Week 1 Homework Problems

Exercise 1.2:

The main distinction is that the specification belongs to the product developer in the development of generic software products. The consumer owns and controls the specification while creating a custom product. The implications of this are significant because, when the customer owns the specification, changes must be negotiated between the customer and the developer and may have contractual implications. The developer can quickly decide to change the specification in response to some external change (such as a competing product).

Exercise 1.3:

Acceptability, dependability and security, efficiency, and maintainability are the four most crucial qualities of professional software. Performance, Scalability, Reusability, Testability, Reliability, and Availability of the software are more instances of quality qualities.

Exercise 1.5:

Expenses and rate of change. Some systems are very expensive to modify, while others need to be altered regularly to meet shifting needs. In order to verify that the requirements are consistent and that the system meets its specifications, systems that are highly expensive to update require considerable upfront study and validation.

For systems that change very quickly, this is not a cost-effective strategy. the most crucial "non-functional" needs. Different systems prioritize non-functional requirements differently. For instance, a real-time control system in an aircraft prioritizes safety; an interactive game prioritizes responsiveness and usefulness. The methods employed to ensure safety are not necessary for interactive gaming, and safety-critical control systems do not require the elaborate UI designs required for games.

The delivery schedule and software lifecycle. While some software systems may last a few years, others can last for tens of years. If some systems are to be beneficial, they must be delivered rapidly. When creating long-lifetime systems, such as design modeling, it is improper to apply the same development methods that were used to create short-lifetime, rapid delivery systems.

Exercise 2.1:

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.

waterfall

A virtual reality system to support software maintenance.

Incremental development

A university accounting system that replaces an existing system

Reuse-based approach can be the best choice, because the requirements are well-known and will be used in conjunction with lots of other systems.

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

environmental impact

incremental

Exercise 2.3:

The purpose of a requirements specification is to identify the system and user requirements for a software system. Reusable components are used to modify the requirements specification, which is known as requirements modification. Since we typically don't have complete control over the reusable components, some form of modification is required.

Exercise 2.4:

This is so because the goal of requirement engineering is to create a contract between the client and the developer. While the consumer is familiar with business jargon, a developer needs very specific instructions before they can begin. As a result, it is vital to have both a high-level technical definition and a thorough user specification.