Q/A Sheet #1- Introduction to SE

date: 09/06 number: 2018312280 name: 이상수

Questions from Prof.

1. On slide 7, why is the cost distribution pattern of the iterative and component-based model different from the waterfall?

waterfall은 이름대로 폭포수처럼 specification , Design, ,Development, Integration and testing 순으로 순차적으로 균형 있게 진행되며 개발하는 과정 까지만 고려하고 고객에게 보낸 이후는 고려하지 않아 전체적으로 고루 분포된 모형을 보입니다.

그에 반해 iterative는 처음 specification을 받은 후 고객 또한 자신이 specification을 준 것이 완벽한지 모르기에, 개발하면서 확인 받고 다시 개발하고 또 다시 확인 받는 과정을 iterate 하기에 iterative development 파트가 매우 길게 나타납니다.

component-based model은 개발은 이미 존재하는 component를 모아서 하는 것이다 라는 방법론으로 그렇기에 component를 모은 다음 testing 하는 “integration and testing” 파트가 가장 큰 비중을 차지합니다.

1. Explain the trade-off between the essential attributes on slide 12.

프로그래머가 Maintainability를 중요시 하면, 나중에 유지보수가 쉬워 지기는 하겠지만, 그만큼 프로그램이 수정하기 쉽고 이해하기 쉬워야 하기에 그렇게 코드를 짜는 것은 어려울 수 있습니다. Dependability and Security를 중요시하면 안전하고 정확한 프로그램이 되지만, encryption이나 데이터 분할 등 추가적인 보호조치가 취해져야 하기에 그만큼 프로그램이 복잡해지고 알아보기 힘들어집니다. Efficiency를 중요시하면 메모리 소모나 계산하는 시간이 줄어들어서 좋지만 코드가 길어지거나 더 알아보기 어려워질 수 있습니다. Acceptability는 유저가 사용하기 좋고 유저가 프로그램 개발자에게 더 호의적이게 되겠지만, 유저의 needs를 전부 만족시키기는 어렵고, 특정 유저의 희망에 더 비중을 두어 만들다 보면 다른 중요한 부분에서 문제가 발생 할 수 있습니다.

1. Compare the difference between validation and verification activities.

verification은 프로그램이 요구에 맞게 만들어졌나 확인하는 과정, validation은 유저가 원하는 프로그램이 맞는지 확인하는 과정입니다.

1. Explain four fundamental software engineering activities, such as specification, development, validation, and evolution.

specification은 고객 또는 개발자가 소프트웨어가 실행해야 할 과제, 그리고 제약사항 등을 정합니다. development은 소프트웨어를 디자인하고 실제 프로그램을 하는 과정입니다. validation은 고객이 요구한 조건을 소프트웨어가 맞는지 확인합니다. evolution은 고객과 시장의 요구가 바뀌면 그에 맞춰 소프트웨어를 바꾸는 과정입니다.

1. Discuss the types of applications on Slide 18-20 and find more than two examples for each type.

Stand-alone application은 특별한 추가기능이나 네트워크 없이 작동하는 프로그램입니다. 예시로는 윈도우 컴퓨터에 있는 캡쳐 프로그램, 스티커 메모장 등이 있습니다.

Interactive transaction-based application은 외부 컴퓨터에서 작동하나 사용자들의 개인 컴퓨터로 접속하여 작동하는 어플리케이션입니다. 예시로는 네이버 검색프로그램, 넷플릭스 등이 있습니다.

Embedded control systems은 하드웨어장치들을 작동하는데 사용되는 시스템입니다. 예시로는 자동차 블랙박스 카메라의 시스템, 자동 판매기의 시스템이 있습니다.

Batch processing systems은 큰 집단의 데이터를 입력받아 처리하는 시스템입니다. 예시로는 은행거래 시스템, 회사 재고 관리 시스템이 있습니다.

Entertainment systems은 유저가 즐거움을 느끼도록 해주는 것을 목적으로 하는 시스템입니다. 예시로는 컴퓨터 게임, 음악 스트리밍 프로그램 등이 있습니다.

Systems for modelling and simulation은 물리현상이나 상황을 모델링하고 시뮬레이션 하기 위해 사용됩니다. 예시로 CFD , Abaqus가 있습니다.

Data collection systems은 특정 환경에서 센서들을 이용해 데이터를 수집하는 시스템입니다. 예시로는 기상관측 시스템, 환경 모니터링 드론 등이 있습니다.

Systems of systems은 다수의 시스템으로 이루어진 복합 시스템을 의미합니다. 예시로는 공항 운영 시스템, 도로 교통 관리 시스템이있습니다.

1. Explain why you should choose different software engineering techniques, methods, and tools depending on the context of the software project.

소프트웨어는 그 용도에 따라 개발 방식이 얼마든지 달라질 수 있기 때문입니다. 간단하게 게임을 만드는 프로그래머와 웹사이트를 만드는 프로그래머가 있습니다. 이 때 게임이 플레이스테이션용 솔로 플레이 게임일 경우 게임은 한번 만들면 이후 버그 수정만 하면 다른 요소는 손을 볼 필요가 없습니다. 그에 반해 웹사이트는 만들고나서 새로운 광고나 새로운 정보 등이 들어오면 그 부분을 지속적으로 업데이트 해주어야 합니다. 이럴 때 유지보수에 더 큰 비중을 두어야하는 프로그래머는 웹사이트 프로그래머입니다. 또한 네트워크 시스템을 포함해야하는지 말아야하는지도 둘의 차이점이 될 수 있습니다. 만약 웹사이트를 만들 때 처럼 유지보수가 쉽게 게임을 만든다면 오히려 악성 이용자들이 게임의 구조를 쉽게 파악해서 크랙버전이나 도용을 위한 조작을 시도할 수 있습니다. 이렇듯 같은 프로그래머라도 만드는 프로젝트에 따라 그 프로그램 제작에 있어 다양한 요소를 취사선택 해야합니다.

1. Think about some fundamental principles that can apply to all types of software systems.

소프트웨어는 어느정도 추상화가 있어야합니다. 아무리 간단하고 보안상 위험이 없는 소프트웨어이더라도 변수의 추상화, 객체의 추상화 정도는 존재 해야지 이를 프로그래밍 하는 사람도, 이를 이용하는 사용자도 편리성을 느낄 것입니다.

1. Describe the distinctive changes in the software development process when using web or web services as a technical platform.

웹을 사용하게 되면 네트워크를 고려해야합니다. 웹에서 데이터를 주고받을 때 네트워크 상태에 따라 해당 프로그램의 성능이 변할 수 있습니다. 그리고 웹과 연결되는 만큼 보안에도 신경을 많이 써야합니다. 특히 사용자 개인정보보안에 관심을 두어야합니다. 대신 웹을 사용하는 만큼 개발과정에서 개발자가 가지고만 있던 데이터나 시스템들 말고 웹에서 가져올 수 있는 데이터나 시스템도 개발에 적극 활용 할 수 있습니다.

1. Investigate the fatal consequences caused by unethical behavior or decisions by software engineers.

대표적으로 최근 사건으로 보자면 넥슨에서 만들던 게임 아이디어와 에셋을 가지고 외부에서 다른 게임을 만든 다크앤다커 게임이 사건이있습니다. 이러한 행위가 지속될 시 게임개발에 있어 개발자들은 점점 그 의욕을 잃게 될 것이고 게임 시장에도 큰 타격을 줄 수 있게 됩니다.

1. Discuss with your colleagues what knowledge and efforts you need to be a competitive software engineer.

제가 생각하기에는 자신의 프로그램을 지속적으로 트래킹하는 능력이 매우 중요하다 생각합니다. 프로젝트가 조금만 커져도 자신이 짠 프로젝트의 코드들을 잘 기억 못하거나 프로젝트의 요소들을 제대로 파악하기 힘들어질 수 있습니다. 그렇기에 사전에 코딩할 때 중간중간에 기록을 하여 코드의 흐름을 잃지 않게 유지하는 것이 중요한 것 같습니다.

Questions from Student

1. Have you ever had bad experiences from programs that are not ethical.

2. Talk about examples of wise choice of software engineering techniques or methods.