

# KEEPING SOFTWARE USERS ON BOARD—INCREASING CONTINUANCE INTENTION THROUGH INCREMENTAL FEATURE UPDATES

*Complete Research*

Amirpur, Miglena, Technische Universität Darmstadt, Darmstadt, Germany,  
amirpur@ise.tu-darmstadt.de

Fleischmann, Marvin, Ludwig-Maximilians-Universität München, München, Germany,  
fleischmann@bwl.lmu.de

Benlian, Alexander, Technische Universität Darmstadt, Darmstadt, Germany,  
benlian@ise.tu-darmstadt.de

Hess, Thomas, Ludwig-Maximilians-Universität München, München, Germany,  
thess@bwl.lmu.de

## Abstract

*Although feature updates are a ubiquitous phenomenon in both 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 incremental feature 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 updates incrementally rather than providing the entire feature set right with the first release. However, we also find that CI diminishes when the number of updates exceeds a tipping point in a given timeframe, disclosing update frequency as crucial boundary condition. Furthermore, we unveil that the beneficial effect of feature updates on CI operates through positive disconfirmation of expectations, resulting in increased user satisfaction. Implications for research and practice as well as directions for future research are discussed.*

*Keywords: Feature Updates, IT Features, Expectation-Confirmation Theory, IS Continuance, IS Post-Adoption.*

## 1 Introduction

In recent years, software vendors have increasingly leveraged feature updates as a measure to enhance their software products. Feature updates are self-contained modules of software, that are provided to the user for free in order to extend and enhance the functionality of software after it has been rolled out and is already in use. Functionality thereby refers to distinct, discernible features which are deliberately employed by the user in accomplishing the task or goal for which he or she uses the software (Benlian 2015). Feature updates are thus no discrete and stand-alone programs themselves but rather integrated into the base software once they are applied to it (e.g., Dunn 2004). Such feature updates that are the focus of the present study, are to be distinguished from other, non-feature update types, such as bug-fixes. These technical non-feature updates do not change the core feature set of software but only correct flaws or change software properties. In contrast to feature updates, they often do not directly affect the user's interaction with the software and are typically not even visible to the user (e.g., improvements in stability, compatibility, security or performance) (Popović et al. 2001).

Feature updates are a particularly prevalent phenomenon in the area of mobile applications and operating systems, but have also been used long before in the desktop space. In a 2013 update, the popular Facebook app for smartphones and tablet computers received a comprehensive instant messaging chat feature (Etherington 2013). On the desktop, web browsers such as Google Chrome and Mozilla Firefox continuously receive feature updates, which extend their functionalities. Here, an example is the ‘tab sync’ functionality, which was added to the browser Google Chrome in 2012 via a feature update. This particular feature enabled users to synchronize opened websites (tabs) across different computers and mobile devices in order to seamlessly continue browsing when switching devices (Mathias 2012).

This ubiquitous use of feature updates by software vendors in practice is reflected in a large body of research on the technical design of software, its maintenance and management. Research on software engineering (Sommerville 2010), including software product lines (Clements and Northrop 2002), software release planning (Svahnberg et al. 2010) and software evolution and maintenance (Mens and Demeyer 2008) 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 focus nonetheless lies on the supply side, primarily exploring technical design aspects of software. There is as yet, however, little understanding of the user’s perspective on updates—the demand side. In particular, the behavioral dimension, i.e. how updates are perceived by users is still an area that has so far received only minimal research attention (Hong et al. 2011; Sandberg and Alvesson 2011).

Investigating the effect of feature 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 (Bhattacharjee and Barfar 2011). For software vendors, shedding light on the role of feature updates for the IS continuance decision can result in a better understanding of how to strategically utilize updates in order to achieve desirable performance outcomes such as higher user loyalty and sustained revenue streams. From a research perspective, a better understanding of feature updates from a user’s perspective has the potential to increase the explanatory and predictive power of existing post-adoption theory. In particular, 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 finer-grained features that are dynamic and alterable over time. However, understanding the granularity of software and its changes through feature 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 (Benbasat and Zmud 2003; Hevner et al. 2004; Orlikowski and Iacono 2001).

We therefore seek to address the discussed research gaps by examining the questions of *whether, when and how feature updates influence users’ IS continuance intentions*.

We contribute to prior research in three important ways. First, we identify a positive and somewhat counterintuitive effect of feature updates on users’ CI. According to this effect, software vendors can increase their users’ CI by delivering functionality via incremental updates rather than providing the entire feature set right with the first release of the software. A key boundary condition of this effect, however, is update frequency. We found that CI diminishes when the number of updates exceeds a tipping point in a given timeframe. Second, we not only investigate the direct effect of feature updates on CI; we also open up the theoretical black box of how feature updates influence IS continuance intention by highlighting the role of affect. Third, 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. From a practitioner's perspective, our study offers implications for software vendors on how to increase their customers' loyalty (i.e., CI) through the delivery of feature updates. 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 Feature updates

In the software engineering literature (e.g., Sommerville 2010), a feature update is the delivery of functionality after the first release of a software and falls within the strategic considerations regarding when to deliver what type of functionality to the user (Svahnberg et al. 2010). A first release is the version of a software that is released to the market for the very first time, as well as the initial release of a new generation of an already established software. As pointed out in the introduction, functionality refers to distinct, discernible features which are deliberately employed by the user in accomplishing the task or goal for which he uses the software (Benlian and Hess 2011; Benlian 2015). From the user's perspective, feature updates occur during the continued use of software and are usually recognized through notifications, required actions during installation or through the display of new or changed functionality. As we will outline later on, we argue that this has the potential to influence users' beliefs, attitudes, and behaviors regarding the focal software in the post-adoption stage of IS usage, including their decisions on continued use or discontinuance in those settings where use is not mandated, such as consumer software.

### 2.2 Information systems continuance

In post adoption research (Karahanna et al. 1999; Bhattacharjee 2001), the term information systems continuance refers to "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 proposes satisfaction (SAT) with a product or service as the essential driver of repurchase intention. 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. According to Bhattacharjee (2001), it is influenced by satisfaction (SAT) and perceived usefulness (PU). SAT is an affective state and the result of a positive disconfirmation of prior expectations (Oliver 1980; Bhattacharjee 2001). 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 which has a positive impact on their satisfaction with the IS. If perceived performance falls short of the initial expectations, negative disconfirmation occurs and users are dissatisfied with the IS (Bhattacharjee and Barfar 2011). Positive (negative) disconfirmation thus consists of two elements—unexpectedness and a positive (negative) experience. Satisfied users intend to continue using the IS, while dissatisfied users discontinue its subsequent use. PU, on the other hand, captures the expectations about future benefits from using the IS (Bhattacharjee and Barfar 2011).

In its original form, the IS continuance model (Bhattacharjee 2001) has a static perspective on the IS continuance setting, failing to account for changing user beliefs and attitudes over time. 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). While this dynamic perspective already provides valuable insights into the drivers of post-adoption behavior, it still neglects the IT artifact's

changing and malleable nature. Evidence from practice shows that information systems are constantly modified over time, for example, when vendors update and change their software or introduce new software generations. Following Bhattacharjee and Premkumar (2004), it is reasonable to assume that a change in the IT artifact may also induce a change in users' beliefs and attitudes toward it. Kim and Malhotra (2005), Kim (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. In order to investigate the changing nature of the IT artifact and its effect on users' beliefs, attitudes and behaviors during post-adoption use, we explore feature updates through the lens of the disconfirmation mechanism in ECT.

### 3 Hypotheses Development

#### 3.1 The effect of unexpected feature updates on users' continuance intentions

We argue that if a free feature update provides additional functionality that directly serves users in accomplishing their IS-based tasks, it will be perceived as a positive experience with the software. Furthermore, it is reasonable to assume that feature updates are usually not anticipated by users and can thus be perceived as unexpected experiences with the software. Even if a software vendor does provide release plans about future feature updates, we suggest that in practice, most users—and especially consumers—are unlikely to follow such update plans in detail for each and every individual software product they have in use. If feature updates are perceived as unexpected and positive experiences during usage, according to ECT, they should consequently induce perceived positive disconfirmation (Oliver 1980). As a result, drawing on ECT and the IS continuance model (Bhattacharjee 2001), it is plausible that perceived positive disconfirmation during software use will increase users' CI regarding the updated software.

In the context of software features, ECT moreover implies that positive disconfirmation from feature updates depends on a relative change in functionality compared to a user's subjective reference point (the initial configuration of the software) rather than an absolute change (Helson 1964; Oliver 1980). According to this logic, a software vendor should thus be able to create positive disconfirmation and therefore increase the user's CI by applying the strategy of simply holding back features (functionality) in the first release of a software package and delivering this functionality only later on, through incremental, free feature updates. Under this incremental feature delivery strategy, a feature-complete software package might be designed and developed by the software vendor, but certain features might not be included in the initially shipped software version. The user is assumed to be unaware of the existence of these remaining features. Once these remaining features are subsequently delivered through updates, they are likely to elicit positive disconfirmation. Consistent with the IS continuance model, this could then lead to an increase in CI. This incremental 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.

Nonetheless, both feature delivery strategies are assumed to overall comprise the same type and number of features. We additionally 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). To summarize, because of the nature of the disconfirmation mechanism in ECT, which operates through an evaluation of relative instead of absolute change, the users of software that receive functionality via incremental feature updates will likely have a higher intention to continue using this software than users who received all these features right with the first release. We accordingly derive our first hypothesis:

*Hypothesis 1: Software that receives functionality via incremental feature updates induces a higher continuance intention compared to software that includes the complete and equivalent set of functionality right with the first release.*

### **3.2 The effect of expected feature updates on users' continuance intentions**

As outlined before, users must perceive updates as unexpected 'small gifts' from the vendor akin to a surprise that surpasses users' expectations (Oliver 1980). However, if these feature updates are delivered too frequently, they will probably no longer be perceived as unexpected by users because the feature updates become a predictable routine. Therefore, it is likely that the anticipated benefits from these expected feature updates will be included in the expectation about the future performance of the software (Kim and Malhotra 2005). The experience with the software would then no longer exceed the expectation, leading to a lack of positive disconfirmation. As a result an increase in CI from feature updates, as suggested in hypothesis 1, would fail to occur. In addition to this lack of positive disconfirmation, a high frequency of updates also increases the likelihood of being perceived as unsolicited interruptions of the workflow (Gluck et al. 2007; Hodgetts and Jones 2007). While additional functionality through updates may be welcomed by the user, with increasing frequency of updates, the accompanied interruptions might reach a point, where they are perceived as disproportionately high compared to the benefits (i.e. functionality) that accompany them. In terms of ECT, such a negative experience with the software from a too high frequency of updates may also diminish or even annihilate the previously discussed positive effect of the added functionality received through the feature update. Based on this logic, we argue that when the number of updates goes beyond a specific tipping point, the positive effect of feature updates on users' CI will decrease again or even completely disappear. Taken together, we thus hypothesize:

*Hypothesis 2: Beyond a threshold level of update frequency, incremental feature updates will no longer increase users' continuance intentions.*

### **3.3 The mediating effect of satisfaction for feature updates**

According to ECT (Oliver 1980) and the IS continuance model (Bhattacharjee 2001), disconfirmation will not have a direct effect on CI but will instead work through a mediation mechanism. Specifically, a positive disconfirmation leads—in a first step—to an affective response: an increase in the user's SAT with the IS. Only in a subsequent, second step will this increased SAT with the IS lead to a higher intention to continue using the IS. In the case of an unexpected feature update (hypothesis 1), the pleasant surprise of this helpful, 'free gift' from the software vendor that exceeds the expectation about this software would induce positive disconfirmation. ECT suggests that the positive disconfirmation from such a feature update then would trigger a positive affect which is reflected in increased SAT. Accordingly, we argue that SAT is the factor that drives and explains this increase in CI regarding the software that receives functionality through incremental feature updates compared to software that includes all features right with the first release.

Even though PU is another core driver of CI in the IS continuance model, we argue that the positive effect from an incremental feature delivery strategy—compared to an all-at-once feature delivery strategy—on CI is not driven by PU. This is because in the continuance model PU represents the user's evaluation of future benefits from using the software, regarding its functionality, i.e. features (Bhattacharjee 2001). According to our initial assumption (hypothesis 1), under both feature delivery strategies, the user's evaluation of PU occurs when the incremental feature delivery strategy is executed, i.e. the feature updates have already been delivered and users are thus endowed with the same set of features as if they had received them right with the first release. In both cases, the prospective benefits from using the software should thus be identical, implying the same level of PU (Bhattacharjee 2001). Moreover, it should be noted, that this assumption likely resembles the real-world use scenario. When users have to make a decision about continuing an IS, they will probably base their decision on

the configuration of the software that they have recently worked with rather than the configuration, which they originally started to work with. To sum up, the specific comparative increase in CI from an incremental feature delivery strategy as proposed in hypothesis 1 is solely mediated by SAT. We thus hypothesize:

*Hypothesis 3: The positive effect of incremental feature updates on users' continuance intentions is mediated by satisfaction with the software.*

## 4 Method

### 4.1 Experimental design

With the goal to examine the effects of feature updates on users' CI as suggested by our hypotheses, we opted for a laboratory experiment that allowed us to investigate and isolate the causal mechanisms that operate between feature updates and attitudinal user reactions. Even though this laboratory setting comes with the downsides of a simplified experimental task and a limited time span of observable usage, it also allows for an accurate identification of the hypothesized effects which we consider as crucial given that this study is the first to explore the effect of feature updates on users' continuance intentions. A second reason for choosing an experiment was the indication from theory that, working through affect, the core mechanism behind our proposed effect of feature updates might be outside of users' awareness, which made a cross-sectional survey with self-reported measures less suitable. Third, the experimental setting enabled us to account for the claims of numerous continuance researchers to put the IT artifact more at the center of investigation in post-adoption research by using an IS as basis for manipulations. We thus conducted a 1 x 3 between-subjects laboratory experiment (see Figure 1) with 90 participants recruited at a large public university in Germany to evaluate the impact of feature updates on the user's SAT, PU and CI. The participants used a word-processing program ('eWrite') with a simplified user interface that was developed and tailored to the purposes of this experiment to complete a text formatting task. The use of a student sample is appropriate for this study, because college students are likely to be familiar with both feature updates and word processing programs and show similar attitudes and beliefs toward the feature updates offered in our experiment compared to non-student samples (Jeong and Kwon 2012).

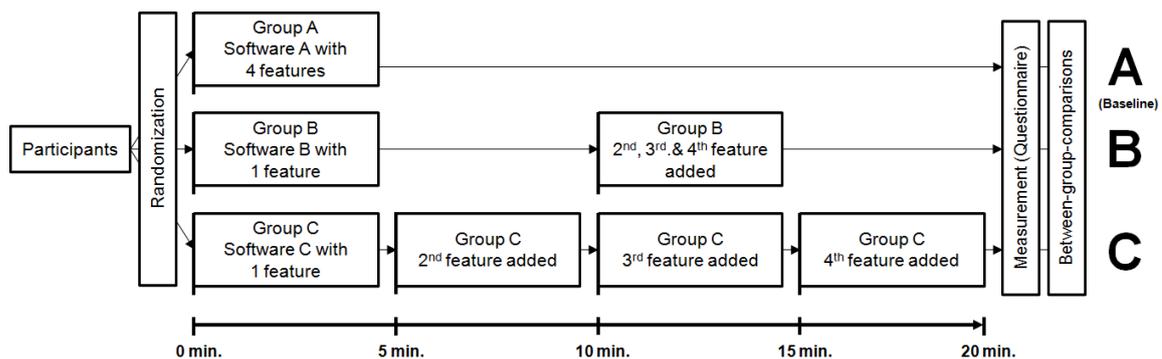


Figure 1. Experimental Setup, Groups, and Treatments.

### 4.2 Manipulation of independent variables

In our experiment, we used a word-processing program for two reasons: Our first criterion was ensuring a basic familiarity with the program of choice 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<sup>1</sup>. Second, to minimize unwanted variance in our response data, we were looking for software features that are preferably value-free, equivalent<sup>2</sup>, and independent (i.e., modular). We used a total of four text formatting features in our word-processing system context: 1) font size, 2) font style, 3) font, and 4) text alignment. The feature updates were directly related to the experimental task by adding new text-formatting functionalities. The available time for task completion was 20 minutes. In the one-feature-update condition (B), participants simultaneously received features 2, 3, and 4 ten minutes after having started to work on the task (see Figure 1). In the three-feature-update condition (C), participants received the first update (with feature 2) after five minutes, the second update (with feature 3) after ten minutes and the third update (with feature 4) after fifteen minutes. Participants in each group were informed about updates via a pop-up notification window at the center of the screen, which contained a brief explanation of the update's content and required them to confirm the update by clicking on an 'Ok' button before they could proceed with their experimental task. After confirming the notification, participants could immediately use the new feature. This notification had been included in order to ensure awareness with the feature update. Figure 2 provides examples of the user interface.

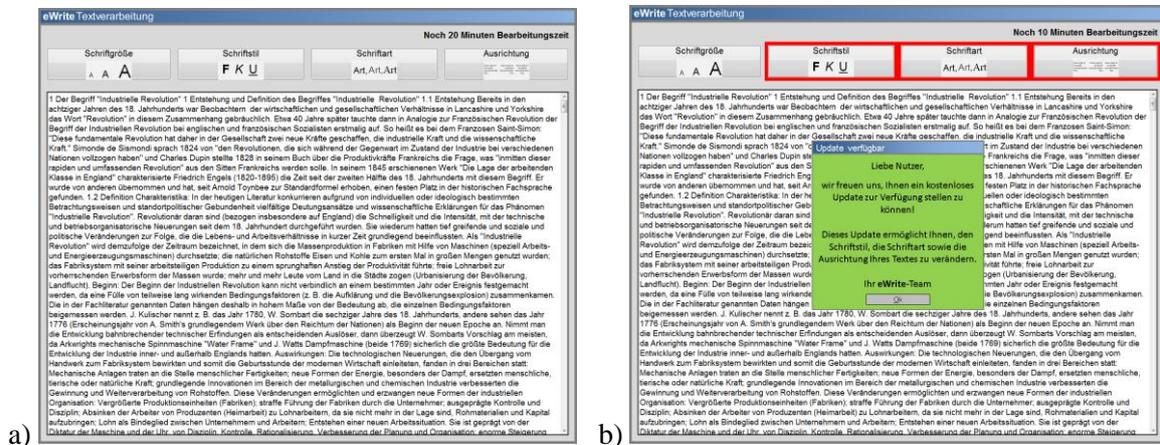


Figure 2. Sample Screenshots of Text Editor – a) Group A (Control Group, no Updates) b) Group B (One Feature Update) after 10 min.<sup>3</sup>

The simplifications in functionality and user interface of our experimental software were made on purpose and followed similar IS studies (e.g., Murray and Häubl 2011). This simplified setting enabled us to establish a controlled environment and unmistakably ascribe any observed changes in the dependent variables (CI, SAT, PU) directly to our experimental treatments. The text which had to be formatted in the experimental task was a historical text about the Industrial Revolution. We consider this type of text, just like the program features, to be a 'neutral', objective one, compared for example to a newspaper article about a current event, which is often an emotive one. Furthermore, the text was

<sup>1</sup> Section 4.4 shows, that this assumption is clearly met in our sample, as the vast majority of our participants indicated a regular use of word-processing programs and reported high levels of competence in the use of word-processing programs.

<sup>2</sup> The scope and importance of the four text formatting functionalities in groups A, B and C were held constant in order to avoid potential confounding effects from the nature of the updates' contents. The functional equivalence of the individual feature updates for the text formatting task were validated in a pre-study with 52 subjects that were recruited using WorkHub, a crowdsourcing platform similar to Amazon Mechanical Turk (Paolacci et al. 2010). The subjects participated online for a small payment. No significant differences emerged among the four text-formatting features (all t<1).

<sup>3</sup> The green notification at the center of the screen informed the participant about the update and its content. In the case of a feature update (groups B and C), it briefly describes the added functionality (e.g. 'This update enables you to change the font style.').

long enough—as the pilot test showed—to keep the participants busy throughout the entire twenty minutes. Thus, we ensured that the participants could not complete their task too quickly and might have had to wait, which could have confounded our results. The participants were also instructed that they did not need to format the entire text, but to focus on the formatting quality, which in turn fostered the comprehensive use of all available program features.

A pilot test with 12 subjects was conducted to ensure that all of the treatments were manipulated according to the experimental design (Perdue and Summers 1986). Specifically, subjects were asked about the functional equivalence of the individual updates, ease of use of the text-formatting editor and comprehensibility of instructions and items. Feedback and suggestions were obtained from participants after they had completed the pre-test experiment. The word-processing program and the questionnaire were accordingly revised for the main test.

## 4.3 Measures

### 4.3.1 Dependent variables

We used validated scales with minor wording changes for all constructs, capturing the core part of the IS continuance model (CI, PU, SAT) (Bhattacharjee 2001). Measures for CI were adapted from Bhattacharjee (2001): CI1. *I intend to continue using eWrite rather than discontinue its use*; CI2. *My intentions are to continue using eWrite than use any alternative means*; CI3. *If I could, I would like to discontinue my use of eWrite* (reverse coded). Measures for PU and SAT were based on Kim and Son (2009): PU1. *Using the features of eWrite enhanced my effectiveness in completing the task*; PU2. *Using the features of eWrite enhanced my productivity in completing the task*; PU3. *Using the features of eWrite improved my performance in completing the task*. SAT1. *I am content with the features provided by the word-processing program eWrite*; SAT2. *I am satisfied with the features provided by the word-processing program eWrite*; SAT3. *What I get from using the features of the word-processing program eWrite meets what I expect 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 7-point-Likert-scales anchored at (1)=strongly disagree and (7)=strongly agree.

### 4.3.2 Control variables and manipulation check

In our study, we control for the impact of usage intensity of word-processing programs in real life, frequency of updates in real life for productivity software/entertainment software and desktop computer/smartphone and computer self-efficacy (Marakas et al. 2007) on CI. Furthermore 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: 1) the instructions in the experiment and 2) the items' formulation. We included these control variables as well as the subjects' demographics as covariates to isolate the effects of the manipulated variables. Finally, we included two questions as manipulation checks: 1) *What was the experimental task? (formatting the entire text or formatting the text as appealingly as possible)* and 2) *How many updates did you receive during the experiment? (no updates, one update, or three updates)*.

## 4.4 Participants, incentives and procedures

90 participants were recruited from the campus of a large public university in Germany. Participants received 5€ for their participation in the lab experiment. In order to align their motivations to properly fulfil the experimental task, 3 x 50€ Amazon vouchers and an iPad Mini were announced as rewards for the four most appealingly edited texts. Five participants were excluded from the sample based on the manipulation checks. We therefore used a sample of 85 subjects in the following analysis. Of the

85 subjects, 31 were females. The participants' age ranged from 18 to 36, with an average value of 23.85 ( $\sigma=3.34$ ). 78 participants were university students, two participants were high school students, five were employees and one was self-employed. Three participants refused to state their occupation. The educational backgrounds of the participants were diverse, including physics, arts, law, management, medical science, biology, geography etc. 51% of the subjects ( $n=44$ ) use word-processing programs from one up to five hours per month, 28% between five and 30 hours ( $n=24$ ), and 14% more than 30 hours per month ( $n=12$ ). 80% rated their computer skills as high to very high. 4% believed they had a rather low competence in using computers.

When participants arrived at the laboratory, they were randomly assigned to a treatment/control group. All instructions and questionnaire items were presented on the computer screen in order to minimize the interaction with the supervisor of the experiment, and thus to reduce error variance to a minimum. They then completed a pre-experimental questionnaire including demographic variables such as gender and age, as well as some control variables such as motivation to process information. In order to ensure comparable initial conditions, participants were further presented with a program tutorial (a program screen similar to that of the actual experimental task). In this tutorial, the initially available features (depending on the experimental condition) were presented and each one was explained in a text bubble. Before they could proceed, all participants had to try out each available feature at least once by formatting a short sample text, ensuring that each participant had understood the program's functionality. On the next two screens, the actual experimental scenario and task, the time available to complete the task, and the results-based incentives were introduced. After having read these instructions, the participants could manually start the actual experimental task by clicking on a button. After having worked 20 minutes on the experimental task, they were automatically redirected to the post-experimental questionnaire, which contained the measurement of all dependent variables (quantitative and qualitative), all remaining control variables, and the manipulation checks. Finally, they were compensated for their participation and debriefed.

## 5 Data Analysis and Results

### 5.1 Control variables and manipulation check

Based on the results of a series of Fisher's exact tests, we could conclude that there was no significant difference across the three experimental conditions in terms of gender ( $p>0.1$ ), age ( $p>0.1$ ), intensity of using word-processing programs ( $p>0.1$ ), attitudes toward productive ( $p>0.1$ ) and entertainment software ( $p>0.1$ ), as well as frequencies of the received updates (desktop/productive:  $p>0.1$ ; desktop/entertainment:  $p>0.1$ ; smartphone/productive:  $p>0.1$ ; smartphone/entertainment:  $p>0.1$ ). Furthermore, based on a series of ANOVA tests, we found no significant differences across the three experimental conditions regarding the task-relevant control variables motivation to process information ( $F=0.05$ ,  $p>0.1$ ), understanding of instructions ( $F=0.07$ ,  $p>0.1$ ) and items' formulations ( $F=0.21$ ,  $p>0.1$ ). It is therefore reasonable to conclude that participants' demographics and task-relevant controls were homogeneous across the three conditions and thus did not confound the effects of our experimental manipulations. Prior to testing the hypotheses, a manipulation check was performed to examine whether our experimental treatments worked as intended. Participants had to state whether they had received 1) one feature update, 2) three feature updates or 3) no update. As mentioned above, in five observations, the wrong condition was ticked, which led to their exclusion from the final sample (three subjects have stated to be in group C, while being in group A and two subjects claimed to be in group B while being in group C). Overall, the results from our manipulation checks suggest that our experimental treatments were successful.

## 5.2 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 for the dependent variables (Levine 2005). Table 1 reports the CFA results regarding convergent validity using SmartPLS, version 2.0 M3 (Chin et al. 2003; Ringle et al. 2005) for the core constructs.<sup>4</sup>

Variables	Number of Indicators	Range of Standardized Factor Loadings*	Cronbachs Alpha	Composite Reliability ( $\rho_c$ )	Average Variance Extracted (AVE)
Continuance Intention (CI)	3	0.783 - 0.900	0.802	0.883	0.716
Satisfaction (SAT)	3	0.898 - 0.928	0.895	0.935	0.827
Perceived Usefulness (PU)	3	0.853 - 0.910	0.845	0.906	0.762

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

Table 1. Results of Confirmatory Factor Analysis for Core Variables.

The constructs were assessed for reliability using Cronbach's alpha (Cronbach 1951). A value of at least 0.70 is suggested to indicate adequate reliability (Nunnally 1994). The alphas for all constructs were well above 0.7. Moreover, the composite reliability of all constructs exceeded 0.70, which is considered the minimum threshold (Hair et al. 2011). Values for AVEs for each construct ranged from 0.709 to 0.889, exceeding the variance due to measurement error for that construct (that is, AVE exceeded 0.50).

## 5.3 Hypotheses testing

To test our hypotheses, we conducted one-way ANOVAs with planned contrast analyses with IBM SPSS Statistics 20. Table 2 presents the mean values of the main constructs for groups A, B and C.

Mean Values for Groups				Mean Differences and Significance Levels	
	No Update (A), n=27	One Feature Update (B), n=30	Three Feature Updates (C), n=28	B-A	C-A
PU	4.51	4.76	4.77	0.25	0.26
SAT	4.71	5.32	5.05	<b>0.61**</b>	0.34
CI	5.66	6.17	5.91	<b>0.51**</b>	0.25

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

Table 2. Mean Differences and Significance Levels.

In hypothesis 1, we claimed that software that receives additional functionality via incremental feature updates will induce higher user CI compared to software that includes all these features right from the first release. The experiment's results indicate that on average, participants' CI in group B (one update) was significantly higher than participants' CI in group A (no updates). Hence, hypothesis 1 is supported. Moreover, hypothesis 2 posits that if delivered too frequently, incremental feature updates do not increase CI any more but rather have an adverse effect on it. As hypothesized, our results show

<sup>4</sup> Computer self-efficacy and other control variables also satisfied the criteria regarding Cronbach's Alpha, AVE and Cross Loadings. Items, scale specifications and results from discriminant validity analysis can be obtained from the authors.

(see Table 2) that a high update frequency (i.e., in our case, three feature updates in the given time-frame; group C) is not perceived more positively than the no update condition (i.e. group A) in terms of CI. Hence, hypothesis 2 is supported. Furthermore in order to test our mediation hypotheses we ran a serial multiple mediator analysis (Hayes 2013) on a sub-sample that included only groups A and B (n=57) with SAT and PU as mediators, while controlling for all direct and indirect paths between the mediators and CI. The results from a bootstrapping analysis in Table 3 reveal that only the indirect effect path (3) from low-frequency feature updates via SAT to CI was significant. Moreover, the direct effect of feature updates on users' CI became insignificant after inclusion of SAT, suggesting full mediation (Hayes 2013). PU, on the contrary, was not influenced by our treatment (i.e., low-frequency feature updates, group B) and was therefore not capable to predict the influence of feature updates on CI. Hence, hypothesis 3 is supported.

Indirect effect paths	<i>Effect z</i>	<i>Boot SE</i>	<i>LLCI</i>	<i>ULCI</i>
(1) Feature Updates → PU → CI	0.015	0.060	-0.049	0.243
(2) Feature Updates → PU → SAT → CI	0.045	0.069	-0.067	0.218
<b>(3) Feature Updates → SAT → CI</b>	<b>0.157</b>	<b>0.120</b>	<b>0.039</b>	<b>0.492</b>

Note: \*We conducted inferential tests for the 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), n=57.

Table 3. Results from Serial Multiple Mediation Analysis, Groups A and B (Bootstrapping Results\* for Indirect Paths).

## 6 Discussion

This study sought to achieve three main objectives: (1) to examine the effects of feature updates on users' intentions to continue using an information system (i.e., whether there is a discernible effect from updates), (2) to investigate crucial boundary conditions (i.e., when there is an effect from updates and when not), and (3) to unravel the explanatory mechanism through which such an effect occurs (i.e., how such an effect from updates operates). To achieve these objectives, we drew on the IS continuance model that is embedded in the expectation-confirmation theory and investigated our hypotheses based on a controlled lab experiment.

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 continuance intention was significantly higher in the update condition (group B) than in the non-update condition (group A). This increase in CI in group B compared to group A can be interpreted as being a somewhat counter-intuitive finding because participants who received feature updates (group B) were objectively disadvantaged compared to the participants who had all functionalities right with the first release (group A): during the limited 20 minutes of the experiment, group B had in sum fewer features per time to accomplish their text-formatting task compared to group A. Despite this objective disadvantage, participants in group B showed significantly higher scores in CI which indicates the presence of a positive, somewhat non-rational effect (Fleischmann et al. 2014) of feature updates on users' CI—a finding that challenges 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). Furthermore, our experiment identifies a crucial boundary condition to the positive effect of feature updates on users' CI: update frequency. In this regard, our results indicate that there is a tipping point for the optimal number of updates in a given time frame. Specifically, a too frequent delivery of feature updates seems to annihilate the mechanism of positive disconfirmation by turning updates into expected events that no longer surprise users. Finally, we could demonstrate that the positive effect of feature updates on CI was fully mediated by user's SAT, emphasizing the role of affect in continuance decisions. The results regarding PU in group B, however, might seem counter-

intuitive at first: Despite the fact that our experimental treatment was a manipulation of the core functionality of the software, we could not observe any significant differences in PU between the treatments. However, a closer analysis reveals that this finding is comprehensible and in line with hypothesis 3. Because participants were asked to state their PU *after* they had completed the experimental task, their evaluation of PU was based on the same set of features, i.e. at this point of time, groups A, B and C had all four and thus the same set of features at their disposal.

## 6.1 Implications for research

The paper makes three main contributions to the literature. First, our main contribution lies in the detection of a positive user reaction to feature updates. Specifically, delivering incremental feature updates in a given timeframe has a stronger and more positive impact on IS users' continuance intentions compared to situations in which the entire feature set is provided at once and right away with the first release. In addition, our findings imply that update frequency is a crucial boundary condition for the identified positive effect of feature updates such that above and beyond a specific tipping point of update frequency, users' CI decreases to a point where they no longer perceive a relative advantage of feature updates compared to non-update versions of the software. Our second main contribution lies in shedding light on the explanatory mechanism behind the identified effect of feature updates on CI. Specifically, we find out that this positive effect primarily works via the affective component (SAT) rather than the cognitive component (PU) of the continuance model. This finding once again emphasizes the still underestimated role of affect in both the IS continuance and IT management literature. Nevertheless, we show that the identified positive effect of feature updates still depends on the presence of PU, so that PU can be seen as necessary and SAT as sufficient condition for its occurrence. Our third contribution consists 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 dynamic perspective by showing how an alterable and malleable information system might influence users' attitudes and behaviors over time. In doing so, we answer several calls of IS scholars (e.g., Jaspersen 2005; Benbasat and Barki 2007 etc.) to consider the granularity of information systems in research studies and how IS evolve over time. As such our study offers a novel complement to the existing IS post-adoption literature by showing that user attitudes and behaviors change over time, as the IT artifact's nature and composition evolves over time through feature updates.

## 6.2 Implications for practice

Our results also have important implications for practice. First, despite the extensive use of feature 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 apparently leaves practitioners puzzled and without guidance. From the results of our experimental study we can conclude that it might be advisable for vendors to distribute software functionality over time via updates, because feature updates can induce a positive affective state of surprise, which, in turn, increases users' CI. For vendors, users with a high CI are a particularly desirable goal because these are the loyal, returning customers who ensure the long term profitability of their businesses in the highly competitive software industry. Moreover, a high CI is particularly important for the increasing share of subscription-based business models in the software industry (Veit et al., 2014). However, while the identified positive effect of feature updates seems to be a useful measure for software vendors to keep their customers satisfied and 'on board', it also needs to be well understood and correctly applied in order to achieve the desired outcomes. The findings of this study reveal that this effect works only if users are really surprised when receiving an update (positive disconfirmation). Too frequently delivered updates seem to cancel out this positive effect, because they are no longer unexpected. Consequently, software vendors can learn from this study's results that there is an optimum corridor for the number of updates delivered in a given time frame that increases users' continuance intentions. They should therefore test where this optimal corridor for their specific software

lies so that updates can be performed repeatedly, while still being perceived as surprising. It should also be noted that vendors should not overdraw holding back functionality. 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. Finally, for vendors, our findings highlight an additional benefit from using a modular architecture for their software. Aside from flexibility in the development and maintenance, a modular architecture also facilitates benefiting from the positive effect of feature updates on customer loyalty, because features that are encapsulated in discrete modules are technically easier to segregate from the software. Moreover, such modules may be delivered in small packages (updates) and can be integrated easily in existing systems that are already being used.

### **6.3 Limitations and future research**

Four limitations of this study are noteworthy and provide avenues for future research. First, in our experiment, we utilized a self-developed, simplified word-processing program with homogeneous and functionally equivalent features to reduce confounding effects and isolate the impact of updates. Nevertheless, to better resemble real-world update practices of software vendors, future studies could investigate more complex programs and deliver more innovative features instead of the basic and well known features that we used. Second, we identified update frequency as a crucial boundary condition to the positive effect of feature updates on users' CI. Future studies are encouraged to specify further possible boundary conditions. For example, they could distinguish between different types of feature updates (e.g. common and extraordinary features), different types of update notifications (e.g. no, unobtrusive or obtrusive notifications), different initial feature endowment, or different competition situations (e.g. many or few competing vendors). Third, the positive effect of feature updates on users' CI was shown to work for productivity software (word-processing). Future research is encouraged to show whether the same effect occurs also for hedonic (e.g. entertainment) software. Because this positive effect of feature updates occurred in software with a low affective quality (word-processing), we are confident that it might have an even stronger impact on CI for entertainment software, which is more emotionally charged. Finally, we conducted a controlled laboratory experiment with the purpose to make a first step towards exploring the causal effect of feature updates on IS continuance, presenting results with a high internal validity. This, however, came at the price of some reasonable but strict assumptions, such as the evaluation of the program taking place at the same time for all users and only after all users had access to the same set of features. Future studies are encouraged to complement the findings of this study by conducting longitudinal field experiments or case studies, in order to advance the external validity of our findings. Also laboratory experiments conducted on longer time spans (e.g. over some weeks) with users' evaluations measured at several points in time could provide additional evidence for the robustness of the positive effect of feature updates on users' CI.

### **6.4 Conclusion**

Feature updates have become a pervasively used instrument of software vendors to maintain, alter and extend their products over time. Despite their prevalence in private and business IT usage contexts, however, feature updates' effects on crucial user reactions in the IS post-adoption context have remained largely unexplored. This study is not only the first to demonstrate that feature updates have the potential to increase users' CI above and beyond a level generated by monolithic software packages that are delivered with the entire feature set at once; it also reveals update frequency as a crucial boundary condition to this phenomenon. Specifically, the identified positive effect on CI is weakened by an increasing update frequency. Furthermore, this study explains the underlying mechanism of why and how feature updates influence users' CI. In summary, it represents an important first step towards better understanding the nature of feature updates and how they affect user reactions over time, and may therefore serve as a springboard for future studies on feature updates in the context of IS post-adoption research.

## References

- Anderson, E. W. and M. W. Sullivan (1993). "The Antecedents and Consequences of Customer Satisfaction for Firms." *Marketing Science* 12 (2), 125-143.
- Benbasat, I. and H. Barki (2007). "Quo vadis, TAM?" *Journal of the Association for Information Systems* 8 (4), 211-218.
- Benbasat, I. and R. W. Zmud (2003). "The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties." *MIS Quarterly* 27 (2), 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), 144-173.
- Benlian, A., and Hess, T. (2011). "The Signaling Role of IT Features in Influencing Trust and Participation in Online Communities." *International Journal of Electronic Commerce* 15 (4), 7-56.
- Bhattacharjee, A. (2001). "Understanding Information Systems Continuance: An Expectation-Confirmation Model." *MIS Quarterly* 25 (3), 351-370.
- Bhattacharjee, A. and A. Barfar (2011). "Information Technology Continuance Research: Current State and Future Directions." *Asia Pacific Journal of Information Systems* 21 (2), 1-18.
- Bhattacharjee, A. and G. Premkumar (2004). "Understanding Changes in Belief and Attitude toward Information Technology Usage: A Theoretical Model and Longitudinal Test." *MIS Quarterly* 28 (2), 229-254.
- Chin, W. W., B. L. Marcolin and P. R. Newsted (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), 189-217.
- Clements, P. and L. Northrop (2002). *Software Product Lines: Practices and Patterns*. 3rd Edition. Boston: Addison-Wesley Reading.
- Cronbach, L. J. (1951). "Coefficient Alpha and the Internal Structure of Tests." *Psychometrika* 16 (3), 297-334.
- Dunn, K. (2004). "Automatic Update Risks: Can Patching Let a Hacker in?" *Network Security* 2004 (7), 5-8.
- Etherington, D. (2013). *Chat Heads Coming To iOS Facebook App Via Update Pushing Out Anytime Now*. URL: <http://techcrunch.com/2013/04/16/chat-heads-coming-to-ios-facebook-app-via-update-pushing-out-anytime-now/> (visited on 05/01/2014).
- Fleischmann, M., Amirpur, M., Benlian, A., & Hess, T. (2014). „Cognitive Biases in Information Systems Research: A Scientometric Analysis.” In: *Proceedings of the 22nd European Conference on Information Systems*. Tel Aviv: AIS.
- Gluck, J., A. Bunt and J. McGrenere (2007). "Matching Attentional Draw with Utility in Interruption." In: *Proceedings of CHI 2007 Conference on Human Factors in Computing Systems*. San Jose: ACM.
- Hair, J. F., C. M. Ringle and M. Sarstedt (2011). "PLS-SEM: Indeed a Silver Bullet." *Journal of Marketing Theory and Practice* 19 (2), 139-151.
- Hayes, A. F. (2013). *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*. New York: Guilford Publications, Inc.
- Helson, H. (1964). *Adaptation-Level Theory: An Experimental and Systematic Approach to Behavior*. New York: Harper and Row.
- Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design Science in Information Systems Research." *MIS Quarterly* 28 (1), 75-105.
- Hodgetts, H. M. and D. M. Jones (2007). "Reminders, Alerts and Pop-ups: The Cost of Computer-Initiated Interruptions." In: *Human-Computer Interaction. Interaction Design and Usability*. Ed. by J.A. Jacko. Berlin and Heidelberg: Springer, 818-826.
- Hong, W., J. Y. Thong, L. C. Chasalow and G. Dhillon (2011). "User Acceptance of Agile Information Systems: A Model and Empirical Test." *Journal of Management Information Systems* 28 (1), 235-272.

- Jasperson, J. S., P. E. Carter and R. W. Zmud (2005). "A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems." *MIS Quarterly* 29 (3), 525-557.
- Jeong, H. J. and K.-N. Kwon (2012). "The Effectiveness of Two Online Persuasion Claims: Limited Product Availability and Product Popularity." *Journal of Promotion Management* 18 (1), 83-99.
- Karahanna, E., D. W. Straub and N. L. Chervany (1999). "Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs." *MIS Quarterly* 23 (2), 183-213.
- Kim, S. S. (2009). "The Integrative Framework of Technology Use: An Extension and Test." *MIS Quarterly* 33 (3), 513-537.
- Kim, S. S. and N. K. Malhotra (2005). "A Longitudinal Model of Continued IS Use: An Integrative View of Four Mechanisms Underlying Postadoption Phenomena." *Management Science* 51 (5), 741-755.
- Kim, S. S. and J.-Y. Son (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), 49-70.
- Levine, T. R. (2005). "Confirmatory Factor Analysis and Scale Validation in Communication Research." *Communication Research Reports* 22 (4), 335-338.
- Locke, E. A. (1976). "The Nature and Causes of Job Satisfaction." In: *Handbook of Industrial and Organizational Psychology*. Ed. by M.D. Dunnette. Chicago: Rand McNally College Pub. Co., 1297-1349.
- Marakas, G. M., R. D. Johnson and P. F. Clay (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), 16-46.
- Mathias, R. (2012). *Keeping tabs on your tabs*. URL: <http://chrome.blogspot.de/2012/05/keeping-tabs-on-your-tabs.html> (visited on 05/01/2014).
- Mens, T. and S. Demeyer (2008). *Software Evolution*. Berlin and Heidelberg: Springer.
- Murray, K. B. and G. Häubl (2011). "Freedom of Choice, Ease of Use, and the Formation of Interface Preferences." *MIS Quarterly* 35 (4), 955-976.
- Nunnally, J. C. (1994). *Psychometric Theory*. 3rd Edition. New York: McGraw-Hill.
- Oliver, R. L. (1980). "A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions." *Journal of Marketing Research* 17 (4), 460-469.
- Oliver, R. L. (1993). "Cognitive, Affective, and Attribute Bases of the Satisfaction Response." *Journal of Consumer Research* 20 (3), 418-430.
- Orlikowski, W. J. and C. S. Iacono (2001). "Research Commentary: Desperately Seeking the "IT" in IT Research - A Call to Theorizing the IT Artifact." *Information Systems Research* 12 (2), 121-134.
- Ortiz de Guinea, A. and M. L. Markus (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), 433-444.
- Ortiz de Guinea, A. and J. Webster (2013). "An Investigation of Information Systems Use Patterns: Technological Events as Triggers, the Effect of Time, and Consequences for Performance." *MIS Quarterly* 37 (4), 1165-1188.
- Paolacci, G., J. Chandler and P. G. Ipeirotis (2010). "Running experiments on Amazon Mechanical Turk." *Judgment and Decision Making* 5 (5), 411-419.
- Perdue, B. C. and J. O. Summers (1986). "Checking the Success of Manipulations in Marketing Experiments." *Journal of Marketing Research* 23 (4), 317-326.
- Popović, M., B. Atlagić and V. Kovačević (2001). "Case study: a maintenance practice used with real-time telecommunications software." *Journal of Software Maintenance and Evolution: Research and Practice* 13 (2), 97-126.
- Ringle, C. M., S. Wende and A. Will (2005). "SmartPLS 2.0 (M3) Beta." Hamburg: University of Hamburg.

- Sandberg, J. and M. Alvesson (2011). "Ways of constructing research questions: gap-spotting or problematization?" *Organization* 18 (1), 23-44.
- Sommerville, I. (2010). *Software Engineering: International Version*. 9th Edition. Boston: Pearson.
- Suri, R. and K. B. Monroe (2003). "The Effects of Time Constraints on Consumers' Judgments of Prices and Products." *Journal of Consumer Research* 30 (1), 92-104.
- Svahnberg, M., T. Gorschek, R. Feldt, R. Torkar, S. B. Saleem and M. U. Shafique (2010). "A Systematic Review on Strategic Release Planning Models." *Information and Software Technology* 52 (3), 237-248.
- Veit, D., E. Clemons, A. Benlian, P. Buxmann, T. Hess, D. Kundisch, J. M. Leimeister, P. Loos and M. Spann (2014). "Business Models - An Information Systems Research Agenda." *Business & Information Systems Engineering* 6 (1), 45-53.
- Zhu, M., D. M. Billeter and J. J. Inman (2012). "The Double-Edged Sword of Signaling Effectiveness: When Salient Cues Curb Postpurchase Consumption." *Journal of Marketing Research* 49 (1), 26-38.