

Revealing the Vicious Circle of Disengaged User Acceptance: A SaaS Provider's Perspective

Completed Research Paper

Sebastian Loss

IT University of Copenhagen
Rued Langgaards Vej 7, 2300
Copenhagen S, Denmark
sebastian.loss.1989@gmail.com

Raffaele Fabio Ciriello

IT University of Copenhagen
Rued Langgaards Vej 7, 2300
Copenhagen S, Denmark
raci@itu.dk

Jürgen Cito

Massachusetts Institute of Technology
32 Vassar Street, Cambridge, MA
jcito@mit.edu

Abstract

User acceptance tests (UAT) are an integral part of many different software engineering methodologies. In this paper, we examine the influence of UATs on the relationship between users and Software-as-a-Service (SaaS) applications, which are continuously delivered rather than rolled out during a one-off signoff process. Based on an exploratory qualitative field study at a multinational SaaS provider in Denmark, we show that UATs often address the wrong problem in that positive user acceptance may actually indicate a negative user experience. Hence, SaaS providers should be careful not to rest on what we term disengaged user acceptance. Instead, we outline an approach that purposefully queries users for ambivalent emotions that evoke constructive criticism, in order to facilitate a discourse that favors the continuous innovation of a SaaS system. We discuss theoretical and practical implications of our approach for the study of user engagement in testing SaaS applications.

Keywords: *user acceptance, testing, software-as-a-service, cloud computing, constructive criticism, continuous delivery, technology acceptance, disengagement*

Introduction

Motivation: Software-as-a-Service (SaaS) providers face constant pressures to innovate their product offering due to increased competition, ever-shortening innovation life cycles, and low switching costs (Ciriello et al., 2018). Hence, SaaS providers often see a user acceptance test (UAT) as a key part of their beta testing procedures in order to increase efficiency and quality of a new release to be rolled out (Humble and Farley, 2011), and to validate incremental product innovations against specified acceptance criteria (ISO, 2010). At first glance, it seems plausible to perform a UAT, as users should have a voice in what will be released.

However, prior research has found that feedback provided via structured elicitation techniques, such as UATs, does not necessarily correspond to the actual experience of a user. For instance, a global case study of customer service interactions among more than 100 companies and 125,000 customers found that 20% of the customers who claimed to be satisfied also expressed intentions to terminate the contract, while 28% of the dissatisfied customers had intended to stay loyal with their provider (Dixon et al., 2013). When positive feedback may correspond a negative user experience, it falls into question whether UATs are actually appropriate in the context of SaaS.

Problematization: In contrast to providers of custom-made, contractually ordered on-premise software products, SaaS providers are technically able to release new features fast and frequently, incorporating continuous delivery approaches (Humble and Farley, 2011). Thus, SaaS providers typically set acceptance criteria themselves while allowing users to experiment with a pre-release to discover problems and to provide further input (Sommerville, 2007). While SaaS providers exercise full authority over the application, they are also highly dependent on their actively subscribed user base. SaaS users tend to churn fast in the face of limited customizability of products, and also due to low switching cost (Chou and Chiang, 2013). For SaaS providers, being fully responsible for the system's maintenance and evolution, this means that they have to keep close contact with their subscribers and monitor their usage, in order to ensure that the provided services and system enhancements will meet actual user needs (Sommerville, 2007). Selective user testing that typically probes a limited pool of users within large scale SaaS systems (Humble and Farley, 2011), vast possibilities of live site monitoring within the cloud architecture (Gabbard et al., 1999), and a need for speed in delivering new services to stay ahead of competition may further limit the appropriateness of a traditional UAT approach in a SaaS context. While it is important for SaaS providers to obtain an appropriate understanding of their users, it falls into question if UATs are the right approach for doing so. Having a deeper understanding of the practices and effects of SaaS providers' UAT usage would therefore be desirable to help SaaS providers elicit appropriate user feedback.

Research Scope: This study focuses on the usage and effects of UATs from a SaaS providers perspective. Our aim is to find out whether formal UAT practices of verifying a software build against a fixed set of acceptance criteria can yield valuable user feedback in a SaaS context. Specifically, we ask the research question:

RQ: *How do SaaS providers use user acceptance tests, and what are the effects of doing so?*

Contribution: We develop a conceptual model of what we term the *vicious circle of disengaged user acceptance*, referring to the adverse tendency of UATs to let users passively conform with formal testing procedures while hiding their actual emotional connection to the SaaS application and its provider. The conceptual model is based on related literature on UATs and empirical data gathered from an exploratory qualitative field study at *CloudEnhancer*, a multinational Danish SaaS provider. Our data shows that user acceptance testing, when conducted in a SaaS context, can create a vicious circle of *disengaged user acceptance*, during which testers and product managers within the SaaS provider make decisions based on incomplete and inaccurate information gathered from a UAT, leading users to detach themselves ever more from the SaaS application. This adverse effect will further be exacerbated when the SaaS provider conducts further UATs to address the problem. Based on these insights, and drawing on related literature, we argue that SaaS providers can break the vicious circle of disengaged user acceptance by asking more uncomfortable questions that cultivate constructive criticism and facilitate ambivalent affective responses.

User Acceptance Testing: Foundations and Current Practice

This section reviews the literature on UATs in different software paradigms. A UAT can be defined as a “formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a product or product component” (Khannur, 2014, p. 91). UATs usually consist of a set of processes to validate the product or its component(s) against clearly defined acceptance criteria, after initial development and release tests have been performed (Sommerville, 2007).

UATs, which originated from plan-driven methodologies and have been transferred to agile, prototyping, and SaaS methodologies, may contradict the characteristics of continuous software evolution. Recent research on Platform-as-a-Service (PaaS) has found that PaaS applications allow for self-organization and continuous feedback, enabling organizations to transition to agile and continuous software development practices (Krancher et al., 2018). It thus falls into question if formal UAT practices are still a valuable

measure for user feedback during beta testing phases, especially in the context of SaaS with its possibilities for continuous delivery and increased feedback communication throughout the entire software development life cycle. While SaaS makes it technically possible to release rapidly, the human factor may be a bottleneck when SaaS providers try to enforce temporary stabilization through linear testing processes (Humphreys, 2005). Could UATs be simply a vestige from plan-driven software development? Could a certain degree of disagreement eventually be beneficial for product innovation?

User Acceptance Tests in Plan-Driven Software Development

UATs originate from plan-driven software development methodologies, such as the waterfall model, V-Model, incremental build model, or spiral model. Here, the result of each development phase is “one or more documents that are approved (‘signed off’)” (Sommerville, 2007, p. 31). Thus, UATs are performed after the delivery of a product-batch, testing a predefined set of objectives and requirements derived from those. The underlying assumption is that requirements are stable and can be formalized by or with a user to develop a software product against clearly defined specifications (Scacchi, 2002). However, in practice, consecutive stages feed information into each other (e.g. when issues with specified requirements are identified during the design phase), leading to significant rework or ignoring processual learnings, which can result in the testing of outdated acceptance criteria (Sommerville, 2007).

User Acceptance Tests in Agile Software Development

In agile software development methodologies, such as Scrum, Extreme Programming, or Lean Software Development, requirement elicitation and testing activities are intertwined with design and development activities via UATs (Sommerville, 2007). For instance, test-driven development processes start with defining tests before developing code. This approach requires that a customer representative is part of the development team, developing tests along the elicitation of requirements in so called user stories (Beck and Gamma, 2000). These user stories are then broken down in incremental tasks, for which UATs are defined between developers, testers, and users. Only then, developers start to validate new code via the previously defined tests to ensure it meets the user’s expectations (Sommerville, 2007). Hence, this is also called acceptance test-driven development, where “the distinction between user acceptance tests and system tests is practically eliminated” (Pugh, 2010, p. 3).

In agile development, UATs take place as part of design and development phases and are merged with other tests, but the underlying assumptions are similar to those in plan-driven development: The user, potentially through the help of a service provider, is able to clearly define requirements at a given time. If validated at a later point in time, those requirements are considered accepted. However, in practice requirements can change for a number of reasons, such as emerging technologies, changing market conditions or new business opportunities, to name a few. Thus, although agile methodologies embrace changing requirements by breaking development down into small increments for rapid delivery, a UAT does not guarantee that what has been validated - how a requirement was defined - is actually what the user (still) wants.

User Acceptance Tests in Software Prototyping

Software prototypes can be used to improve consistency and mutual understanding of requirements in a user-developer interaction, which can potentially lead to a better fit between defined acceptance criteria and actual user requirements. Because they provide a tangible preview of the prospective systems to users, prototypes can function as a basis for further requirement specification (Kordon, 2002). Thereby, acceptance criteria of an envisioned system can already be retrieved from users at an early stage of the development (Davis and Venkatesh, 2004). Thus, software prototypes can serve as a tool to generate, refine, verify, and finalize requirements, as the prototype itself contains a wide range of implicit knowledge (Ciriello et al., 2017). For instance, in participatory design methodologies, such as human-centered design, design thinking, and user co-creation, UATs are seen as a synchronization in the reflection and building process between designers and users (INUSE Research Group, 2017).

Here, the underlying assumption is that the design of software systems is a self-iterative process, as the result of any design activity becomes the starting-point for another, where learning and change result from a dialogue between users, developers, and testers (Floyd, 1992). This view recognizes that software development is an evolutionary process, where user acceptance can only be tentatively negotiated and has

to be continuously (re-)constructed (Lehman et al., 1997). A prototype changes the foundation on which new specifications, ideas and change requests appear, constructing a new reality domain to be re-evaluated and re-verified once again in consecutive development phases. Requirements are then based on individual perspectives and values, enacted in interaction with peers and the environment. Hence, acceptance criteria are ever-evolving and can never be final. In this view, a software system can only remain acceptable in its functionality through the continuous evolution of acceptance criteria. User acceptance testing is therefore considered as a boundary-crossing activity, taking place through the deliberate creation of knowledge spaces for mutual learning activities in the continuous software design (Suchman, 2002). Here, users do not only provide feedback on a product as part of a formal UAT, but during actual usage of a prototype or the software systems in their day-to-day activities.

User Acceptance Tests in Software-as-a-Service

With the rise of cloud computing came highly virtualized shared knowledge spaces that allow service providers to access, transmit, and utilize user feedback. One example is the SaaS licensing and delivery model, in which the customer subscribes to a centrally hosted software from the SaaS provider. As SaaS providers are able to handle a large volume of client applications over which they have full control, they typically also face more diverse customer expectations for new features, higher demands for continuous service improvement, and lower switching costs compared to on-premise software (Chou and Chiang, 2013).

In contrast to custom-made, contractually ordered software products, where customers determine whether or not a system shall be deployed to their production environment in the course of a UAT, the customer does not have this decision-making authority in SaaS applications. Here, the SaaS provider typically sets acceptance criteria while allowing users to experiment with a pre-release to discover problems and to provide further input, with UATs being either guided or unguided (Sommerville, 2007).

With the advent of SaaS and according licensing models emerge also new practices allowing for increased collaboration, automation, measurement, and knowledge-sharing between providers and their users. For instance, continuous delivery is a practice that enables frequent and continuous deployment by taking a software build to production environment through automated and manual tests, such as exploratory testing, usability testing, and user acceptance testing (Humble and Farley, 2011).

However, the evaluation of a SaaS application does not end at a formative level throughout the development phase but continues with summative evaluations after deployment (Gabbard et al., 1999). Hence, user acceptance testing in SaaS is not only an act of conservation of what has been developed, but an evolutionary practice of measurements, feedback handling, and continuous enhancements to further innovate a software system (Dittrich, 2014).

Research Method

Our aim is to understand how SaaS providers use UATs to elicit user feedback, along with the subjective meanings people assign to this practice. Thus, our research approach is guided by the principles of interpretive field research, which aims to understand the deeper structure of a phenomenon from a participant's perspective (Orlikowski and Baroudi, 1991).

Sampling and Relationship with the Research Site

Based on the above-summarized literature, our selection criteria were that the case organization needs to be 1) a SaaS provider who 2) performs UATs with 3) a substantial and diverse user base. We chose a single case study to be able to obtain and maintain long-term access to the company and its user base, where we could study the phenomenon in depth (Walsham, 2006). This led us to engage in a close research relationship with the following company (we use a pseudonym, as per the company's wish):

CloudEnhancer, headquartered in Denmark, is a multinational SaaS provider currently operating in 18 markets. CloudEnhancer's cloud-based software products automate the process of identifying errors and problems within content management systems. The company's web platform constitutes a collection of integrated tools for controlling online content and compliance regulations, improving web accessibility and search indexing, as well as monitoring website performance and analytics data. The company currently

employs more than 500 employees in several offices worldwide, while all development for their partly customized, off-the-shelf software products is performed in-house at the Danish office site. Their actively subscribed business-to-business client base includes over 6000 organizations worldwide.

Over the course of the research project, the first author of this paper became deeply involved within the practices at the research site. By becoming an active member of the case organization, he had access to confidential and sensitive information and also had the possibility to observe day-to-day practices in action (Walsham, 1995). Additionally, he was able to obtain access to documentation of past situations and projects, schedule on-the-fly meetings with participants, familiarize himself with the company-internal jargon, and build relationships through informal networking, through which he was able to also engage with CloudEnhancer's client base.

Data Collection

Our primary data sources were interviews, as these allow to access the different perceptions and interpretations of participants (Klein and Myers, 1999), complemented by observations and archival documents. We purposefully accepted some vagueness at the outset of our exploration, selecting the next data slices based on what was necessary to further elaborate on emerging concepts until a coherent picture emerged (Walsham, 2006, 1995).

The first author of this paper was the primary responsible for collecting all data to ensure consistency in the process (Walsham, 1995). In all, he conducted 11 expert interviews, ranging from 28 to 53 minutes (384 minutes in total), which he transcribed ad verbatim to capture a full description of what was said. He also conducted 9 participant observations, which allowed us to complement subjective interview statements with observations of actual practices. Data collection proceeded between February and May 2018 and started out by joining weekly release management meetings to explore relevant concepts and make connections with relevant stakeholders. The first interview was with the *Director of Release Management*, whose affiliated team plays an intermediary role in the handover between development and operations, and who also has extensive experience and overview on testing procedures and participants' practices due to joining the company in the year it was founded.

From there, we selected further participants based on the principle of theoretical sampling, which aims “to understand the nature and dimensions of emerging conceptualizations further, usually by sampling data in a way that varies a particular set of dimensions that emerge from prior data analysis” (Birks et al., 2013, p. 3). In a first exploratory phase, this led us to interviewing a preliminary group of four participants over the course of eight weeks, including an *Agile Coach*, a *Product Operations Manager*, a *Product Manager*, and a *Technical Writer*. These participants were closely affiliated with the releases of new features and regularly involved in user acceptance testing practices throughout beta stages. The topics discussed with them were not bound to any specific project but more conceptual, focusing on areas that have emerged in the first exploratory interview, such as acceptance criteria, collaboration, criticism, and transparency. In these interviews, we applied laddering interview techniques, which aim to elicit “core beliefs and values that drive behavior in organizations” (Schultze and Avital, 2011, p. 13). This helped us to capture the content and structure of personal constructs in order to identify the subjective sensemaking of current practices (Schultze and Avital, 2011). We prompted participants with the above-identified themes to ensure topical focus while also allowing them to freely express views outside our emerging conceptual framework (Walsham, 1995).

Several weeks into the research, we became aware that the *Customizable Dashboards* project, which aims at enabling users to create their own customized reporting interface within the CloudEnhancer platform, was currently preparing for UATs. The project met our research objectives well, as it had a fairly large pool of cross-functional testers and users involved. Subsequently, we dove into this project by interviewing seven people involved in the project's UAT practices during the beta phase, being a *Customer Success Manager*, an *Account Executive*, a *Partner Success Manager*, a *Customer Experience Operations Manager*, the responsible *Product Manager* together with a *UX/UI Designer*, as well as one *Customer*. Thereby, we could analyze the UAT practices associated with this project in depth, ensuring that memories of the testing experience were freshly present in the participants' minds.

Data Analysis and Interpretation

We inductively developed emerging interpretations of the research domain by collecting, analyzing, and reflecting iteratively on the data (Birks et al., 2013). We recorded and fully transcribed all interviews ad verbatim for detailed analysis, adapting them only slightly to make them anonymous via aliases (Birks et al., 2013). Next, we imported the transcripts into the qualitative data analysis software MAXQDA to initiate a process of data-driven coding. Codes, being descriptive abstractions of intersubjective realities, can originate from existing theories, research goals as well as data. Relying on grounded theory principles, our coding was primarily data-driven while still comparing the emerging codes with existing theories that appeared plausible along the way (DeCuir-Gunby et al., 2011). We added, transformed, and reconceptualized codes through iterative processes of open and axial coding (Birks et al., 2013), leading to the codebook in table 1, consisting of 9 dominant themes and 60 sub-themes connected to them.

| Table 1: Overview of Main Codes. | | |
|---|---|--|
| Dominant Theme | Explanation | Examples of sub-themes |
| Testing Roles | Roles and responsibilities throughout UAT processes. | Clients, development team, users, user representatives |
| Testing Procedures | SaaS provider's intention for conducting UATs and execution of those. | Test format, UAT evolution, selection of testers |
| Feedback Elicitation | Gathering of input and transmission of information from conducted UATs. | Communication, information loss, clarification, transparency |
| Culture | Perspectives on the corporate culture and stakeholder relationships. | Agility, practices, expertise |
| Acceptance vs. Use | Discrepancy between testing feedback and actual system usage. | Engagement, satisfaction, usability |
| Criticism | Handling of users' critique during both formative and summative evaluations on the part of the SaaS provider. | Rejection, resistance, disengagement |
| Feedback Processing | Subsequent processing of collected input from stakeholders. | Validation, prioritization, measurement difficulties |
| User Involvement | Degrees of user inclusion and authority within UATs. | Inclusion, co-design, closure |
| Emotions | Intrinsic feelings during testing and usage of a software system. | Instinct, intuition, feelings |

Table 1: Overview of Main Codes.

Figure 1 provides an overview of our research approach and the overall analytical process that led us to our findings. Based on our collected data primarily consisting of interviews and complementary observations, we identified possible categories (i.e. candidates for dominant themes) and their properties (i.e. candidates for sub-themes) through open coding techniques. Throughout our data analysis, we continuously refined our emerging concepts in an iterative process of collecting new data and integrating it into our previous conceptualizations. By means of axial coding techniques, through which we established connections between emerging themes, we were able to construct a comprehensive scheme of CloudEnhancer's practices and motivations in relation to conducting UATs. This led us to our findings, in which we describe how and during which phases presumed acceptance arises in user acceptance testing and how disengagement then continues during live site usage. This vicious circle of disengaged user acceptance shows how superficial user feedback is being transmitted, bearing the risk of users gradually detaching themselves from the continuous design process of a SaaS system while leaving the SaaS provider under the impression of doing the right thing. Ultimately, this can hinder SaaS providers to make user-centered system improvements.

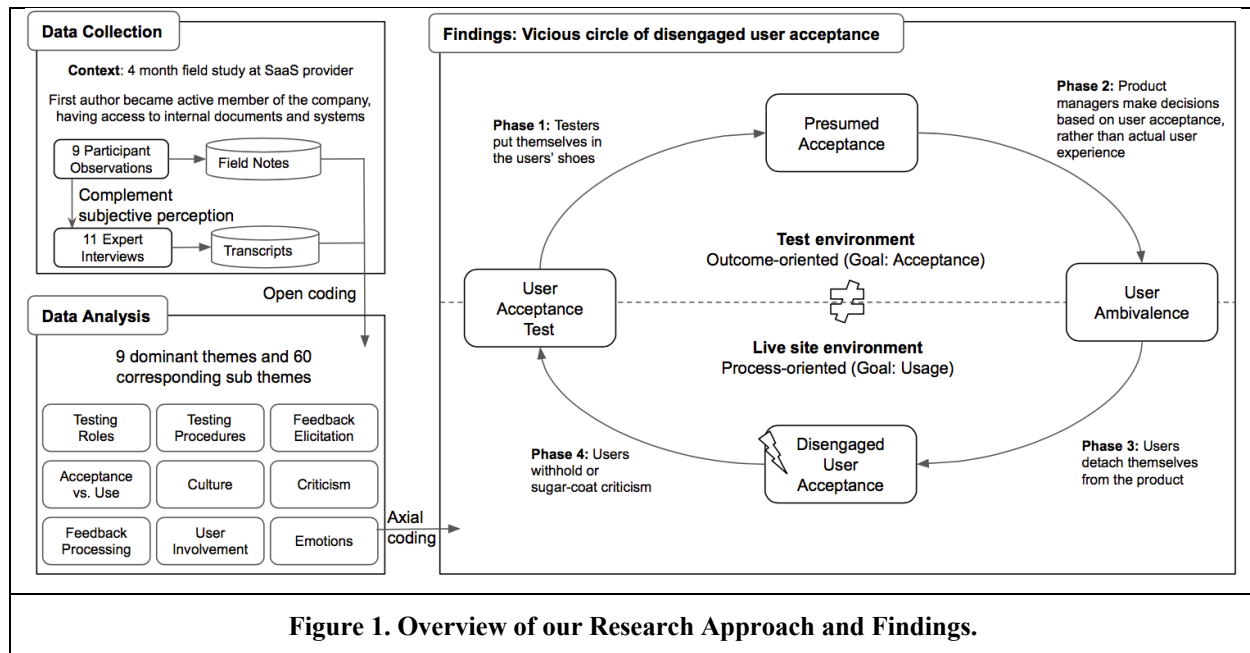


Figure 1. Overview of our Research Approach and Findings.

Results: The Vicious Circle of Disengaged User Acceptance

This section presents the findings from our qualitative case study of user acceptance testing at CloudEnhancer. The results are structured along the following four phases:

Phase 1: We start by showing that UATs serve the dual purpose of a legal signoff and a usability test, in which testers put themselves in the shoes of an imagined end user, leading to *presumed acceptance* that does not necessarily relate to user acceptance. Neither does presumed acceptance provide product managers with sufficient pre-release feedback to make informed decisions during beta stages.

Phase 2: Next, we show that UATs might address the wrong problem when being mostly targeted towards user acceptance rather than the actual user experience, which is the predominant goal of live site usage, including factors such as engagement, interaction, and emotional attachment. This can lead to *user ambivalence* towards the superficially tested SaaS product.

Phase 3: We then show that UATs can hinder constructive criticism, as they do not allow for emotional reactions and communicating negative feedback appropriately. Following a continued state of user ambivalence, users may eventually fall into *disengaged user acceptance*, meaning that they passively conform with UAT procedures while hiding their actual emotional connection to the tested service. Ultimately, this can have a negative impact on a user's affiliation towards a SaaS system.

Phase 4: Finally, we show that cultural factors may contribute to the spreading of disengaged user acceptance, further hindering constructive criticism. Although the company culture at CloudEnhancer was perceived as open and supportive, negative feedback was held back, sugar coated, or diluted. Critique being minimized and not expressed directly turned out to be a major problem for continuous product innovation. Eventually, this may mislead SaaS providers to thinking that the issue at hand could be solved by conducting further UATs, thereby exacerbating the problems of disengagement during testing and decreasing adoption of product innovations.

Phase 1: From User Acceptance Test to Presumed Acceptance

As we learned from our study, UATs at CloudEnhancer represent a weakly defined hybrid form between a legal signoff (something that has to be done) and a usability test (something that has to be satisfactory). As it can be difficult to involve many customers in the testing procedures, UATs may be performed with only a limited set of users, and product managers need to make many decisions based on informed guesses rather than on well-founded testing feedback. When conducting UATs, product managers may encounter minimal

engagement from testers and receive weakly defined requirement change requests. This form of acceptance could be described with the term *presumed acceptance*: Testers put themselves in the shoes of the end user, imagining the usefulness of a new functionality for a third party and not themselves, without knowing whether that third party actually finds that functionality useful.

From studying how UATs were conducted at CloudEnhancer, we learned that there is no strictly defined process implemented. In the past, a lot of releases have also been deployed to production even without any prior user testing. But frustration during past releases in regard to exclusion resulted in a corporate decision that, moving forward, UATs always have to be performed for larger, non-contractually ordered projects. The intended purpose is essentially to include users for open engagement, as one participant explains:

“basically, it's more like a kind of open response to what they're seeing, [...] less specific in regards to how they should navigate around the process [...], but trying to elicit more open engagement” [i1, Director of release management].

In the case of the ‘customizable dashboards’ project, testers received introductory steps via email or a presentation of the new feature, without defining specific testing procedures. For that particular project, the testing focus has been on usability, identifying bugs, but also on gathering additional feature requests and spotting functional inconsistencies. However, the testers often did not comply with that semi-structured test procedure, but focused mostly on usability aspects. One tester from the sales department notes positively that *“it was interesting for me to see the product before it goes live and just go through the various workflows on how a customer may use it”* [i11, Account executive]; but also mentioned in regards to the testing format *“I think that was pretty open ended [...] I would have liked maybe a few more steps or few more different workflows [...] to run through it, just because [...] I wasn't really sure how to attack it”* [i11, Account executive].

Other participants have mixed feelings about the feedback that was provided during those tests.

“feedback was, 'oh this is great, perfect. Now I can do whatever I want to'. But I would say that's not feedback [...]. We know it's great. That's why we built it. [...] We want to know what's the next step.” [i7, Partner success manager].

The user experience (UX) department’s impression of that was that *“we learned [...] a ton, so there were a lot of [...] small usability issues that we had to fix”* [i8, UX/UI designer]. In regards to online testing, the number of customers sending their feedback directly to the product departments was low, although some of them voluntarily participated, especially because they were unsatisfied with the prior dashboard version. Internally, the received feedback varied and showed insufficient volume and quality overall:

“most people do end up supplying feedback [...] and then the quality of the feedback varies. [...] I think some people just take a really rough go at it and provide some very general input” [i2, Product operations manager].

Yet still, the product managers consider UATs as an important preliminary step before releases, especially due to negative experiences in the past when not conducting any user tests at beta stages. While one product manager generally states that *“I should also get the green light from the customer saying that, okay it works, for us”* [i4, Product manager], the project manager steering the ‘customizable dashboards’ project states that *“mainly it was to get it out to the customers [...] just so someone could have a look before we [...] launch”* [i8, Product manager]. A customer, who had access to the new feature for several weeks before being involved in an onsite testing session with two CloudEnhancer employees, provided many suggestions for the new feature, which according to them could add a lot of value to their business needs. Yet this participant concludes that *“it was a standard meeting. And [...] the dashboards worked, it made sense”* [i10, Customer], showing a high degree of imagination when it comes to evaluating the general usefulness of the ‘customizable dashboard’ feature, rather than testing it for themselves specifically. Summarizing the onsite testing session, the customer articulates a neutral, detached reflection to testing that feature, which they previously evaluated as a valuable addition to the product offering: *“there was nothing [...] special, [...] just them asking questions for their dashboard feature”* [i10, Customer].

The responsible project manager states that there were still many uncertainties regarding the new release after conducting the UATs:

“Is this what people are expecting, or [...] is this something that, at least, the customers can actually use and get value of and is it something that you can come out with [...] that does not have too much technical debt [...] it's many things you can scope” [i8, Product manager].

As a result, product managers might have to rely on their intuition due to the insufficient testing feedback available, as one participant explains: *“one thing is, just you are [...] feeling in, the gut, [...] in your stomach [...], at some point you feel that, okay, people are really liking this, [...] and users are actually using it, and they get value of it” [i8, Product manager].*

Phase 2: From Presumed Acceptance to User Ambivalence

As we also learned from our study, UATs might address the wrong problem when being targeted towards user acceptance, rather than user experience. This can result in a mismatch between test environment goals, which are steered towards acceptance, and live site environment goals, which are steered towards usage and user experience. When product managers rely too much on presumed acceptance to make decisions on product improvements and releases, users may become increasingly ambivalent towards the product over time, without product managers being aware of it.

At CloudEnhancer, the UAT results of the ‘customizable dashboards’ feature were *“generally [...] positive, I think no one didn't agree that this was a good idea” [i8, UX/UI designer].* When the project has been pushed to the live environment of all customers, the evaluations did not end there - also because many customers saw it for the very first time. Anonymous responses submitted through an in-tool feedback module within the first week after the go live showed mixed feedback to the new feature, although it was encountered fairly well during the UAT.

One participant sees this as a confirmation of his concern that high user acceptance does not seem to indicate a positive user experience, but may actually disguise an ambivalent user experience that, if negatively increasing, may even lead to customer churn over time. Referring to the metrics of a market that has the best net promoter score (NPS) of all, the participant explains: *“we don't need customers to be happy [...]. We need them to be successful [...]. We don't really care too much about happy customers because uh, they can be happy. They love the tool [...]. But they don't utilize it to the fullest. They think it's great. They think we're great. They love us. They give us very high NPS scores [...]. But we can see that they're churning” [i9, Customer experience operations manager].*

Another customer-facing employee states a very similar opinion, seeing user acceptance only as an added bonus, but putting a very high emphasis on user experience including criticizing rather than superficially verifying a new release:

“The markets that sent in more support questions had less churn [...] than the ones that didn't. So, one would think: well they don't send in support tickets, so they're all happy, everything is great. But that's actually not the case [...]. Because if they really use the tool, if they get the value out of it, if they really want to use it they will bump into issues” [i7, Partner success manager].

In fact, the market with the best NPS results had the lowest volume of support requests submitted on average per customer in 2017. One possible explanation is that the performed UATs did not appropriately elicit both positive and negative emotional reactions, due to the test setup not being realistic. As one product manager explains: *“when we do (user acceptance) testing we can't get people's frustration right? We can't get how they are using it, or if they're just, ‘F***, it doesn't work.’ You can't get this from (acceptance) testing unless they're really vocal about it” [i4, Product manager],* highlighting that product managers may not be able to gather insights into an ambivalent user experience from UATs solely.

These findings indicate that the previously described presumed acceptance might not correspond to actual (and potentially negative) user experience in a live scenario, where also all system users have the opportunity to test. This poses the question how ambivalent feedback during live site usage should be communicated and how it should be received. Thus, we extended the research scope from its initial focus on pre-release activities to also include delivery stages.

Phase 3: From User Ambivalence to Disengaged User Acceptance

As we learned from studying live site evaluations, UATs can hinder constructive criticism in that they do not appropriately enable users to share emotional reactions and negative feedback. CloudEnhancer took many active steps to foster a culture of participation, transparency, and communication to gather user feedback. While responsible product managers mentioned criticism as a step towards user participation and continuous product improvement, the feedback received in a live setting can rather be described as a form of *disengaged user acceptance*. Low emotional reactions and reservation made product managers hold on to their initial intuitions gained from testing, because they were not sufficiently challenged due to not being aware of the persisting ambivalent user experience. Even when customers were critical, that criticism was mostly received anonymously in the form of writing or through intermediaries, hindering the possibility to follow up on it. The previous conception of dissatisfied users, claiming that usage in a realistic setting would have led into higher engagement to report dissatisfying system behaviors, was not necessarily verified in live site evaluations. In contrast, the users' passive conformance throughout testing and continuously during usage has gradually led to user disengagement and detachment from the licensed SaaS system.

One customer for instance, who was part of the onsite testing session, did not use the 'customizable dashboards' feature once it went live, even though they considered it generally valuable during the UAT: *"I haven't really worked with the new features afterwards"* [i10, Customer], along with other features they tested out during prototyping phases. When we asked if they had provided more feedback to the dashboard functionality that just went live, they responded that they think *"our feedback has been already enough for them [...]. If not, they can ask of course"* [i10, Customer]. The responsible project lead for the 'customizable dashboard' feature, reflecting on the scarce but positive feedback he received during testing, said that *"at least they have the option to try it out [...] and if there was something critical or catastrophic, we would probably have heard about it."* [i8, Product manager].

A customer representative, who received positive feedback during the UAT and assumed the new release was well received, similarly argued: *"I haven't gotten back to them since we released the dashboard [...]. But I guess they love it"* [i7, Partner success manager]. Another customer representative reviewed missing criticism more critically though, as customers *"just stick to their routine and don't think about it."* [i6, Customer success manager].

Thus, CloudEnhancer employees were under the impression that the 'customizable dashboards' project turned out well due to not directly receiving any negative feedback. However, the users actually did not engage with the feature in depth, neither during testing nor during live site usage, and thereby did not criticize nor adopt the feature in any way. The product manager's intuitions from the testing phase were then once more reconfirmed within the production phase based on generally positive, but emotionally neutral reactions originating from perceptively distanced user standpoints. While leaving product managers in the dark, users progressively fell into a state of disengaged user acceptance, conforming to functionalities to be rolled out to their live environment while actually hiding their ambivalence towards a system they evaluated during testing and subsequent usage phases.

The only explicitly negative feedback CloudEnhancer received from was in the form of bug reports:

*"When we signed up a new site, the deep link [...] was not working. So, I was writing to them often [...]. I mean, it could've been easy just to make the f***ing deep link work. Like, sitting and writing every time is just really annoying and, um yeah, we need the deep linking to work"* [i10, Customer].

Such strong emotional reactions to an erroneous functionality were an exception, as most clients considered the relationship with the service provider to be helpful and supportive. When we asked the client for their general impression, the response was: *"I think we shouldn't complain because it's a really valuable tool to us. So, in the bigger picture it's good."* [i10, Customer]. Because the presumed form of user acceptance during testing did not change throughout live site usage, but in fact disengagement to state critique was continued along, we also focused more in-depth on how other opportunities for directly communicating negotiable criticism have been utilized.

Phase 4: From Disengaged User Acceptance back to User Acceptance Test

As we furthermore learned from studying UATs at CloudEnhancer, customers often held negative feedback back in order not to damage the good relationship with CloudEnhancer, even though attempts have been made to bypass disengaged acceptance and elicit actual emotional reactions. Although the product development department emphasized the importance of gathering unfiltered feedback directly and the corporate culture was being perceived as generally open and supportive, users still sugar coated their negative feedback to be polite. This adverse effect was further exacerbated when CloudEnhancer tried to obtain better feedback by conducting further UATs.

Because usage metrics from live site monitoring did not reveal if retention or sales had been increasing, product managers gathered qualitative feedback to complement the quantitative usage metrics. One product manager stated that the quantitative tracking of key performance indicators “*doesn't really give me how people are using it, just telling me what people are using.*” [i4, Product manager]. Active involvement to gather user feedback seemed key to increase collaboration between users and product development, potentially also through “*techniques like design sprints or other ways of involving [...] customers and co-creation [...] we will more likely hit the dart board, at least, but [...] preferably the bull's eye*” [i3, Agile coach]. This has been also confirmed from a UX perspective, though putting particular emphasis on feedback being provided directly and instantly: “*the feedback we get through [...] word of mouth, or [...] what they write to us is always through a filter, because you don't observe them directly interacting with the product*” [i8, UX/UI designer]. In fact, a lot of user feedback did get filtered retrospectively through intermediaries, internally mostly through product operation managers and technical support, externally mostly through customer success managers. Those intermediaries voiced their peers' opinion on their behalf.

However, when reviewing how users or user representatives utilized feedback opportunities after a recent release, not when being simply enabled to give feedback but when being actively asked for feedback, another problem became apparent. According to the product development department, users and user representatives showed a high degree of reservation when being asked directly for their opinion to both beta and live versions, as users were trying to not offend the other:

“people don't speak up at meetings [...] I can stand in front of people: ‘anyone, any questions?’ [...] But they feel like they would offend me by telling me that it's rubbish and they can't use it. That's an issue for us [...] and I do believe we have a development department that aren't sticking their fingers in the ears - they're just literally not hearing it, because it's not getting through” [i1, Director of release management].

Even though the corporate culture was generally transparent and (maybe even because) the service level provided was very supportive, cultural barriers hindered open feedback communication for continuous product improvements.

The difficult situation of negative feedback not being transmitted to product development departments was exacerbated when users took part in multiple UATs throughout their active usage phase. At CloudEnhancer, there was a pool of customers who voluntarily agreed to be available for UAT purposes, the so called *FirstEnhancers*. According to the development department, users from this list showed an increasingly high degree of reservation when being asked directly for their critique to both beta and live versions: “*do they feel that they'd be kind of, putting us down if they didn't like what [...] we gave them? [...] ‘We've given you early access [...] you're a FirstEnhancer you're like a star person, do you really want to offend me?’*” [i1, Director of release management]. In fact looking at the statistics, over three-fourths of the accounts in that ‘FirstEnhancer’ list had a low risk level being the subjective opinion of the assigned account manager indicating how likely the customer is going to churn. Considering the earlier identified potentially positive correlation between good satisfaction survey ratings and churn rates, our collected data prefigures that further UATs might even worsen a user's insufficient product usage and detachment from collaborative product design, and thereby complete the process from presumed acceptance to user ambivalence into a vicious circle aggravating disengagement and passive compliance when conducting UATs.

Discussion

Based on the insights gleaned from the above findings at CloudEnhancer, this section discusses and integrates the findings with existing literature. Specifically, we discuss the dangerous tendency of UATs to address the wrong problem when measuring user experience, which can potentially lead to a vicious circle of disengaged user acceptance. We also reflect on the role of emotions in user acceptance testing and how service providers can query users for ambivalent emotions in particular. We then discuss the limitations of our study and outline a path for further research before we sum up the key takeaways in the conclusion.

User Acceptance Tests: Providing the ‘Right Answer to the Wrong Question’

As we learned from our study, positive outcomes in a UAT and the absence of criticism do not necessarily indicate actual use or user experience. One particular issue with UATs is that they are used to replicate a momentary snapshot of a user journey (Zaki et al., 2016). Furthermore, UATs may be used to conclusively verify user experience, instead of seeing user experience as something that must be continuously questioned and re-evaluated as part of a larger journey (Pham, 2007). Our data indicates that user acceptance and experience are neither equal nor verifiable during one time testing procedures, but rather a continuous practice of making necessary system changes along the way. This finding points to the existence of a ‘say-do gap’ in user acceptance testing: While respondents may state to do one thing when specifically queried during a UAT, they may in fact behave completely differently while unobserved (for a discussion of this phenomenon in survey studies, see Bertrand and Mullainathan, 2001).

These findings suggest that that UATs alone cannot provide conclusive answers on how the users actually perceive a SaaS application, and should thus play only a minor role in evaluating the overall user experience, as they can only provide a brief snapshot of a static situation. Instead, SaaS providers need to query users for unfiltered and mixed emotional responses to the product, rather than just re-confirming what the provider already assumed.

Implication: See Merely Positive Feedback in UAT as Early Warning Sign of Disengagement

The question to ask should not be whether the customer accepts the product, and neither is it conducive to get positive user acceptance test results at all costs. Quite the contrary, successful UATs that only report positive feedback can be an early warning sign of disengagement and passive compliance, while criticism and complaints can be an indicator that the customer is engaged and values the product. So, while UATs may still be necessary as a first indication of whether the new release complies with the identified requirements, SaaS providers should also consider criticism as a positive form of feedback. Therefore UATs should be merely seen as one of many evaluation methods in the continuous quest for qualitative software delivery. Instead of resting on positive UAT outcomes, continuously evaluating the overall user experience is more important. Besides the currently non-utilized opportunities for active user participation and collaboration, UATs should not be considered as a means to an end for finalizing a project, but as means for ending one iteration to be further adapted and revised on production environment, as a continuous self-iterative process. Against this backdrop, software development in a SaaS context should be seen as an evolutionary process, in which the adaptation, requirements, and configuration of a SaaS system constantly changes, involving both the SaaS provider and customer in participatory development practices (cf. Dittrich, 2014). The question to ask should essentially not be "is it good?", which prompts users for acceptance. Instead, SaaS providers should rather ask the uncomfortable question "what troubles you?", which prompts users for constructive criticism.

Breaking the Vicious Circle by Cultivating Constructive Criticism

As the relationship between the customer and the service provider in our case study was perceived as supportive and trustful, the customers often either sugar coated or withheld negative feedback, in order not to harm the positive relationship. As a result, constructive criticism was difficult to obtain and uncertainty arose whether users were really satisfied or just disengaged. This drawback was even exacerbated when users conducted various UATs for several system enhancements.

Implication: Encourage Negative Opinions to Cultivate Open Feedback Cycles

SaaS providers should not rest on the assumption that lack of negative feedback indicates satisfied customers. Instead, absence of criticism may well indicate the proverbial calm before a storm (for instance customer churn). Hence, SaaS providers should deliberately seek out negative feedback and try to regard it as constructive criticism. The important part is then to frame negative feedback, which “negatively evaluates another’s products, performances, or attributes” (Fong et al., 2018, p. 1), as constructive criticism. For this to happen, it is important to establish an environment of trust, where customers can freely express their opinion and frame negative feedback as improvement-oriented, well-intended, and unfiltered constructive criticism (Fong et al., 2018). From the SaaS provider’s perspective, this requires embracing constructive criticism, rather than reacting to it with defensiveness and counter-provocations (Fast et al., 2014). From the customer’s perspective, the attempt to provide direct constructive criticism often leads to applying mitigation strategies, softening criticism by being indirect in order not to harm an interpersonal relationship through negative judgment. However, this form of sugar-coating criticism potentially leads to a lack of understanding or misinterpretation of the underlying information in the recipient’s mind (Hyland and Hyland, 2001). More indirect forms of transmitting feedback, such as writing anonymously to the intended target subject should therefore be avoided, as follow-up questions or negotiations cannot be facilitated in the co-creation process of a SaaS system (Leung et al., 2001).

Implication: Enable Peer-based Discourse Instead of Intermediary-based Feedback

Intermediaries are ill-suited for conveying criticism, because then not only could the transferred message potentially get diluted or altered, but the criticism also lacks authenticity in the perspective of the receiver (Nemeth et al., 2001). Thus, constructive criticism should be communicated directly, openly, honestly, and in full. This calls for a redesign of organizational structures, environments, practices and attitudes when evaluating a software build that shift the locus of control from the top of the hierarchy towards peer-based discourse (McDonald, 2011). Through establishing mutual trust as a basis for dialogical design, software providers can encourage users to explicitly disclose negative emotions (Floyd, 1992). In order to give credibility to someone’s negative feedback, cultivating high trust environments is about creating a space in which peers can speak clearly for themselves (McDonald, 2011). Socially supportive milieus can then also be challenging when negative feedback is disclosed, yet being constructive not by phrasing critique differently, but by framing it differently (Grant, 2016).

Implication: Elect ‘Bad Cops’ to Actively Seek Out Edge Cases in Real Usage Scenarios

One potential approach to facilitate constructive criticism would be to encourage selected customers to serve as ‘bad cops’, being explicitly instructed to criticize the SaaS application. This could be especially effective if conducted in a live site evaluation setting, during which customers use the application to execute their main use cases, while testers and product managers from the SaaS provider are present, either physically or via co-browsing. Customers could then share their ambivalent emotions and negative experiences ‘in vivo’, as well as reflect on their experience in a ‘cross-examination’ session where they play the ‘bad cop’ role, thereby enabling testers and product managers to obtain a more accurate picture of the customer’s emotional connection to the SaaS application. When the testing focus is placed on the customer’s actual day-to-day work practices with the SaaS application, rather than on open ended testing aspects such as usability or bugs, even a UAT could provide reliable results that accurately indicate actual user experience. SaaS providers could draw on principles from the ‘Six Thinking Hats’ method, whereby participants are assigned different roles to look at an issue from several angles (De Bono, 1985).

Facilitating Constructive Criticism via Affective Responses

As we further learned from our study, continuous delivery of SaaS applications should not only be based on quantitative metrics, but also be enhanced with qualitative user feedback (cf. Guckenheimer, 2016). Qualitative feedback can enrich quantitative data and makes it inspiring as well as convincing, as it is adding emotional factors to the decision-making process, and is helpful when SaaS providers want to know what they do not already know (Wang, 2013). By getting qualitative customer feedback in the form of stories, emotions can be shared, which play an important role in understanding users’ affiliation with a product (Stein et al., 2015). Because relationships between a stakeholder and an organization are emotional and not rational, organizations should focus on users, their actual stories, and their unfiltered user experiences when evaluating their current and future product offering (Wang, 2013). For this purpose, SaaS providers could draw on principles from design thinking, such as using structured yet open questions to elicit deep-level information (Stanford D.School, 2010).

Implication: Augment Quantitative Metrics with Qualitative Emotional Responses

Accounting for the emotional factor, meaning that users do not make purely rational decisions, can be difficult to accommodate for SaaS providers. For instance, examining the role of emotions in emerging IT usage patterns, Stein et al. (2015) suggest that gathering in-depth feedback from users is highly important, especially in a post-production scenario, as “it is often difficult to elicit genuine feedback from users prior to going live because the project is not salient to them at that stage” (Stein et al., 2015, p. 45). Particularly interesting here are the identified consequences between uniform and ambivalent as well as positive and negative affective responses. While users respond to uniform dissatisfaction and satisfaction with clear refusal or adaptation strategies respectively, the coexistence of both satisfaction and dissatisfaction increases engagement, which offers an opportunity for software providers to make user-centered system improvements (Stein et al., 2015).

Against this backdrop, negative emotional responses should not be seen as merely unfavorable. Negative affections are neither directly related to a user’s usefulness perception, nor are they directly related to the usability of an IT application, but could also be related to the degree of involvement throughout the software implementation process (Stein et al., 2015). Although software development methodologies are still primarily concerned with usability and user acceptance targets (Vermeeren et al., 2010), our findings relate to prior research, which found that ambivalent emotions can lead to active and positive user experience (Stein et al., 2015). Because these emotions in a user-system interaction have to be identified, communicated, and addressed with appropriate means, relating a user’s emotional connection to a product should not only be seen as something evaluable after once interacting with it (Vermeeren et al., 2010). Instead, mixed emotions can have a positive impact on engagement while “uniform satisfaction [...] leads to minimal user engagement and rather passive compliance” (Stein et al., 2015, p. 44).

Implication: Enable Immediate Feedback Channels to Counter Improvisational Sensemaking of Ambivalent Emotional Responses

Reflecting on emotions retrospectively may be difficult for users, especially if the emotional reaction was not uniform. Because users who experience mixed emotions throughout the use of a software system tend to be unable to describe their affective responses clearly, they may then provide developers with generally positive feedback, as they have worked out their own solution (Stein et al., 2015). In that regard, having the opportunity to provide open feedback freely is one important aspect, though another equally important factor is that emotions can be communicated instantly when they occur. This is important to ensure that ambivalent emotions will not be diluted through improvisational sensemaking, in order to provide designers with emotional stories they need for planning future product iterations. It is through these stories that decisions upon product changes can be influenced constructively (Ciriello et al., 2017).

The issue of filtered feedback becomes especially apparent when feedback is provided via intermediaries. At CloudEnhancer, product operation managers have acted as user representatives, and so the feedback lacked authenticity. Instead, SaaS providers should be aware of the importance of gathering unfiltered, emotionally-charged criticism. This would also be beneficial for all users that are not part of a testing phase for non-contractually ordered functionalities, as they could then participate in the design community in practice throughout live site usage (Heiskanen et al., 2010).

Limitations

This study has to be seen in the light of its limitations. First, as typical for an in-depth qualitative field study, we deeply immersed ourselves with the practices of a single company. Although we have provided rich contextual details to maximize the study’s transferability, the generalizability of our findings is limited. As such, our suggested model of a vicious circle is highly contextualized. For instance, we identify salient user and usage context factors such as easy churn, low switching costs, distributed vendor-client locations, and a competitive global landscape, which all may exacerbate repercussions of negative user experiences. Further theorizing studies in other contexts would be useful to identify potentially less contextualized implications (cf. Hong et al., 2013). As we closely followed the principles of interpretive field research (Walsham, 2006), our results are likely transferable to similar contexts, namely other business-to-business enterprise SaaS vendors with a product offering serving a diverse and globally distributed client base with limited customization options. Yet, other context factors might the transferability of our findings, such as cultural aspects related to how openly critique is being communicated.

Second, having one author deeply embedded at the case company allowed us to “go native” within the research site. Although we critically reflected, challenged, and discussed the findings as much as possible in the author team, the findings are by nature of the research method tainted towards the subjective interpretations of the participants.

Third, although our study provided useful suggestions for CloudEnhancer to reflect on and improve their own testing practices, other companies may have to interpret and customize these suggestions according to their own needs. For further research, we therefore recommend to include the learnings gathered from this research in a long-term multi case study, following up on this exploratory study to also investigate deeper on the business impact of disengaged user acceptance. In our study, we focus primarily on the SaaS provider’s perspective. Although we conducted one in-depth interview with a client and additionally gathered archival documents from other actively or previously licensing customers, a larger volume of clients might be necessary to obtain deeper insights into their perspective on the phenomenon. Another promising research stream would be design science research for developing socio-technical solutions that better encourage constructive criticism while minimizing the adverse effects of disengaged user acceptance.

Conclusion

This paper explores the practices and effects of user acceptance testing in SaaS through an exploratory qualitative field study at a multinational SaaS provider in Denmark. We find that user acceptance testing alone cannot provide an accurate picture of how the user actually feels about the tested product or feature. Even worse, being overly reliant on UATs may put service providers at risk of checking the wrong boxes, substituting actual user experience with user acceptance while users fall into *disengaged user acceptance*.

Disengaged user acceptance can be understood as passively conforming with UAT procedures while hiding one’s actual emotional connection to the tested product or service. Our data shows that this may result from negative feedback being sugar coated or withheld in order not to damage the relationship between customer and SaaS provider, a false sense of security from receiving positive user acceptance feedback, but also (and perhaps most importantly) due to UATs providing the “right answer to the wrong question”.

Once disengaged user acceptance emerges, it continues to spread throughout the live site usage. This can potentially lead to a vicious circle, during which users detach themselves even further from the SaaS application and its provider, leading to ever less engagement as well as product management tapping in the dark. Not only is a UAT insufficient to counter this problem, but it may make the problem even worse. In our study, UATs reinforced existing power inequalities, preventing users to participate more actively in the design and decision-making process from the very beginning of a system rollout. Hence, a testing approach targeted towards acceptance might even increase the gap between what users say during a UAT and what they do throughout later system usage. The test environment itself even further promoted user detachment from the design process and has put users into the role of passive bystanders. Even though customers perceived the corporate culture in our case company as open and supportive, disengaged user acceptance prevented them from communicating constructive criticism.

Based on these findings, we suggest that UATs can only provide a brief snapshot of a static situation throughout beta testing and in evaluating the overall user experience. Rather than relying mainly on UATs throughout pre-release testing, cultivating constructive criticism can be more helpful for the continuous evolution of SaaS applications. For this, SaaS providers need to query users for unfiltered and mixed emotional responses to the product, rather than just re-confirming what the provider already assumed. The question to ask should essentially not be “is it good?” (which prompts users for acceptance), but rather asking the uncomfortable question “what troubles you?” (which prompts users for constructive criticism).

On a concluding note, our study may also have interesting implications for technology acceptance research. Our findings indicate that perception-based agreement scales are not only fragile per se, but also that the type of values these scales measure is questionable. Agreement scales are not only used in UATs but also in related models such as the Technology Acceptance Model. As such, agreement scales may reveal unrealistic acceptance effects. We hope that our study inspires further research on disengagement and emotional responses in technology acceptance.

References

- Beck, K., and Gamma, E., 2000. "Extreme programming explained: embrace change." *Addison-Wesley professional*.
- Bertrand, M., and Mullainathan, S. 2001. "Do people mean what they say? Implications for subjective survey data." *American Economic Review*, 91(2), 67–72.
- Birks, D.F., Fernandez, W., Levina, N., and Nasirin, S. 2013. "Grounded theory method in information systems research: its nature, diversity and opportunities." *European Journal of Information Systems* 22, 1–8. <https://doi.org/10.1057/ejis.2012.48>
- Chou, S.-W., and Chiang, C.-H. 2013. "Understanding the formation of software-as-a-service (SaaS) satisfaction from the perspective of service quality." *Decision Support Systems* 56, 148–155.
- Ciriello, R.F., Richter, A. and Schwabe, G. 2018. "Digital Innovation." *Business & Information Systems Engineering* 60(6), pp.563-569.
- Ciriello, R.F., Richter, A., and Schwabe, G. 2017. "When Prototyping Meets Storytelling: Practices and Malpractices in Innovating Software Firms", in: *Proceedings of the 39th International Conference on Software Engineering*, Buenos Aires, Argentina, pp. 163–172. <https://doi.org/10.1109/ICSE-SEIP.2017.24>
- Davis, F.D., and Venkatesh, V., 2004. "Toward preprototype user acceptance testing of new information systems: implications for software project management." *IEEE Transactions on Engineering Management* 51, 31–46.
- De Bono, E., 2017. Six thinking hats. Penguin UK.
- DeCuir-Gunby, J.T., Marshall, P.L., and McCulloch, A.W. 2011. "Developing and using a codebook for the analysis of interview data: An example from a professional development research project." *Field Methods* 23, 136–155.
- Dittrich, Y. 2014. "Software engineering beyond the project—Sustaining software ecosystems." *Information and Software Technology* 56, 1436–1456.
- Dixon, M., Toman, N., and DeLisi, R. 2013. "The effortless experience: conquering the new battleground for customer loyalty". Penguin.
- Fast, N.J., Burris, E.R., and Bartel, C.A. 2014. "Managing to stay in the dark: Managerial self-efficacy, ego defensiveness, and the aversion to employee voice." *Academy of Management Journal* 57, 1013–1034.
- Floyd, C. 1992. "Software development as reality construction", in: *Software Development and Reality Construction*. Springer, pp. 86–100.
- Fong, C.J., Schallert, D.L., Williams, K.M., Williamson, Z.H., Warner, J.R., Lin, S., and Kim, Y.W. 2018. "When feedback signals failure but offers hope for improvement: A process model of constructive criticism." *Thinking Skills and Creativity*, 30, pp.42-53.
- Gabbard, J.L., Hix, D. and Swan, J.E. 1999. "User-centered design and evaluation of virtual environments." *IEEE Computer Graphics and Applications*, 19(6), pp.51-59.
- Grant, A. 2016. "How to build a culture of originality." *Harvard Business Review* 94, 18.
- Guckenheimer, S. 2016. "Our journey to Cloud Cadence, lessons learned at Microsoft Developer Division." Technical Report. <https://www.microsoft.com/en-us/download/details.aspx>.
- Heiskanen, E., Hyysalo, S., Kotro, T., and Repo, P. 2010. "Constructing innovative users and user-inclusive innovation communities." *Technology Analysis & Strategic Management* 22, 495–511.
- Hong, W., Chan, F.K., Thong, J.Y., Chasalow, L.C. and Dhillon, G. 2013. "A framework and guidelines for context-specific theorizing in information systems research." *Information Systems Research*, 25(1), pp.111-136.
- Humble, J., and Farley, D. 2011. "Continuous delivery: reliable software releases through build, test, and deployment automation." Addison-Wesley Boston.
- Humphreys, L. 2005. "Reframing social groups, closure, and stabilization in the social construction of technology." *Social epistemology* 19, 231–253.
- Hyland, F., and Hyland, K. 2001. "Sugaring the pill: Praise and criticism in written feedback." *Journal of second language writing* 10, 185–212.
- INUSE Research Group, 2017. "CoDesign Journey Planner." [WWW Document]. URL <http://codesign.inuse.fi/approaches> (accessed 8.28.18).
- ISO, 2010. ISO 9241-210:2010(en), Ergonomics of human-system interaction [WWW Document]. URL <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en> (accessed 8.28.18).
- Khannur, A. 2014. "Structured Software Testing: The Discipline of Discovering." New Delhi: Partridge.

- Klein, H.K., and Myers, M.D. 1999. "A set of principles for conducting and evaluating interpretive field studies in information systems." *Management Information Systems Quarterly* 67–93.
- Kordon, F. 2002. "An introduction to rapid system prototyping." *IEEE Transactions on Software Engineering* 28, 817–821.
- Krancher, O., Luther, P. and Jost, M. 2018. "Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback." *Journal of Management Information Systems*, 35(3), pp.776-812.
- Lehman, M.M., Ramil, J.F., Wernick, P.D., Perry, D.E., and Turski, W.M. 1997. "Metrics and laws of software evolution-the nineties view", in: *Software Metrics Symposium*, pp. 20–32.
- Leung, K., Su, S., and Morris, M.W. 2001. "When is criticism not constructive? The roles of fairness perceptions and dispositional attributions in employee acceptance of critical supervisory feedback." *Human Relations* 54, 1155–1187.
- McDonald, P. 2011. "It's time for management version 2.0: Six forces redefining the future of modern management." *Futures* 43, 797–808.
- Nemeth, C., Brown, K., and Rogers, J. 2001. "Devil's advocate versus authentic dissent: Stimulating quantity and quality." *European Journal of Social Psychology* 31, 707–720.
- Orlikowski, W.J., and Baroudi, J.J. 1991. "Studying information technology in organizations: Research approaches and assumptions." *Information Systems Research* 2, 1–28.
- Pham, H. 2007. "System software reliability." Springer Science & Business Media.
- Pugh, K. 2010. "Lean-Agile Acceptance Test-Driven-Development: Better Software Through Collaboration." Pearson Education.
- Scacchi, W. 2002. "Process models in software engineering." *Encyclopedia of software engineering*.
- Schultze, U., and Avital, M. 2011. "Designing interviews to generate rich data for information systems research." *Information and Organization* 21, 1–16.
- Sommerville, I. 2007. "Software engineering." Addison-wesley.
- Stanford D.School. 2010. Bootcamp Bootleg.
- Stein, M.-K., Newell, S., Wagner, E.L., and Galliers, R.D. 2015. "Coping with information technology: mixed emotions, vacillation and non-conforming use patterns." *Management Information Systems Quarterly* 39(2), pp.367-392.
- Suchman, L. 2002. "Located accountabilities in technology production." *Scandinavian Journal of Information Systems* 14(2), p.7
- Vermeeren, A.P., Law, E.L.-C., Roto, V., Obrist, M., Hoonhout, J., and Väänänen, K. 2010. "User experience evaluation methods: current state and development needs", in: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction*, pp. 521–530.
- Walsham, G. 2006. "Doing interpretive research." *European Journal of Information Systems* 15, 320–330.
- Walsham, G. 1995. "Interpretive case studies in IS research: nature and method." *European Journal of Information Systems* 4, 74–81.
- Wang, T. 2013. "Big data needs thick data." *Ethnography Matters* 13.
- Zaki, M., Kandeil, D., Neely, A., and McColl-Kennedy, J.R. 2016. "The fallacy of the net promoter score: Customer loyalty predictive model." *Cambridge Service Alliance* 1–25.