2: Discuss a number of software process models and when they might be used, advantages, disadvantages …

As we have learnt, in general, there are five types of software process models, which are “Build and fix”, “Waterfall’, “Incremental”, “Spiral” and “Rapid prototype”. Every of them has its own merits and demerits.

First come to build and fix model. Briefly, when a product is being programming without enough instructions and design, programmers always need to fix it several times to meet the needs of the clients, which means it fits for the small products. Its advantage is that it has limited complex and often cost a little, however, it has a number of disadvantages. For example, as the product needs so many times fixes, it’s unpredictable when it can be finished and delivered, and even though it has been completed, the quality is low sometimes. Moreover, as no document and enough specification will be produced, maintenance can be extremely difficult.

Secondly, the waterfall model, fits for the product whose requirements are clear and easy to understand. It is a great model that every stage will be verified before proceeding, and faults detected can cause rolls-back to certain stage, maintenances rolls-back process appropriately. While its merits are obvious, its demerits are also clear. If discovering faults late, they may cost a lot to fix them and there is no requirement to illustrate the client anything but documents until the end.

Next is the incremental model. It divides the whole project into several blocks, and when we finish one of them, we can show it to clients to ask for their advice, so it can reduce the cost of the changes raised from clients and it’s easy to get feedback from them anytime, which can help to produce a product that is truly useful. It has some disadvantages, for instance, the process is invisible when proceeding, and the structure of system may degrade when adding new increments.

Then the Spiral model. It is like the mixture of the Waterfall model and Incremental model, so it includes all of their merits. While it still needs long time to complete the task, it is easy to be controlled so it fits for the product that may meet risks during programming.

Last comes to the rapid prototype model. It need the programmers to establish a brief model at first to understand the details of the requirements and it can help them to reach a consensus with clients on the requirements at the beginning, however, it means it needs clients to involve a lot and sometimes the initial prototypes cannot be used.

The words above describe the five types of lifecycle models briefly, and we come to conclusion that if we fully understood the use of them, we could finish the tasks whatever we meet perfectly.