

CptS 322 Software Engineering Principles I

Homework 2 – example solution

1. (10 points) Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:

- A system to control anti-lock braking in a car

This is a safety-critical system so requires a lot of up-front analysis before implementation. It certainly needs a plan-driven approach to development with the requirements carefully analyzed. A waterfall model is therefore the most appropriate approach to use, perhaps with formal transformations between the different development stages.

- A virtual reality system to support software maintenance

This is a system where the requirements will change and there will be an extensive user interface components. Incremental development with, perhaps, some UI prototyping is the most appropriate model. An agile process may be used.

- A university accounting system that replaces an existing system

This is a system whose requirements are fairly well-known and which will be used in an environment in conjunction with lots of other systems such as a research grant management system. Therefore, a reuse-based approach is likely to be appropriate for this.

- An interactive travel planning system that helps users plan journeys with the lowest environmental impact.

System with a complex user interface but which must be stable and reliable. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.

2. (10 points) Provide three examples of software projects that would be amenable to the waterfall model. Be specific.

Generally, the waterfall model is appropriate for projects with the following characteristics: (1) the problem is well understood (requirements are well-defined); (2) the delivery date is realistic; (3) it's unlikely that major changes in requirements will be requested as the project proceeds.

Specific examples might be: (1) a well understood modification to an existing program; (2) a straightforward implementation of a numerical calculation or business rule, even if it's complex; (3) a constrained enhancement to an existing program.

3. (10 points) Provide three examples of software projects that would be amenable to the prototyping model. Be specific.

Software applications that are relatively easy to prototype almost always involve human-machine interaction and/or heavy computer graphics. Other applications that are sometimes amenable to prototyping are certain classes of mathematical algorithms, subset of command driven systems and other applications where results can be easily examined without real-time interaction.

Specific examples include (1) a shooting game application, (2) a garment design CAD (computer-aided design) system, and (3) a scientific calculator. (The Automaton tool can be an example too.)

On the other hand, applications that are difficult to prototype include control and process control functions, many classes of real-time applications and embedded software.

4. (10 points) Spiral model and incremental model are two software process models. Explain how these two models are different and related.

Relation: both models allow for iterations, feedback from users, and refinement of analysis and design; both models can adapt to changes occurred during software development.

Differences: These two models can be differentiated from three aspects.

1. In terms of nature and process flow, spiral model is evolutionary while incremental model is iterative.
2. In terms of product delivery, incremental model ends up with a functional product at the end of each iteration, while spiral model may not end up with a functional product at the end of a cycle
3. In terms of focus, spiral model focuses on risk analysis/reduction, while incremental model focuses on incremental delivery of working products.