SciForschen
Open HUB for Scientific Research

International Journal of Multimedia

Editorial Volume: 1.1 Open Access

Why is User Experience Design treated as an After Thought?

Bruce R Maxim*

Computer and Information Science, University of Michigan-Dearborn, USA

***Corresponding author:** Bruce R Maxim, Computer and Information Science, University of Michigan-Dearborn, USA, **E-mail:** bmaxim@umich.edu

Received date: 02 Jan 2016; **Accepted date:** 06 Jan 2016; **Published date:** 10 Jan 2016.

Citation: Maxim BR (2016) Why is User Experience Design treated as an After Thought? Int J Multimedia 1(1): doi: http://dx.doi.org/10.16966/ijm.e101

Much of the motivation for improving software engineering practice comes from trying to avoid the costs of unnecessary work caused by delivering a product that does not meet the customer's needs or expectations. Yet time and time again we hear complaints about software products that look great and are very hard to use. Trying to ignore this problem by adding features to an existing product will not stop users from switching to alternative products that are easier to use. Software usability is not really a matter of personal opinion or simply following a check list of things to include. Usability means a software product is easy to learn and easy to use. So how can project managers be encouraged to pay attention to spending more user experience design and how can software developers be encouraged to focus on meeting customer expectation sooner rather than later?

Many times it is easy to predict that a product will deliver a poor user experience. When programmers rely on their intuition or their knowledge of how an application behaves, they are less likely to create user experience that meets the customer's expectations. Allowing marketing pressures to add rarely used product features make it hard to develop products that users find easy to use. Not including sufficient time and budget for usability testing makes it hard to gauge whether the completed product will be accepted by the end users.

Managers will only be convinced to change their practices if they see reduced costs and shorter development times. If customers are not happy with the delivered product they are likely to request changes which require rework before acceptance takes place. Late changes to products are always more costly to implement than changes requested early in the design process. Algorithm errors and performance problems are likely to be caught and repaired during the testing process, without requiring design changes and significant rework. Customers may not be able to detect algorithm errors until end-user complaints start arriving after a product is in use. Problems with the user experience are likely to be immediately obvious to the customers, even those with technical backgrounds.

Games provide a good example of multimedia software products that people often have usability misconceptions about. Some people think usability is not important for games, because games they believe games are intended to be challenging. However, poor user experience design often makes even the simplest player task into a trial and error hunt for user controls which destroys the immersive character of the game environment valued by customers. Some managers may think that when sales are good and there are few complaints that the user experience must be OK. The reality may be that the software product may be enjoying a temporary monopoly while competitors are building easier to use products. Many software developers believe that a shift to user-centered design practices can help improve the quality of the user experience.

Agile software developers believe the key to user-centered design is to involve a customer's user representative early in the development process as a full team member. The user representative helps frames the product requirements as user stories that define the customer's vision for using the final software product. User stories are written in the customer's own words without regard to implementation details. This makes it easy for the customer to verify the validity of the incremental prototypes created as the product evolves and reduces the likelihood of unnecessary rework being required. Using paper prototypes instead of coding a throwaway prototype can also be helpful.

Unless the developers are not confident in their abilities to manage the technologies needed to implement the proposed software product, there is no need to wait to engage with customers during the design process. In fact modern software engineering practices include the frequent delivery of incremental prototypes. The use of agile processes to manage customer changes can reduce development times and improve software quality. The principles of good user experience design include similar practices that can assist in reducing both delivery times and the need for unnecessary rework.

The purpose of a user interface is to provide users with enough information with enough information that they select the next actions needed to help them achieve their goals. This means that a good user interface provides representation of the system state that matches the user's conceptual model of the system and its behavior. Action alternatives should be easily identified in the user interface by a typical user. A good user interface design should include facilities that provide the use with continuous feedback as they progress toward their goals.

Where do user interfaces typically fail? Users may have goals or expectation that cannot be met using the interface features. The users may fail to identify the relevant interface objects needed to complete their task due to poor labeling or bad layout. Users may not know how to create or execute a plan needed to meet their goals using the interface objects. Users may receive poor feedback from their actions. Most of these problems can be avoided by careful analysis of the users' goals.

Software project managers often do not budget either enough time or money to allow for usability testing early in the product development timeline. The first step in developing any software product should be to perform analysis of the user's task as outlined in the user stories. It is important for developers to understand what the user really needs to do to accomplish his or her goals. It is important to keep in mind that the user's goals ae not likely to include pushing buttons or typing commands. If the product has an entertainment goal, then the task becomes to determine which features need to be present to make using the product fun and which do not. Usability testing needs to take place early in the development process to identify any difficulties that are not part of the user's entertainment goals. Usability design heuristics can be used to remove problem features once they are defined.

It is important set usability targets (e.g. learn program basics in 15 minutes) early in the design process. Quantifiable target measures are more persuasive then personal opinion. Measurement should begin as soon as a minimum viable product prototype is ready for testing. This prototype should be assessed for ease of learning, speed of task completion, and user error rates as part of the usability assessment process. When usability problems are identified they should be corrected. The usability assessment should be repeated until the targets are met.

It is interesting that many of the problems that results in the creation of poor user experience design can be addressed by modern software engineering practices. Developers need to involve customers early in the design process and pay attention to their stated needs and preferences. The early creation of product prototypes and engaging customers in the evaluation of the prototypes can help reduce the requests for last minute changes when it is time to deliver the final product. Software engineers have learned that every team member must have a quality mindset at all times to create high quality products. The same is true for designing good user experiences.