Towards Usability Maintenance

Parmit K. Chilana
The Information School
University of Washington
pchilana@uw.edu

Abstract

Current usability practices focus on improving the design of software upfront through prototyping, user testing, and other activities. However, a number of problems emerge after software is deployed in the user's environment. In fact, software maintenance and software support activities incur high costs in time and resources in the post-deployment phase. At the same time, it appears that the role of usability diminishes after deployment. We propose that within the field of usability, there needs to be an orientation towards usability maintenance. Unlike the concept of software maintenance that largely focuses on the correctness and performance of the software artifact, the goal of usability maintenance is to support and improve user experience after deployment.

Keywords

Usability maintenance, usability practices, software development lifecycle

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Motivation

The software engineering community has long recognized that software is constantly evolving: in many projects, 70% or more of the operating costs and activities are spent on maintaining software after deployment [3]. The IEEE formally defines *software maintenance* as, "the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment."[1]

In addition to bug fixing and other maintenance activities, a major focus of the post-deployment phase is also on providing end-user software support [6]. Users often seek help for resolving a range of issues, such as recovering from errors, understanding functionality, learning new ways to customize settings, among others. Companies often have to create specialized support structures and tools to help users resolve such software issues [1,6].

To prevent post-deployment issues from occurring in the first place, usability practices have had a significant impact on changing upfront design processes [4,5]. Still, software maintenance and software support activities continue to incur high costs after deployment, suggesting that not all software issues can be anticipated or prevented through upfront design alone. Even though a key premise of the usability engineering lifecycle is a commitment to usability principles throughout all stages of development, in practice we find that the role of usability appears to diminish after a product has been deployed [2].

In light of current usability practices and demands on software development, we propose an orientation towards what might be called *usability maintenance*. The concept of usability maintenance parallels software maintenance, but emphasizes maintaining the overall post-deployment user experience rather than correcting code-level defects or performance issues. In doing so, a number of questions are raised for the HCI community about the role of usability and whether we need to reconsider usability's relationship with software maintenance and software support activities.

Current state of post-deployment usability

We recently carried out a survey [2] to better understand the state of post-deployment usability activities. We received 333 responses from usability professionals working in large and small corporations representing a variety of industries in North America and abroad. We asked respondents questions about direct involvement in the different phases of the user-centered design process and asked respondents to specify their particular pre-deployment and post-deployment activities. We also asked respondents about their interactions with software support and software development teams in the post-deployment phase. (More details on our survey instrument and findings can be found in [2]).

Usability professionals and development phases

The majority of our survey respondents (87.7%) said that they were regularly involved in the design phase, but only 50.9% of respondents reported any direct involvement in the post-deployment phase. Some of the common post-deployment activities (indicated by about a third of the respondents) included analysis of usage log data, satisfaction surveys, and monitoring of product-specific discussion forums. The results indicated that after deployment, most of the respondents (70.3%) started working on another product and/or the next version of the current product. There appeared to be little direct involvement from the usability professionals once a product had been deployed in the user's setting.

[1] *http://standards.ieee.org/findstds/standard/1219-1998.html*

Usability professionals and software support

The before/after findings of usability activities showed that only 34.8% of respondents appeared to leverage customer support data in the post-deployment phase, even less than the pre-deployment phase. Furthermore, over 50% of respondents never or rarely talked to support specialists. Since support specialists are at the front lines of directly interacting with end-users and helping them troubleshoot or learn about product features, it is possible that usability practitioners are missing several opportunities for learning about user experience from the field. As today's systems are becoming more complex and enabling idiosyncratic customizations, it is likely that the role of support will continue to be integral in supporting and evolving user experience. Thus, our survey findings highlight the need for the usability community to consider ways in which customer support data can be leveraged to guide iterative design tasks.

Usability professionals and software maintenance

Software developers spend most of their time triaging and fixing bugs and maintaining the software [3]. Only 18.9% of usability professionals said that they were directly involved in helping triage bugs, and about 30.0% of respondents never or rarely talked to software developers after a product had been deployed. Since a number of bugs that arise in the post-deployment phase are potential design and usability bugs, there is opportunity for exploring how usability professionals can play a more influential role in the bug triaging process.

The need for *usability maintenance*

Our survey results indicate that, as a whole, the role of usability in current practice appears to diminish after a product has been deployed. This finding is somewhat troubling given that 1) most of the high-cost activities in the software engineering lifecycle occur after deployment [3], and 2) iteration and user feedback are advocated as core components of all phases in the usability engineering lifecycle [4,5]. Given the increased uptake of usability in industry, it is not a surprise that the value of getting upfront design into organizations has paid off. However, despite sincere intentions in tackling potential usability problems upfront, current industry practices suggest that a number of issues emerge in the post-deployment phase. To what extent, should post-deployment usability (or lack thereof) be a concern for HCI practitioners and researchers? What can we do differently to play a more influential role in all phases of development?

We propose that within the field of usability, there needs to be an orientation towards *usability maintenance*. Unlike the concept of software maintenance where the focus is on the correctness and performance of the software artifact, usability maintenance should focus on the post-deployment user experience. Since software is constantly evolving, usability should not be considered as a state that can be achieved with upfront design activities. Rather, it should be recognized that a focus on the current version of the deployed product is perhaps as equally important as upfront design.

Although there is need for more empirical studies to further develop the idea of usability maintenance, as a start, we propose looking towards software support and software maintenance. For example, to truly know how well a product succeeds in adhering to usability principles of learnability, efficiency, memorability, recovery from errors, or satisfaction, the usage of the current version of the product needs to be monitored. One way that usability professionals could learn about users and usage patterns more closely would be by collaborating with support specialists and monitoring support requests. Usability professionals could also be involved in monitoring bug reports and influencing triaging decisions that involve issues related to the product's user experience.

We realize that organizational cultures and product delivery schedules may not be amenable to an orientation towards usability maintenance. For example, many projects still face resistance in incorporating any kind of usability work. But, given the potential benefit that usability maintenance would bring in streamlining existing software support and maintenance activities, we believe that there is potential in further exploring this concept. In our future work, we will be investigating other dimensions of usability maintenance by gathering data from the field and inventing opportunities for usability professionals to be directly involved in the post-deployment phase.

References

1. Chilana, P.K., Grossman, T., and Fitzmaurice, G. Modern software product support processes and the usage of multimedia formats. *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 3093-3102.
2. Chilana, P.K., Ko, A.J., Wobbrock, J.O., Grossman, T., and Fitzmaurice, G. Post-deployment usability: a survey of current practices. *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 2243-2246.
3. Lehman, M.M. and Belady, L.A., eds. *Program evolution: processes of software change*. Academic Press Professional, Inc., 1985.
4. Mayhew, D.J. *The usability engineering lifecycle: a practitioner's handbook for user interface design*. Morgan Kaufmann, 1999.
5. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, 1993.
6. Tourniaire, F. and Farrell, R. *The art of software support*. Prentice-Hall, 1997.