**Software Engineering Q/A Sheet (#4)**

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

**Questions from Prof.**

1. Discuss and write your own ideas on why rapid development and delivery is important from a business and technology perspective.

빠르게 개발하고 배포하면 requirements를 명확히 하기 더 좋기 때문이다.

1. Explain the principle of agile methods with each reason.

Customer involvement : customers가 개발에 참여하면 requirements의 우선순위를 정하는 것이 쉽고 system을 바로 평가할 수 있다.

Incremental delivery : software는 customer가 각 increment에서 특정한 요구사항을 위주로 점진적으로 개발한다.

People not process : 팀원끼리 공유하여 함께 개발하기 때문에 서로의 개발 능력을 알고있어야 한다.

Embrace change : system requirements가 바뀔 수 있기에 변화에 맞출 수 있도록 design 해야한다.

Maintain simplicity : 개발된 software와 개발과정은 변화를 쉽게 수용할 수 있게 간단하게 만들어야 한다.

1. Discuss the strengths and problems of agile methods.

장점 : customer가 개발에 참여하여 clear한 요구사항을 알 수 있다. 개발과 배포가 빠르다.

단점 : customer가 개발에 오래 참여하기 어렵다. Stakeholders가 많으면 우선순위를 정하기가 어렵다.

1. Summarize the determinants when deciding on a plan-driven or agile approach to system development.
2. 상세한 명세서나 design이 필요하면 plan-driven이 좋다.
3. customer에게 feedback을 빠르게 받는 incremental deliver 전략이면 agile이 좋다.
4. 큰 team에서 큰 system을 개발하는 것이면 plan-driven이 좋다.
5. Implementation 전에 분석이 많이 필요하면 plan-driven이 좋다.
6. Lifetime이 긴 system은 더 정교한 design이 필요하므로 plan-driven이 좋다.
7. Evolving을 지속할 좋은 기술이 있으면 agile이 좋다.
8. Development 팀이 분산되어 있거나 development의 일부를 outsourced하면 plan-driven이 좋다.
9. Traditional engineering organizations이 있으면 plan-driven이 좋다.
10. 높은 기술 수준을 가지고 있어야 agile이 좋다.
11. System이 외부적인 제약이 많으면 plan-driven이 좋다.
12. Discuss the release cycle of XP and compare it to the waterfall model.

XP ‘select user stories fore this release’ -> ‘break down stories to tasks’ -> ‘plan release’ -> ‘develop/integrate/test software’ -> ‘release software’ -> ‘evaluate system’ 순으로 진행되고 waterfall model과 다르게 남는 stories가 없을 때까지 위의 과정을 반복한다.

1. Summarize the important features on practices of XP.

Incremental planning : requirements을 story cards로 만들고 배포 가능 시간과 우선 순위에 따라 tasks로 나눈다.

Small releases : business 가치가 있고 최소한으로 사용하는 것을 먼저 개발한다. 계속 기능을 추가하며 빈번하게 배포한다.

Simple design : 현재 개발에 넘치지 않고 만족할 만큼만 design한다.

Test-first development : test code를 먼저 작성하고 application code를 작성한다.

Refactoring : code를 보기 쉽게 만들기 위해 refactoring한다.

Pair programming : driver과 observer가 짝을 지어 개발한다.

Collective ownership : 조직의 모든 사람이 개발에 참여하여 인원이 변경되어도 지장이 없게 한다.

Continuous integration : task를 완성하면 즉시 system에 통합하여 모든 unit tests를 확인해본다.

Sustainable pace : 너무 많은 작업을 하지 않는다.

On-site customer : customer은 개발팀에 참여하여 requirements를 선정하는데 책임감을 가져야한다.

1. Read the story card on slide 22 carefully and look for tasks 4 or above.

처음 시작인 Medication field를 열어서 ‘current medication’, ‘new medication’, ‘formulary’ 중 하나를 선택하는 과정이 추가되어야 한다.

‘current medication’을 선택했을 때 dose를 바꿀 것인지 확인하는 과정이 추가 되어야한다.

‘new medication’을 선택했을 때

1. 약의 첫 몇 글자를 입력한다.

2. 약 list 중에 원하는 것을 고른다.

3. 고른 약이 맞는지 다시한번 확인한다.

과정이 추가되어야한다.

마지막 단계인 Prescription을 확인한 후 ‘OK’와 ‘Change’의 선택지를 주어

‘OK’ : 선택한 prescription이 audit database에 기록된다.

‘Change’ : 다시 ‘prescribing medication’ 단계로 돌아간다.

내용이 추가 되어야 한다.

1. Explain test-driven development and think about the impact on SW quality.

Code를 만들기 전에 test를 먼저 만들어 구현할 requirements를 명확하게 한다. Tests는 자동으로 실행되게 programs으로 만든다. TDD를 사용할 경우 버그가 많이 줄어들고 코드가 simple해져 유지비용이 감소한다.

1. Summarize the advantages of pair programming.

조직에 모든 개발자가 개발에 참여하고 모든 내용을 공유하여 일부 사람이 교체되어도 개발 진행에 지장이 없다.

Observer가 개발과정에서 review를 해주어 refactoring에 도움이 된다.

1. Explain the SCRUM process on slide 40.

Outline planning and architectural design에서 story cards를 명세화하고 project 목표와 system architectural을 design한다.

Sprint cycle은 outline planning and architectural design 단계에서 정해진 story cards을 우선 순위에 따라 선택, 개발, review를 모든 story cards가 끝날 때까지 반복한다.

Project closure은 필요한 문서를 작성하며 마무리하는 단계이다.

1. Summarize the benefits of the SCRUM.

Product가 관리가 편하고 이해하기 쉽게 나누어진다. 불필요한 요구사항이 진행을 방해하지 않는다. Visibility가 좋아지고 team communication이 향상된다. Customers가 제시간에 배포를 받고 feedback을 바로 해주어 customers과 developers 사이의 신뢰가 높아진다.

1. Summarize the strategies for scaling up and scaling out of agile methods.

Scaling up은 agile method를 large software systems에서 사용하기 위해 사용하는 방법이고 scaling out은 software을 개발 경험 오래된 large organization에서 사용하기 위한 방법이다.

Questions from your ownself

1. scrum master가 무언인지

sprint cycle 동안에는 team이 customer가 단절된다. 이때 channel 역할을 하는 사람이 scrum master이다.

1. Refactoring을 하는 이유

Code의 visibility를 높여 한번에 이해하기 쉽게 만들어 나중에 변경사항이 있을 때 수정하기 쉽게 해준다.(maintenance의 약점을 보완)