Influence of Software Updates on Hedonic Software Users

Full Paper

Daniel Franzmann
Goethe University Frankfurt
franzmann@wiwi.uni-frankfurt.de

Abstract

Lately, a more and more frequently used method to enhance and maintain software is through software updates. These updates are distributed over the Internet in order to fix bugs, improve base-software, or add new functionalities. This research paper extends theory in the IS topic of post-adoption and examines the effect of software updates on the individual hedonic software user. We develop a digital game and use it in a web-based experiment with 225 participants who are randomly assigned to three distinct groups. We adapt the IS continuance model and assess the effects of a functional software update and a placebo update notification through inter group comparisons. Our study unveils that while a functional software update leads to an increase in perceived enjoyment, satisfaction, continuance intention, and disconfirmation, albeit the placebo update notification does not. Finally, implications for research and practice are discussed.

Keywords

Software updates, hedonic software, IS continuance model, experiment.

Introduction

Software development, publishing, and maintenance has changed over the last years. While earlier software was developed until completion (e.g., final release version) and released on physical-media, today's software development works differently. Agile software development, beta-test programs, digital distribution, and early access marketing models have made software available to the users at an earlier stage of development (Microsoft 2017; Time 2013; Valve 2016). Sometimes, software or hardware is even released so early that core functionalities are not yet functional: for example, the operating system of the PlayStation 4 was not able to process Blu-ray discs after the initial release (Volpe 2013). To solve this discrepancy between the advertised features of a software and the features available at release, developers frequently use the instrument of software updates. Even though updates and bugfixes are nothing new in software development and maintenance, the installation process was often inconvenient and annoying for the users (Amirpur et al. 2015). Today, it is far easier for developers to create and roll out updates that enhance the base-software or add new functionalities without creating much effort on the users' side (Fleischmann, Benlian, et al. 2015; Sommerville 2010).

This study addresses the research call by Fleischmann et al. (2016) and evaluates how users of hedonic entertainment software perceive software updates in a post-adoption setting: How do players of digital games perceive feature improvements after they initially adopted the software? Do software updates lead to an increase in satisfaction or continuance intention? Can players be tricked into thinking that the game has improved by displaying a placebo update notification or is it counterproductive? It is crucial for game publishers and developers to understand the mechanisms behind the users' decision to keep using their software. For example, digital games have high development costs and often use subscription- or freemium[1]-based marketing models to generate revenue over an extended period of time after the initial

[1] The software is provided free of charge, but user have to pay for additional, premium features.

release (Davidovici-Nora 2014; Voigt and Hinz 2016). Therefore, it is essential to keep the players content, prevent them from discontinuing the game or switching to a game of a competitor. One way to achieve this goal could be through a smart, long-term focused software update strategy. Its goal could be to publish new and free content that adds features (e.g., new game maps), changes the balancing of the game, or improves the gameplay. Such strategies can lead to long term success because the players receive new content and are more satisfied. As a result the total number of sales and subscriptions can be increased, and good publicity induced as several examples have shown (Brown and Cairns 2004; GitHyp 2016; PlayStationLifeStyle.net 2016). While the technology acceptance model (TAM), the hedonic-motivation systems adoption model (HMSAM), or the unified theory of acceptance and use of technology (UTAUT) are theories that assess the initial adoption of an information system (IS), the IS continuance model (ISCM) is used for evaluating the user perception and aims to explain continuance intention in the post-adoption stage. Therefore, we use it as a framework for our study.

In this study, we conduct a web-based experiment using a self-developed game with one control and two treatment groups. We aim to answer the following research question by assessing how players of digital games perceive a functional software update versus a placebo update notification:

> *RQ: How do players of a hedonic game perceive feature updates?*

The remainder of this research paper is structured as follows: In Section 2, we provide a general overview of the theoretical background of software updates, digital games as a part of hedonic software, and the ISCM that we use as a framework for our analysis. In Section 3, we develop our hypothesis and key constructs used in this study. In Section 4, we describe our research methodology, the design of the experiment, and the data collection process. In Section 5, we present our analysis and results. In Section 6, we conclude the paper by discussing our results, and outlining our contribution to IS research and practice. Finally, we show practical implications and acknowledge limitations.

Theoretical Background and Related Work

Hedonic Software

In contrast to utilitarian software, hedonic software tries to entertain the user rather than providing an often quantifiable benefit (van der Heijden 2004). The distinction between these two categories of software originates from the consumer behavior literature (Hirschman and Holbrook 1982; Holbrook and Hirschman 1982). IS research has adopted this concept as it regards users as consumers that can choose between a variety of different software. For example, van der Heijden (2004) studies the effect of how users adopt hedonic IS through the use of the TAM. Later, Lowry et al. (2013) use van der Heijden's theory to create the hedonic-motivation system adoption model that can be used as a validated framework for studying the adoption of hedonic-motivation systems.

Gamification, on the other hand, is a relatively new trend in IS research that tries to explain the benefits of using specific hedonic elements in a regular IS (Blohm and Leimeister 2013). These elements should positively influence and enhance the users' experience with the IS in order to create positive short-term and long-term effects. While Bui et al. (2015) assess a gamification framework, they also state that gamification is just "a new label for similar research areas that have existed for decades" (Bui et al. 2015, p. 16).

However, the total number of users or players of digital games, a sub-category of hedonic software, is growing rapidly (Liu et al. 2013). People play them mostly for entertainment purposes respectively, as a favorite pastime activity. Therefore, game developers try to create an immersive gaming experience to create a flow or immersion for the player that tries to keep the user interested (Chen 2007). This immersion is an essential component for the continuance of the game (Brown and Cairns 2004).

Software Updates

Software updates are enhancements or changes of the base-software that are provided by publishers or developers free of charge (Amirpur et al. 2015). They are often based on bug reports, specific user feedback (e.g., forum feedback), or can be part of a long-term software maintenance strategy. The Internet and interconnectivity of devices, such as computers, laptops, or smartphones, has simplified the rollout process of those updates. Earlier, software updates required a manual installation process by the user that was often inconvenient because the device had to be rebooted by the user after the installation or because the installation process was complicated and time-consuming. This has changed over the last years, as today's updates are often rolled out over-the-air and are sometimes even overlooked by the users because the installation runs in the background of the device, respectively operating system (Aghera et al. 2004; Fleischmann et al. 2016; van der Storm 2005). While the software engineering and software maintenance literature has already covered most technical parts, only scarce research has been conducted concerning how users perceive these updates (Fleischmann, Hess, et al. 2015).

Recently, IS research has started to acknowledge the importance of software updates as an instrument to trigger specific user behavior patterns. For example, software updates can be used as a tool to increase user satisfaction and continuance intention. However, those effects depend on the type of update, as several studies outline (Amirpur et al. 2015; Fleischmann et al. 2016). Therefore, a classification of the different categories of software updates is necessary for providing a deeper understanding of how users perceive and experience these updates. Fleischmann et al. (2016) are among the first who assigned updates into two categories: feature updates and non-feature updates. This classification is required as their experimental study outlines that the positive effects of an update only occur for feature updates. They state that "feature updates change the core functionality of software to which they are applied" and name the introduction of the Facebook instant messaging feature in 2013 as an example of such an update (Fleischmann et al. 2016). Contrary, non-feature updates do not change the core functionality and are often improvements or bug fixes. Such updates can be the fixing of security vulnerabilities, general stability improvements, or solving bugs in the base software. To conclude, feature updates are mainly visible to the user, while non-feature are less visible and might not even be noticed by the user.

Expectation-Confirmation Theory and IS Continuance Model

Oliver (1977) develops and validates the expectation-confirmation theory in the context of repurchase decisions of consumer products. Similar to the definition of hedonic software, it also originates from the consumer behavior literature and evaluates to what extent the confirmation or disconfirmation of the expected and the real perceived product performance influence the consumer's repurchase decision (Bhattacherjee 2001). The main construct of the theory is confirmation that has a direct effect on satisfaction, while satisfaction has a direct effect on the repurchase intention (Oliver 1980).

The original expectation-confirmation theory was adapted for the post-exposure context of an IS, commonly known as post-adoption. Bhattacherjee (2001) is the first who empirically validates the theory in the post-adoption context of an online-banking service and extends the original model by including a variable from the TAM: perceived usefulness. Lately, the theory is commonly used for assessing the continuance intention of an IS from the individual user-perspective (Bhattacherjee and Barfar 2011). For this purpose the theory is adapted for the specific case of analysis: Fleischmann et al. (2015) and Amirpur et al. (2015) are the first who used the theory to explain the effects of software updates on the user.

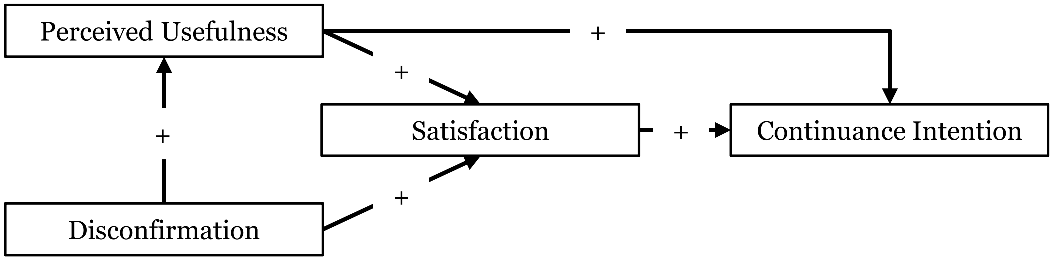

Figure 1. IS Continuance Model by Fleischmann et al. (2016)

Fleischmann et al. (2016) use an experimental setting and focus on utilitarian software to measure differences in satisfaction and continuance intention between control and treatment groups. Overall, the ISCM is constantly extended, improved, and has shown its generalizability for many post-adoption contexts (Bhattacherjee and Barfar 2011; Fleischmann et al. 2016).

Hypothesis Development

To test our hypotheses and answer our research question, we decided to use the ISCM as a framework for evaluating the influence of software updates on the individual user-perspective. We adapted the theory for the context of a hedonic, digital game and identified the following constructs through a literature review and a qualitative, interview-based pre-study: perceived enjoyment, satisfaction, continuance intention, and disconfirmation. Building on this theoretical framework, we came up with two hypotheses per construct. Our study pits the case of an actual feature update (hypothesis a) against the case of a placebo update notification which states that "some optimization and bug fixing" occurred (hypothesis b). In the case of the unexpected feature update, we hypothesize that the player will experience a positive disconfirmation effect because the new feature is perceived as something new and exciting. However, after displaying the placebo update notification the player will expect some sort of change and will experience negative disconfirmation as no change is visible, that leads to a negative effect.

Perceived enjoyment (PE) is an important user belief (Davis et al. 1992). Adoption literature argues that it plays a great role in the overall success of the software and it is defined as the degree to which the user enjoys using the IS (van der Heijden 2004; Thong et al. 2006). While perceived usefulness is necessary for the context of utilitarian software, we do not include it in our context of a hedonic software. As new features provided by a software update extend the content of the hedonic software, the placebo update notifications do not. Therefore, we hypothesize that:

> **H1a**: An unexpected feature update for hedonic software increases perceived enjoyment.
> **H1b**: An unexpected placebo update notification for hedonic software decreases perceived enjoyment.

Satisfaction (SAT) is defined as the "function of expectation and expectancy disconfirmation" (Oliver 1980). It is one of the core constructs of the ISCM in the context of IS by Bhattacherjee (2001). For example, Fleischmann et al. (2016) show that feature updates have a positive influence on satisfaction. Although they used the context of a word-processing program, we hypothesize that:

> **H2a**: An unexpected feature update for hedonic software increases satisfaction.
> **H2b**: An unexpected placebo update notification for hedonic software decreases satisfaction.

Continuance intention (CI) is the main dependent construct in the expectation-confirmation theory. It is the extent to which the user will continue using the IS after the initial adoption (Bhattacherjee 2001; Limayem and Cheung 2008). Like satisfaction is has also been empirically validated in the context of software updates (Amirpur et al. 2015; Fleischmann et al. 2016; Fleischmann, Benlian, et al. 2015). Therefore, for our hedonic software context, we hypothesize that:

> **H3a**: An unexpected feature update for hedonic software increases continuance intention.
> **H3b**: An unexpected placebo update notification for hedonic software decreases continuance intention.

Disconfirmation (DISC) is the discrepancy between the consumers' expectation and the actual performance of a good (Oliver 1977). Bhattacherjee (2001) shows in his first validation of the ISCM that disconfirmation is the core independent variable of the research model. Unexpected feature updates trigger the comparison between the expected and actual performance after the installation of the update. Therefore, it is the key construct for an examination of the post-adoption individual user perspective. We hypothesize that:

> **H4a**: An unexpected feature update for hedonic software increases disconfirmation.
> **H4b**: An unexpected placebo update notification for hedonic software decreases disconfirmation.

Research Method

To test our hypotheses, we used an HTML5 endless runner, platform game that we specifically developed for our experiment in which we evaluated the effect of software feature updates on the players.

Experimental Design and Data Collection

We used an open source online survey tool (Limesurvey v.2.05) that was hosted on a western university server to create our experimental survey. The anonymity of all participants was ensured by using appropriate privacy settings in the survey tool. The survey was distributed via clickworker.com, a crowdsourcing Internet market place similar to Amazon Mechanical Turk (Clickworker.com 2017). Paolacci et al. (2010) and Buhrmester et al. (2011) conclude that these services are a cheap and easy way to conduct surveys or experiments and that the data quality does not differ compared to traditional methods. All participants were randomly distributed to one of three groups and asked for their demographics. Afterwards, the participants of the *control group* played the self-developed HTML5 game that was embedded in the survey. For this group, the game offered the functionalities A (raw version that included the possibility to run and jump) and B (collectable coins and background images) right from the start. Participants in the *treatment group I* played the same game but it lacked the functionality B. *Treatment group II* received the full game with functionalities A and B as did the control group, but differed in the treatment, see below.

After playing the game for exactly 120 seconds our treatment was applied and all groups had to restart the game. The control group continued with the game still consisting of features A and B. Treatment group I got a pop-up notification that informed the participant of a software update that added functionality B, while treatment group II also received a pop-up notification that informed the participants that a software update had been released that consisted of general performance and quality improvements but, in reality, did not change anything. After an additional playtime of 120 seconds, all participants were redirected to the second part of the survey and had to answer the identical questionnaire. For an overview of the experimental setting, see Figure 2.

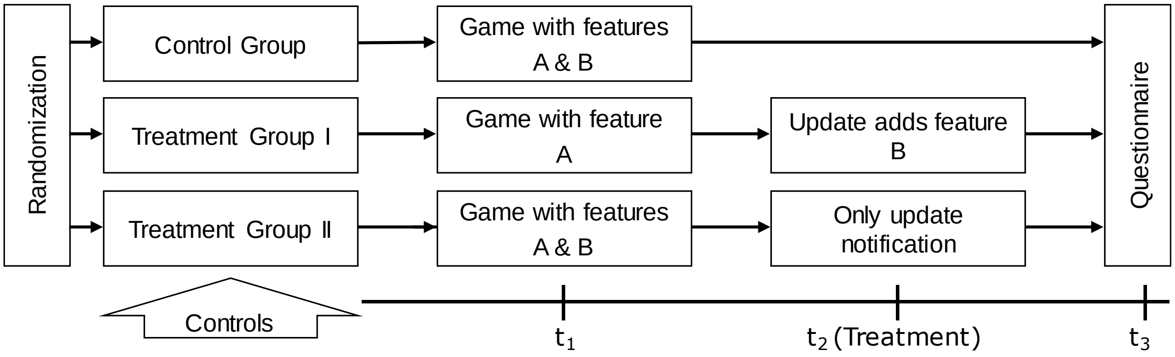

Figure 2. Experimental Design, Groups, and Treatments

Instrument Development, Participants, and Manipulation Check

Our questionnaire contained 29 questions that covered five constructs, demographics, and control questions. The response format was standardized using a 7-point Likert scale ranging from "strongly disagree" – 1 to "strongly agree" – 7. Only the construct satisfaction was measured as a semantic-differential with 6-points, see Table 4 in the Appendix.

The entire experiment was conducted in December 2016 and by the due date, out of 376 subjects who participated in the experiment, 242 completed it successfully. These 242 received a monetary compensation of 0.40$ each. We had to drop 17 responses because they did not answer the control questions correctly. The average duration of the experiment was 9.1 minutes. Table 1 outlines the characteristics, controls, and demographics of all 225 remaining participants. As successful randomization is a key requirement of the whole experiment, we used Person's Chi-squared test as a manipulation check between our three groups. None of our control variables differed significantly between the groups, see Table 1. Therefore, we do not reject the assumption that the three groups originate from the same population and the test confirms the robustness of our experimental design.

Controls		**Count**	**Percentage**	**Pearson's Chi-Square Test**
Gender	Male	109	48.44%	P = 0.372
	Female	116	51.56%	
Age	< 18 years	1	0.44%	P = 0.203
	18 – 24 years	50	22.22%	
	25 – 29 years	69	30.67%	
	30 – 39 years	84	37.33%	
	40 – 49 years	12	5.33%	
	50 – 69 years	8	3.56%	
	> 69 years	1	0.44%	
Marital Status	Single	96	42.31%	P = 0.805
	In a relationship	57	21.79%	
	Married	72	35.90%	
Education	Middle school	2	0.89%	P = 0.469
	Secondary school	5	2.22%	
	High school	100	44.44%	
	Bachelor's degree	90	40.00%	
	Master's degree	2	0.89%	
	PhD	26	11.56%	
Profession	Student	34	15.11%	P = 0.766
	Employed	92	40.89%	
	Self-employed	55	24.44%	
	Out of work	40	17.78%	
	Retired	4	1.78%	
P-value: * significant at p ≤ 0.05; ** significant at p ≤ 0.01; *** significant at p ≤ 0.001				

Table 1. Demographics and Manipulation Check

Data Analysis and Results

In the first section, we tested our measurement model, using SPSS v.23 and AMOS v.23 to conduct a confirmatory factor analysis (CFA). We checked all instruments for convergent and discriminant validity and evaluated the model fit. The second section describes our analysis conducted in Stata v.14 to validate our hypotheses.

Measurement Model Assessment

All our constructs and items stem from IS literature and were adapted for the setting of our experiment. In order to test the reliability of our measurement model, we conducted a CFA with maximum likelihood as the estimation method. The factor loadings of all items scored between 0.83 and 0.98 on their latent constructs. In line with the recommendation of Straub et al. (2004) and Hair et al. (2010), we checked for internal consistency and sufficient reliability. The Cronbach's Alpha of all constructs exceeded the threshold of 0.70 and all composite reliability scores were greater than 0.80 (Bhattacherjee 2012; Hair et al. 2010). The crossloadings of our constructs were rather high, but the variance inflation factors were all less than 10 as commonly suggested (Hair et al. 2010; O'Brien 2007).

Construct	Cronbach's Alpha	Composite Reliability	AVE	PE	SAT	CI	DISC
PE	0.977	0.979	0.903	**0.957**			
SAT	0.929	0.929	0.765	0.747	**0.864**		
CI	0.983	0.984	0.938	0.865	0.761	**0.968**	
DISC	0.965	0.965	0.817	0.925	0.708	0.790	**0.935**
The square root of the average variance extracted (AVE) is displayed on the diagonal of the last four columns.							

Table 2. Results of Confirmatory Factor Analysis for Latent Constructs

Construct validity was assessed by evaluating the convergent validity and the discriminant validity (Straub et al. 2004). While the convergent validity measures the degree of how distinct two constructs are, the discriminant validity describes how the measurement between constructs differ. Research considers the convergent validity sufficient when the AVE is greater than 0.50. Further, discriminant validity is adequate when the square root of the AVE exceeds the correlation among the research constructs, see Table 2 (O'Leary-Kelly and J. Vokurka 1998). The CFA showed an adequate and acceptable model fit with CMIN/DF 1.696, CFI 0.989, GFI 0.925, AGFI 0.888, RMSEA 0.056 and PCLOSE 0.255 (Bagozzi and Yi 1988; Hair et al. 2010; Hu and Bentler 1999).

Hypotheses Testing

In order to answer our research question and test our hypothesis, we compared the mean differences between the three groups. In a first step, we calculated the mean of each observation based on the items tested in the CFA. Second, we used the Shapiro-Wilk test to assess the data distribution. For the constructs that were presumable normally distributed, we used one-sided t-tests for our mean comparison tests, all other constructs were tested using the Wilcoxon-Mann-Whitney test that can be used for nonparametric data (Kothari 2004). The results of the inter group comparisons are shown in Table 3.

Hypotheses	Construct	Control Group	Treatment Group I	Difference	Wilcoxon-Mann-Whitney test / t-test
		Mean	Mean		
H1a – supported	PE	3.95	4.61	+ 0.66	0.0244*
H2a – supported	SAT	3.71	4.11	+ 0.40	0.0368*
H3a – supported	CI	3.36	3.88	+ 0.52	0.0455*
H4a – supported	DISC	4.08	4.73	+ 0.65	0.0321*
		Control Group	**Treatment Group II**	**Difference**	**Wilcoxon-Mann-Whitney test / t-test**
		Mean	**Mean**		
H1b – not supported	PE	3.95	3.78	- 0.17	0.6310
H2b – not supported	SAT	3.71	3.74	+ 0.03	0.7248
H3b – not supported	CI	3.36	3.22	- 0.14	0.3331
H4b – not supported	DISC	4.08	4.03	- 0.05	0.9441
P-value: * significant at p ≤ 0.05; ** significant at p ≤ 0.01; *** significant at p ≤ 0.001					

Table 3. Treatment Effects and Hypotheses Overview

Discussion, Conclusion, and Limitations

Our study utilized an experimental setting in combination with a survey to evaluate how the players of a digital game perceive functional software updates and sole placebo update notifications. We used the

ISCM as a framework that we adapted for the case of hedonic software. The test of our hypotheses confirms that an unexpected functional update leads to a significant increase in perceived enjoyment (H1a), satisfaction (H2a), continuance intention (H3a), and disconfirmation (H4a). This was shown through a mean comparison between the control group and treatment group I. In line with the results of Fleischmann et al. (2016) and Amirpur et al. (2015), we demonstrate that an unexpected addition of a feature leads to a positive effect in the user perception of the hedonic software. However, in contrast to our hypotheses, displaying a placebo update notification – treatment group II – does not significantly decrease perceived enjoyment (H1b), satisfaction (H2b), continuance intention (H3b), or disconfirmation (H4b). None of the constructs' means differed significantly compared to the constructs' means of the control group; therefore; no hypotheses of treatment group II were supported. This is an interesting finding, because the players did not seem to experience a negative disconfirmation and thus, no effects on the other three constructs were measured.

This paper has theoretical and practical contributions. First, our extension of the ISCM in the context of unexpected updates for hedonic software contributes to the post-adoption research stream. The results of our analysis show that an unexpected feature update increases the constructs of perceived enjoyment, satisfaction, continuance intention, and disconfirmation. Second, as a practical contribution, we state that users of hedonic software are aware of the features a game offers. Players react positively when new, unexpected features enhance the game. Publishers and developers can use the results of our study to deepen their knowledge about feature updates. For example, games that suffer from problems, such as an unhappy player base or a decreasing number in overall players, could be targeted with an unexpected feature update that improves the players' perception. However, it is not a sufficient strategy to display fake update notifications without any actual enhancements as it leads to neither benefits nor drawbacks and is ethically highly questionable.

The findings of our study are consistent with the IS literature assessing software updates in the post-adoption field. However, we acknowledge several limitations. First, this paper relied on a controlled web-based experiment using a self-developed game offering the advantage of high internal validity. Future studies should conduct field experiments to increase the external validity and examine different types of updates, for example feature updates, bug fixing updates, or security updates. Second, all participants were American or British citizens, thus cultural differences could be an issue. Finally, the results could vary depending on the user's individual gaming experience. Future research should examine how the individual experience with the use of hedonic software influences the effects of software updates by using structural equation modeling. Such a method could also help to validate and examine the inter-construct relationships of our proposed research model.

REFERENCES

Aghera, P., Bok, A., Chintada, S., Rao, S., and Rinaldi, A. 2004. "Over the air mobile device software management," *Patent Application Publication*, Google Patents.

Amirpur, M., Fleischmann, M., Benlian, A., and Hess, T. 2015. "Keeping Software Users on Board — Increasing Continuance Intention Through Incremental Feature Updates," in *ECIS 2015 Proceedings*, pp. 1–16.

Bagozzi, R. P., and Yi, Y. 1988. "On the evaluation of structural equation models," *Journal of the Academy of Marketing Science* (16:1), pp. 74–94.

Bhattacherjee, A. 2001. "Understanding Information Systems Continuance: an Expectation Confirmation Model," *MIS Quarterly* (25:3), pp. 351–370.

Bhattacherjee, A. 2012. *Social Science Research: principles, methods, and practices Textbooks collection* (Vol. 9).

Bhattacherjee, 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.

Blohm, I., and Leimeister, J. M. 2013. "Gamification: Design of IT-based enhancing services for motivational support and behavioral change," *Business and Information Systems Engineering* (5:4), pp. 275–278.

Brown, E., and Cairns, P. 2004. "A grounded investigation of game immersion," *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Au), pp. 1297–1300.

Buhrmester, M., Kwang, T., and Gosling, S. D. 2011. “Amazon’s Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?,” *Perspectives on Psychological Science* (6:1), pp. 3–5.

Bui, A., Veit, D., and Webster, J. 2015. “Gamification – A Novel Phenomenon or a New Wrapping for Existing Concepts?,” in *ICIS 2015 Proceedings*, pp. 1–21.

Chen, J. 2007. “Flow in games (and everything else),” *Communications of the ACM* (50:4), p. 31.

Clickworker.com. 2017. “Clickworker about us,” (at https://www.clickworker.com/about-us/; retrieved January 5, 2017).

Davidovici-Nora, M. 2014. “Paid and Free Digital Business Models Innovations in the Video Game Industry,” *Communications & Strategies* (94), pp. 83–102.

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. 1992. “Extrinsic and Intrinsic Motivation to Use Computers in the Workplace,” *Journal of Applied Social Psychology* (22:14), pp. 1111–1132.

Fleischmann, M., Amirpur, M., Grupp, T., Benlian, A., and Hess, T. 2016. “The role of software updates in information systems continuance - An experimental study from a user perspective,” *Decision Support Systems* (83), Elsevier B.V., pp. 83–96.

Fleischmann, M., Benlian, A., Hess, T., München, L., Grupp, T., Amirpur, M., and Benlian, A. 2015. “When Updates Make a User Stick: Software Feature Updates and their Differential Effects on Users ’ Continuance Intentions,” in *ICIS 2015 Proceedings*, pp. 1–20.

Fleischmann, M., Hess, T., and Benlian, A. 2015. “Gains and Losses in Functionality – An Experimental Investigation of the Effect of Software Updates on Users ’ Continuance Intentions,” in *ICIS 2015 Proceedings*, pp. 1–21.

GitHyp. 2016. “Ubisoft Claiming The Division’s Player Counts Are Back To Launch Numbers,” (at http://www.githyp.com/ubisoft-claiming-the-divisions-player-counts-are-back-to-launch-numbers/; retrieved January 10, 2017).

Hair, J. F., Black, W. C. ., Babin, B. J., and Anderson, R. E. 2010. *Multivariate Data Analysis*, Prentice Hall.

van der Heijden, H. 2004. “User Acceptance of Hedonic Information Systems,” *MIS Quarterly* (28:4), Management Information Systems Research Center, University of Minnesota, pp. 695–704.

Hirschman, E. C., and Holbrook, M. B. 1982. “Hedonic consumption: Emerging concepts, methods and propositions,” *Journal of Marketing* (46:3), pp. 92–101.

Holbrook, M. B., and Hirschman, E. C. 1982. “The experiential aspects of consumption: Consumer fantasies, feelings, and fun,” *Journal of Consumer Research* (9:2), pp. 132–140.

Hong, W., Thong, J. Y. L., 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.

Hu, L., and Bentler, P. M. 1999. “Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives,” *Structural Equation Modeling: A Multidisciplinary Journal* (6:1), pp. 1–55.

Kothari, C. R. 2004. *Research methodology: methods and techniques New Age International.*

Limayem, M., and Cheung, C. M. K. 2008. “Understanding information systems continuance: The case of Internet-based learning technologies,” *Information and Management* (45:4), pp. 227–232.

Liu, D., Li, X., and Santhanam, R. 2013. “Digital Games and Beyond: What Happens When Players Compete,” *MIS Quarterly* (37:1), pp. 111–124.

Lowry, P. B., Gaskin, J. E., Twyman, N. W., Hammer, B., and Roberts, T. L. 2013. “Taking ‘Fun and Games’ Seriously: Proposing the Hedonic-Motivation System Adoption Model (HMSAM),” *Journal of the Association for Information Systems* (14:11), pp. 617–671.

Microsoft. 2017. “Windows Insider,” (at https://insider.windows.com/; retrieved January 10, 2017).

O’Brien, R. M. 2007. “A caution regarding rules of thumb for variance inflation factors,” *Quality and Quantity* (41:5), pp. 673–690.

O’Leary-Kelly, S. W., and J. Vokurka, R. 1998. “The empirical assessment of construct validity,” *Journal of Operations Management* (16:4), pp. 387–405.

Oliver, R. L. 1977. “Effect of expectation and disconfirmation on postexposure product evaluations: An alternative interpretation.,” *Journal of applied psychology* (62:4), American Psychological Association, p. 480.

Oliver, R. L. 1980. “A cognitive model of the antecedents and consequences of satisfaction decisions,” *Journal of Marketing Research* (17:4), pp. 460–470.

Paolacci, G., Chandler, J., and Ipeirotis, P. 2010. “Running experiments on amazon mechanical turk,” *Judgment and Decision making* (5:5), pp. 411–419.

PlayStationLifeStyle.net. 2016. "Rainbox Six Siege's Daily Player Count Is Higher Now Than It's Ever Been," (at http://www.playstationlifestyle.net/2016/08/11/rainbow-six-sieges-daily-player-count-higher-now-ever/; retrieved January 10, 2017).

Sommerville, I. 2010. *Software Engineering Software Engineering*.

Spreng, R. A., MacKenzie, S. B., and Olshavsky, R. W. 1996. "A Reexamination of the Determinants of Consumer Satisfaction," *Journal of Marketing* (60:3), p. 15.

van der Storm, T. 2005. "Continuous release and upgrade of component-based software," *Proceedings of the 12th international workshop on Software configuration management - SCM '05*, pp. 43–57.

Straub, D., Boudreau, M.-C., and Gefen, D. 2004. "Validation Guidelines for Is Positivist," *Communications of the Association for Information Systems* (13:24), pp. 380–427.

Thong, J. Y. L., Hong, S. J., and Tam, K. Y. 2006. "The effects of post-adoption beliefs on the expectation-confirmation model for information technology continuance," *International Journal of Human Computer Studies* (64:9), pp. 799–810.

Time. 2013. "Should You Subscribe to Microsoft Office 365?," (at http://techland.time.com/2013/01/30/office-365/; retrieved January 1, 2016).

Turel, O., Serenko, A., and Giles, P. 2011. "Integrating Technology Addiction and Use: An Empirical Investigation of Online Auction Users.," *MIS Quarterly* (35:4), pp. 1043–1061.

Valve. 2016. "Early Access Titles," (at http://store.steampowered.com/genre/Early Access/; retrieved September 11, 2016).

Voigt, S., and Hinz, O. 2016. "Making Digital Freemium Business Models a Success: Predicting Customers' Lifetime Value via Initial Purchase Information," *Business and Information Systems Engineering* (58:2), Springer Fachmedien Wiesbaden, pp. 107–118.

Volpe, J. 2013. "Sony's PlayStation 4 will require a Day One update, mobile app to be released pre-launch," *AOL Inc* (at https://www.engadget.com/2013/10/25/sonys-playstation-4-will-require-a-day-one-update-mobile-app-t/; retrieved December 27, 2016).

Appendix

Construct	Code	Item	Reference
Perceived Enjoyment	PE1	Playing the game was enjoyable.	van der Heijden (2004); Turel et al. (2011)
	PE2	Playing the game was pleasurable.	
	PE3	Playing the game was fun.	
	PE4	Playing the game was exciting.	
Satisfaction	Please rate your satisfaction with the game.		Bhattacherjee (2001); Hong et al. (2011); Spreng et al. (1996)
	Sat1	Very dissatisfied – very satisfied	
	Sat2	Very displeased – very pleased	
	Sat3	Very frustrated – very content	
	Sat4	Absolutely terrible – absolutely delighted	
Continuance Intention	CI1	I intend to continue using the game rather than discontinue its use.	Bhattacherjee (2001); Fleischmann et al. (2016)
	CI2	If I could, I would like to continue my use of the game.	
	CI3	Overall, I would intend to continue playing the game.	
	CI4	In the near future, I plan to keep playing the game.	
Disconfirmation	Disc1	My experience with using the game was better than what I expected.	Bhattacherjee (2001); Fleischmann et al. (2016)
	Disc2	The level of fun provided by the game was better than what I expected.	
	Disc3	The functionality provided by the game was better than what I expected.	
	Disc4	The implementation of the game was better than what I expected.	

Table 4. Items and References