programing) method에서도 requirements는 incremental하게 user story등을 이용하여 파악합니다.

이러한 방법은 개발이전에 많은 사전작업이 필요하거나, 여러 팀으로 구성된 개발에는 어울리지 않습니다.

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

Preface : 문서의 버전에 관한 정보. 버전에서 변화된 사항 기재

Introduction : 시스템 요구사항(작동을 위한 요구사항), 시스템 기능, 다른 시스템과의 연동방식, 조직에서 시스템의 역할 등을 기재

Glossary : 용어사전(누구든지 이해할 수 있게)

User requirements definition: 유저에게 제공되는 기능 소개. 자연어, 다이어그램 그리고 프로세스 기반등이 설명됨. 여기에 nonfunctional requirements 또한 포함됨

System architecture : 기능들이 어떤 모듈들에 적용 되어있는지 등의 시스템 구조 설명

System requirements specification:

functional ,nonfunctional requirements에 대한 자세한 설명.

System models:

시스템이 구동 환경에서 돌아가는 그래픽 모델

시스템의 구성요소간에 동작이나 환경과의 상호작용등이 묘사됨

System evolution :

시스템이 기반을 둔 내용과 거기서 변화된 내용을 기재

이 부분을 보고 시스템 디자이너들이 어떠한 부분이 미래에 시스템 변화에 영향을 줄지 파악함

Appendices :

Appendices는 자세하게 기록되어 개발에 관련된 정보를 잘 전달해야 합니다. 예를 들어 하드웨어 요구사항, 데이터 베이스의 로직등

Index :

Document에 관한 index. 다이어그램, 함수 등의 인덱스

1. Describe the overlapping features between requirement and design.

Requirements는 무엇을 해야하는지, design은 어떻게 해야하는지를 정의합니다.

시스템 디자인은 requirements를 구조화하기 위해 이루어집니다.

시스템은 design requirements를 추가로 요구하는 다른 시스템과 상호작용 할 수 있습니다.

non-functional requirements를 달성하기 위해 domain requirement에 필요한 특정한 구조를 디자인으로 활용해야 할 수 있습니다.

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

Spiral view는 requirement engineering process의 순환 과정을 보여줍니다.

Business requirements specification 부터 시작해서,

이를 실현 가능한지 validation합니다.

그 다음 User requirements를 이끌어내고,

다시 user requirements를 specific하게 세분화합니다.

그 다음 prototyping으로 validation을 진행하고

System requirements를 이끌어 냅니다.

System requirements 또한 specific하게 명세화 하고,

Review를 통해 이 requirements 들이 제대로 명세화 되었나 확인합니다.

이러한 과정을 필요하는한 계속 순환하며 최종 document가 만들어질 때까지 진행합니다.

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

Requirements discovery : stakeholders들과 협력하여 그들의 requirements를 알아내고 동시에 domain requirements를 이 때 파악합니다.

Requirements classification and organization

연관된 요구사항끼리 묶거나 체계화합니다

Prioritization and negotiation

요구사항의 우선순위를 정하고 requirements conflict를 해소합니다.

Requirements specification

Requirements를 document로 만들고 다음 단계로 넘어갑니다.

문제점:

Stakeholder도 스스로 원하는 것을 잘 모를 수 있다.

요구사항을 domain에서 사용되는 언어로 알려줄 수도 있다.

stakeholder끼리 충돌이 일어날 수 있다.

조직 또는 정치적 requirements가 있을 수 있다.

분석중에 requirements에 변화가 일어날 수 있다.

새로운 stakeholder가 요구사항을 추가할 수 있다.

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

Interviewing은 stakeholder들이 실제로 어떤 일을 하는지 등을 파악하여 그들이 시스템을 어떻게 활용할지를 예측하게 해줍니다.

하지만 interviewing으로는 domain requirements에 대한 지식을 얻기는 힘듭니다. 개발자들이 domain에 대한 지식을 이해하기 어렵고 어떨 때는 이를 설명해줄 필요를 못 느껴 제대로 설명해주지 않을 수 있습니다.

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.

Start : 코드의 탄소 배출량을 알기 위해 사이트에 접속

Normal : 코드를 input으로 넣으면 서버에서 이를 분석하고 필요한 cpu값 등의 상수값을 데이터베이스에서 가져와서 탄소 배출량을 계산해 알려줍니다.

What can go wrong: 코드가 너무 runtime이 길거나 오류가 있을 수 있습니다. 이럴 때는 에러를 유저에게 알리고 코드를 수정해서 올려달라고 알림을 띄웁니다.

Other activities : 이미 서버에서 처리중인 코드가 있을 때 유저가 코드를 수정하거나 코드를 다시 제출하려고 하여도 이는 적용되지 않습니다.

System state on completion : 유저가 사이트에 접속 되어있습니다. 유저의 코드와 탄소 배출량이 데이터베이스에 기록됩니다.

Actor 유저 : 탄소발생량 확인.

서버 매니저 : 유저의 접근 기록 확인, 유저의 코드와 탄소 배출량 확인.

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

직접 관찰하는 만큼, 미리 알지 못했던 사회적, 구조적으로 중요한 요인들을 알 수 있습니다.

이를 통해 기존에 시스템 모델을 통해 인식했던 것 보다 일의 복잡성과 난이도를 확인할 수 있습니다.

하지만 결국 관찰을 통해 얻는 정보인 만큼 관찰로 이해할 수 없는 정보는 얻을 수 없습니다.

이는 예를 들어 가게에 주문을 받는 시스템을 만들 때, 해당 가게에 가서 직접 어떠한 방식으로 주문을 받고 일을 하는지 알아 보는데 활용할 수 있습니다.

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

Requirements를 정의하는데서 에러가 생기면 이를 수정하는 작업은 엄청난 비용을 요구합니다. 그렇기에 이를 validate하는 과정은 매우 중요합니다.

Validation을 위해서는 몇가지 확인해야할 사항이 있습니다.

Validity 시스템이 고객의 요구를 만족하는 최적의 방식인지 확인합니다.

Consistency 요구사항끼리 충돌하거나 하는 일은 없는지 확인합니다.

Completeness 고객이 원한 모든 기능이 다 포함되었나 확인합니다.

Realism 비용과 기술을 고려했을 때 requirements가 실현 가능한지 확인합니다.

Verifiability requirements를 달성 하였는지 test가 가능한지 확인합니다.

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

Requirements reviews에서는 requirements가 현실적이고 테스트 가능한지, 이 요구사항이 제대로 이해 되었는지, 요구사항을 처음에 누가 제시했는지, 그리고 요구사항에 변화가 일어나도 요구사항 전체에 영향을 많이 주지는 않는지 체크할 수 있습니다.

Prototyping으로는 모델의 프로토타입을 만들어 시스템에 적용된는 requirements를 미리 확인해 볼 수 있습니다.

Test-case generation방법은 requirements에 대한 test를 직접 만들어보며 test 가 test가능한지 확인할 수 있습니다.

1. Describe the process of requirement change management.

– Problem and change specification analysis

이 단계에서는 문제와 변화 요구를 분석하여 이게 가능한지 판단합니다. 이 때 분석을 다시 요청자에게 전달하여 좀 더 자세한 requirements 변화 요청을 받거나 요구를 불응할 수도 있습니다. 결국 change request를 받을지 안받을지 판단하는 과정.

– Change analysis and costing

시스템에 대한 정보와 요구사항을 기반으로 변화에 영향력을 판단하여 requirements change를 진행 할지 판단합니다. 결국 변화를 실제로 시행할지 판단하는 과정.

– Change implementation

변화를 수행하고 나면, requirements document와 필요하면 시스템 디자인과 활용도 수정해야합니다. 그래서 미리 document를 수정하기 용이하게 체계화 해두어야합니다.

Questions from your ownself

1. 요구사항의 충돌이 해소가 어려운 경우는 어떻게 할까?

충돌하는 요구사항중 하나를 포기한다.

1. 요구사항이 어떤 경우 agile이 어울릴까?

요구사항이 처음부터 주어지는 것은 적지만 사용자들이 이용하다보면 요구사항이 점점 늘어나는 시스템