Q/A Sheet #1 – Introduction to SE

2016314576 Choi Jaehyuk

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

- Iterative model is used when the user's requirements are not clear. This reduces the cost of specifying requirements and since series of development work is iteratively proceeded, development cost is much more than that of waterfall model. Also, in the case of a component-based model, since software is developed in small units called components, it requires a lot of cost in the process of integration.

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

- Maintainability to continuously meet user requirements is in a trade-off relationship with Dependability. In order to continuously satisfy the user's requirements, the program becomes heavier and more branching occurs. Therefore, when a system failure occurs, the damage received increases and the recovery function becomes more complex.

3. Compare the validation and verification activities.

- Validation is task that check whether finally developed software satisfies customer’s requirements. For validation, software engineers used to do usability test. On the other hand, verification is checking whether the functions required at a specific stage of the development cycle are normally performed. One of the activities for verification is verify the behavior of the code at each development sequence.

4. Explain four fundamental software engineering activities.

- First, there is a software specification that clarifies the customer's requirements and the constraints that must be met. Second, software development is a comprehensive term that includes design and programming. At this stage, the source code can be turned into executable files or the software can be designed in a form that meets the requirements. Third, software validation is activity that check whether the user's requirements are truly satisfied. Finally, software evolution is also evolving software to meet the constantly changing needs of customers and markets.

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

- Stand-alone applications: Applications that run on a local computer, which includes all necessary functionality. (Ex. Calculator, Notepad)
- Interactive transaction-based applications: Applications that execute on a remote computer. It needs method for communicating with other PCs. (Ex. Mail, Phot sharing)
- Embedded control systems: Software control systems that control and manage hardware devices. There should be few changes after development. (Ex. Software systems in air-conditioner, television)
- Batch processing systems: It is processing systems dealing with data in large batches. (Ex. Phone billing systems, Bank transaction system)
- Entertainment systems: Systems that are for personal use and which are intended to entertain the user. Quality of the user interaction offered is the most important thing. (Ex. Games, Netflix)
- Systems for modelling and simulation: Systems that are developed by scientists and engineers to model physical processes or situation. (Ex. Solar activity models, Highway and street traffic models.)
- Data collection systems: Systems that collect data from their environment using a set of sensors and send that data to other systems for processing. (Ex. Data registries, case management system)
- Systems of systems: Systems that are composed of a number of other software systems. (Ex. Embedded Automotive Systems, Manufacturing Supply Chains)

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

- Each software project has a different final product to implement. Therefore, the business model, development cost, replacement cycle, and fatality level when a system failure occurs are all different. This is the reason why we cannot proceed development in the same way every time.

7. Think about some fundamental principles that can apply to all types of software Systems.

- Specify development process and customer's requirement.
- Dependability: ability of the software to perform its requested functions accurately, consistently and without error.
- By reusing software that has already been used, reliability can be secured and development costs can be reduced.

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

- By using web or web service as a technical platform, many individual developers have been able to share their code externally. Therefore, they can work together on large-scale projects without being affiliated anywhere.

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

Flight crashes

- In 1994 in Scotland, a Chinook helicopter crashed and killed all 29 passengers. While initially the pilot was blamed for the crash, that decision was later overturned since there was evidence that a systems error had been the actual cause. Another example of a software-induced flight crash happened in 1993, when an error in the flight-control software for the Swedish JAS 39 Gripen fighter aircraft was behind a widely publicized crash in Sweden.
- This isn't a case of software engineers intentionally acting maliciously, but it teaches the lesson that small mistakes they make can lead to catastrophic accidents.

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

- After I decided on my career as a software engineer, I heard the most saying is, "A software engineer must study constantly." Studying can be computer science knowledge, a new language, or a new platform environment. The software industry is changing very rapidly. Users' needs change from year to year, and the level of technology required is constantly increasing. In order to save my competitiveness, I will always have to accept new knowledge with an open mind.

Questions from Student

1. **Pros and cons of Waterfall model, iterative model, and component-based model**

	Pros	**Cons**
Waterfall model	-Timescales are kept -No financial surprises -Deal with issues in the design	-Potential lack of flexibility -Longer delivery time
Iterative model	-Potential defects are spotted and dealt with early -Functional prototypes are developed early in the project life cycle	-More resources may be required -The need for more intensive project management may be required
Component-based model	-Increase in reusability -Reduction of development time -Cost Reduction	-Ambiguous requirements -Choice of components

2. **What are the advantages of cloud computing among web service-based technologies?**

- The initial purchase cost and cost are low, and the portability is high.
- It is possible to use various devices as a terminal, and a consistent user environment can be implemented through the service.

- By storing user data on a reliable server, it can be safely stored.

- It can be used easily without knowledge of specialized hardware.