

Gains and Losses in Functionality – An Experimental Investigation of the Effect of Software Updates on Users’ Continuance Intentions

Completed Research Paper

Marvin Fleischmann

Ludwig-Maximilians-Universität
München
Ludwigstr. 28, 80539 Munich,
Germany
fleischmann@bwl.lmu.de

Tillmann Grupp

Technische Universität Darmstadt
Hochschulstraße 1, 64289 Darmstadt
grupp@ise.tu-darmstadt.de

Miglena Amirpur

Technische Universität Darmstadt
Hochschulstraße 1, 64289 Darmstadt
amirpur@ise.tu-darmstadt.de

Thomas Hess

Ludwig-Maximilians-Universität
München
Ludwigstr. 28, 80539 Munich,
Germany
thess@bwl.lmu.de

Alexander Benlian

Technische Universität Darmstadt
Hochschulstraße 1, 64289 Darmstadt
benlian@ise.tu-darmstadt.de

Abstract

Although software updates are ubiquitous in professional and private IS usage, their impact on user behaviors has received little attention in post-adoption research. Based on expectation-confirmation-theory and the IS continuance model, we investigate the effects of gaining and losing features through updates on expert and novice users’ continuance intentions (CI). In a vignette based experiment, we find that updates which add features to software after its release increase novices’ CI above and beyond a level generated by a monolithic software package that contains the entire feature set from the beginning. With diminished CI, experts show a contrary reaction to the same update. Losing features through an update, on the other hand, severely diminishes CI for experts and novices alike. Mediation analysis reveals positive disconfirmation of previous expectations as psychological mechanism behind novices’ counter-intuitive and somewhat non-rational responses to gaining features through an update. Implications for research and practice are derived.

Keywords: Software updates, IT features, IS expertise, expectation-confirmation theory, IS continuance model, IS post-adoption

Introduction

The software industry and its business models have changed over the last years. This particularly applies to the market for consumer software. Traditionally, software vendors developed discrete programs and sold them as pre-packaged software at fixed prices. During the time of its use, this software remained largely unchanged and the user eventually replaced it with a new generation of this software, once it became available. This new generation of software was again sold at fixed prices. The popular office suites from Microsoft are a typical example for this practice. Recently, however, many software vendors have adopted a different practice. Often, a first, rudimentary version of an application is developed and sold. Then, over the course of time, this initial application is frequently changed through software updates. This practice is often (but does not have to be) accompanied by subscription based or ad-based revenue models that require a large and active user base in order to generate reoccurring revenues for vendors from renewed subscriptions or ad sales. This has not only become common practice in the app economy for smart phones and tablet computers but has also been adopted in the more mature field of software for desktop computers and web services. For example, Microsoft recently announced that it planned to shift to this practice with the version “10” of its operating system Windows, constantly enhancing and extending the software through updates over time while their customers already use it (Myerson 2015). As this example shows, vendors usually update their software in order to enhance it by correcting flaws or extending its functionality (i.e. features). In practice, however, the quality of software is sometimes also diminished by updates. A vendor can do this intentionally, for strategic reasons or when licensing deals run out, for instance. One example for this is the mapping functionality on the iPhone. In 2012, Apple updated its iOS operating system and, amongst other changes, removed the access to Google’s maps service (Apple 2015). Mapping functionality was replaced with a functional inferior, in-house developed service (Sherr 2012). Another example is an update to Google’s Android operating system from 2013. It removed privacy features which had previously allowed users to control the degree of personal data that could be accessed by third party applications (Constantin 2013). In other cases, software functionality is sometimes lost or diminished through updates unintentionally, when faulty updates corrupt features or render them useless.

However, despite the ubiquitous use of software updates in practice and many vendors’ dependency on their customers’ loyalty (i.e. continued use), there is little research on the impact of updates on users’ beliefs, attitudes and specifically their continuance intentions regarding the updated software (Hong et al. 2011; Claussen et al. 2013). Most of the existing research neglects the user perspective and explores software updates from a purely technical perspective. This includes research on software engineering (Sommerville 2010), software product lines (Clements and Northrop 2002), software release planning (Svahnberg et al. 2010) and software evolution and maintenance (Mens and Demeyer 2008). Because updates are the means by which the characteristics of software are changed over time, during use, they may have the potential to alter users’ beliefs, attitudes and behaviors regarding this software in the post-adoption stage (Karahanna et al. 1999; Bhattacharjee 2001). A better understanding of software updates from a user’s perspective thus has the potential to increase the explanatory and predictive power of existing post-adoption theory. However, researchers studying post-adoption phenomena often tend to conceptualize information systems (IS) as a monolithic and coarse-grained black box, rather than as a collection of specific and finer-grained features that are dynamic and alterable over time (Jasperson et al. 2005). Only few studies have explored IS usage at a feature level (Benlian 2015). These studies have considered that different users employ different feature sets (DeSanctis and Poole 1994; Leonardi 2013), value them differently (Hiltz and Turoff 1981) and that the breadth and depth of a feature set that is utilized may change over time (Kay and Thomas 1995; Sun 2012). Nonetheless this stream of research does usually not consider changes in the available feature set over time, during usage, such as the addition or removal of features through software updates. Understanding the granularity of software and its changes through software updates would help to explain how users’ beliefs, attitudes, and behaviors fluctuate over time as a result of the flexible nature of information systems. Moreover, there are several calls for research from IS scholars who criticize the negligence of the IT artifact’s role in IS research (Benbasat and Zmud 2003; Hevner et al. 2004; Orlikowski and Iacono 2001). They suggest focusing on changes in beliefs, attitudes and behaviors emanating from the IT artifact itself rather than from other IT-unrelated environmental stimuli. Another issue that arises from the increasing ubiquity of software (and consumer software specifically), is the potential diversity in a software’s user base. As more and more people gain access to information technology, it is increasingly also used by late adopters and non-experts

(Rogers 1995). This can be seen as a contrast to the usage in organizations, where information systems are often operated by specialists or employees who receive specific training. To theoretically account for these developments, it becomes increasingly important for IS research to explore the heterogeneity in different users' beliefs, attitudes and behaviors regarding IT. Past IS research has already accounted for this (e.g. Kim and Son 2009; Venkatesh et al. 2012) but when investigating new phenomena or use cases, this issue has to be considered consistently. This study therefore raises the following research questions:

RQ1: *Does gaining or losing features through software updates impact users' continuance intentions?*

RQ2: *How and why do novices and experts differ in their responses to software updates?*

Drawing on expectation-confirmation theory (Oliver 1980) which is embedded in the IS continuance model (Bhattacharjee 2001), we conducted a vignette based online experiment with 178 participants to answer these questions. This study contributes to prior research in three important ways. First, we increase the understanding of users' post-adoption behaviors by identifying differential reactions of novice and expert users to software updates. While experts show rational reactions, our findings regarding novices' responses are counter-intuitive and may be characterized as non-rational. We identify update type and user expertise as crucial moderators for explaining the use of agile information systems. By investigating the mediating role of disconfirmation of expectations, our second main contribution is shedding light on the explanatory mechanism behind these different responses to updates. This has not been explicated in such a nuanced way in previous continuance literature. Our third and overarching contribution lies in the extension of the predominant view of information systems in post-adoption literature from a mostly monolithic and static one to a finer-grained and more flexible perspective by showing how an alterable information system might influence users' attitudes and behaviors during use.

Software vendors can learn from this study's results that holding back functionality in the first release of a software to deliver it only later on through updates has the potential to please customers and increase their loyalty. This measure, however, may not work for expert users and even be counterproductive. Vendors should thus be well aware of their customer base's software specific expertise. Moreover, vendors should avoid removing features from software after its first release by any means. It may severely raise their customers' likelihood to stop using the software and switch to a competitor's product.

Theoretical Foundations

Software Updates

Software updates can be defined as self-contained modules of software that are provided to the user for free in order to modify or extend software after it has been rolled out and is already in use (e.g. Dunn 2004). Software updates are no 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 (e.g. operating systems, drivers, 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 (Sommerville 2010), such as software release planning, software maintenance and evolution and software product lines (Weyns et al. 2011). In this context, software release planning or strategic release planning refers to the "*idea of selecting the optimum set of features or requirements to deliver in a release within given constraints*" (Svahnberg et al. 2010, p. 1). Following this definition, an update is the delivery of features after the first release of a software and also falls within the strategic considerations regarding when to deliver what type of functionality to the user. Literature on software evolution and maintenance addresses the later stages in the software lifecycle, where updates are utilized to adjust software to changing requirements or repair emerging flaws in the software while it is already in use (Shirabad et al. 2001). In contrast to this rich stream of technical literature dealing with software updates from the developers' perspective, the users' beliefs and attitudes regarding updates have so far been explored only sparsely. This reflects in few IS studies dealing with updates (Amirpur et al. 2015). Hong et al. (2011), for example, explore user's acceptance of information systems that change through the addition of new functionality. Benlian (2015), 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. This also applies to other studies at the feature level which have considered that different users employ different feature sets

(DeSanctis and Poole 1994; Leonardi 2013), value individual features differently (Hiltz and Turoff 1981) and that the breadth and depth of the utilized feature set may change over the time of usage (Kay and Thomas 1995; Sun 2012). And while other IS studies have found updates to influence usage behaviors, they have often pushed them to the sidelines, treating them as control variables for investigating other phenomena (e.g. Claussen et al. 2013).

In the present study which investigates the user perspective, we distinguish two basic types of software updates: feature updates and non-feature updates. 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 popular 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 (Etherington 2013). 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) (Popović et al. 2001). Examples for this type of update are the prominent ‘hot fixes’ that Microsoft regularly distributes via its Windows Update service. Because they occur during the use of software and are usually recognized by users through notifications, required actions during installation and new or changed functionality, specifically feature updates have the potential to affect users’ post-adoption beliefs, attitudes and behaviors regarding software, including continuance intentions.

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 (Jasperson et al. 2005). In the context of post-adoption research (Karahanna et al. 1999; Bhattacharjee 2001; Benlian et al. 2011), the term information systems continuance refers to the “*sustained use of an IT by individual users over the long-term after their initial acceptance*” (Bhattacharjee and Barfar 2011, p. 2). To explore IS users’ intentions to continue or discontinue using an IS, Bhattacharjee (2001) adopts expectation-confirmation theory (ECT) (Locke 1976; Oliver 1980, 1993; Anderson and Sullivan 1993). ECT puts customers’ repurchase intentions at the center of investigation. In Bhattacharjee’s (2001) model, repurchase intention is replaced by a user’s intention to continue using an IS (CI)—the core dependent variable in his model. Following ECT, the IS continuance model suggests that users compare their pre-usage expectations of an IS with their perception of the performance of this IS during actual usage (Bhattacharjee 2001). If perceived performance exceeds their initial expectations, users experience positive disconfirmation (DISC) which has a positive impact on their satisfaction regarding the IS. If perceived performance falls short of the initial expectations, negative disconfirmation occurs and users’ satisfaction with the IS is reduced (Bhattacharjee and Barfar 2011). Satisfied users intend to continue using the IS, while dissatisfied users discontinue its subsequent use (Oliver 1980; Bhattacharjee 2001).

The concept of positive (negative) disconfirmation thus has two prerequisites—unexpectedness and a positive (negative) experience. Moreover, ECT posits expectations as a relative, subjective reference point or baseline (i.e. not an absolute, objective value) upon which the user makes his comparative judgment (Oliver 1980). This idea of a subjective, relative reference point is based on Helson’s (1964) adaptation level theory. It proposes that human beings perceive stimuli relative to or as a deviation from an ‘adapted level’ or baseline stimulus level. “*This adapted level is determined by the nature of the stimulus, the psychological characteristics of the individual experiencing that stimulus, and situational context*” (Bhattacharjee 2001, p. 354).

The IS continuance model has made valuable contributions to post-adoption research (Bhattacharjee 2001). However, in its original form, the IS continuance model has a static perspective on the IS continuance setting, failing to account for changing user beliefs and attitudes during use. In response to this limitation, Bhattacharjee and Premkumar (2004) introduce a more dynamic perspective by showing that beliefs and attitudes do not only change from pre-usage to actual usage but also during the ongoing usage of an IS. Kim and Malhotra (2005), Kim and Son (2009), Ortiz de Guinea and Markus (2009) and Ortiz de Guinea and Webster (2013), for instance, have provided evidence that the IS itself can shape users’ beliefs, attitudes and even their affect regarding the IT in later usage stages. Following

Bhattacharjee and Premkumar (2004), it is reasonable to assume that a change in the IT artifact can also induce changes in users' beliefs and attitudes towards it. To investigate the changing nature of the IT artifact and its impact on users' beliefs, attitudes and behaviors during post-adoption use, we explore software updates through the lens of the disconfirmation mechanism in ECT.

Information Systems Expertise

Due to superior knowledge and abilities regarding a subject matter, experts make better decisions and perform tasks more successfully than novices (Alba and Hutchinson 1987). Individuals' expertise has been explored in various research fields such as auditing (Shanteau and Steward 1992) and political science (Voss et al. 1983). Expertise is, however, not a general trait but specific to a certain subject or domain (Anderson 1982). Chess experts, for instance, *"do not appear to be better general thinkers for their genius in chess"* (Nelson et al. 2000, p. 477). Research on consumer decision making, for example, has repeatedly identified an individual's product related expertise to significantly influence product choices (e.g. Lynch et al. 1991) and the use of products (e.g. Blackler et al. 2010). One major finding of this stream of research is that past experience and knowledge about a product or class of products allows experts to make comparisons with previous evaluations when making decisions (Ghoshal et al. 2014). This can lead to more objective evaluations and make experts less prone to bias regarding product choice and use. In experiments, experts have been found to rely on extra experimental information retrieved from their memory to supplement the experimentally supplied information. Novices, on the other hand, are more stimulus-bound in their decision making (Lynch et al. 1991). Due to a lack of experience and knowledge regarding a product, novices' decisions are more bound to the immediate situation or product at hand. As a result, their decisions are often more subjective and they may fall prone to bias in their product related decision making more easily (Mishra et al. 1993).

Expertise has also been shown to affect beliefs, attitudes and behaviors in IS usage. Research has repeatedly found users' expertise with an information system to moderate the relationship between independent and dependent IS usage variables, significantly affecting their strength or direction (Venkatesh and Davis 1996, 2000; Szajna 1996; Venkatesh et al. 2003; Kim and Malhotra 2005). In IS research, there have been various conceptualizations of expertise, emphasizing its different dimensions such as knowledge or abilities. These conceptualizations include IS expertise (Nelson et al. 2000), IS competency (Huff et al. 1992; Munro et al. 1997; Marcolin et al. 2000; Eschenbrenner and Nah 2014) and computer self-efficacy (Marakas et al. 1998; 2007; Rhee et al. 2009). According to Munro et al. (1997, p. 45), user competence *"is composed of an individual's breadth and depth of knowledge of end user technologies, and his or her ability to creatively apply these technologies"*. The concept of computer self efficacy is related to expertise with an information system and has been found to be a strong predictor of end-user performance (Marakas et al. 2007). In particular, past use and the resulting user experience are known to play important roles as moderators in IS post-adoption phenomena (Venkatesh and Davis 2000; Jaspersen et al. 2005; Kim and Malhotra 2005; Kim and Son 2009). In the case of continued use, a user's earlier evaluations of an information system affect later evaluations because knowledge gained from experience with an IS is utilized in the decision making on its continuation (Hogarth and Einhorn 1992, Bolton 1998; Kim and Malhotra 2005). Eschenbrenner and Nah (2014, p. 1366) moreover point out that *"competency in the domain of IS is unique considering IS are continuously evolving, in development, and periodically upgraded (i.e., being updated, replaced, and modified)"*. However, despite its important role for understanding how users' beliefs, attitudes and behaviors might change over time, as the IT artifact's nature and composition evolves through software updates, user expertise has only been explored sparsely in post-adoption research so far. Especially in the consumer domain of IS usage, this constitutes a research gap, considering the abovementioned insights from consumer decision making research which highlight significant differences between experts' and novices' product related choice and use behaviors. This study thus addresses the moderating role of expertise in users' post-adoption perceptions of software updates and their potential impact on continuance intentions.

Hypothesis Development

In this section, we develop our hypotheses about how and under which conditions updates can influence users' beliefs and attitudes in post-adoption software usage. Specifically, we explore decisions on continued use or discontinuance in settings where use is not mandated, such as consumer software. We

therefore focus on feature updates which are recognized by the user during usage through explicit notification and ignore updates that are implemented 'behind the scenes'. Within this scope, we further distinguish between feature updates that add functionality and feature updates that remove functionality. We also distinguish expert and novice users. In our theorizing, we assume updates to deliver common features with functional equivalence across the hypothesized conditions. We make this assumption to properly reflect the practice (free updates do usually not deliver uniquely extraordinary features) and because previous research has found that uncommon, unique features may bias decisions and thus interfere with our attempt to conceptually isolate the psychological mechanism through which software updates might influence users' continuance intentions (e.g. Dhar and Sherman 1996).

IS Novices' Response to Gaining a Feature through a Software Update

We argue that receiving feature updates during the post-adoption use of an IS can induce positive disconfirmation and increase a novices' CI (Bhattacharjee 2001). According to ECT, the occurrence of positive disconfirmation requires an *unexpected* and *positive* experience (Oliver 1980). Overall, the experience must constitute an unanticipated, relative improvement compared to a baseline, i.e. it must exceed an individual's subjective reference point (Helson 1964). In the context of software updates, that means that a surprising update must lead to a perceived improvement in the functionality of a software compared to its pre-update state. Following research on product expertise, it is reasonable to assume that due to a lack of knowledge and past experiences (Alba and Hutchinson 1987), novices do usually not anticipate feature updates, making them surprising, unexpected experiences with the software. Even if a vendor provides release plans about future updates, in practice, novices are unlikely to follow such update plans in detail for each software product they use. Moreover, when assessing the value of gaining a feature through an update, novices simply compare a software's functionality after the update to the functionality before the update, using the software at hand as primary reference point. According to research on product expertise and IS user competence, novices' evaluations are bound to this immediate stimulus because they lack other reference points from domain specific knowledge or previous use experience (Lynch et al. 1991; Eschenbrenner and Nah 2014). Novices cannot assess if the received feature might be overdue, if it is maybe already available in competing software products, if the vendor has developed the feature long before and delivered it only later, with an intentional delay and if it took the vendor much effort to develop. In sum, it is thus likely that novices will perceive a feature update as *unexpected* and *positive* experience during use, inducing positive disconfirmation in the sense of ECT (Oliver 1980).

According to this logic, a software vendor should be able to create positive disconfirmation and thus increase IS novices' CI by applying the strategy of holding back features (functionality) in the first release of a software package and delivering this functionality only later on, through free software updates. Under this deferred feature delivery strategy, a feature-complete software package might be designed and developed by the vendor, but certain features might not be included in the initially shipped software version. As outlined above, the novice user is assumed to be unaware of the existence of these remaining features. Once these remaining features are subsequently delivered through updates, they likely elicit positive disconfirmation. Consistent with the IS continuance model, this could then lead to an increase in CI. This deferred feature delivery strategy is thus to be distinguished from an all-at-once feature delivery strategy under which all developed features are delivered in the first release.¹ To summarize, because of the subjective nature of the disconfirmation mechanism in ECT, which operates through an evaluation of relative instead of absolute change, and a lack of software specific knowledge and past experiences, novice users of software that receives functionality via feature updates will likely have a higher intention to continue using this software than novice users who received all these features right with the first release. We accordingly derive our first hypothesis:

H1a: IS novices have a higher continuance intention regarding software that receives features through updates compared to software that includes the complete and equivalent set of features right with the first release.

¹ Nonetheless, we assume that both feature delivery strategies overall comprise the same type and number of features. We also assume that under both strategies, the user's evaluation of the software regarding CI takes place at the same point in time, which is after the incremental feature delivery strategy has been executed (i.e. when users are endowed with the same set of features as if they had received them right with the first release)

IS Experts' Response to Gaining a Feature through a Software Update

We moreover argue that while ECT also applies to IS experts, it implies a different response to receiving feature updates. Following again research on product expertise and IS user competence, in contrast to novices, IS experts have more knowledge and past use experience about the updated software or this class of software (Alba and Hutchinson 1987; Eschenbrenner and Nah 2014). First, experts are thus more likely to anticipate updates or follow release plans if available. This reduces the likelihood that experts are surprised by an update and perceive it as unexpected event. Second, even if experts are surprised by a feature update, when evaluating this gain of functionality, they will use a different baseline against which they compare the post-update state of the software. Due to their superior knowledge and past usage experience with the software or type of software, experts do not only compare the new functionality to the pre-update state of the immediate software at hand, but also consider information about other, competing or similar software products or general technological developments to assess the value of the added feature. Overall, compared to novices, experts' evaluations of a feature update will be more objective, making them less subject to a biased perception of the new functionality (Lynch et al. 1991). Therefore, we argue, that experts do not fall prey to a vendors' deferred feature delivery strategy of holding back functionality in order to deliver it only later on and increase users' CI as easy as novices would. In practice, experts may even show a negative response to such a strategy of deferring features, when they identify the delivered functionality as common feature that is not a true innovation by the vendor but was developed long before and only held back intentionally. We therefore derive the following hypothesis:

H1b: Experts have a lower continuance intention regarding software that receives features through updates compared to software that includes the complete and equivalent set of features right with the first release.

Novices' and Experts' Response to Losing a Feature through a Software Update

Hypotheses 1a and 1b proposed different user reactions to *gaining* a held back feature through an update for experts and novices due to different levels of experience and knowledge regarding the functionality of a software or class of software. We argue that this different reaction of experts and novices will, however, not be present when *losing* a feature through an update. When losing a feature through an update during the use of a software, the formation of CI will also be influenced by the ECT mechanism (Oliver 1980; Bhattacharjee 2001). However, in this case, the functional baseline against which the updated software with reduced (lost) functionality will be compared includes the removed (lost) feature for experts *and* novices (Kim and Malhotra 2005). In their pre-update use of the software, they both have experienced the feature and are thus assumed to be aware of its presence and helpfulness in task completion. When a feature is removed from the software through an update, this leads to a perceived deterioration of the software for experts and novices. The updated software then lacks a specific feature which may previously have served as a tool for accomplishing a certain task. Assuming that this task can still be accomplished using the deteriorated software, its completion should become more difficult or time consuming. The user might have to substitute the lost functionality with another feature in the software or compensate for the lost feature by conducting previously automated steps of his task manually. As a consequence of this loss of functionality, the updated software should be perceived as comparatively less valuable by experts *and* novices. This should subsequently reduce their satisfaction with the software and intention to continue using it. We thus propose the following joint hypothesis for experts and novices:

H2: Both, experts and novices, have a lower continuance intention regarding software that loses features through updates after the first release compared to software that keeps these features.

The Mediating Effect of Disconfirmation

As pointed out before, we argue that the difference in IS novices' and experts' responses regarding CI from gaining a feature through an update originates in their different evaluations of the software through the ECT mechanism (i.e. different subjective baselines or reference points). According to the continuance model, compared to losing a feature, the novice's positive response should thus be mediated by a positive disconfirmation of their subjective, previous expectations regarding the software, i.e. DISC (Bhattacharjee 2001). Moreover, the ECT mechanism also suggests that such a positive disconfirmation of previous expectations (DISC) would not directly affect CI but in turn be mediated by SAT, which ultimately leads to

the proposed change in CI. Due to their different response to gaining a feature through an update, experts should not show this mediating effect. We thus propose the following mediation hypothesis:

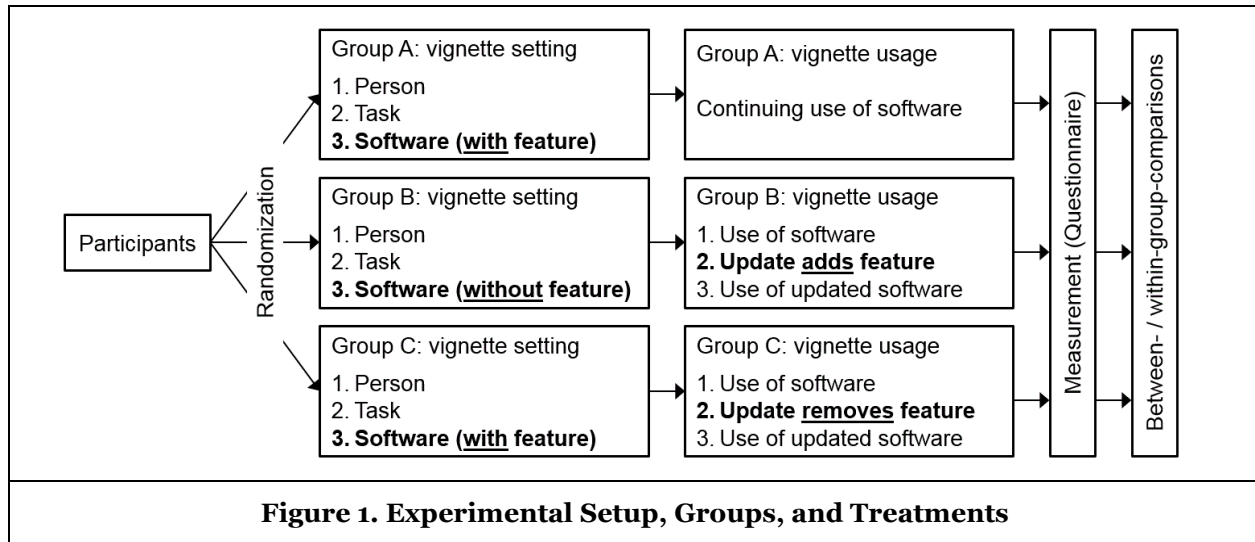
H3: The positive response of novices to gaining a feature through an update compared to losing a feature is mediated by DISC and SAT.

Method

Experimental Design

With the goal to examine the effects of software updates on users' CI as suggested by our hypotheses, we opted for a vignette based online experiment. It allowed us to investigate and isolate the causal mechanisms that operate between software updates and attitudinal user reactions. We presented participants with carefully constructed textual scenarios (vignettes) that precisely described a person (user), task, software, software usage and a conditional update (see Figure 1). We opted for the experimental vignette methodology (EVM) because it provides consistent and identical treatments for all participants and reduces unwanted effects such as social desirability bias (Aguinis and Bradley 2014). Even though this method comes with downsides such as a fictitious setting, it also allows for an accurate identification of the hypothesized effects. By being able to design a quasi-real scenario, the vignettes allowed us to ensure a high external validity, compared to a laboratory experiment. Nonetheless, researchers have shown that individuals respond quite similarly to hypothetical situations in vignettes compared to traditional laboratory experiments, making this method suitable for our needs (Rahman 1996; Shaw et al. 2003; De Cremer et al. 2007; Dennis et al. 2012).

We thus conducted a 1 x 3 between-subjects experiment (see Figure 1) with manipulations of update (no update vs. retained feature gained through update vs. feature lost through update). 178 participants from a large public university in Germany evaluated the impact of software updates on the user's continuance intentions. The participants read textual vignettes which described usage scenarios of a fictitious word-processing program ('xText') used by a fictitious student ('Tom') who had to write a term paper in group work together with classmates. Participants were randomly assigned to one of the three groups. Depending on the experimental condition, halfway during the described overall eight week use of the program, Tom received a feature through an update (group B) or lost a feature through an update (group C). In the control group, he used the software without any update (group A). Using a student sample was appropriate for this study, because students are likely to be familiar with word-processing programs, collaboration in group work tasks and software updates. They should also show similar attitudes and beliefs toward the treatments offered in our experiment compared to non-student samples (Jeong and Kwon 2012).



Manipulation of Independent Variables

In our experiment, we used a word-processing program for two reasons: We sought to ensure a basic familiarity with the program for all participants. Because nowadays almost any young person, especially students, needs to work with word-processing programs, we considered this criterion to be met. Second, for the update, we were looking for a software feature that was easily understandable through a textual description, preferably value-free and directly helpful in achieving the task but not indispensable so that the task could be completed also without the feature. Moreover, our hypothesis also required the update to deliver a feature that could technically be held back in the vendors' deferral strategy and was not an extraordinarily unique feature (Dhar and Sherman 1996). To identify this common feature for our treatment, we conducted a pre-study. In this pre-study, 52 subjects rated the relative importance of the 54 text editing features that are provided by the open source online text editor TinyMCE on seven-point-Likert scales (TinyMCE 2015). The subjects for this pre-study were recruited using WorkHub, a crowdsourcing platform similar to Amazon Mechanical Turk and participated online for a small payment (Paolacci et al. 2010). As a result, a feature for spell checking and grammar correction was selected. It met the requirements for our study best.

In the main study, the textual scenarios were presented to the participants in an online questionnaire that comprised several consecutive pages. On a first page, we described Tom, his task and the software with which he had to accomplish this task (see vignette setting, Figure 1). Tom had to write a term paper and work *“together with three classmates in a team. Their professor demands to write their term paper in English [which is not their native language].”* They had eight weeks to complete the paper. *“Because two team members are abroad during the entire working time, personal meetings are not possible”*. Therefore, they *“use the text editing program xText.”* The program only had *“a basic [yet sufficient] set of functionality but allows for collaboration in one text document by several users over the internet which is necessary...”*. Based on the information provided in the vignettes, the use of ‘xText’ was thus mandatory for this specific project. Depending on the experimental condition, the described software included—among other features which were listed in the vignette²—the feature for English spell checking and grammar correction (groups A and C) or not (group B). The use of this spell checking and grammar correction feature, however, was not mandated. Like any other feature in ‘xText’ its utilization was optional but—if available—obviously helpful for achieving the task. On a second page, we described Tom’s experience with the software during the entire time of the task completion, i.e. from starting to work on the term paper to handing in the final paper (see vignette use, Figure 1). Depending on the experimental condition, the description included an update of the software that added (group B) or removed (group C)

² These features were also identified from the results of the previously mentioned pre-study and resemble the basic functionality of web based text editing tools such as Google Docs in earlier versions. They were *“Changing font style and size”, “Underline, italics, bold”, “Alignment of text”, “Insert pictures and tables”*.

the feature for English spell checking and grammar correction or no update at all (group A). In group B, after four weeks of working on the paper, when opening the program, “*Tom is notified about an update that is automatically executed [...] and adds [for free] a feature for spell checking and grammar correction to the program.*” The new feature is described to save time for Tom because “*now he does not need to check the text word for word.*” In group C, after four weeks of working on the paper, when opening the program, “*Tom is notified about an update that is automatically executed [...] and removes [for free] the feature for spell checking and grammar correction from the program.*” As an explanation, it was stated that the vendor of ‘xText’ had only licensed this feature and it had to be removed because “*the licensing deal was not renewed*”. “*After the feature is removed, Tom has to check the text word for word for errors which costs time.*” Except for the description of the update, the usage of the program was described identically in group A. Except for the manipulated parts, we kept the scenario identical across the three groups. Each vignette ended with the group handing in the term paper after eight weeks. After this second page, participants started to answer the questionnaire. This included their evaluation of the protagonist’s intention to continue using ‘xText’ for future term papers when its use would no longer be mandated. Participants could only proceed to the next page when all questions were answered and returning to previous pages, including the vignettes, was not possible.

A pilot test with six subjects was conducted to ensure that the treatments were manipulated according to the experimental design (Perdue and Summers 1986). Specifically, subjects were asked about the comprehensiveness of the instructions, the vignettes and the questions in the following questionnaire. Suggestions were obtained from the participants and the vignettes and the questionnaire were revised accordingly for the main experiment.

Measures

Dependent Variables

We used validated scales with minor wording changes for all constructs. Measures for CI were adapted from Bhattacharjee (2001). Participants were asked to evaluate what they thought Tom would do, if after the completion of this term paper, he would have to write another paper in the future: ci1. *Tom intends to continue using xText rather than discontinue its use*; ci2. *Tom’s intentions are to continue using xText than use any alternative means*; ci3. *If Tom could, he would like to discontinue his use of xText* (reverse coded). DISC was also adopted from Bhattacharjee (2001): disc1. *Tom’s experience with using the word-processing program xText was better than what he expected*; disc2. *The functionality provided by the word-processing program xText was better than what Tom expected*; disc3. *Overall, most of Tom’s expectations from using the word-processing program xText were confirmed*. Measures for SAT were based on Kim and Son (2009): sat1. *Tom is content with the features provided by the word-processing program xText*; sat2. *Tom is satisfied with the features provided by the word-processing program xText*; sat3. *What Tom gets from using the features of the word-processing program xText meets what he expects for this type of programs*. Because constructs were measured with multiple items, summated scales based on the average scores of the multi-items were used in group comparisons (Zhu et al. 2012). Unless stated otherwise, the questionnaire items were measured on seven-point-Likert-scales anchored at (1)=strongly disagree and (7)=strongly agree.

Control Variables

In our study, we examined participant’s *motivation to process information* with one item (Suri and Monroe 2003), because this variable may also influence the response behavior of the participants and, thus, the validity of the results. Moreover, after conducting the experimental task, participants were asked to what extent they had understood *the items’ formulation*, to what extent they were *able to put themselves in the hypothetical setting described in the vignette*, if *the setting in the described story was realistic* and if they knew *what the goals of this survey were*. We included these control variables as well as the subjects’ demographics as covariates to isolate the effects of the manipulated variables. The participants’ expertise regarding word-processing programs was captured on an established four item, seven-point semantic differential scale with the items *know very little about/know very much about*, *inexperienced/experienced*, *uniformed/informed*, *novice buyer/expert buyer* (Mishra et al. 1993).

Participants, Incentives and Procedures

Participants for the final experiment were members of a large, public university in Germany. In December 2014, 6039 members of the university received an email, inviting them to participate in "an online survey about software usage". The email contained a link to the online experiment and stated that ten Amazon vouchers worth 10 € and one Amazon voucher worth 50 € were drawn in a lottery among all participants, once the study had been completed. Overall, 254 subjects started the experiment. 76 participants did not complete the experiment. They were excluded from our analysis. We thus used a sample of 178 subjects in the following analysis. Of these 178 subjects, 60 were males. The participants' average age was 25.12 ($\sigma=6.80$). 148 participants were students, 27 were employees or self-employed and three were seeking work. The educational backgrounds of the participants were diverse, including management, medical science, law, education, biology, physics, philosophy etc. Across the four seven-point semantic differential items, the mean score of the self-stated expertise with word-processing software was 3.96 ($\sigma=0.45$) on average. Based on this mean value across the four items for each participant, a median split was performed to classify subjects as experts and novices for the later hypothesis testing regarding expertise (Lynch et al. 1991). This resulted in the following group sizes: group A, no update, $n=57$ (30 experts, 27 novices); group B, feature gained, $n=63$ (42 experts, 21 novices); group C, feature lost, $n=58$ (31 experts, 27 novices). Across all groups, the participants indicated that they were able to put themselves in the hypothetical setting described in the vignette ($\bar{x}=5.40$, $\sigma=1.57$) and that they thought the described setting was realistic ($\bar{x}=5.23$, $\sigma=1.49$). Participants also indicated a high motivation to process information ($\bar{x}=6.42$, $\sigma=1.03$) and understood the questionnaire items well ($\bar{x}=6.11$, $\sigma=1.36$). On average, they stated that they did not know what the goals of this survey was ($\bar{x}=3.37$, $\sigma=1.75$). This indicates that we were successful in designing the experiment according to its purpose.

Data Analysis and Results

Control Variables

Based on the results of Fisher's exact tests, it can be concluded that there was no significant difference across the three experimental conditions in terms of gender ($p>0.1$) and profession ($p>0.1$). Furthermore, based on ANOVA tests, no significant differences were found across the six experimental conditions regarding age ($F=0.14$, $p>0.1$), and Mishra et al.'s (1993) self-evaluation of expertise on the seven-point semantic differentials ($F=0.88$, $p>0.1$). Furthermore, there was no significant difference across the three experimental conditions regarding the task-relevant control variables motivation to process information ($F=0.15$, $p>0.1$), the extent to which subjects were able to put themselves in the hypothetical situation described in the experimental task ($F=0.47$, $p>0.1$), the evaluation of the vignette's realism ($F=1.83$, $p>0.1$), the comprehensiveness of the items' phrasing ($F=0.74$, $p>0.1$), and knowing what the goals of the survey were ($F=1.11$, $p>0.1$). It is therefore reasonable to conclude that participants' demographics and task-relevant controls were homogeneous across the three conditions and did not confound the effects of our experimental manipulations.

Measurement Validation

Because we adopted established constructs for our measurement, confirmatory factor analysis (CFA) was conducted to test the instrument's convergent and discriminant validity (Levine 2005). Table 1 reports the CFA results using SmartPLS version 3.0 (Chin et al. 2003; Ringle et al. 2014) for the core constructs.

Table 1. Results of Confirmatory Factor Analysis for Core Variables

| Latent construct | Number of Indicators | Range of Standardized Factor Loadings* | Cronbach's Alpha | Composite Reliability (ρ_c) | Average Variance Extracted (AVE) |
|----------------------------|----------------------|--|------------------|------------------------------------|----------------------------------|
| Continuance Intention (CI) | 3 | 0.792-0.906 | 0.833 | 0.901 | 0.753 |
| Satisfaction (SAT) | 3 | 0.805-0.951 | 0.885 | 0.930 | 0.816 |
| Disconfirmation (DISC) | 3 | 0.782-0.920 | 0.844 | 0.907 | 0.766 |

Note: *All factor loadings are significant at least at the $p < 0.01$ level

All items loaded on the target factors and scored above the threshold of 0.7, indicating proper construct validity (Cook and Campbell 1979; Bartholomew et al. 2008). AVE values for each construct ranged from 0.753 to 0.818, exceeding the variance due to measurement error for that construct (AVEs exceeded 0.5). The constructs were assessed for reliability using Cronbach's alpha (Cronbach 1951). A value of at least 0.7 is suggested to indicate adequate reliability (Nunnally et al. 1994). The alphas for all constructs were well above 0.8. The composite reliability of all constructs exceeded 0.7, which is considered the minimum threshold (Hair et al. 2011). Thus, all of the constructs met the norms for convergent validity. For satisfactory discriminant validity, the square root of AVE from the construct should be greater than the variance shared between the construct and other constructs in the model (Fornell and Larcker 1981). As seen from the factor correlation matrix in Table 2, all square roots of AVE exceeded inter-construct correlations, providing strong evidence for discriminant validity. Hence, the constructs in our study are both theoretically and empirically distinguishable.

Table 2. Means, Standard Deviations, and Correlation Matrix for Core Variables

| Latent construct | M | SD | 1 | 2 | 3 |
|--------------------------------|-------|-------|--------------|--------------|--------------|
| (1) Continuance Intention (CI) | 4.118 | 1.626 | 0.868 | | |
| (2) Satisfaction (SAT) | 4.642 | 1.537 | 0.512*** | 0.875 | |
| (4) Disconfirmation (DISC) | 4.541 | 1.525 | 0.564*** | 0.756*** | 0.903 |

Note: Bolded diagonal elements are the square root of AVE. These values should exceed inter-construct correlations (off-diagonal elements) for adequate discriminant validity; *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Hypotheses Testing

In order to test our hypotheses, we conducted one-way analyses of variance (ANOVA) with contrast analyses using StataCorp Stata 12. Continuance intention (CI) was analyzed as function of update and expertise. There was a significant main effect for update ($F=25.94$, $p < 0.01$) but not for expertise ($F=2.01$, $p > 0.1$). However, the interaction between expertise and update had a significant effect on CI ($F=2.73$, $p < 0.05$). Contrast analysis revealed that experts and novices showed different reactions to gaining a feature. Novices showed a significant *higher* CI when gaining the feature (\bar{x} 's = 5.24 vs. 4.44, $p < 0.05$). This supports our hypothesis 1a. Experts, on the other hand exhibited a significant *lower* CI when gaining the same feature through an update (\bar{x} 's = 4.31 vs. 4.77, $p < 0.1$), supporting our hypothesis 1b. When losing a feature during use, both novices and experts had a significant lower CI (\bar{x} 's = 3.21 vs. 4.44, $p < 0.01$ and \bar{x} 's = 2.88 vs. 4.77, $p < 0.01$). This supports our hypothesis 2. Table 3 provides an overview over the effects of different update types and expertise on CI. Figure 2 visualizes the different user reactions to software updates, indicating mean values of CI for experts and novices across groups.

| Table 3. Means, Mean Differences and Significance Levels for Continuance Intention | | | | | |
|--|------------------------|---|---|--|-----------------|
| Expertise with Software | Mean Values for Groups | | | Mean Differences and Significance Levels | |
| Experts / Novices | No Update (A) n=57 | Feature Gained through Update (B) n=63 | Feature Lost through Update (C) n=58 | B-A | C-A |
| Experts | 4.77 | 4.31 | 2.88 | -0.46* | -1.89*** |
| Novices | 4.44 | 5.24 | 3.21 | 0.80** | -1.23*** |

Note: *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$ (one-sided); ANOVA-tests with contrast analyses

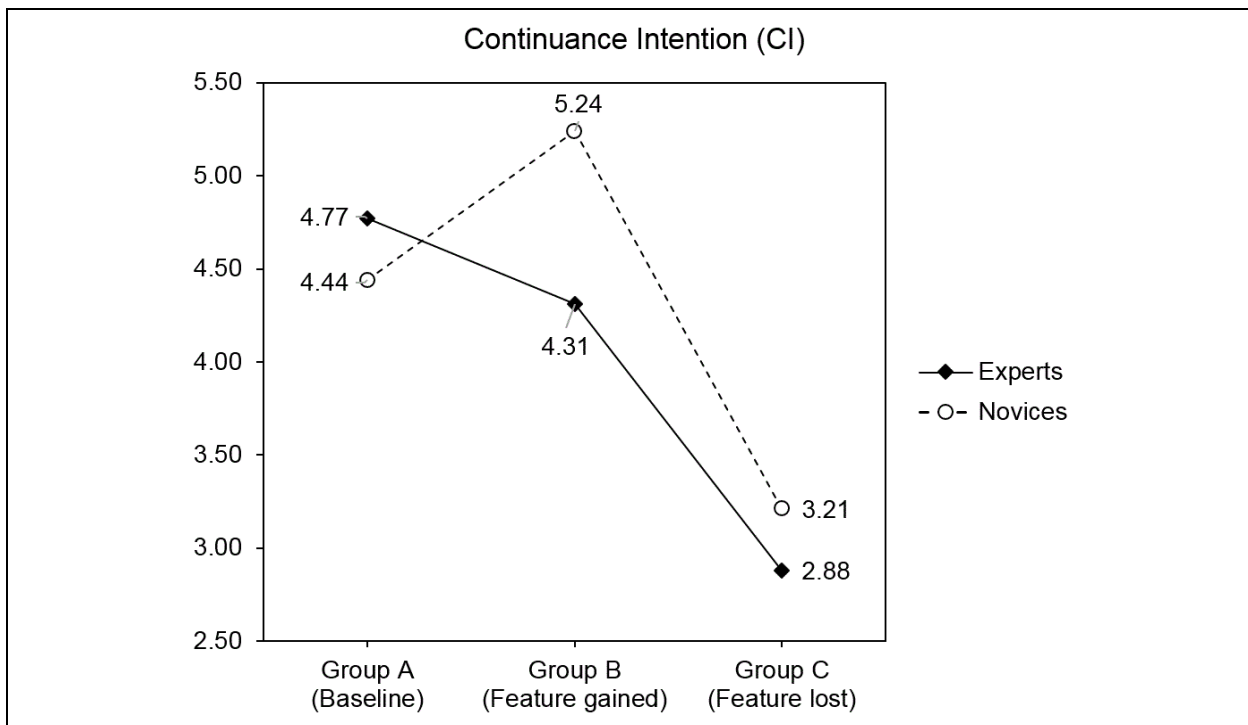
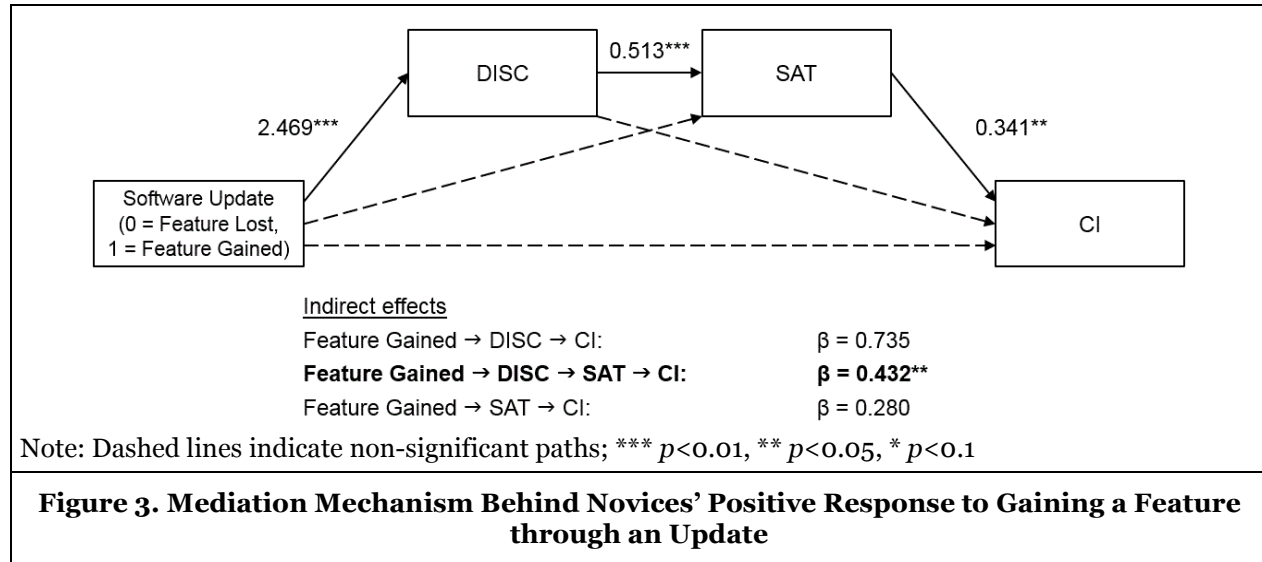


Figure 2. Expert and Novice Responses to Gaining and Losing Features from an Update

In order to investigate hypothesis 3 and explore the psychological mechanism behind the novices' different responses to gaining and losing a feature, a mediation analysis of the continuance model's core variables (Bhattacharjee 2001) was performed for novices in groups B (gaining a feature) and C (losing a feature). To analyze the mediating effects of DISC and SAT, we used PROCESS, a regression-based approach developed by Hayes (2013). PROCESS uses bootstrapping procedures for estimating direct and indirect effects. Figure 3 provides an overview of the analyzed conceptual model with direct and indirect paths. As recommended by Hayes (2013), path coefficients are unstandardized because the independent variable (software update) is dichotomous.



The results from bootstrapping analysis in Table 4 revealed that only the (unstandardized) indirect effect path (2) from gaining a feature through an update via DISC and SAT to CI was significant. Moreover, the direct effect of gaining a feature through an update on users' CI became insignificant after including DISC and SAT, suggesting full mediation (Hayes 2013). This mediation analysis was also performed separately for experts. As also expected from hypothesis 3, due to their different response to gaining a feature through an update, this mediation was not found for experts. Hence, hypothesis 3 is supported.

| Indirect effect paths | Effect z | Boot SE | LLCI | ULCI |
|---|--------------|--------------|--------------|--------------|
| (1) Feature Gained → DISC → CI | 0.735 | 0.596 | -0.210 | 2.211 |
| (2) Feature Gained → DISC → SAT → CI | 0.432 | 0.257 | 0.093 | 1.207 |
| (3) Feature Gained → SAT → CI | 0.280 | 0.319 | -0.133 | 1.086 |

Note: Inferential tests for indirect effect paths based on 1.000 bootstrap samples generating 95% bias-corrected bootstrap confidence intervals (LLCI=Lower Limit/ULCI=Upper Limit of Confidence Interval)

Discussion

This study sought to achieve three main objectives: (1) to examine the effects of different types of software updates on users' intentions to continue using an information system compared to monolithic software (i.e. whether there are discernible effects from gaining or losing features through updates), (2) to investigate the moderating role of IS expertise (i.e. if novices perceive updates differently than experts) and (3) to unravel the explanatory mechanism behind such different responses to updates (i.e. how and why such an effect from updates occurs). To achieve these objectives, we drew on the IS continuance model, the underlying expectation-confirmation theory and theory on IS user expertise and investigated our hypotheses based on a vignette based online experiment with 178 participants.

Drawing on the advantages of the experimental method, which allows to isolate the effects of manipulated stimuli on user responses from other confounding variables and thus to unveil causal relationships, we found that expert and novice users showed different reactions to updates. In the case of experts, any type of update led to a decrease in CI (groups B and C). Not only losing a feature through an update (group C) but even gaining a feature (group B) significantly lowered their intention to continue using the software. The response to losing a feature is comprehensible. Halfway through task completion, the user is deprived of a helpful functionality in the utilized program. This reduction in functionality makes his present task

more difficult and the program less valuable for any future use. Consequently, the user's intention to continue using the program beyond the current project (CI) is diminished. The experts' response to gaining a feature, on the other hand, may seem surprising at first, because it seemingly increases the value of the program to the user. However, the gained feature was artificially held back and intentionally delivered only later on, through an update. As suggested in hypothesis 1b, theory on product expertise (Alba and Hutchinson 1987) and information systems expertise (Eschenbrenner and Nah 2014) implies that expert users are likely to identify the delivered functionality as common feature that is not a true innovation by the vendor but was developed before and only held back on purpose. In line with this reasoning, experts in group B did not fall prey to the deferred feature delivery strategy, overall showing a rational behavior.

Novices on the other hand showed different reactions. While they also had a lower CI when losing a feature through an update (group C), their CI was significantly higher in the positive update condition (group B) than in the non-update condition (group A). This increase of novices' CI in group B compared to group A can be interpreted as being a somewhat counter-intuitive finding because the user described in the vignette with a feature gained through an update (group B) was objectively disadvantaged compared to the user who had all functionalities right with the first release (group A): during the time span of usage as described in the vignette, the user in group B had in sum fewer features per time to accomplish his text-formatting task compared to group A. Despite this objective disadvantage, novice participants in group B showed significantly higher scores in CI. This suggests the presence of a somewhat non-rational effect (Fleischmann et al. 2014). When comparing the absolute values of the novices' responses to gaining and losing a feature, their evaluations seem even less rational. Considering the non-update condition (group A) as reference point, the perceived loss from removing a feature from the software through an update (mean difference between responses by novices in group A and C) was higher in magnitude than the perceived gain from receiving the exact same feature through an update (mean difference between responses by novices in group A and B). This suggests the presence of loss aversion in novices (Kahneman and Tversky 1979). As such, both findings of novices' responses to updates challenge the idea of a 'rational user' in the IS continuance literature (Ortiz de Guinea and Markus 2009; Bhattacharjee and Barfar 2011; Ortiz de Guinea and Webster 2013).

Finally, we could demonstrate that the positive response to gaining a feature through an update regarding CI (novices in group B) is fully mediated by the ECT core variables DISC and SAT. Due to a lack of experience and outside knowledge, novices seem to be unable to objectively evaluate the gain of a retained feature from an update. In terms of ECT, novices only use their immediate, subjective perception of the software's functionality before the update as reference point. Exceeding this subjective reference point induces positive disconfirmation of previous expectations (DISC) which initiates a psychological process by which increases in SAT eventually lead to a higher CI.

Implications for Research

The paper makes three main contributions to the literature. First, we identify different user reactions to software updates. These responses crucially depend on the type of update and the users' expertise regarding the updated software. Losing a feature through an update decreases CI for experts and novices. Gaining a retained feature through an update, on the other hand, induces a positive reaction in novices. This has even a stronger and more positive impact on novices' continuance intentions compared to situations in which the entire feature set is provided at once and with the first release. Expert users, however, do not show this positive response. The gain of a feature can therefore be seen as necessary and the lack of expertise with the software as sufficient condition for this positive response to software updates that deliver features which have been held back at the initial release of software. Conceptually, update type and expertise regarding the updated software thus seem to moderate the effect of updates on CI. This interaction emphasizes the importance of a *joint* consideration of IT artifacts' and the users' characteristics when investigating usage behaviors. Our second main contribution is shedding light on the explanatory mechanism behind the identified positive effect of updates on CI for IS novices, which could not be ascertained for IS experts. Specifically, we find that this positive effect for IS novices is fully mediated by a positive disconfirmation of previous expectations regarding the software due to the update (DISC) and SAT. This finding once again highlights the pivotal role of ECT within the IS continuance model. Our third and overarching contribution lies in showing how a malleable information system might influence users' attitudes and behaviors during post-adoption use. This answers the calls of several IS

researchers by extending the still predominant view of post-adoption literature on the IT artifact as a static and monolithic block to a more flexible and finer-grained perspective which considers information systems as a modular composition of features that may change over time (Jasperson et al. 2005; Benbasat and Barki 2007 etc.). We complement existing IS post-adoption literature through the notion that users' beliefs and attitudes might fluctuate over time, in conjunction with changes in the used information system.

Implications for Practice

Our results also have important implications for practice. First, despite the extensive use of software updates by vendors to maintain, alter and extend their products after they have already been rolled out, it is surprising to find that insights on how these updates are perceived and evaluated by users are still scarce. This leaves practitioners without guidance. From the results of our experimental study we can conclude that vendors should avoid removing features from software after its release. This also includes well-intentioned updates which unintentionally damage the software and render certain features useless. When vendors remove functionality from their software, they significantly increase their customers' likelihood to discontinue using their product (and perhaps switch to a competitor's product). In the already highly competitive market for consumer software, vendors may want to avoid this by any means.

Adding helpful features through free updates, on the other hand, might seem as a straightforward measure for vendors to please customers and increase their loyalty (i.e. CI). More specifically, our findings suggest, that it could even be advisable for vendors to hold back software functionality and distribute it over time via updates, instead of delivering all features right with the first release of a software. Feature updates have the potential to increase users' CI above and beyond a level generated by software packages that are delivered with the entire feature set at once. However, the findings of this study reveal that this effect seems to work only for novice users. Software vendors can learn from this study's results that they should be well aware of their customer base and its expertise regarding the software. Utilizing customer data or conducting market research can be helpful in this regard. It should also be noted that vendors should not overdraw the holding back of functionality when applying the deferred feature delivery strategy. Starting out with a too small feature set might render the first release of a software almost useless and lead to discontinuation before the program can be updated or even prohibit the adoption in the first place. Especially vendors who face direct competition from other, similar software products should carefully evaluate what type and number of features they can afford to hold back under this strategy and which ones ought to be provided immediately in order to win or retain customers. In practice, each vendor will have to determine this sufficient amount of features for his own, specific case.

Finally, when maintaining their software after its first release, software vendors should not only focus on their own product but also keep track of connected or compatible programs from other vendors. In today's interconnected but quickly changing software industry, many programs rely on interoperability through interfaces, plug-ins and compatibility. When other connected or compatible software is changed through updates, the own interfaces and plug-ins may stop working and compatibility may vanish, rendering some features useless. In order to avoid losing customers' from such a loss in functionality (even if only temporary), vendors should closely monitor the integrity and functionality of these interfaces, plug-ins and compatibility and quickly respond to restore or repair them if necessary.

Limitations and Future Research

Four limitations of this study are noteworthy and provide avenues for future research. First, our experiment utilized textual vignettes to describe software usage scenarios. While this is a proven methodology, it also has some limitations (Aguinis and Bradley 2014). Our constructed setting was fictitious and it required subjects to put themselves in the position of the scenario's protagonist. Moreover, because the study was conducted online, there was no instructor who could have answered any questions regarding the described vignette scenario. We thus controlled for motivation to process information, perceived realism of the scenario and how well participants understood the questions and thought that they were able to put themselves in the hypothetical setting. Based on the results regarding these measures, we are confident that our vignettes worked as intended and our study's implications are applicable to real usage settings. Nonetheless, future studies could investigate actual usage experiences with real software to validate our findings. Second, we identified update type and user expertise as crucial

moderators for the effect of updates on users' CI. Future studies are encouraged to further differentiate update types (e.g. several features in one update) and explore additional user characteristics (e.g. different cultural backgrounds). Furthermore, complementary qualitative studies (e.g. thought-listing) could further substantiate our theoretical reasoning behind the identified moderators e.g., why experts disliked the deferred delivery of features through an update (Ma and Roesse 2014). Third, the demonstrated effects of updates on users' CI were shown to work for productivity (word-processing) software. Future research could show whether the same effects also occur for hedonic (e.g. entertainment) software. Finally, we conducted a controlled experiment with the purpose of presenting results with a high internal validity. This required some reasonable but strict assumptions, such as exploring a common feature, an identical and linear course of events for all users and ex-post measurement of variables. Future studies are encouraged to complement the findings of this study by investigating different types of features (e.g. extraordinary features) and conducting longitudinal field experiments, to advance the external validity of our findings. Also settings with repeated updates over longer time spans with participants evaluations measured at several points in time could provide additional evidence for the robustness of our findings. Specifically, a field experiment using an online service similar to Google Docs or Microsoft Office Online would be well suited to collect panel data from real usage over a longer period of time.

Conclusion

Software updates have become a pervasively used instrument for vendors to maintain, alter and extend their products over time. Despite this prevalence, their effects on crucial post-adoption user reactions have remained largely unexplored. This study's diverse findings highlight the importance of a profound understanding of updates for both researchers and practitioners. Updates that add features to a software after its first release, while it is already in use, have the potential to increase users' CI above and beyond a level generated by a monolithic software package that is released with the entire feature set at once. However, this only applies for novice users but not for experts. Losing a feature through an update, on the other hand, severely diminishes CI and raises a user's likelihood of switching to a competitor's product. Furthermore, this study explains the psychological mechanism behind the different user responses to updates. It works through disconfirmation of previous expectations regarding the updated software.

References

- Aguinis, H., and Bradley, K. J. 2014. "Best Practice Recommendations for Designing and Implementing Experimental Vignette Methodology Studies," *Organizational Research Methods* (17:4), pp. 351-371.
- Alba, J. W., and Hutchinson, J. W. 1987. "Dimensions of Consumer Expertise," *Journal of Consumer Research* (13:4), pp. 411-454.
- Amirpur, M., Fleischmann, M., Benlian, A., and Hess, T. 2015. "Keeping Software Users on Board—Increasing Continuance Intention Through Incremental Feature Updates," *Proceedings of the 23rd European Conference on Information Systems*, Münster, Germany, pp. 1-16.
- Anderson, E. W., and Sullivan, M. W. 1993. "The Antecedents and Consequences of Customer Satisfaction for Firms," *Marketing Science* (12:2), pp. 125-143.
- Anderson, J. R. 1982. "Acquisition of Cognitive Skill," *Psychological Review* (89:4), pp. 369-406.
- Apple. 2015. "iOS 6 Software-Update." Retrieved 05/01/2015 from <https://support.apple.com/kb/DL1578>
- Bartholomew, D. J., Steele, F., Moustaki, I., and Galbraith, J. 2008. *Analysis of Multivariate Social Science Data* (2nd ed.), London: CRC Press.
- Benbasat, I., and Barki, H. 2007. "Quo vadis TAM?," *Journal of the Association for Information Systems* (8:4), pp. 211-218.
- Benbasat, I., and Zmud, R. W. 2003. "The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties," *MIS Quarterly* (27:2), pp. 183-194.
- Benlian, A. 2015. "IT Feature Use over Time and its Impact on Individual Task Performance," *Journal of the Association for Information Systems* (16:3), pp. 144-173.
- Benlian, A., Koufaris, M., and Hess, T. 2011. "Service Quality in Software-as-a-Service: Developing the SaaS-Qual Measure and Examining Its Role in Usage Continuance," *Journal of Management Information Systems* (28:3), pp. 85-126.
- Bhattacharjee, A. 2001. "Understanding information systems continuance: an expectation-confirmation model," *MIS Quarterly* (25:3), pp. 351-370.

- Bhattacharjee, A., and Barfar, A. 2011. "Information Technology Continuance Research: Current State and Future Directions," *Asia Pacific Journal of Information Systems* (21:2), pp. 1-18.
- Bhattacharjee, A., and Premkumar, G. 2004. "Understanding Changes in Belief and Attitude toward Information Technology Usage: A Theoretical Model and Longitudinal Test," *MIS Quarterly* (28:2), pp. 229-254.
- Blackler, A., Popovic, V., and Mahar, D. 2010. "Investigating Users' Intuitive Interaction with Complex Artefacts," *Applied Ergonomics* (41:1), pp. 72-92.
- Bolton, R. N. 1998. "A Dynamic Model of the Duration of the Customer's Relationship with a Continuous Service Provider: The Role of Satisfaction," *Marketing Science* (17:1), pp. 45-65.
- Chin, W. W., Marcolin, B. L., and Newsted, P. R. 2003. "A Partial Least Squares Latent Variable Modeling Approach for Measuring Interaction Effects: Results from a Monte Carlo Simulation Study and an Electronic-Mail Emotion/Adoption Study," *Information Systems Research* (14:2), pp. 189-217.
- Claussen, J., Kretschmer, T., and Mayrhofer, P. 2013. "The Effects of Rewarding User Engagement: The Case of Facebook Apps," *Information Systems Research* (24:1), pp. 186-200.
- Clements, P., and Northrop, L. 2002. *Software Product Lines: Practices and Patterns* (3rd ed.), Boston, MA: Addison-Wesley Reading.
- Constantin, L. 2013. "EFF criticizes Google for removing 'vital privacy feature' with Android 4.4.2 | PCWorld." Retrieved 09/04/2015 from <http://www.pcworld.com/article/2080241/eff-criticizes-google-for-removing-vital-privacy-feature-with-android-442.html>
- Cook, T. D., and Campbell, D. T. 1979. *Quasi-Experimentation: Design & Analysis Issues for Field Settings*, Boston, MA: Houghton Mifflin.
- Cronbach, L. J. 1951. "Coefficient Alpha and the Internal Structure of Tests," *Psychometrika* (16:3), pp. 297-334.
- De Cremer, D., Van Dijke, M., and Bos, A. E. 2007. "When Leaders Are Seen as Transformational: The Effects of Organizational Justice¹," *Journal of Applied Social Psychology* (37:8), pp. 1797-1816.
- Dennis, A. R., Robert Jr, L. P., Curtis, A. M., Kowalczyk, S. T., and Hasty, B. K. 2012. "Research Note- Trust Is in the Eye of the Beholder: A Vignette Study of Postevent Behavioral Controls' Effects on Individual Trust in Virtual Teams," *Information Systems Research* (23:2), pp. 546-558.
- DeSanctis, G., and Poole, M. S. 1994. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization science* (5:2), pp. 121-147.
- Dhar, R., and Sherman, S. J. 1996. "The Effect of Common and Unique Features in Consumer Choice," *Journal of Consumer Research* (23:3), pp. 193-203.
- Dunn, K. 2004. "Automatic update risks: can patching let a hacker in?," *Network Security* (2004:7), pp. 5-8.
- Eschenbrenner, B., and Nah, F. F.-H. 2014. "Information Systems User Competency: A Conceptual Foundation," *Communications of the Association for Information Systems* (34:1), pp. 1363-1378.
- Etherington, D. 2013. "Chat Heads Coming To iOS Facebook App Via Update Pushing Out Anytime Now." Retrieved 05/01/2014, from <http://techcrunch.com/2013/04/16/chat-heads-coming-to-ios-facebook-app-via-update-pushing-out-anytime-now/>
- Fleischmann, M., Amirpur, M., Benlian, A., and Hess, T. 2014. "Cognitive Biases in Information Systems Research: A Scientometric Analysis," *Proceedings of the 22nd European Conference on Information Systems*, Tel Aviv, Israel, pp. 1-21.
- Fornell, C., and Larcker, D. F. 1981. "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error," *Journal of Marketing Research* (18:1), pp. 39-50.
- Ghoshal, T., Yorkston, E., Nunes, J. C., and Boatwright, P. 2014. "Multiple Reference Points in Sequential Hedonic Evaluation: An Empirical Analysis," *Journal of Marketing Research* (51:5), pp. 563-577.
- Hair, J. F., Ringle, C. M., and Sarstedt, M. 2011. "PLS-SEM: Indeed a Silver Bullet," *Journal of Marketing Theory and Practice* (19:2), pp. 139-151.
- Hayes, A. F. 2013. *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*, New York, NY: Guilford Publications, Inc.
- Helson, H. 1964. *Adaptation-Level Theory: An Experimental and Systematic Approach to Behavior*, New York, NY: Harper and Row.
- Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-105.
- Hiltz, S. R., and Turoff, M. 1981. "The Evolution of User Behavior in a Computerized Conferencing System," *Communications of the ACM* (24:11), pp. 739-751.

- Hogarth, R. M., and Einhorn, H. J. 1992. "Order Effects in Belief Updating: The Belief-Adjustment Model," *Cognitive Psychology* (24:1), pp. 1-55.
- Hong, W., Thong, J. Y., Chasalow, L. C., and Dhillon, G. 2011. "User Acceptance of Agile Information Systems: A Model and Empirical Test," *Journal of Management Information Systems* (28:1), pp. 235-272.
- Huff, S. L., Munro, M. C., and Marcolin, B. 1992. "Modelling and measuring end user sophistication," *Proceedings of the 1992 ACM SIGCPR Conference*, Cincinnati, OH, pp. 1-10.
- Jaspersen, J. S., Carter, P. E., and Zmud, R. W. 2005. "A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems," *MIS Quarterly* (29:3), pp. 525-557.
- Jeong, H. J., and Kwon, K.-N. 2012. "The Effectiveness of Two Online Persuasion Claims: Limited Product Availability and Product Popularity," *Journal of Promotion Management* (18:1), pp. 83-99.
- Kahneman, D., and Tversky, A. 1979. "Prospect Theory: An Analysis of Decision under Risk," *Econometrica* (47:2), pp. 263-292.
- Karahanna, E., Straub, D. W., and Chervany, N. L. 1999. "Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs," *MIS Quarterly* (23:2), pp. 183-213.
- Kay, J., and Thomas, R. C. 1995. "Studying Long-Term System Use," *Communications of the ACM* (38:7), pp. 61-69.
- Kim, S. S., and Malhotra, N. K. 2005. "A Longitudinal Model of Continued IS Use: An Integrative View of Four Mechanisms Underlying Postadoption Phenomena," *Management Science* (51:5), pp. 741-755.
- Kim, S. S., and Son, J.-Y. 2009. "Out of Dedication or Constraint? A Dual Model of Post-Adoption Phenomena and its Empirical Test in the Context of Online Services," *MIS Quarterly* (33:1), pp. 49-70.
- Leonardi, P. M. 2013. "When does Technology Use Enable Network Change in Organizations? A Comparative Study of Feature Use and Shared Affordances," *MIS Quarterly* (37:3), pp. 749-775.
- Levine, T. R. 2005. "Confirmatory Factor Analysis and Scale Validation in Communication Research," *Communication Research Reports* (22:4), pp. 335-338.
- Locke, E. A. 1976. "The Nature and Causes of Job Satisfaction," in *Handbook of Industrial and Organizational Psychology*, M.D. Dunnette (ed.), Chicago, IL: Rand McNally College Pub. Co., pp. 1297-1349.
- Lynch Jr, J. G., Chakravarti, D., and Mitra, A. 1991. "Contrast Effects in Consumer Judgments: Changes in Mental Representations or in the Anchoring of Rating Scales?," *Journal of Consumer Research* (18:3), pp. 284-297.
- Ma, J., and Roese, N. J. 2014. "The maximizing mind-set," *Journal of Consumer Research* (41:1), pp. 71-92.
- Marakas, G. M., Johnson, R. D., and Clay, P. F. 2007. "The Evolving Nature of the Computer Self-Efficacy Construct: An Empirical Investigation of Measurement Construction, Validity, Reliability and Stability Over Time," *Journal of the Association for Information Systems* (8:1), pp. 16-46.
- Marakas, G. M., Yi, M. Y., and Johnson, R. D. 1998. "The Multilevel and Multifaceted Character of Computer Self-Efficacy: Toward Clarification of the Construct and an Integrative Framework for Research," *Information Systems Research* (9:2), pp. 126-163.
- Marcolin, B. L., Compeau, D. R., Munro, M. C., and Huff, S. L. 2000. "Assessing User Competence: Conceptualization and Measurement," *Information Systems Research* (11:1), pp. 37-60.
- Mens, T., and Demeyer, S. 2008. *Software Evolution*, Berlin and Heidelberg: Springer.
- Mishra, S., Umesh, U. N., and Stem Jr, D. E. 1993. "Antecedents of the Attraction Effect: An Information-Processing Approach," *Journal of Marketing Research* (30:3), pp. 331-349.
- Munro, M. C., Huff, S. L., Marcolin, B. L., and Compeau, D. R. 1997. "Understanding and measuring user competence," *Information & Management* (33:1), pp. 45-57.
- Myerson, T. 2015. "The next generation of Windows: Windows 10." Retrieved 05/01/2015 from <http://blogs.windows.com/bloggingwindows/2015/01/21/the-next-generation-of-windows-windows-10/>
- Nelson, K. M., Nadkarni, S., Narayanan, V., and Ghods, M. 2000. "Understanding Software Operations Support Expertise: A Revealed Causal Mapping Approach," *MIS Quarterly* (24:3), pp. 475-507.
- Nunnally, J. C. 1994. *Psychometric Theory* (3rd ed.), New York, NY: McGraw-Hill.
- Oliver, R. L. 1980. "A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions," *Journal of Marketing Research* (17:4), pp. 460-469.

- Oliver, R. L. 1993. "Cognitive, Affective, and Attribute Bases of the Satisfaction Response," *Journal of Consumer Research* (20:3), pp. 418-430.
- Orlikowski, W. J., and Iacono, C. S. 2001. "Research Commentary: Desperately Seeking the "IT" in IT Research - A Call to Theorizing the IT Artifact," *Information Systems Research* (12:2), pp. 121-134.
- Ortiz de Guinea, A., and Markus, M. L. 2009. "Why Break the Habit of a Lifetime? Rethinking the Roles of Intention, Habit, and Emotion in Continuing Information Technology Use," *MIS Quarterly* (33:3), pp. 433-444.
- Ortiz de Guinea, A., and Webster, J. 2013. "An Investigation of Information Systems Use Patterns: Technological Events as Triggers, the Effect of Time, and Consequences for Performance," *MIS Quarterly* (37:4), pp. 1165-1188.
- Paolacci, G., Chandler, J., and Ipeirotis, P. G. 2010. "Running Experiments on Amazon Mechanical Turk," *Judgment and Decision Making* (5:5), pp. 411-419.
- Perdue, B. C., and Summers, J. O. 1986. "Checking the Success of Manipulations in Marketing Experiments," *Journal of Marketing Research* (23:4), pp. 317-326.
- Popović, M., Atlagić, B., and Kovačević, V. 2001. "Case study: a maintenance practice used with real-time telecommunications software," *Journal of Software Maintenance and Evolution: Research and Practice* (13:2), pp. 97-126.
- Rahman, N. 1996. "Caregivers' Sensitivity to Conflict: The Use of the Vignette Methodology," *Journal of Elder Abuse & Neglect* (8:1), pp. 35-47.
- Rhee, H.-S., Kim, C., and Ryu, Y. U. 2009. "Self-efficacy in information security: Its influence on end users' information security practice behavior," *Computers & Security* (28:8), pp. 816-826.
- Ringle, C. M., Wende, S., and Becker, J.-M. 2014. *SmartPLS 3.0*, Hamburg: SmartPLS GmbH (<http://www.smartpls.com>).
- Rogers, E. M. 1995. *Diffusion of Innovations* (4th ed.), New York, NY: Free Press.
- Shanteau, J., and Stewart, T. R. 1992. "Why Study Expert Decision Making? Some Historical Perspectives and Comments," *Organizational Behavior and Human Decision Processes* (53:2), pp. 95-106.
- Shaw, J. C., Wild, E., and Colquitt, J. A. 2003. "To Justify or Excuse?: A Meta-Analytic Review of the Effects of Explanations," *Journal of Applied Psychology* (88:3), pp. 444-458.
- Sherr, I. 2012. "Apple Makes a Wrong Turn as Users Blast Map Switch – WSJ." Retrieved 05/01/2015 from <http://www.wsj.com/articles/SB10000872396390443890304578008712527187512>
- Shirabad, J. S., Lethbridge, T. C., and Matwin, S. 2001. "Supporting Software Maintenance by Mining Software Update Records," *Proceedings of the International Conference on Software Maintenance*, Florence, Italy, pp. 22-31.
- Sommerville, I. 2010. *Software Engineering: International Version* (9th ed.), Boston, MA: Pearson.
- Sun, H. 2012. "Understanding User Revisions When Using Information System Features: Adaptive System Use and Triggers," *MIS Quarterly* (36:2), pp. 453-478.
- Suri, R., and Monroe, K. B. 2003. "The Effects of Time Constraints on Consumers' Judgments of Prices and Products," *Journal of Consumer Research* (30:1), pp. 92-104.
- Svahnberg, M., Gorschek, T., Feldt, R., Torkar, R., Saleem, S. B., and Shafique, M. U. 2010. "A Systematic Review on Strategic Release Planning Models," *Information and Software Technology* (52:3), pp. 237-248.
- Szajna, B. 1996. "Empirical Evaluation of the Revised Technology Acceptance Model," *Management Science* (42:1), pp. 85-92.
- TinyMCE. 2015. "TinyMCE - Home." Retrieved 05/01/2015, from <http://www.tinymce.com/>
- Venkatesh, V., and Davis, F. D. 1996. "A Model of the Antecedents of Perceived Ease of Use: Development and Test," *Decision Sciences* (27:3), pp. 451-481.
- Venkatesh, V., and Davis, F. D. 2000. "A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies," *Management Science* (46:2), pp. 186-204.
- Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. 2003. "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly* (27:3), pp. 425-478.
- Venkatesh, V., Thong, J. Y., and Xu, X. 2012. "Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology," *MIS Quarterly* (36:1), pp. 157-178.
- Voss, J. F., Greene, T. R., Post, T. A., and Penner, B. C. 1983. "Problem-Solving Skill in the Social Sciences," in *The Psychology of Learning and Motivation*, Vol. 17, G.H. Bower (ed.), New York, NY: Academic Press, pp. 165-213.

- Weyns, D., Michalik, B., Helleboogh, A., and Boucke, N. 2011. "An Architectural Approach to Support Online Updates of Software Product Lines," *Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture*, Boulder, CO, pp. 204-213.
- Zhu, M., Billeter, D. M., and Inman, J. J. 2012. "The Double-Edged Sword of Signaling Effectiveness: When Salient Cues Curb Postpurchase Consumption," *Journal of Marketing Research* (49:1), pp. 26-38.