



The role of software updates in information systems continuance – An experimental study from a user perspective



Marvin Fleischmann^{a,*}, Miglena Amirpur^b, Tillmann Grupp^b, Alexander Benlian^b, Thomas Hess^a

^a University of Munich (LMU Munich), Munich, Germany

^b Darmstadt University of Technology (TU Darmstadt), Darmstadt, Germany

ARTICLE INFO

Article history:

Received 22 October 2014

Received in revised form 18 November 2015

Accepted 30 December 2015

Available online 16 January 2016

Keywords:

Software updates

IT features

IS continuance

IS post-adoption

Expectation–confirmation theory

ABSTRACT

Although software updates are a ubiquitous phenomenon in professional and private IT usage, they have to date received little attention in the IS post-adoption literature. Drawing on expectation–confirmation theory and the IS continuance literature, we investigate whether, when and how software updates affect users' continuance intentions (CI). Based on a controlled laboratory experiment, we find a positive effect of feature updates on users' CI. According to this effect, software vendors can increase their users' CI by delivering features through updates after a software has been released and is already used by customers. We also find that users prefer frequent feature updates over less frequent update packages that bundle several features in one update. However, the positive effect from updates occurs only with functional feature updates and not with technical non-feature updates, disclosing update frequency and update type as crucial moderators to this effect. Furthermore, we unveil that this beneficial effect of feature updates operates through positive disconfirmation of expectations, resulting in increased perceived usefulness and satisfaction. Implications for research and practice as well as directions for future research are discussed.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, software vendors have increasingly leveraged software updates as a measure to modify and enhance their software products, while they are already being used by their customers. This phenomenon is particularly prevalent in the area of mobile applications and operating systems, but updates have also been used long before in the desktop space. Apple iPhone users, for instance, regularly receive updates for their apps. On the desktop, web browsers such as Google Chrome and Mozilla Firefox continuously receive updates, which extend their functionalities. Other examples include Microsoft Windows, the Adobe Reader and Sun's Java platform which all regularly receive updates to close security gaps or fix minor flaws.

This ubiquitous use of updates by software vendors in practice reflects in a large body of research on the technical design of software, its maintenance and management. Research on software engineering [59], including software product lines [9], software release planning [61] and software evolution and maintenance [39] explores how and when software functionality should be developed and delivered in order to maintain the technical integrity of the software and optimize the vendor's production process. While this stream of research does

account for customer needs, its primary focus lies on the supply side, exploring technical design aspects of software. There is as yet, however, little understanding of the user's perspective on software updates—the demand side. In particular, the behavioral dimension, i.e., how updates are perceived by users is still an under-explored area that has so far received only minimal research attention [25,56]. Investigating the effect of software updates on users' beliefs, attitudes and behaviors regarding an information system (IS), however, might be beneficial for software vendors and of particular interest in the post-adoption context, because users' continuance decisions (i.e., customer loyalty) are strongly influenced by their experiences made during actual IS use [6]. For software vendors, shedding light on the role of software updates for the IS continuance decision can thus result in a better understanding of how to deliver updates to users in order to achieve desirable performance outcomes such as higher user loyalty and sustained revenue streams.

From a research perspective, a better understanding of software updates from a user's perspective has the potential to increase the explanatory and predictive power of existing post-adoption theory. In conjunction with pre-adoption and adoption, post-adoption research constitutes IS usage, one of the most mature fields in IS [27]. However, compared to research on pre-adoption and adoption decisions, post-adoption studies still remain sparse. Many scholars have thus called for studies that explicitly focus on post-adoptive phenomena (e.g., [2]). Furthermore, researchers studying IS post-adoption phenomena often tend to conceptualize information systems as a monolithic and coarse-grained black box, rather than as collection of specific and

* Corresponding author at: Ludwig-Maximilians-Universität München (LMU), Munich School of Management, Institute for Information Systems and New Media, Ludwigstrasse 28, 80539 Munich, Germany. Tel.: +49 89 2180 6394; fax: +49 89 2180 99 6394.

E-mail address: fleischmann@bwl.lmu.de (M. Fleischmann).

finer-grained features that are dynamic and alterable over time. However, understanding the granularity of software and its changes through software updates would help explain how users' beliefs, attitudes, and behaviors fluctuate over time as a result of the dynamic nature of information systems. In addition, the focus on changes in beliefs, attitudes and behaviors, emanating from the IT artifact itself rather than from other IT-unrelated environmental stimuli, is a response to several calls for research from IS scholars who criticize the negligence of the IT artifact's role in IS research [3,21,48]. From a theoretical perspective, it is not only important to explore *whether* software updates have an effect on users' beliefs, attitudes and behaviors toward the software and their continuance intentions (CI) in particular. It is equally important to examine *when* and *how* these effects might occur, thus providing a profound theoretical explanation as well as the possibility to predict user reactions toward software updates. Against this backdrop, our objective is to study software updates as a measure by which a vendor can provide maintenance for or extend the functionality of its software over time, while it is already being used by customers. To the best of our knowledge, software updates and their effects on users' IS continuance decisions are thus far still underexplored in the IS post-adoption context. We therefore seek to address this research gap by examining the questions of *whether, when and how software updates influence users' IS continuance intentions*.

In line with the mentioned research gaps, we contribute to prior research in three important ways. First, our overarching contribution is to advance the predominant view of information systems in post-adoption literature from a mostly monolithic and static to a finer-grained and more dynamic perspective by showing how a functionally malleable information system might influence users' beliefs, attitudes and behaviors over time. As such, we also accentuate the changing nature of the IT artifact for users' CI and thus explicitly consider the software product lifecycle in our theorizing. Second, we identify substantially different user reactions to different update types and modes of delivery. While feature updates increase users' continuance intentions, technical non-feature updates (e.g. bug fixes) have no effect on the intention to continue using the software. Moreover, we find that users prefer features to be delivered in individual updates over a delivery of features in larger but less frequent update packages comprising several features. Update type and frequency thus seem to moderate the effect of software updates on users' continuance intentions. Third, we not only investigate the direct effect of software updates on CI; we also open up the theoretical black box of *how* software updates influence IS continuance intention by highlighting the complementary roles of cognition and affect. From a practitioner's perspective, our study offers implications for software vendors on how to deliver software updates in order to increase their customers' loyalty (i.e., CI). We not only provide guidelines on which actions to take, but also on which measures to avoid in order to benefit from the positive effect of feature updates on users' CI.

2. Theoretical foundations

2.1. Software updates

Consistent with previous research (e.g., [14]), we consider software updates to be self-contained modules of software that are provided to the user for free in order to modify or extend a software after it has been rolled out and is already in use. Software updates are thus not discrete and stand-alone programs but rather integrate into the base software once they are applied to it. In practice, software updates are applied to different types of software, such as system software (e.g., operating systems, drivers) or application software (e.g., office suites) and on different platforms (e.g., desktop computers, mobile devices). With varying terminology (e.g. update, upgrade, patch, bug fix, or hotfix), the concept of software updates is repeatedly addressed throughout the software engineering literature [59], such as software

release planning, software maintenance and evolution and software product lines [57,61,67].

In contrast to this rich stream of technical literature dealing with software updates from the developers' perspective, the customer perspective has received less attention [41]. Specifically users' perceptions of updates have so far been explored only sparsely. This reflects in few IS studies dealing with updates. Hong et al. [25], for example, explore user's acceptance of information systems that change through the addition of new functionality. Benlian [4], on the other hand, explores different IT feature repertoires and their impact on users' task performance, but does not consider changes in functionality through updates. Other IS studies that found updates to influence usage behaviors, have often pushed them to the sidelines, treating them as control variables for investigating other phenomena (e.g., [10]). Existing IS research has, however, not explored the specific impact of updates on users' beliefs and attitudes regarding an IS. Specifically, the impact of different modes of delivery (e.g., frequency of updates) and different update types have so far not been explored.

Concerning the present study, we distinguish between two basic types of software updates: feature updates and non-feature updates (e.g., [40]). Feature updates change the core functionality of software to which they are applied. Functionality can be added to or removed from the original version of the software and refers to distinct, discernible features which are deliberately employed by the user in accomplishing the task for which he uses the software. The Facebook app for smartphones and tablet computers provides an example for this type of update. In a 2013 update, it received a comprehensive instant messaging feature [15]. An example from the desktop space is the 'tab sync' functionality, which was added to the browser Google Chrome in 2012 via a feature update. It enabled users to synchronize websites (tabs) across different computers and mobile devices to seamlessly continue browsing when switching devices [38]. In contrast to feature updates, technical non-feature updates do not change the core functionality of software but only correct flaws (e.g., bug fixes) or change software properties that are not directly related to its core functionality (e.g., improvements in stability, security or performance) [53]. Thus non-feature updates usually do not directly affect the user's interaction with the software and therefore the changes in the software are often not even evident to the user. Moreover, non-feature updates often fix problems that concern only a small number of users, use cases or setups but have no consequence for the majority of users. Examples for this type of update are the 'hotfixes' that Microsoft regularly distributes via its Windows Update service.

2.2. Information systems continuance

Together with research on users' pre-adoption activities and the adoption decision, post-adoption research constitutes the research field IS usage—one of the most mature fields in IS [27]. Post-adoption research explores users' beliefs, attitudes, and behaviors around the continued use of an IS [5,28]. In this regard, the term information systems continuance refers to "sustained use of an IT by individual users over the long-term after their initial acceptance" ([6], p. 2). To explore users' intentions to continue or discontinue using an IS, Bhattacharjee [5] adopts the expectation–confirmation theory (ECT) [1,35,46,47]. In Bhattacharjee's [5] model, a user's intention to continue using an IS (CI) is the core dependent variable. It is positively influenced by satisfaction (SAT) and perceived usefulness (PU). PU captures the expectations about future benefits from IS usage [6] and has a positive impact on SAT and CI [5]. While SAT represents the affective part of the continuance model, PU rather represents the cognitive one. The concept of PU has been carried over from adoption theory [12]. Perceived ease of use (PEoU), which is the second main driver of technology adoption is, however, not part of the IS continuance model. While ease of use is an important determinant of individual technology adoption decisions (i.e., at earlier stages of use), research has found ambiguous results

Download English Version:

<https://daneshyari.com/en/article/552423>

Download Persian Version:

<https://daneshyari.com/article/552423>

[Daneshyari.com](https://daneshyari.com)