

Incremental process model and extreme programming critical thinking examples

[Business](#), [Customers](#)



Incremental model can be said to be an enhancement of the waterfall model. There are several waterfall approach done repeatedly in the development of the software. The software development cycles are split down into smaller pieces that are easily managed. Each of the iterations or the software version released go through the whole waterfall steps; requirements, design, implementation and testing phase (Pressman 52).

Each iteration produces a working version of the software which is then tested. The software version is tested by the users and the developers to capture the misleading or misunderstood requirements. The following software development or iterations use the previous version. Below is a figure for incremental model.

Fig. 2 Incremental Life cycle model

Incremental model creates a working version very early and quickly in the life cycle. It is also more flexible and less expensive in changing the scope and requirements.

Developing a mobile software application requires that the model offers a chance to improve on the current version. Mobile applications are very small projects with small number of teams. This the application requires earlier versions so that they can be tested by the users. The feedback from the users can be used to correct bugs and improve on the user requirements. Mobile based platform requires a model such as incremental model.

Incremental model is suitable because it gives an earlier version in the development cycle of the software. Incremental model is used in software development where user requirements cannot be defined in the initial stages of the project. It gives software developers an opportunity to add new

requirements during the next development phase. Mobile application requirements are sometimes complex where the user would like to add new features in the application once they test it. Hence the incremental model is suitable (Pressman 22).

eXtreme Programming (XP)

This is a software engineering model that was developed by Kent Beck in 1996. There are two practices in this methodology that are regarded as controversial; pair programming and test-driven development. This model is aimed at reducing risks in the software development and management process.

XP methodology is made of releases just like iterations of the incremental model where the first release gives a working version. Further releases just build up on the initial release where additional features are made and errors from the previous version are fixed. This methodology allows the stakeholders to add requirements because the requirements are not fixed. Before the commencement of the project customer requirements are grouped into stories. The customers select the story with highest priority during the start of a next iteration.

Fig. 2 XP Programming

The customers get a working version early and developers are required to meet frequently. Frequent meeting ensures that the final project is correct and what was intended. Since the customers are also involved in the process, developers pass on risks to them.

This model is applicable for small and medium size projects. It is also most suitable in situations where requirements are not very clear and there are

chances of additional requirements later in the development of the software. Some of the advantages of this methodology are that an earlier working version is produced after first release. This reduces the chances of final product failing to meet the user requirements because they are improved based on user feedback.

Works Cited

Pressman, Roger. Software engineering: A practitioner's approach. NY: McGrawHill. 2010