**Software Engineering Q/A Sheet (#3)**

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

**Questions from Prof.**

1. Explain the strengths and weaknesses of the Waterfall model.

Waterfall model은 parallel development와 management가 쉽다는 장점이 있다. 단점은 중간에 변경이 힘들다는 것이다.

1. Explain the strengths and weaknesses of the incremental model.

장점으로는 requirements가 변할 때 수용하는데 드는 비용이 적다는 것이다. 따라서 고객들에게 software를 배포하는 것이 빠르다. 단점은 process가 명확하지 않고, 지속적으로 수정할 경우 system이 무너져 내린다.

1. Look for refactoring concepts and find examples of them.

Refactoring은 외부 동작을 바꾸지 않으면서 내부 구조를 바꾸는 방법이다.(코드의 가독성을 높임, 새로운 기능 추가하는 것 x)

Refactoring의 예시로는 Extract Method, Move Method, Introduce Explaining Variable등의 방법론이 있다. ex ) eclipse, android studio, visual studio

1. Compare the waterfall model with the incremental model for the workflow.

Waterfall model은 미리 세워진 계획에 따라 순서대로 진행되고, incremental model은 specification, development, validation이 반복 교차하여 지속적으로 수정하면 진행한다.

1. Describe four fundamental process activities.

Specification은 고객이 가지고 있는 요구사항을 명확히 또는 시스템을 개발하는데 충족해야 하는 제약사항들을 명확히 하는 작업이다.

Development는 Designed and programmed로 Specification을 포함하기도, 하지 않기도 한다.

Validation은 개발된 소프트웨어가 실질적으로 고객의 요구사항, 제약사항을 모두 충족하는지 확인하는 작업이다.

Evolution은 많은 변화요인에 맞추어 시스템을 변화시켜가는 과정이다.

1. Describe the requirement engineering process with key activities and their work products.

Feasibility study, report에서 business적으로 이득이고, technically 가능한지, budget안에서 가능한지 타당성 조사를 한다.

Requirements elicitation and analysis 에서 요구사항 중 반영할 것을 추출하여 system descriptions을 만든다.

Requirements specification은 요구사항을 명세화하는 과정이다.

Requirements validation은 요구사항을 확인하는 과정이다. Requirements document를 작성한다.

1. Describe the differences between component, system and acceptance tests.

Components testing : 각각의 components를 독립적으로 test한다. System testing : 전체 시스템을 test한다.(비기능적 요구사항 포함) acceptance testing : customer의 data로 test한다.

1. Slide On page 28, discuss when to plan each test with why.

Acceptance testing은 고객의 환경에서 진행해야 하므로 초기에 고객의 requirements를 specification할 때, 이에 대한 testing plan을 같이 세우기 때문이다.

System integration test와 sub-system integration test도 마찬가지로 system specification과 system design단계에서 plan을 세우기 때문이다.

1. Change is inevitable in every large software project. There are two related approaches you can use to reduce rework costs. First, explain the concept of rework and describe two approaches.

Rework는 이미 만들어진 것에 대해 새로운 requirements가 발생하였을 때 수정과정을 거치는 것이다.

Change avoidance : prototype을 만들어 고객에게 빠르게 보여주어 불명확한 부분을 최대한 줄이는 것이다.

Change tolerance : incremental development의 형태로 부분적으로 개발하고 평가하는 과정을 반복하여 추후에 수정할 부분을 수용하기 쉽게 만드는 것이다.

1. Discuss the benefits of prototyping. And explain the reasons for improved maintainability on page 38 of the slide.

Prototyping을 사용하면 system usability, design quality, maintainability가 늘어나고 고객의 요구를 맞추기 쉽고, 개발하는 데 드는 노력이 줄어든다.

고객의 requirement를 더 명확하게 개발하면 추후의 rework 비용이 줄어 maintainability가 늘어난다.

1. Compare throw-away prototyping with exploratory prototyping.

Throw-away prototyping은 requirement를 명확하게 하는 과정이므로 만족하였을 때 바로 버려진다.

반면에 Exploratory prototyping은 prototype model에서 계속 추가하여 최종 단계까지 개발하는 것이다.

1. With incremental delivery, customers can feel more confident about the system. Guess why.

Incremental delivery은 부분적으로 개발하기 때문에 고객에게 빠르게 결과를 보여줄 수 있고, 가장 중요한 요구가 먼저 개발되어 testing도 많이 받아 고객들이 더 만족할 수 있다.

1. Complete testing is virtually impossible. Discuss cost-effective testing strategies in development environments where time and cost are inadequate. Hint: Take advantage of available statistics.

이전에 비슷한 system이 있는지 확인해보고 있으면, 그 system을 개발하였을 때 사용했던 test cases와 사용 결과들을 참고하면 시간과 돈을 아낄 수 있을 것이다.

Questions from your ownself

1. Cleanroom sw development model이 도입된 이유?

많은 양의 defect를 제거해도 사용자의 reliability는 별로 늘지 않기 때문

1. Back-to-back tests가 무엇인지?

동일한 제품을 여러 방식으로 구현한 후 동일한 test case를 사용하는 방법.