Software Engineering Homework 3

Jude Jang

All software development should follow some sort of process model. Following a process model can improve consistency, understanding, and ease of maintenance. Three common software process models are the waterfall model, spiral model, and agile.

The waterfall model was one of the first process development models, is suited towards well understood problems, and is simple and easy to explain to customers. The waterfall model’s steps are: requirements analysis, system design, program design, coding, unit & integration testing, system testing, acceptance testing, and operation & maintenance.

The spiral model was initially suggested in 1988 by Barry Boehm. The spiral model minimizes and control risks by combing development activities with risk management. This model is presented as a spiral where each iteration is represented as a circuit around the four major steps. The four major steps of the spiral model are: planning, determining goals, evaluating alternatives and risks, and finally developing and testing.

Agile methods emphasize on being flexible while producing software quickly. Key values of agile are to value individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

Each of these software process models can be very useful while building a software. Although each of these software processing models have their unique advantages, they also have respective disadvantages.

One major disadvantage of the waterfall model is that it does not account for any changes to requirements. Another disadvantage is that is treats software development not as a creative process, but a manufacturing process. There are also no iterations in the waterfall method and there is usually a long wait before the final product.

The spiral model also has certain disadvantages. One disadvantage of the spiral model is that it works well with internal software development, but is less effective with contracted software. Another disadvantage is that it relies on risk-assessment expertise of the software developers. In the case that a team of inexperienced developers incorrectly assess a high-risk in the spiral model, it could lead to disastrous results. The spiral model also typically requires further elaboration to software developers who are inexperienced with the spiral model.

The agile software process is the most flexible of the three process models, but is also the most unpredictable. There is also a greater demand on clients and developers due to requiring constant communication between the two which results in more time being spent. Agile also lacks detailed documentation which can lead to difficulties if new developers join.

Although each of these software processing models are suited towards different specifications from each other, it is still possible to use them in tandem. In a large and complex system, there can be many different components and subsystems that may use the different software process models effectively.

For example, when, a component or subsystem requires customer interaction, it might be useful to use agile methods to effectively communicate with the client and use the waterfall method to illustrate progress. In a team of experienced software developers for an internal system that needs to be completed quickly, it might be effective to combine the spiral model and agile methods. Since the team of software developers are experienced, the disadvantage of the spiral relying on risk-assessment expertise is not an issue. If the internal system needs to be completed in a timely manner, agile methods would be effective due to it favoring producing working software quickly at the cost of comprehensive documentation, but if the team consists of experienced software developers who all understand the internal system, lack of documentation may not be a problem.

[1] B.Cheng “Software Development Processes”

[2] W. Royce “Managing the Development of Large Software Systems”

[3] B. Boehm “A Spiral Model of Software Development and Enhancement”

[4] M. Aoyama “Agile Software Process and Its Experience”