View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DigitalCommons@USU

Blame Attribution after Failures within Platform Ecosystems

ABSTRACT

Increasingly, new hardware and software are embedded within ecosystems that include a platform and modules. Ideally these ecosystems perform reliably. However, if an ambiguously sourced failure occurs within one of these ecosystems, users are left to distribute blame across the various components of the ecosystem. The actual distribution of this blame, however, can be difficult to predict. This study investigates attribution of blame and discontinuance recommendations for ecosystem components after an ambiguously sourced failure. To extend platform ecosystems and attribution theory, we conducted a scenario-based experiment investigating the negative consequences of failure for platform and module components and the contingent effects from design elements (border strength) and contextual factors (task goal-directedness, disruption severity). Results demonstrated a diffusion of negative consequences for failure across ecosystem components, but ecosystem modules (apps) received the majority of the blame and highest discontinuance recommendations. High border strength shifted negative consequences for failure away from the OS to the device. Low goal-directedness resulted in users taking more of the blame for the failure, and higher disruption severity resulted in higher discontinuance recommendations for the OS and device. Importantly, the amount of blame attributed to one component in an ecosystem predicted discontinuance recommendations for other components.

Manuscript Word Count: 10,067

KEYWORDS

mobile platforms, digital platforms, mobile platform ecosystems, failure, ambiguous failure, blame, blame attribution, discontinuance

Blame Attribution after Failures within Platform Ecosystems

INTRODUCTION

As platform-based ecosystems become increasingly common, owners of components of those ecosystems are finding it important to understand how to manage their products within these ecosystems successfully. Given the interdependency among components, the function of any one component can be tied to the success and proper functioning of other components. Similarly, and central to our study, failure on the part of one component may affect end-user perceptions of other components. With this complex set of relationships and related risk in mind, these ecosystems present a major challenge for a component's sponsor to influence how the sponsor is perceived.

Although the nature and consequences of failures vary, previous research suggests that when a system fails, users search for someone or something to hold accountable for their frustration and any related negative consequences [56]. When users have a clear idea regarding responsibility for failure, this accountability is relatively straightforward to assess. However, within a platform ecosystem, where interdependence is the norm and responsibility among platform components may be unclear, understanding which component to hold accountable and to what degree becomes more complicated as the source of failure may be ambiguous and/or undiscernible by end users.

For example, consider a motorist using the Waze app running on the Android OS on an HTC smartphone to navigate the streets in a foreign city. At a crucial intersection, the operation of the system stops entirely, leaving the motorist confused as to where to go, or, worse, in a dangerous traffic situation. Which component of the ecosystem does the motorist hold accountable and consider discontinue using? Similarly, a user interacting with the Hulu application on a Roku device may encounter sub-optimal functioning (e.g., content that should be available cannot be found). Again, which component of the ecosystem is held accountable and how does that affect the user's perception and continued use of that component?

In such situations, it is possible that multiple components are considered similarly culpable, multiple components are considered culpable by the user but in unequal amounts, or a single component could be considered the primary source of the problem. Within a complex enivornment, however, this assessment may or may not correspond with actual culpability. Thus, when platform ecosystem failures of an ambiguous nature occur, negative assessment assigned for the failure may spread over ecosystem components regardless of whether or not they are actually at fault.

In this study, we address two broad research issues. First, we examine how users distribute blame among platform ecosystem components when they encounter a failure from an ambiguous source. Second, we consider how components within such ecosystems may be able to reduce the negative implications of such failures. In doing this, we use concepts from the platform ecosystems and platform markets literature [17, 57] to help understand the tight coupling within a platform ecosystem. We then integrate the concepts of digital borders and border strength [14] with attribution theory [20] to theorize the manner in which users choose to apportion blame after the occurrence of an ambiguous failure.

Using the context of the smartphone platform, we conducted a scenario-based experiment in which we focused on how border strength, or the extent to which the boundaries around objects (in this case, the device, its OS, and an application), affects user blame attribution and discontinuance recommendations given an ambiguous failure. We also tested the effects of different types of task (goal-directed vs. less goal-directed) being conducted when the failure occurred and differences in the severity of the failure. Our findings show that while negative consequences for failure are shared across components, apps receive the majority of the blame and the highest discontinuance recommendation. Increasing the border strength between platform components, we found, shifted blame away from the OS and toward the device. Importantly, the amount of blame attributed to one component in an ecosystem predicted discontinuance recommendation for other components.

THEORETICAL BACKGROUND

Failure and Attribution

While the complexity inherent in a platform ecosystem can create coordination and governance challenges [e.g., 28, 51], users still expect technology systems to function properly [56]. The context discussed here, of consumer-facing product-based ecosystems, aligns strongly with the concept of *product failures*. In simple (non-ecosystem) products, these failures have been found to result in negative consequences for the party deemed responsible, including refund-seeking [20], negative brand evaluation [51], distrust toward related products [10], and brand sabotage [29]. Therefore, understanding how the user identifies and attributes blame within a complex ecosystem, where the source of the failure may be particularly unclear, is vital to understanding how ecosystem failures affect user perceptions of the platform components' sponsors.

According to attribution theory, identifying the party to be held responsible occurs based on the user's perception of situations and events [18]. In other words, an actor that purposefully exerted effort resulting in a negative outcome accrues more negative sentiment (e.g., blame) than one that was not capable of preventing the action from occurring. Therefore, for individuals to make meaningful attributions, the intentionality behind actions leading to the negative event must be clear [50]. This intentionality has been further decomposed into attribution theory's three key dimensions: locus, controllability, and stability.

The *locus* of a failure captures the extent to which the action that caused the failure was internal or external to the individual making the assessment [60].

Platform Ecosystems

Platform ecosystems are systems that require both a core platform as well as modules built around the platform [58]. The platform supplies core functionality, such as access to input and output devices, data processing, and accesses to sensors. Modules, on the other hand, extend the functionality of the platform. Together, the platform and the modules that run on the platform form the platform ecosystem [57].

Such platform ecosystems are becoming increasingly common. For example, consider the streaming video ecosystem where the hardware device (e.g., a Roku box or properly-equipped television) is the platform through which modules, in this case streaming service applications (e.g., Netflix, Amazon Prime Video, YouTube), provide access to content. Video game consoles provide another example, wherein the devices (e.g., the PlayStation or Xbox console) are the platform through which game modules provide content and interactivity. As another example, smartphones and mobile devices form a platform ecosystem, whereby the handset and operating system together form the platform upon which applications (the modules) extend the functionality.

This form of ecosystem is attractive to both module creators and end users in that it facilitates easier adoption. For instance, a video game studio does not have to produce the system itself, input/output devices, or other protocols, but instead can focus on developing the entertainment content. Or, from the user perspective, the user does not have to purchase multiple devices or learn multiple interfaces to receive the benefits of differing applications.

Platform ecosystems can vary in complexity. In some cases, software alone can be considered to comprise the platform. For instance, Microsoft Word facilitates access to many different plug-ins (modules), while an internet browser can be considered the platform that provides access to numerous website modules. However, platforms can also be considerably more complex and consist of a combination of both hardware and software [57]. In such cases, the software component of the platform runs on top of the hardware, forming a sort of "stack" that together comprises the full platform. This more-complex form of ecosystem is common in every day computing devices such as PCs (which require both the computer hardware as well as an operating system) and smart phones (which require both a handset as well as an operating system).

With this added complexity, the necessary coordination and governance to facilitate proper function becomes more challenging [e.g., 28]. This may be particularly true for application developers who develop for a given operating system (e.g., developing apps for Android), but must also understand that there can be considerable variance with regard to the hardware part of the platform — for instance, the devices may have various screen sizes, memory capacity, and clock speeds. Further, in such complex platforms, both the software and hardware components of the platform must work together successfully, again while considering that the other components of the platform may vary (e.g., there are multiple versions of Android that may eventually appear on a handset and there are multiple different handsets that run Android).

Such interaction and interdependence are a key characteristics of platform ecosystems [57]. Given the variability possible among components, there are increased opportunities for failures to occur due to the interdependence among the components. Even if one component fails on its own, the performance of the other components is nevertheless tied to the failure of the one. To date, however, antecedents and consequences of user perceptions specific to the components in such ecosystems have not been widely considered despite the fact that qualities of a user's experience have been shown to be crucial to the formation of user attitudes toward technology [e.g., 27, 37, 43].

Failure and Attribution

While the complexity inherent in a platform ecosystem can create challenges with coordination and governance, users still expect these ecosystems to function properly. When the system fails to perform to expectations, users will search for something to blame [56]. Various forms of failure have long been studied within the information systems field. Scholars define a *system failure* as any occurrence in which an information system fails to meet expectations or requirements [e.g., 19, 38, 56]. Using this definition, much research has addressed organization-wide information systems that failed to satisfy their intended purpose to deliver value [3, 11, 15, 47]. Causes of failure include project escalation [30, 31], organization-system fit [55], and user resistance [4, 23, 33]. While important as avenues of inquiry, these studies consider failure in a way different from that proposed in our research. In particular, these studies focus on organization-wide systems (vs. personal technology) and consider known sources of failure (vs. ambiguous sources).

The failure of personal products used by individuals, however, has been studied at length in the marketing field, where it has been termed *product failure.* These failures have been found to result in negative consequences for the party responsible for the failure itself [5, 20, 39]. But the assignment of blame has been shown to be contingent on characteristics of the failure and the entities (individuals, organizations) involved [50]. According to attribution theory, finding a given party to be responsible for a product failure — in other words, blaming a given party — occurs based on the user's perception of situations and events [18]. What the user perceives in terms of causality and responsibility for failure influences the user's attribution of blame [50]. The party to whom blame is attributed has been found to suffer negative consequences as a result, such as anger and refund-seeking [20], negative brand evaluation [51], distrust toward related products [10], and brand sabotage [29].

Early theory on attribution focused on ordinary individuals understanding the meaning behind the actions of others [22]. For individuals to make meaningful attributions about others' dispositions based on observable actions, intentionality behind the actions must be clear [50]. In other words, one who purposefully exerted effort resulting in harmful actions would accrue more blame than one who was forced to perform harmful actions or unable to prevent harmful actions. In subsequent research, intentionality necessary to attribute blame was further decomposed into three characteristics: locus, controllability, and stability dimensions [60, 61].

In interpersonal attribution research, the *locus* of a failure captures whether the action that caused failure was internal or external to the individual [60]. When considering product failures, the locus has been conceptualized as an evaluation of direct culpability when a product fails to provide its intended function and is estimated on a continuum between the product and the consumer [20, 49]. When locus for a failure is estimated to be near the product, consumers perceive that the product is directly responsible and therefore subject to more blame. For example, failure experienced while using a product outside of its intended purpose (e.g., the car stops running after the user continued driving despite seeing the gas gauge on empty) may result in locus near the consumer and minimal blame attributed to the product. However, blame attributed to the product would be much higher for failure experienced during appropriate product use (e.g., the car stops running despite normal maintenance and use).

The *controllability* dimension captures the degree to which failure was the result of volitional or non-volitional action and is also traditionally conceptualized along a continuum [20]. Controllability indicates the degree to which an entity has the capacity to carry out an intended action. When controllability in product failure is high, individuals are likely to perceive that the negative consequences of the failure could have been avoided. Therefore, blame for the failure is also likely to be high. Alternatively, failure caused by unanticipated factors or lack of ability may not accrue as much blame because there was little control over the nature of the failure [60]. Although controllability often coincides with an internal locus, these dimensions can differentially affect blame. For example, a product manufacturer may contractually control the actions of a partner and therefore receive more blame for the partner's actions in a failure even though the partner is external. The extent to which a party is perceived to have had control over a failure outcome is a key determinant of the product user's adverse reaction [20].

Finally, the *stability* dimension is the degree to which the cause of the failure is temporary (e.g., could fluctuate over time) or permanent (e.g., is relatively stable) [20]. Perceptions of stability provide individuals making attributions in response to failure an estimate of how expected the failure was and how likely it will be in the future. For example, in the course of making attributions, one might consider: Is the failure the result of repeated action that is likely to continue or is it the result of transitory actions unlikely in the future? When the causes for failure are relatively stable, blame attributions tend to be more severe.

Together, locus, controllability, and stability have successfully explained a host of attributions that lay people make in response to observable actions of [32, 60]. However, applying attributions within platform ecosystems, where components are interdependent, presents a new challenge. The stability of each component within the ecosystem will likely remain observable. For example, users will notice repeated failures involving the ecosystem components. However, controllability and locus will likely be much more difficult to assess and may be more fluid. For example, in ecosystems with multiple components, the locus can be shared among the components (and the user). Additionally, the resources over which each component has control in the ecosystem are often unclear and consumers might have difficulty determining if a failure was avoidable. Given the interdependence in a platform ecosystem, this fault ambiguity would be highly likely any time a failure occurs — even error messages purporting to explain the failure may miss the mark or mislead the user, who has few resources available (and likely lacks the time, patience, and necessity) to research root causes of ambiguous failures.

HYPOTHESES AND MODEL

To explore ambiguously sourced failures in mobile platform ecosystems, we first consider how locus, controllability, and stability can be used to attribute blame to ecosystem components and ultimately affect discontinuance recommendations. We then introduce a new characteristic, border strength, which we argue can alter the locus and controllability and thereby affect blame attribution and discontinuance recommendations. Finally, we explore contextual contingencies of disruption severity and goal-directedness and how they can influence blame and discontinuance.

Platform Ecosystem Components

To understand how components within a platform ecosystem may be perceived differently by users, it is important to revisit each component's position in the ecosystem.

Specifically, we draw a distinction between platforms (e.g., the device and OS working together) and modules (e.g., apps) in the ecosystem. When individuals experience failure within a platform ecosystem, they may not be aware of technical reason for the failure, thus possibly obscuring actual locus and controllability. However, users are aware of *their* actions as they interact with ecosystem components and these actions make apps the most likely target for blame attribution and recommended discontinuance should a failure occur. Any ambiguously sourced failure will occur during operation of an app, and, prior to that failure, users will have deliberately opened and used the app. Therefore, the app and its potential role in the failure would be highly salient and locus for the failure would likely be closer to the app. Further, the operating environment of the platform is likely to be common and accessible to all developers who create apps. Therefore, app developers will be attributed a greater degree of control over the unique experience their apps provide. If failure occurs, users will likely contrast the failure with successful operation of other apps in the same ecosystem (which ostensibly had similar control). Since platform components provide similar resources to all apps, the increased locus and controllability for the app would lead to higher blame attributed to app than to other components in the ecosystem stack.

Finally, the purpose of platform components is to create a stable operating environment that facilitates access to and management of the digital resources available in the platform [57]. In comparison to experience with apps, users likely will have had many more interactions with platform components during which failure did not occur. In fact, one of the distinguishing characteristics of platforms and an important reason why developers create apps for the platform is stability [57]. With modules of the ecosystem being ascribed greater locus and controllability and less stability than platform components, we anticipate greater blame and recommended discontinuance in response to ambiguous system failures. Therefore,

> *H1: Mobile platform modules (i.e., apps) will be attributed (a) greater blame and (b) higher discontinuance recommendation after an ambiguously sourced system failure than platform components (i.e., device, OS).*

Border Strength

To consider how attributions of blame and discontinuance recommendations may be altered by organizations supporting ecosystem components, we draw on the concept of the digital border. A *digital border* is the specific boundary around a digital artifact such as a website or an application [14]. The prominence of a digital border has predicted recognition of websites, with consequences resulting from recognition (or non-recognition). These findings have particular salience for branding on the Web; websites with higher borders are more likely recognized and credited for their contributions to a task, potentially leading to greater user loyalty and brand recognition. In the Web context, border recognition and attribution can be influenced by *border strength*, or the extent to which a virtual location is indicated and reinforced (e.g., through notifications, visual cues, or instructions) [14].

Given the findings regarding the effects of border strength in the Web context, we expect that border strength will exhibit a similar effect within more complex mobile platform ecosystems by making the potential locus of the failure more evident and raising awareness of potential sources of controllability. Each component within a platform ecosystem, whether part of the platform or module, has opportunity to better differentiate itself and, thus, strengthen the border between itself and the remainder of the ecosystem. These borders, for instance, might be strengthened through stronger, better-differentiated, and potentially interrupting design choices. An application, for instance, can include a branded “splash screen” to raise the user’s awareness regarding the app’s identity and features within the app may reinforce this identity. Similarly, through design choices, the OS and device may raise greater awareness of themselves. Such designs have been found to strengthen borders within a multi-site Internet session context [14], and we expect that stronger borders will play a similar role within the mobile platform ecosystem. Strengthening of borders in this way will make apparent and distinguish the multiple components that could appear monolithic to users. Increasing the prominence of ecosystem components that could potentially be at fault should an failure occur will facilitate the generation and direct perceptions of locus and controllability. For example, increasing border strength will increase salience of the boundaries of controllability for each component. Strengthened borders will also make more clear which components are in operation at the time of failure and, thus, will likely alter perceptions of locus. Therefore,

> *H2: The border strength separating components within the mobile platform ecosystem will significantly affect (a) the amount of blame assessed the component and (b) the discontinuance recommendation of the component after an ambiguously sourced system failure.*

Goal-Directedness

Among contextual factors, the objective of the task being performed by a user interacting with an ecosystem is critical to consider. A user may perceive the components differently depending on the user's specific activity: finding an answer to a closed-ended question, researching a topic of interest to form an opinion, or passing time in pursuit of hedonic interests. The range of potential activities that can be performed within a digital environment has been described by several typologies. These include hedonic vs. instrumental/utilitarian [e.g., 9, 34, 37, 59], telic vs. paratelic [12], hedonic vs. intrinsic vs. extrinsic system proposed by Lowry et al. [36], and the multi-dimensional task complexity spectrum originally introduced by Campbell [7].

In this work, we conceptualize task using a binary categorization of experiential vs. goal-directed [e.g., 25, 42, 44, 62]. This conceptualization is among the most commonly used and permits exploration of task performance failure in some degree of structure. Further, the conceptualization captures the greatest dichotomy among potential task types that are widely performed using mobile platform ecosystems. Past research using this dichotomy has shown not only its usefulness, but also its effects on user perceptions and intentions related to information systems. Deng and Poole [12] found varying levels of pleasantness were perceived due to interactions between a user's meta-motivational state and the goal-directedness of a task. Nadkarni and Gupta [42] found that goal-directedness affected user satisfaction with an online system when considered along with the system's visual complexity. Finally, Novak et al. [44] found that goal-directedness affected the amount of flow (immersion in an activity) experienced by Web users.

In a similar vein, we anticipate that goal-directedness of a task will also alter the blame and discontinuance recommendations for ecosystem components after an ambiguously sourced failure. Blame occurs in response to actions for which individuals will suffer negative consequences [50]. In contrast to experiential tasks, when failure occurs during goal-directed tasks users are denied achieving a defined their aims and must suffer anticipated consequences. Therefore, when goal-directedness is high, blame attributions based on locus, controllability, and stability are likely to be stronger. For example, in a focused task with a concrete objective, users will be more likely to take note of obstacles and who placed them (i.e., locus, controllability) that prevent them from reaching their objectives. If failure within a platform ecosystem makes goal achievement impossible, users will likely respond negatively by attributing blame. In contrast, users engaged in more experiential tasks would be less likely to note the failure as an obstacle and be more likely to simply move on to other tasks. Therefore,

> *H3: The goal-directedness of the interaction with a mobile platform ecosystem will significantly affect (a) the amount of blame assessed the component and (b) the recommended discontinuance of the component after an ambiguously sourced system failure.*

Disruption Severity

Ecosystem failures can result in a variety of consequences for the user. For example, some failures may cost the user only a few moments of inconvenience, while others may require considerably more time and effort to resolve. Indeed, Galletta et al. [21] found that delay within a website context was a cost which negatively impacted a user's future intentions. Other studies have yielded similar findings, where perceived and actual delays negatively impacted the quality of an experience [16, 54], increased user frustration [8, 48], and hampered system success [45]. As the negative consequences for failure increase we expect to see blame attributions as the result of locus, controllability, and reliability increase. These attributions should also be evident in discontinuance recommendations. Therefore,

> *H4: The disruption severity caused by an ambiguously sourced system failure will (a) increase the amount of blame assessed to components of a mobile platform ecosystem and (b) increase the recommended discontinuance of components.*

Blame Attribution and Continuance

Finally, within the context of a user's interaction with information technology, we expect blame resulting from failure within a mobile platform ecosystem to have important consequences. Indeed, in the ecommerce context and as noted by Tan et al. [56], service failures result in negative consequences for the sites where the failures transpired. A substantial literature has found significant relationships between negative perceptions of a technology and future intentions with regard to that technology [e.g., 27, 37, 43]. We expect a similar relationship to emerge with mobile platform ecosystems. Therefore:

H5: The greater the blame assessed a mobile platform ecosystem component after an ambiguously sourced system failure, the greater the recommended discontinuance of that component.

Model

As described above, our exploration consists of the evaluation of the model shown in Figure 1. We examined blame attribution to and discontinuance recommendations for the device, OS, and app. Consistent with our conceptualization of blame, we also captured blame of self.

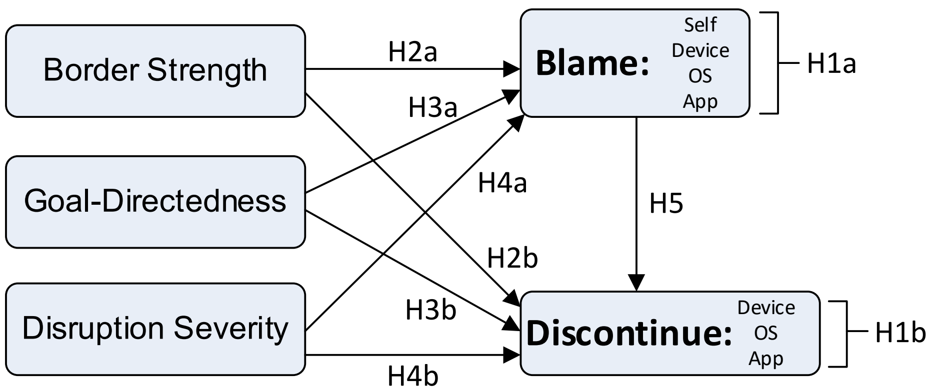

Figure 1. Experimental model for blame and discontinuance

METHOD

To test our hypotheses, we conducted two scenario-based experiments within the mobile platform ecosystem. The first experiment (n = 142) confirmed how attributions are made after failures and explored effects of border strength on discontinuance recommendations for platform components (H1, H2). The second experiment (n = 367) revisited border strength and examined additional effects from goal-directedness and disruption severity on discontinuance (H1-H4). Four pilot studies, including over 500 participants, tested the scenarios and measurement prior to the main experiments. Data were collected in a single session for each participant via the Qualtrics survey system. After providing consent, participants were randomly assigned a condition and were asked to complete a pre-survey including control variables. They were then presented a multi-part scenario (customized by condition) in which they had been given a smartphone (device and OS) with an app that they were expected to use for a new job. While using the smartphone and app, a failure occurred which resulted in the temporary inoperability of the smartphone. After finishing the scenario and completing attention checks, participants then answered questions about components in the mobile platform ecosystem. Appendix A contains the items that were used in the experiments. Appendix B includes the full text and treatment conditions in the scenarios.

Experiment 1

Experiment 1 checked manipulations, revealed attributions in response to failure, and examined the effects of border strength (weak border vs. strong border-unfamiliar app developer vs. strong border-familiar app developer) on discontinuance recommendations (H1, H2).

Participants

Participants were recruited through Amazon's Mechanical Turk (MTurk) service, which has been found to provide participants similar in quality to other frequently-used sources, such as university students and commercial panel recruiting services [e.g., 6, 35, 53]. Participation was limited to United States residents who had completed more than 100 MTurk assignments, but fewer than 1,000 [46]. Participants were compensated $1.00 to complete the experiment. We sourced 200 participants, but 58 participants failed attention checks and comprehension tests. Therefore, our final sample included 142 participants. The mean age of participants was 33.8 (*SD* = 10.7); 60.0% were female, and 59.1% completed at least a bachelor's degree. Every participant reported owning a smartphone and 96.5% of participants reported using their smartphones several times each day.

Independent Variables

Border strength was manipulated in the beginning of the scenario. Participants were given some background about a job they had just started and the expectation of using a company-assigned smartphone and app. As a robustness check for border strength, we tested both familiar and unfamiliar app developers. Teleduke is a fictitious company created for this experiment and Oracle is an established company that is widely known. In the strong borders conditions the brands of the device, OS, and app were named in the scenario, which read:

> *Imagine you have started a job for a new company working in the service department. The company has given you a new smartphone that they expect you to use as your primary mobile device. This particular smartphone is made by Motorola and is the Moto E model. The phone uses the Android operating system developed by Google (version 4.4). Your company has also required you to install and use a third-party app called ComMentor from the Google Play store. This app allows you to monitor and collect data regarding customer comments and was developed by a company called [Teleduke (Unfamiliar app developer)/ Oracle (Familiar app developer)].*

The identity of these brands was then reinforced through attention-check questions, which asked participants to name the smartphone manufacturer, the operating system, and the name of the app developer (only responses where the participant correctly answered these questions were included in the analysis). In the weak borders condition, the background omitted brand names and was followed by questions asking participants to name the department where they worked, where the company's headquarters was located, as well as the kind of building in which the job took place.

Dependent and Control Variables

Following the scenario, participants were given the following prompt:

> *Your company is considering making changes to the smartphone, smartphone operating system, and the app that you used in the scenario. This change would affect you and all other employees in your department. Each of these components could be changed separately (i.e., the company could change smartphone devices, but retain the same operating system and app).*

Following this prompt, participants were asked to recommend whether the company should change the device, the OS, and the app. This recommendation was on a seven-point scale, ranging from strongly recommend against (changing the component) to strongly recommend.

Our investigation also considered several control variables that were included based on past literature indicating that they may affect user attribution of failure with an ambiguous source as well as discontinuance recommendation. Since we used actual brands, we captured attitudes about those brands. Prior to starting the scenario, participants provided their impressions of the smartphone manufacturer, OS, and app brands that would be referenced in the scenario to come. Impressions were captured on a seven-point scale ranging from very negative to very positive [e.g., 13]. The four scenario brands were randomly mixed with eight additional brands to ameliorate priming effects for the scenario brands. Participants then completed scales for propensity to blame, mobile device self-efficacy [1, 41], product involvement [63], and normative and informational susceptibility to interpersonal influence (SII) [2].

Results

To check the theoretical rationale for how borders function to alter attributions following ambiguous failure, we first examined perceptions of locus, controllability, and stability (see Table 1).[1] In the rationale for H1, we argued that perceptions of stability would differ between modules (i.e., app) and the platform (e.g., OS, device). Consistent with this argument, within-subjects comparisons from a repeated analysis of variance (ANOVA) demonstrated differences in the level of stability attributed to the app, OS, and device, $F(2, 280) = 4.126$, $p = .017$. Furthermore, in the rationale for H2, we argued that making borders salient within the platform

[1] Prior to reporting discontinuance recommendations, participants rated the app, OS, and device using the following items taken from <<CITE>>. "The following questions concern the [app, OS, device]. The problem you read about above is something…" Locus: "That reflects an aspect about the [app, OS, device]" … "That reflects something about the situation"; Controllability: "That the [app, OS, device] can regulate" … "That the [app, OS, device] cannot regulate"; Stability: "That is stable over time" … "That varies over time". All items were on a 7-pt scale.

ecosystem would alter locus and controllability. Consistent with this argument, between-subjects comparisons from a repeated ANOVA demonstrated that border strength significantly affected locus, $F(2, 139) = 4.304$, $p = .015$. However, border strength did not affect controllability, $F(2, 139) = 1.605$, $p = .205$. Therefore, effects from border strength were associated with locus, not controllability.

		Component		
Border Conditions	**Attributions**	**Device Mean (SD)**	**OS Mean (SD)**	**App Mean (SD)**
Weak Border	Stability	4.93 (1.52)	4.63 (1.43)	5.24 (1.37)
	Locus	3.90 (1.84)	3.71 (1.42)	2.71 (1.42)
	Controllability	3.98 (1.59)	3.22 (1.28)	3.17 (1.55)
Strong Border – Teleduke	Stability	4.69 (1.49)	4.78 (1.45)	5.24 (1.24)
	Locus	4.73 (1.62)	4.58 (1.64)	2.75 (1.60)
	Controllability	4.49 (1.67)	4.00 (1.82)	2.84 (1.63)
Strong Border – Oracle	Stability	5.15 (1.33)	4.80 (1.52)	4.98 (1.42)
	Locus	4.93 (1.61)	3.89 (1.72)	2.76 (1.61)
	Controllability	4.33 (1.78)	3.74 (1.72)	3.13 (1.71)

Table 1. Experiment 1 stability, locus, and controllability mean values.

Following checks of attribution, we then examined our hypotheses by testing effects of borders on discontinuance. Means of discontinuance recommendations are shown in Table 2. To test H1, we performed a repeated ANOVA to compare the discontinuance recommendations for the app, OS, and device. The repeated ANOVA accounted for the nonindependence of observations and demonstrated significant differences between ecosystem components, $F(2, 278) = 38.342$, $p < .001$, $\eta_p^2 = 0.22$. In support of H1, post-hoc tests with a Bonferroni correction for repeated tests demonstrated that discontinuance recommendations were higher for the app than they were for the OS and device (both at p < .001).

Treatment	**Conditions**	**N**	**Mean Device Discontinuance (SD)**	**Mean OS Discontinuance (SD)**	**Mean App Discontinuance (SD)**
Borders	Weak Border	46	3.89 (1.55)	4.41 (1.57)	5.76 (1.10)
	Strong Border – Teleduke	55	4.65 (1.61)	4.69 (1.67)	5.84 (1.21)
	Strong Border – Oracle	41	4.56 (1.52)	5.17 (1.36)	5.51 (1.05)

Total	143	4.38 (1.59)	4.74 (1.57)	5.72 (1.13)

Table 2. Experiment 1 discontinuance recommendation model mean values.

To test H2, we performed a multivariate analysis of covariance (MANCOVA) with border strength as the independent variable, discontinuance recommendations for each component as dependent variables, and control variables as covariates. Since discontinuance recommendations among ecosystem components are conceptually related, MANCOVAs are appropriate analysis techniques [40]. The complete results from the MANCOVA are presented in Appendix C. Multivariate tests, $F(6, 258) = 2.083$, $p = .056$, indicated a significant effect for border strength. Consistent with H2, follow up univariate tests demonstrated significant effects from borders on continuance recommendations for the OS, $F(2, 130) = 4.027$, $p = .020$, $\eta_p^2 = 0.06$, and for the device, $F(2, 130) = 3.127$, $p = .047$, $\eta_p^2 = 0.05$. Post hoc tests with a Bonferroni correction revealed that strong borders resulted in higher discontinuance recommendations the OS (Strong Border-Oracle compared with Weak Boarder: $p = .014$) and device (strong border-teleduke compared with weak boarder: $p = .057$).

Several covariates demonstrated a significant influence on the dependent variables; therefore, abbreviated significant results are reported next. Brand impressions of the device ($p = .017$) and the app developer (Teleduke: $p = .008$) influenced discontinuance of the app. Propensity to blame ($p = .021$), product involvement ($p = .036$), and SII (normative) ($p = .044$) influenced discontinuance of the OS. Full results are reported in Appendix C.

Experiment 2

Experiment 2 expanded on Experiment 1 and tested H1 – H4 and followed a 3 (border strength: weak border vs. strong border-teleduke vs. strong border-oracle) × 2 (goal-directedness: experiential vs. goal-directed) × 2 (disruption severity: low vs. high) factorial design

Participants

Participants were recruited through Amazon's Mechanical Turk (MTurk) service. We sourced 480 participants via MTurk with the same restrictions and incentive as in Experiment 1. Removing participants who did not complete the experiment and those who failed these attention checks resulted in $n = 367$, with 29 or more participants in each cell. The mean age of participants was 33.2 (SD = 8.8), 61.3% were female, and 52.4% completed at least a bachelor's degree. Every participant reported owning a smartphone and 94.0% of participants reported using their smartphones several times each day.

Independent Variables

In Experiment 2, border strength was manipulated in the same manner as in Experiment 1 and participants who failed to correctly identify the app developer, OS developer, or device manufacturer were excluded. Following border strength, goal-directedness was manipulated and included two conditions: goal-directed or experiential [42, 44]. In the goal-directed condition, the scenario continued by describing a circumstance in which the participant was asked to use the smartphone and app to complete a goal-directed task (finding examples of an employee's work as part of an award nomination process as in Experiment 1). In the experiential condition, the scenario continued by describing a circumstance in which the participant looked for entertaining exchanges between customers and support employees. Participants in the goal-directed condition reported that in the scenario they had more of a distinct, identifiable purpose ($M = 6.27$, $SD = .91$; $t(362) = 15.20$, $p < 0.001$) and looked up more specific information ($M = 6.31$, $SD = .96$; $t(365) = 16.15$, $p < 0.001$) than participants in the experiential condition (purpose $M = 4.24$, $SD = 1.56$; specific information $M = 4.03$, $SD = 1.65$).[2] These significant differences indicated a successful manipulation for goal-directedness.

[2]Manipulation check items for goal-directedness and disruption severity were measured on a 7-point Likert type scale with Strongly Disagree and Strongly Agree as endpoints.

Lastly, participants were also assigned to either a low or high disruption severity condition. In both conditions, the scenario described an ambiguous failure in the mobile platform ecosystem during which the app, OS, and device froze and became unresponsive during the task. In the low disruption severity condition, participants were told that after freezing, they restarted the smartphone and it became operable again. In the high disruption severity condition, however, participants were told:

> *You cannot get the phone to turn off and restart. After taking the phone to your company's IT group, it takes three days to get your phone back in working order, during which time you miss several important calls from your boss, who is out of the country.*

Indicating a successful manipulation, participants in the high disruption severity condition reported their disruption as more severe (M = 5.75, SD = 1.15; $t(365)$ = 13.15, p < 0.001) and serious (M = 5.88, SD = 1.18; $t(365)$ = 13.00, p < 0.001) than participants in the low disruption condition (severe M = 3.82, SD = 1.61; serious M = 3.95, SD = 1.63).

Dependent Variables

The dependent variables and control variables in Experiment 2 mirrored those from Experiment 1 and included continuance recommendations as the dependent variable and attitudes about the smartphone manufacturer, OS, and app brands, propensity to blame, mobile device self-efficacy [1, 41], product involvement [63], and normative and informational SII [2] comprised the control variables.

Analysis

The analysis approach for Experiment 2 mirror the approach for Experiment 1. Descriptive statistics discontinuance recommendations are shown in Table 3.

Treatment	**Conditions**	**n**	**Mean Device Discontinuance (SD)**	**Mean OS Discontinuance (SD)**	**Mean App Discontinuance (SD)**

Borders	Weak Border	123	4.04 (1.339)	4.48 (1.276)	5.59 (1.145)
	Strong Border – Teleduke	123	4.64 (1.466)	4.07 (1.524)	5.60 (1.122)
	Strong Border – Oracle	121	4.53 (1.461)	4.28 (1.629)	5.49 (1.239)
Goal-Directedness	Experiential	184	4.33 (1.340)	4.36 (1.434)	5.56 (1.172)
	Goal-Directed	183	4.48 (1.540)	4.19 (1.541)	5.56 (1.165)
Disruption Severity	Low	182	4.30 (1.411)	4.18 (1.462)	5.34 (1.289)
	High	185	4.50 (1.471)	4.37 (1.513)	5.77 (.990)
Total		367	4.40 (1.443)	4.28 (1.489)	5.56 (1.167)

Table 3. Experiment 2 discontinuance recommendation model mean values.

To test H1, we conducted a repeated ANOVA comparing the discontinuance recommendations among the three components (device, OS, and app). Results demonstrated that, like in Experiment 1, discontinuance recommendations in Experiment 2 also differed across components, $F(2, 365) = 113.469$, $p < .001$, $\eta_p^2 = 0.38$. Post-hoc tests with a Bonferroni correction indicated that discontinuance recommendations for device and OS did not differ from each other ($p = .480$). However, as shown by the total means in Table 2, discontinuance recommendations for the app were higher than recommendations for the device ($p < .001$) and for the OS ($p < .001$). These findings replicate support for H1.

To test H2, H3, and H4, a MANCOVA was performed using border strength, goal-directedness, and disruption severity as independent variables, discontinuance recommendations for the device, OS, and app as the dependent variables, and control variables as covariates. The complete results from the both MANCOVAs are presented in Appendix C.

Multivariate tests indicated significant main effects for border strength, $F(6, 690) = 3.745$, $p = .001$, and for disruption severity, $F(3, 344) = 3.897$, $p = .009$. The lack of significant effects from goal-directedness failed to support H3. Univariate tests indicated a significant main effect of border strength on recommended discontinuance for the manufacturer, $F(2, 346) = 4.264$, $p = .015$, $\eta_p^2 = 0.02$, and for the OS, $F(2, 346) = 3.770$, $p = .024$, $\eta_p^2 = 0.02$. These findings are consistent with H2. Additionally, there was a significant main effect of disruption severity on discontinuance recommendations for the app, $F(1, 346) = 10.537$, $p = .001$, $\eta_p^2 = 0.03$. Post hoc tests with a Bonferroni correction revealed that those in the weak border condition reported lower manufacturer discontinuance recommendations than those in both strong border conditions (Oracle: $p = .054$; Teleduke: $p = .023$). But, those in the strong border Teleduke condition reported lower OS discontinuance recommendations than those in the low border condition ($p = .019$). Finally, consistent with H4, those in the severe disruption condition reported higher discontinuance recommendations for the app ($p = .001$).

Abbreviated significance tests are reported for significant covariates. Higher impressions of Android decreased the discontinuance recommendations for the OS ($p = .001$). SII (informational) increased the discontinuance recommendations for manufacturer ($p = .036$) and OS ($p = .011$). Full results are reported in Appendix C.

DISCUSSION

The objective of this paper was to explore the attribution of responsibility after an ambiguous failure in a platform ecosystem as well as understand the consequences from such attributions. The results (see Table <<REF>>) provide several important theoretical and practical advances regarding attribution and discontinuance recommendations. We discuss each below.

Hypotheses	**Experiment 1 Results**	**Experiment 2 Results**
H1: Mobile platform modules (i.e., apps) will be attributed higher discontinuance recommendation after an ambiguously sourced system failure than platform components (i.e., device, OS).	Supported	Supported
H2: The border strength separating components within the mobile platform ecosystem will significantly affect the discontinuance recommendation of the component after an ambiguously sourced system failure.	Supported	Supported
H3: The goal-directedness of the interaction with a mobile platform ecosystem will significantly affect the recommended discontinuance of the component after an ambiguously sourced system failure.	-	Not Supported
H4: The disruption severity caused by an ambiguously sourced system failure will increase the recommended	-	Supported

discontinuance of components.		

Contributions to Theory

First among the contributions of this paper is the formalization of ambiguous failures in platform ecosystems. Although previous research has explored consequences related to IT systems failure [e.g., 26, 56], none to our knowledge has explored the effects of failure in platforms where the source of the failure is unclear. Platforms and modules are designed to integrate seamlessly, but it is critical to understand the consequences should integration fail. Platform ecosystems are becoming increasingly prevalent for both consumers and organizations [57] . More and more, companies must compete within the context of their platform membership [58]. With the interconnectedness of components within such an ecosystem, responsibility for ecosystem function is distributed across multiple components, including both platform components and modules. Prior work [24] has treated in isolation perceptions of the form and function of various components in digital ecosystems. Our study provides evidence that a focus on a single ecosystem component may neglect critical aspects regarding how consumers actually use and experience mobile platform ecosystems.

Second, with the prevalence of mobile platform ecosystems and their attempts at tight integration, ambiguous failure of one or more components of the ecosystem is likely to be a recurring issue. When failures arise, how do users attribute responsibility? Both Experiment 1 and Experiment 2 demonstrated that when ecosystem failure occurs, the app was recommended much more strongly than other platform components for discontinuance. Yet, this finding was also was intriguing because the app was also attributed much *less* locus and controllability than the OS or device (see Table 1). According to attribution theory, greater locus and controllability are typically needed for assignment of greater responsibility. Yet, this was not the case with ambiguous failure within platform ecosystems. Instead, the last dimension of attribution theory offers clues to how responsibility is assigned to the app. Users perceived the app as less stable than other components in the ecosystem and on this basis, were more likely to assign greater responsibility for failure. The implications of this finding suggest that the three dimensions of attribution theory are not equally weighted when determining fault for ambiguous ecosystem failure. In this case, stability may be more important than locus or controllability.

Although the app bears the brunt of negative consequences during ecosystem failure, we show that the OS and device are not absolved of culpability. Results suggest that following ambiguous failure, multiple parties share in the perceived responsibility. Across conditions, discontinuance recommendations were at or above the midpoint for all ecosystem components, implying that failure of the ecosystem, regardless of the component originating the failure, will have a negative impact on all components in the ecosystem. These findings suggest that digital ecosystems may often be at the mercy of the weakest component used by the consumer.

Third, we find that design elements (e.g., borders) and contextual factors (e.g., disruption severity) are important contingencies in the attribution of responsibility for failure subsequent discontinuance recommendations. Although the app remained the most likely to be discontinued regardless of border condition, results of both experiments demonstrated that borders altered discontinuance recommendations for the digital platform itself (not ecosystem modules). In Experiment 1, salient borders increased discontinuance for the OS and device. Experiment 2 replicated the findings for the device, but border salience was shown to *decrease* discontinuance for the OS. These findings support the idea that the digital platform (e.g., OS, device) is most susceptible to the effects of borders and that the device manufacturer faces greater negative consequences from failure when borders are salient. However, the effects of borders for the OS are complex and warrant additional attention. Results regarding how attributions are made may offer clues to a potential explanation: Experiment 1 demonstrated that the effect of borders was most closely associated with locus (and not controllability). In other words, the effect of borders appears to operate more through estimates of direct culpability and less through estimates of the capacity and intention to avoid failure. Other researchers have argued that the OS is the central component of platform [57] and its boundaries may be obscured. Positioning of the locus in response to failure may be a more difficult (and ambiguous) task for the OS than for the other components in an ecosystem, but this speculation requires additional research.

Among the two contextual factors we examined, effects from disruption severity were more pronounced. We uncovered no effects on discontinuance recommendations from goal-directedness, which suggests individuals are likely to harbor similar attitudes about discontinuance regardless of the task they were performing at the time of failure. Consistent with our expectations, disruption severity increased discontinuance recommendations for the app, but the effect did not spread to the OS or device. This finding implies that, in addition to the tendency for the app to be most likely discontinued following failure, the consequences from severity of the disruption (as tested here) also fall disproportionately on the app.

Implications for Practice

Should ambiguous failure of a mobile platform ecosystem occur, blame is shared among all components of the ecosystem. A better integrated system that experiences fewer faults, therefore, benefits all members of the ecosystem. This finding supports a tighter integration among all components of the stack to create a more functionally problem-free system where good apps run confidently on an easy-to-develop-for OS that is then run on hardware devices well equipped to handle the requirements of both the OS and the apps that may be available for it. In fact, some operating systems may already be taking pains to ensure this occurs, for instance Microsoft has reportedly created specific hooks within its Windows PC OS to facilitate successful interaction with certain software and hardware vendor products [52].

When ambiguous failures arise, app developers should be aware that their products are most likely to be held responsible and other components of the ecosystem (e.g., OS and device) are more insulated from responsibility. Apps are perceived as less stable compared to other components in the ecosystem and the worse the disruption, the greater the likelihood users will opt to discontinue using the app. Therefore, in the eyes of users, app developers are likely to bear the largest portion of responsibility for delivering a problem-free experience. As the level of attributed responsibility grows, so too will a disproportionate incentive for tight and robust integration between apps and platform components. Fortunately for app developers, the type of app or what users are doing with the app (experiential or goal-directed) seems to matter less than other failure contingencies.

Finally, design decisions that make borders between ecosystem components salient alter discontinuance recommendations. Salient borders prior to failure will harm attitudes toward the device, but its effects are mixed toward the OS. Thus, device manufacturers and OS developers may be incentivized to obscure borders in situations where probability of failure is high. App developers, on the other hand, may be incentivized to promote borders to differentiate themselves from other app developers and other components in the ecosystem, but also to spread responsibility should failure occur.

Limitations and Future Research

There are some important limitations to our findings. First, we chose one of several ecosystems (mobile) and taxonomies for understanding goal-directedness. While this enabled us to ground our scenarios in familiar contexts and have distinct differentiation between experimental conditions, other ecosystems and taxonomies could present different outcomes in response to failure. Therefore, replication with other ecosystems and task taxonomies is recommended. Additionally, while we found significant main effects for border strength, our manipulation was simplified as a result of the scenario-based data collection approach. Other, real-world attempts at creating border strength may have amplified effects on the results and should be a subject for future inquiry. Finally, our data collection was based entirely on a scenario-based experiment in which participants had to pretend to have participated in the events described to them. While we found significant results, we expect that these results were dampened by the requirement to imagine the experience. Future research may find even more pronounced results from an experimental or archival dataset based on actual failure experienced by participants.

Conclusion

Platform ecosystems continue to grow in prominence both for organizations and consumers. The markets created by these ecosystems also continue to grow, despite possible complications caused by an increased level of interconnectedness among components within these systems. Our study has made significant contributions to the understanding of such systems, particularly when ambiguous failures occur within them. Further, we have uncovered interesting results regarding the nature of user perceptions of components within mobile platform ecosystems in the context of a system failure. As such ecosystems proliferate, the relevance and importance of this research for component creators and consumers will increase.

REFERENCES

1. Barbeite, F.G. and Weiss, E.M. Computer self-efficacy and anxiety scales for an Internet sample: testing measurement equivalence of existing measures and development of new scales. *Computers in Human Behavior*, *20*, 1 (2004), 1–15.

2. Bearden, W.O., Netemeyer, R.G., and Teel, J.E. Measurement of consumer susceptibility to interpersonal influence. *Journal of Consumer Research*, *15*, 4 (1989), 473–481.

3. Benlian, A. Service quality in software-as-a-service: Developing the SaaS-Qual measure and examining its role in usage continuance. *Journal of Management Information Systems*, *28*, 3 (2011), 85–126.

4. Bhattacherjee, A. and Hikmet, N. Physicians' resistance toward healthcare information technology: a theoretical model and empirical test. *European Journal of Information Systems*, *16*, 6 (2007), 725–737.

5. Bitner, M.J. The service encounter: diagnosing favorable and unfavorable incidents. *Journal of Marketing*, *54*, 1 (1990), 71–84.

6. Buhrmester, M. and Kwang, T. Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, *6*, 1 (2011), 3–5.

7. Campbell, D.J. Task complexity: A review and analysis. *Academy of Management Review*, *13*, 1 (1988), 40–52.

8. Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., and Shneiderman, B. Determining causes and severity of end-user frustration. *International journal of human-computer interaction*, *17*, 3 (2004), 333–356.

9. Cyr, D., Head, M., Larios, H., and Pan, B. Exploring human images in website design: a multi-method approach. *MIS Quarterly*, *33*, 3 (2009), 539–566.

10. Darke, P.R., Ashworth, L., and Main, K.J. Great expectations and broken

promises: Misleading claims, product failure, expectancy disconfirmation and consumer distrust. *Journal of the Academy of Marketing Science*, *38*, 3 (2010), 347–362.

11. DeLone, W.H. and McLean, E.R. The DeLone and McLean model of information systems success: A ten-year update. *Journal of management information systems*, *19*, (2003), 9–30.

12. Deng, L. and Poole, M.S. Affect in web interfaces: A study of the impacts of web page visual complexity and order. *MIS Quarterly*, *34*, 4 (2010), 711–730.

13. Dodds, W.B. and Monroe, K.B. The effect of brand and price information on subjective product evaluations. *NA-Advances in Consumer Research Volume 12*, (1985).

14. Dunn, B.K., Ramasubbu, N., Galletta, D.F., and Lowry, P.B. Digital Borders, Location Recognition, and Experience Attribution within a Digital Geography. *Journal of Management Information Systems*, *36*, 2 (2019), 418–449.

15. Dwivedi, Y.K., Wastell, D., Laumer, S., et al. Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, *17*, 1 (2015), 143–157.

16. Egger, S., Reichl, P., Hoßfeld, T., and Schatz, R. “Time is bandwidth”? Narrowing the gap between subjective time perception and Quality of Experience. In IEEE, 2012, pp. 1325–1330.

17. Eisenmann, T., Parker, G.G., and Van Alstyne, M.W. Strategies for Two-Sided Markets. *Harvard Business Review*, *84*, 10 (2006), 92–102.

18. Fiske, S.T. and Taylor, S.E. *Social cognition*. McGraw-Hill, New York, NY, 1991.

19. Flowers, S. *Software failure: Management failure*. John Wiley, Chichester, UK,

1996.

20. Folkes, V.S. Consumer reactions to product failure: An attributional approach. *Journal of consumer research*, *10*, 4 (1984), 398–409.

21. Galletta, D.F., Henry, R.M., McCoy, S., and Polak, P. When the wait isn't so bad: The interacting effects of website delay, familiarity, and breadth. *Information Systems Research*, *17*, 1 (2006), 20–37.

22. Heider, F. *The psychology of interpersonal relations*. Wiley, New York, NY, 1958.

23. Hirschheim, R. and Newman, M. Information systems and user resistance: theory and practice. *The Computer Journal*, *31*, 5 (1988), 398–408.

24. Hoehle, H. and Venkatesh, V. Mobile Application Usability: Conceptualization and Instrument Development. *MIS Quarterly*, *39*, 2 (2015).

25. Hoffman, D.L. and Novak, T.P. Marketing in hypermedia computer-mediated environments: Conceptual foundations. *Journal of Marketing*, *60*, 3 (1996), 50–68.

26. Holloway, B.B. and Beatty, S.E. Service failure in online retailing a recovery opportunity. *Journal of service research*, *6*, 1 (2003), 92–105.

27. Hu, P.J., Brown, S.A., Thong, J.Y., Chan, F.K., and Tam, K.Y. Determinants of service quality and continuance intention of online services: The case of eTax. *Journal of the American Society for Information Science and Technology*, *60*, 2 (2009), 292–306.

28. Huber, T.L., Kude, T., and Dibbern, J. Governance Practices in Platform Ecosystems: Navigating Tensions Between Cocreated Value and Governance Costs. *Information Systems Research*, *28*, 3 (2017), 563–584.

29. Kähr, A., Nyffenegger, B., Krohmer, H., and Hoyer, W.D. When hostile

consumers wreak havoc on your brand: The phenomenon of consumer brand sabotage. *Journal of marketing*, *80*, 3 (2016), 25–41.

30. Keil, M. Pulling the plug: software project management and the problem of project escalation. *MIS Quarterly*, *19*, 4 (1995), 421–447.

31. Keil, M., Cule, P.E., Lyytinen, K., and Schmidt, R.C. A framework for identifying software project risks. *Communications of the ACM*, *41*, 11 (1998), 76–83.

32. Kelley, H.H. and Michela, J.L. Attribution theory and research. *Annual Review of Psychology*, *31*, 1 (1980), 457–501.

33. Klaus, T. and Blanton, J.E. User resistance determinants and the psychological contract in enterprise system implementations. *European Journal of Information Systems*, *19*, 6 (2010), 625–636.

34. Li, X., Hsieh, J.P.-A., and Rai, A. Motivational differences across post-acceptance information system usage behaviors: An investigation in the business intelligence systems context. *Information Systems Research*, *24*, 3 (2013), 659–682.

35. Lowry, P.B., D'Arcy, J., Hammer, B., and Moody, G.D. "Cargo Cult" science in traditional organization and information systems survey research: A case for using nontraditional methods of data collection, including Mechanical Turk and online panels. *The Journal of Strategic Information Systems*, *25*, 3 (2016), 232–240.

36. Lowry, P.B., Gaskin, J., and Moody, G.D. Proposing the Multi-Motive Information Systems Continuance Model (MISC) to Better Explain End-User System Evaluations and Continuance Intentions. *Journal of the Association for Information Systems*, *16*, 7 (2015), 515–579.

37. Lowry, P.B., Gaskin, J., Twyman, N.W., Hammer, B., and Roberts, T.L.

Proposing the hedonic-motivation system adoption model (HMSAM) to increase understanding of adoption of hedonically motivated systems. *Journal of the Association for Information Systems*, *14*, 11 (2013), 617–671.

38. Lyytinen, K. and Hirschheim, R. Information systems failures—a survey and classification of the empirical literature. In Oxford University Press, Inc., 1988, pp. 257–309.

39. Maxham, J.G. and Netemeyer, R.G. Modeling customer perceptions of complaint handling over time: The effects of perceived justice on satisfaction and intent. *Journal of Retailing*, *78*, 4 (2002), 239–252.

40. Meyers, L.S., Gamst, G., and Guarino, A.J. *Applied multivariate research: Design and interpretation*. Sage Publications, Thousand Oaks, CA, 2006.

41. Murphy, C.A., Coover, D., and Owen, S.V. Development and validation of the computer self-efficacy scale. *Educational and Psychological measurement*, *49*, 4 (1989), 893–899.

42. Nadkarni, S. and Gupta, R. A task-based model of perceived website complexity. *MIS Quarterly*, *31*, 3 (2007), 501–524.

43. Nah, F.F.-H., Eschenbrenner, B., and DeWester, D. Enhancing brand equity through flow and telepresence: A comparison of 2D and 3D virtual worlds. *MIS Quarterly*, *35*, 3 (2011), 731–747.

44. Novak, T.P., Hoffman, D.L., and Duhachek, A. The influence of goal-directed and experiential activities on online flow experiences. *Journal of Consumer Psychology*, *13*, 1 (2003), 3–16.

45. Palmer, J.W. Web site usability, design, and performance metrics. *Information Systems Research*, *13*, 2 (2002), 151–167.

46. Peer, E., Vosgerau, J., and Acquisti, A. Reputation as a sufficient condition for data quality on Amazon Mechanical Turk. *Behavior Research Methods*, *46*, 4 (2014), 1023–1031.

47. Petter, S. Information systems success: The quest for the independent variable. *Journal of Management Information Systems*, *29*, 4 (2013), 7–62.

48. Rose, G.M., Meuter, M.L., and Curran, J.M. On-line waiting: The role of download time and other important predictors on attitude toward e-retailers. *Psychology & Marketing*, *22*, 2 (2005), 127–151.

49. Russell, D. The causal dimension scale: A measure of how individuals perceive causes. *Journal of Personality and Social Psychology*, *42*, 6 (1982), 137–145.

50. Shaver, K.G. *The attribution of blame: Causality, responsibility, and blameworthiness*. Springer Verlag, New York, NY, 1985.

51. Song, S., Sheinin, D.A., and Yoon, S. Effects of product failure severity and locus of causality on consumers' brand evaluation. *Social Behavior and Personality: an international journal*, *44*, 7 (2016), 1209–1221.

52. Spolsky, J. *Joel on Software*. Apress, Berkeley, CA, 2004.

53. Steelman, Z., Hammer, B.I., and Limayem, M. Data collection in the digital age: Innovative alternatives to student samples. *MIS Quarterly*, *38*, 2 (2014), 355–378.

54. Strohmeier, D., Jumisko-Pyykkö, S., and Raake, A. Toward task-dependent evaluation of web-QoE: Free exploration vs."who ate what?" In IEEE, 2012, pp. 1309–1313.

55. Strong, D.M. and Volkoff, O. Understanding Organization—Enterprise System Fit: A Path to Theorizing the Information Technology Artifact. *MIS quarterly*, (2010), 731–756.

56. Tan, C.-W., Benbasat, I., and Cenfetelli, R.T. An Exploratory Study of the

Formation and Impact of Electronic Service Failures. *MIS Quarterly*, *40*, 1 (2016), 1–29.

57. Tiwana, A. *Platform ecosystems: Aligning architecture, governance, and strategy*. Elsevier, Wlatham, MA, 2014.

58. Tiwana, A., Konsynski, B., and Bush, A.A. Research Commentary--Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics. *Information Systems Research*, *21*, 4 (2010), 675–687.

59. Van der Heijden, H. User acceptance of hedonic information systems. *MIS Quarterly*, *28*, 4 (2004), 695–704.

60. Weiner, B. *Human Motivation*. Holt, Rinehart & Winston, New York, NY, 1980.

61. Weiner, B. Attributional thoughts about consumer behavior. *Journal of Consumer Research*, *27*, 3 (2000), 382–387.

62. Wells, J.D., Fuerst, W.L., and Palmer, J.W. Designing consumer interfaces for experiential tasks: An empirical investigation. *European Journal of Information Systems*, *14*, 3 (2005), 273–287.

63. Zaichkowsky, J.L. Measuring the involvement construct. *Journal of Consumer Research*, *12*, 3 (1985), 341–352.