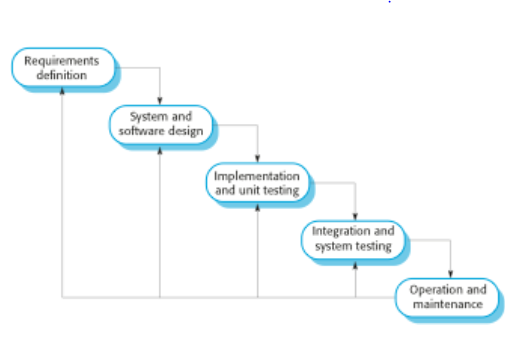
**"Discuss at least two reasons as to why the classical waterfall model can be considered impractical and cannot be used in real projects ?"**

According to Sommerville (2011), the development of most large software systems requires a process that includes elements from several different software process models. One of these models is the traditional waterfall model (described below). This particular model is plan-driven and usually not sufficient in practice. Instead it is often augmented or replaced by incremental development and/or reuse-oriented software engineering (Sommerville, 2011).

**The Waterfall Model**

The traditional waterfall model has several sequential development stages including, system requirement analysis and definition, the design of the system and software, implementation and testing of individual units, larger system integration and testing, and, finally, operation and maintenance. The diagram below shows the cascade from one stage to the next in the waterfall development model.



(excerpted from Sommerville, 2011).

The waterfall model can be useful when there is a need for organization in planning. For example, if very large systems are being built across different locations, the waterfall model can help to keep the project focused and moving according to plan (Sommerville, 2011). However, there are problems with the waterfall method, the most significant of which being that it doesn’t allow for changes past the initial planning stage. In fact, each phase needs to be done according to the original plan sequentialy, with each phase completed before the next phase begins.

**Changing Business Needs**

If all contingencies and requirements could be thought through in advance during the planning phase, then the waterfall model could work. But, in practice, customer requirements may change midstream during the development project, making this inflexible model ineffective. According to Sommerville (2011), there are not many business systems that are stable enough over the development phase for the waterfall model to be effective.

**Advances in Technology**

Not only do changing business requirements make the waterfall model ineffective, but there is also the matter of technological advances also affecting software development. For example new, useful technologies may need to be incorporated into a system or perhaps upgrades to current applications and platforms may need to be accommodated (Somerville, 2011).

**The Cost of Changes**

Regardless of the underlying cause of changing requirements, there is a monetary cost if the original plan needs to be altered or re-worked to accommodate new requirements (Sommerville, 2011). For this reason, many system development projects do not follow the waterfall model. Instead, other, more flexible development models are recommended to mitigate the cost of changing system requirements.

Reference

Sommerville, I. (2011) "Chapter 2 - Software Processes: Lecture 1 -2.” Software Engineering, Ed. 9. Retrieved from<http://my.uopeople.edu/mod/resource/view.php?id=57330>