

CS554 - Designs for Software and Systems

Professor Ho-Jin Choi

Date: 4 January 2013

Student name / ID: _____ / _____

(You may answer either in English or in Korean.)

[Questions 1-5] Answer the following questions. (10 points each)

[1] What are the essential characters of software artifacts that make it difficult to adopt design techniques developed for other engineering disciplines such as civil or automobile engineering?

[2] In his book, Fred Brooks makes seemingly self-contradictory arguments. In one place he argues for delivering the second system because usually the first system built is barely usable. In other place, though, he argues for avoiding the second system, characterizing it as the most dangerous system one ever designs. Clarify his points.

[3] What is a paradigm shift? Explain and give examples of two or three major scientific paradigm shifts in the history of science.

[4] Michael Jackson argues that the root cause of most complexity in software development is premature composition, and that the key to protecting yourself against this complexity is postponing composition until you have mastered the composition. State your position for or against this argument, and explain the reasons why.

[5] CelsiusTech – the ship-building company – is known to be the first pioneer of the product line approach. What motivated CelsiusTech to develop a product line approach?

[Questions 6-7] As for methods dealing with software quality attributes, answer the following questions. (10 points each)

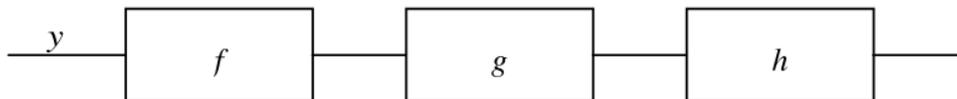
[6] Explain the reasons why a structured tradeoff analysis method such as ATAM is useful.

[7] The ATAM process helps developers discover tradeoffs and sensitivity points. In the context of ATAM, what are tradeoffs and sensitivity points? Give definitions.

[Questions 8-10] As for software architectural styles, answer the following questions. (10 points each)

[8] Discuss the importance of software architecture in the process of software development and also in the process of software maintenance.

[9] How do you reason about the functionality of pure pipe-and-filter systems? For example, if filter f_1 with input stream x delivers the output stream $f_1(x)$, what does the following combination in the figure compute? Explain why. (In the figure, y is the input.)



[10] Suppose someone told you they were using implicit invocation, except that by convention there is always exactly one recipient of each event. And further, the announcer of the event expects a return event from the receiver carrying a result. Is this really implicit invocation, or not? Explain the reasons of your answer.

(The End)