Usability Requirements

Usability is the customer's or user's point of view of software and failing to provide them with what they need can be very costly. If software is considered unusable, then it may be abandoned altogether by users, representing a sunk cost to software vendors. Oftentimes, project failure can be attributed significantly to poor implementation of usability requirements, among other issues. Studies have indicated that poor user requirements represent 62.5% of the causes of project failure (Pew, Mavor, 2007, p. 191). With such a negative impact on finished software products, usability requirements should constitute a major part of projects, but despite being well-documented, they are not usually implemented very well (Pew, Mavor, 2007, p. 197).

To truly implement usability requirements, it is imperative to tie them to system requirements. Doing so emphasizes the importance of the human-computer relationship and interaction and helps system developers to maintain focus on building software that meets a level of quality in terms of user needs. Usability requirements run the gamut of certain quality characteristics, including ease of use, understandability, learnability, operability, attractiveness, effectiveness, efficiency, and satisfaction (Pew, Mavor, 2007, p. 192). The last three usability requirements (effectiveness, efficiency, satisfaction) are each measures of how well a user can work with the software from simply performing the job to the speed with which the job is completed.

Usability requirements are metrics for determining how successful software will be in relating humans to computers, and there are methods that can be used to help determine the requirements. One method is to use an existing system as a model and to create a baseline that contains metrics that indicate success in meeting business needs (Pew, Mavor, 2007, p. 193). The usability of a working system can help system developers to have perspective when creating a new system with emphasis on usability. Another way to build usability into a new system is to explicitly establish the requirements, which can be elicited through prototypes, data modeling, and understanding the environment in which users will be putting the software to use (Pew, Mavor, 2007, p. 193). Speaking with users can aid system developers to understand their needs rather than just speculating on what users will want the system to do. These discussions can happen informally or they can be more formal. Formal methods of gathering usability requirements from users include questionnaires, interviews, and observations. Additionally, usability requirements can be tested to determine if they are feasible, as well as to establish whether or not they have been achieved (Pew, Mavor, 2007, p. 193). It is not enough to simply incorporate usability requirements into a project requirements document. They must serve to validate the success of software or to identify areas for improvements in future releases.

As with any human endeavor, determining software usability requirements is a team effort, which requires significant communication to ensure cohesion during development and success of the software. According to the Common Industry Specification for Usability Requirements, communication should be supported in terms of context of use, usability measures, and a test method (Pew, Mavor, 2007, p. 194). It is especially important that communication flows among the development team, as they are at the heart of constructing software. They must understand usability requirements before design begins so that risk of product failure will be reduced; development effort will be less; and costs can be controlled (Pew, Mavor, 2007, p. 195). The importance of communication can be extended to that which happens between users, or customers, and system developers, or the organization.

With usability requirements providing an avenue for delivering software that people can effectively use, it is essential for system developers to initiate user-focused activities. As mentioned earlier, quality is an important ingredient and must be defined for the software to be successful. User-centered activities also include creating user personas to help identify stakeholders, analyzing scenarios that occur in daily system activity, examining context of use, and recognizing the interaction that takes place between the users and the system (Pew, Mavor, 2007, p. 196). Focusing software development on the user helps to determine all the necessary functionality that must be built into the system. Several people can be recognized as stakeholders in a software development project, especially users, which makes usability requirements of utmost importance when creating software. Usability requirements can act as an "advance warning" to prevent serious gaps that can result from poor planning during a software development project (Pew, Mavor, 2007, p. 197). Such warnings can save an organization both time and money and prevent budgetary overruns. Usability requirements provide a contract between users and system developers that can limit scope creep in a project. Overall, usability requirements offer strategic advantages without taking much away from the software development process.

References

Pew, R., Mavor, A. (2007). Human-System Integration in the System Development Process: A New Look. Washington, D.C.: The National Academies Press.