Sustaining Complement Quality for Digital Product Platforms

Hilbolling, Susan; Berends, Hans; Deken, Fleur; Tuertscher, Philipp

published in
Journal of Product Innovation Management
2021

DOI (link to publisher)
10.1111/jpim.12555

document version
Publisher's PDF, also known as Version of record

J PROD INNOV MANAG 2020;0(0):1–28

DOI: 10.1111/jpim.12555

Sustaining Complement Quality for Digital Product Platforms: A Case Study of the Philips Hue Ecosystem

Susan Hilbolling, Hans Berends, Fleur Deken, and Philipp Tuertscher

Innovation in a digital world increasingly revolves around open platforms that consist of a core technology and a large variety of complementary products developed by an ecosystem of independent complementors. The platform ecosystem literature has mainly focused on indirect network effects arising from the quantity of complements, with little attention to the quality of complements, despite the importance of quality for the complementary value that drives platform ecosystems. Because digital products are malleable and dependent on the ever-evolving ecosystem, we advance a relational and dynamic conceptualization of complement quality. Drawing on a systematic, in-depth qualitative case study of the Philips Hue connected lighting platform and its complementary third-party apps, we study how and why complement quality is sustained over time. By analyzing apps and their updates, we developed a process model that explains pathways through which complement quality is enhanced, maintained, or deteriorates. Changes in the platform core, changes in other ecosystem elements, and idiosyncratic connections by users result in expanding affordances, materializing glitches, and emerging obsolescence. Without further action, glitches and obsolescence lead to deteriorating quality. Joint action of complementors, platform owners, and users is needed to act upon affordances, glitches, and obsolescence, in order to maintain integrity and enhance functionality. This paper contributes to the literature on innovation in platform ecosystems by explaining the dynamic and relational nature of complement quality in a digital platform ecosystem and showing the interdependence of ecosystem members (the triad between platform owner, complementors, and users) in sustained development efforts.

Practitioner Points

- The quality of a platform complement (e.g., a third-party app) concerns how well it interoperates with the core platform and the additional functionality it provides.
- In digital product platforms, complement quality is not fixed, but changes as the platform and the ecosystem evolve over time. Ecosystem dynamics lead to new affordances of complements, glitches in their functioning, and the obsolescence of complements.
- To sustain quality, platform owners depend on the efforts of complementors and other ecosystem actors (e.g., users), over whom they have limited or no control. Platform owners can use indirect measures to alleviate the burden of maintaining complements and create new opportunities for ongoing innovation.
- Platform owners should not only be concerned about attracting new complementors, but also ensure these complementors stay engaged over time.

Address correspondence to: Susan Hilbolling, Department of Management, Aarhus School of Business and Social Sciences, Aarhus University, Fuglesangs Allé 4, 8210 Aarhus, Denmark. E-mail: susan@mgmt.au.dk.

Introduction

The digital age has brought a new innovation paradigm to the forefront: organizing product development in platform ecosystems (Gawer and Cusumano, 2014; Mäkinen, Seppänen, and Ortt, 2014; Nambisan, Lyytinen, and Majchrzak, 2017). The value of platform ecosystems depends on the development of complementary products and services by independent complementors, such as third-party developers that create apps for smartphones (Boudreau, 2012; Eaton, Elaluf-Calderwood, Sørensen, and Yoo, 2015; Tiwana, Konsynski, and Bush, 2010). A core assumption in the literature on platform ecosystems is that platform value is driven by self-reinforcing indirect network effects: the availability of more complements attracts more users, and more users attract more complementors. Thus, a key challenge for platforms is to attract complementors to rapidly realize

a large *quantity* of complements (e.g., Boudreau and Jeppesen, 2015; Jacobides, Cennamo, and Gawer, 2018; Katz and Shapiro, 1986). Researchers have only recently pointed out that the *quality* of complements also affects the value of platforms (e.g., Cennamo, 2018; McIntyre, Srinivasan, Afuah, Gawer, and Kretschmer, 2020), showing that not all complements are equally valuable.

So far, however, we know little about how complement quality evolves over time. Complement quality refers to the *functionality* that a specific complement adds to a platform and how smoothly it *interoperates* with the platform core (Cennamo, Ozalp, and Kretschmer, 2018; Cennamo and Santalo, 2019). The few available empirical studies on complement quality treated it as a fixed characteristic, as, for instance, reflected in an average customer review score (Binken and Stremersch, 2009; Cennamo, 2018; Gretz, Malshe, Bauer, and Basuroy, 2019). Other studies documented how platform owners can ensure complement quality at the time of their introduction (e.g., Ghazawneh and Henfridsson, 2013; Tiwana, 2015a). Such a static conceptualization is problematic, however, specifically in light of *digital* platforms and complements.

In this paper, we advance a relational and dynamic conceptualization of complement quality in digital platform ecosystems, emphasizing that quality emerges from interactions between social and technical aspects over time (Garud, Tuertscher, and Van de Ven, 2013). The generative, reprogrammable, and connected nature of digital technologies calls for such a conceptualization. Because the value of digital platforms and complements depends on technical connections for enabling interoperability, digital products need to be considered relationally, that is, in connection to other products and services (Adner, Puranam, and Zhu, 2019; Henfridsson, Nandhakumar, Scarbrough, and Panourgias, 2018). Furthermore, the reprogrammable nature of digital platforms and complements implies their continued evolution, such as newly added features, after being launched (Yoo, Boland, Lyytinen, and Majchrzak, 2012). For example, Tesla cars were remotely updated with a new autopilot functionality. These ongoing dynamics enable new combinations that may impact the functionality that a complement adds to the platform, as well as its interoperability. A relational and dynamic conceptualization also considers the social interactions between actors involved. To ensure complement quality over time, platform owners depend on the sustained efforts of their complementors, over whom they have no direct control. We address this challenge through the following research question: *How and why is the quality of existing complementary products in digital platform ecosystems sustained over time*?

We performed an inductive, in-depth field study of the ecosystem around the Philips Hue smart lighting platform. Philips Hue is a consumer LED lighting system that can be controlled from a smart device. Besides the official Hue app, hundreds of complementary third-party apps have been developed for the Hue platform. Informed by theory on platform ecosystems, we collected data from multiple sources, including interviews with and observations of the Philips Hue team, archival app data, and interviews with third-party app developers. By systematically comparing complementary products (i.e., apps and their updates

BIOGRAPHICAL SKETCHES

Susan Hilbolling is a Postdoctoral researcher at the Department of Management, Aarhus School of Business and Social Sciences, Aarhus University (Denmark). She obtained her Ph.D. at the KIN Center for Digital Innovation, Vrije Universiteit Amsterdam and holds a Master of Science degree in Strategic Product Design from Delft University of Technology, The Netherlands. Her research interests include the dynamics of collaborative multiparty (digital) innovation, platform ecosystems, and process research.

Hans Berends is Professor of Innovation and Organization at the KIN Center for Digital Innovation, School of Business and Economics, Vrije Universiteit Amsterdam. He received his Ph.D. from Eindhoven University of Technology for a dissertation on knowledge sharing in industrial research. His research interests concern processes and practices of innovation and interorganizational collaboration, focusing in particular on digital innovation. His work has appeared in Academy of Management Journal, Organization Science, Organization Studies, Journal of Product Innovation Management, among others.

Fleur Deken is Associate Professor of Innovation and Organization at the KIN Center for Digital Innovation, School of Business and Economics, Vrije Universiteit Amsterdam. She has been studying collaborative innovation for the development of digital innovations. Her recent work focuses on innovation ecosystems, organizational routines, and discovery-driven experimentation. Fleur has published in outlets such as Academy of Management Journal, Organization Science, and Strategic Organization.

Philipp Tuertscher is Associate Professor of Technology and Innovation at the KIN Center for Digital Innovation, School of Business and Economics, Vrije Universiteit Amsterdam. His research explores organizational mechanisms and social practices for collaborative innovation in a variety of settings. Besides studying innovation processes on collaborative crowdsourcing platforms and online communities such as Linux, Wikipedia, and Threadless, Philipp has been studying innovation and the creation of digital infrastructure in large-scale scientific collaborations at CERN. His work has appeared in the Academy of Management Annals, Information Systems Research, Organization Science, and Organization Studies, among others.

over time) as embedded units of analysis, we developed detailed insights on the actions through which complement quality is sustained over time and what drives actors in the ecosystem to do so.

Our findings explain the sustained development of complementary products to ensure complement quality and how these relate to ecosystem dynamics over time. We differentiate between complementors that (1) maintained integrity, (2) enhanced functionality, and (3) abandoned their complements. These three actions are explained by ecosystem dynamics that triggered *expanding affordances*, *materializing glitches*, and *emerging obsolescence*. Our process model explains how their ultimate effect depends on how the ecosystem actors (complementors as well as the platform owner and users) act upon these mechanisms. Without further complementor action, these mechanisms lead to deteriorating quality; however, complementors can maintain or enhance complement quality by enhancing functionality and maintaining the integrity of their complement; platform owners and users can facilitate these actions by reactively and proactively providing input to complementors.

Our findings have important implications for theory and practice of innovation in platform ecosystems. First, we contribute a relational and dynamic conceptualization of complement quality. We show how complement quality not only depends on the bilateral relation between platform core and complement, but also on the multilateral connections with other elements of the platform ecosystem. We uncover the mechanisms that underlie the deterioration of complement quality over time, the need for maintaining it, and opportunities for enhancing quality. Second, we explain how sustained development is a distributed innovation process involving interactions between all three platform actors: platform owner, complementors, and users. This interplay becomes more complicated to manage when ecosystems grow because platform owners cannot fully control the actions of others—most notably, whether complementors continue to develop and maintain their complements. In light of this challenge, our study revealed actions through which platform owners can proactively alleviate the burden of and create new opportunities for sustained development by complementors. Third, we offer new insights on indirect network effects, suggesting that low-quality complements may negatively affect platform ecosystems and that interactions between users and complements may enable the process of sustaining complement quality over time. Overall, our paper sheds light on the digital transformation of incumbent, product-centric firms, who need to develop new practices and approaches to deal with the complex interdependencies associated with organizing innovation in digital platform ecosystems (Nambisan et al., 2017).

Theoretical Background

Platform Ecosystems

Platform ecosystems evolve around open product platforms (Cennamo, 2018; Jacobides et al., 2018). Research from a technology management perspective focused on technological characteristics of platforms (Gawer, 2014; McIntyre and Srinivasan, 2017), according to which open product platforms consist of a core product, complementary products, and standardized interfaces that enable interoperability between a platform and its complements (Baldwin and Woodard, 2009; Tiwana, 2013). Examples are smartphones and apps (Ghazawneh and Henfridsson, 2013), gaming consoles and video games (Zhu and Iansiti, 2012), web browsers and extensions (Tiwana, 2015b), or e-readers and e-books (Wang and Miller, 2020). Strategy scholars focused on the actors engaging with platforms (McIntyre and Srinivasan, 2017). Accordingly, the platform ecosystem comprises the platform owner who controls the core product and its interfaces, the complementors that provide complementary products, and their users (Jacobides et al., 2018; Selander, Henfridsson, and Svahn, 2013).

Theory about platform ecosystems suggests that their value is rooted in the *complementarity* of the platform and its complements (Jacobides et al., 2018). Complementarity refers to the condition that when things are used together, their value exceeds the sum of their parts (Baldwin, 2018). In the context of platform ecosystems, complementarity goes both ways: complementary products have limited value without the platform core (e.g., an app has no value without an operating system), and the value of a platform depends on the availability of complementarity products that extend the functionality of the platform. Such complementarity requires interoperability between the platform core and complements, which is typically realized through standardized *interfaces* (Baldwin and Woodard, 2009). For example, USB interfaces connect a PC and its peripherals, and application

programming interfaces (APIs) provide apps access to the platform's core modules (Ghazawneh and Henfridsson, 2013).

Scholars studying platform ecosystems have typically assumed that platform value depends on the *quantity* of complements available or expected (e.g., Clements and Ohashi, 2005; Katz and Shapiro, 1986; Song, Parry, and Kawakami, 2009; Stremersch, Tellis, Franses, and Binken, 2007; Zhu and Iansiti, 2012). Many platform owners open their platform to independent third-party developers to increase the number of complementary products and services available to users (e.g., Boudreau, 2010; Schilling, 2002). Independent developers—ranging from firms to hobbyists (Boudreau and Jeppesen, 2015)—may have insights into specific user needs and may possess the distinct knowledge and skills required to meet these, thereby providing variety in functionality for users.

Consequently, open platform ecosystems are multisided markets, driven by indirect network effects (Gawer, 2014; Rochet and Tirole, 2003). A growing user base attracts more complementors to the platform, while an increasing number of complements positively affects the size of the user base. For example, video game developers are attracted to platforms that have a sufficiently large user base to warrant the upfront investment of developing a game, and players are drawn to platforms that offer a variety of games (Eisenmann, Parker, and Van Alstyne, 2006). To actuate these self-reinforcing indirect network effects, platform owners often employ aggressive "get big fast" strategies (Cennamo and Santalo, 2013) and attract complementors through, for example, providing subsidies and technical resources for developers (Boudreau and Jeppesen, 2015; Ghazawneh and Henfridsson, 2013).

Complement Quality and the Value of Platforms

This focus on complement *quantity* by scholars and platform owners, however, is insufficient and potentially misleading. Some recent work has pointed out that platform value not only depends on the quantity of complementary products, but also on their *quality* (Cennamo, 2018; Hagiu, 2011; McIntyre et al., 2020; Panico and Cennamo, 2020). Complements differ in the value they provide for the platform ecosystem. Some complements are "superstars" that generate high sales, thereby stimulating platform growth (Binken and Stremersch, 2009; Gretz et al., 2019). Furthermore, when users experience high-quality complements, they may also buy other complementary products because they anticipate that their quality will be high too (Cennamo and Santalo, 2019). However, other complements may be of low quality. As Wareham, Fox, and Cano Giner (2014, p. 2012) note: "if a thousand flowers grow, inevitably, some will be undesirable and harmful to the ecosystem. In the extreme, the unconstrained growth of low-quality innovations can kill a platform." Thus, complementary value is not given by the mere existence of a complement. Low-quality complements may reflect negatively on the platform as a whole and scare off users (Cennamo and Santalo, 2019; Wareham et al., 2014).

To avoid negative consequences of low-quality complements for the overall ecosystem, platform owners employ various governance and control mechanisms with the ultimate goal to ensure quality (Tiwana et al., 2010; Tiwana, 2015a; Wareham et al., 2014). Although independent complementors are beyond their direct control, platform owners may influence complement quality through gatekeeping. They may act as "bouncers" who decide what complements are admitted to their platform (Ghazawneh and Henfridsson, 2013; Tiwana, 2013; 2015a) and through "soft quality incentives," such as selective promotion (Rietveld, Schilling, and Bellavitis, 2019) and rewarding high quality (e.g., by lifting API restrictions) (Claussen, Kretschmer, and Mayrhofer, 2013).

Sustaining Quality in Digital Platform Ecosystems

Our study advances beyond the above-reviewed studies by taking a dynamic and relational perspective on complement quality that fits the characteristics of digital technologies. The digital technologies that underlie platform ecosystems are inherently *dynamic* as they are reprogrammable, for instance, by implementing new APIs or updating the firmware (Yoo et al., 2012), which allows for continuous evolution and novel combination with other complements even while they are in use (Garud, Jain, and Tuertscher, 2008; Zittrain, 2006). Moreover, digital products call for a *relational* approach as they often gain in value when used in connection with other products associated with the core platform and beyond (Adner et al., 2019; Henfridsson et al., 2018; Yoo, Henfridsson, and Lyytinen, 2010). Relationality entails both technological and social aspects (Faulkner and Runde, 2012; Garud et al., 2013), which fits calls for the joint consideration of products

and actors in platform ecosystems (Gawer, 2014; McIntyre and Srinivasan, 2017).

Aligned with our *relational* focus on the complementarity of the platform and its complements and following arguments of Cennamo and Santalo (2019, p. 639) and Cennamo et al. (2018, p. 463), we discern two dimensions of complement quality: (1) the *integrity* of the complement (i.e., how well it interoperates with the platform) and (2) the *functionality* provided by the complement in the platform ecosystem.

First, quality in terms of *integrity* concerns how well complements technically interoperate with a platform (e.g., Garud and Kumaraswamy, 1995). When there are inconsistencies between the complement and the platform, compatibility issues arise and the integrity of the overall system decreases. For example, smartphone apps may vary in "the degree to which an app faithfully uses standards and protocols predefined by the platform owner (e.g., platform-specific APIs, data formats, and protocols) to interact with the platform" (Tiwana, 2015b, p. 47). Compliance with standardized interfaces ensures a minimum quality and reduces the likelihood of glitches as a result of suboptimal integrations (Hoopes and Postrel, 1999).

Integrity remains at stake over time, even for complementary products that have been in use. Although platform interfaces are ideally fixed and stable (Baldwin and Woodard, 2009), the reprogrammability of digital platforms results in changes over time, for example, to enhance the platform's security. Such changes may affect a complement's integrity, as these may jeopardize its interoperability with the platform (Tiwana, 2015b). Therefore, it has been argued that the integration between a digital platform and a complement is not a "one-shot task" but an ongoing process (Tiwana et al., 2010).

Second, quality in terms of *functionality* regards the unique content or utility complements provide vis-a-vis the platform ecosystem. The functionality delivered by a complement affects how much value it contributes to the platform as well as its fate in the competition among complements: some complements may never get traction, or may just serve a small niche, whereas others become "killer apps" (Boudreau, 2015; Evans, Hagiu, and Schmalensee, 2006). At the same time, other complements may have "dubious value" (Eaton et al., 2015) that negatively affect the overall platform.

Quality in terms of functionality also needs to be considered from a dynamic and relational perspective. The functionality provided by the platform itself evolves and changes over time (Evans et al., 2006; Gawer, 2014). As a consequence, third-party complements can become obsolete when a platform incorporates similar functionality in core components (e.g., when Apple introduced "Spaces" as an integral feature of its OS X platform, they enveloped functionality that was earlier provided by third-party software packages).

In sum, platform ecosystems thrive on the value of complements, which depends on the quality of such products vis-a-vis other elements of the platform ecosystem and changes over time. Therefore, we advance a dynamic and relational conceptualization of quality. Ensuring quality is challenging because the developers of complementary products are independent members of platform ecosystems and, therefore, cannot be fully controlled by a platform owner (Wareham et al., 2014). Thus, we study how and why the quality of existing complementary products is sustained in digital platform ecosystems.

Research Methods

Our objective is to elaborate theory (Fisher and Aguinis, 2017) on the quality of complementary products in digital platform ecosystems. Informed by a critical realist research philosophy (Sayer, 1992; Van de Ven, 2007), we seek to identify underlying generative mechanisms through a case study (Tsoukas, 1989). Case study research is highly appropriate for such a theory development objective as it enables examining new challenges and mechanisms underlying innovation management in novel settings (e.g., Goffin, Åhlström, Bianchi, and Richtnér, 2019). Moreover, a case study strategy enables examining phenomena in their real-life context (Yin, 2009), which is needed for the investigation of complement quality from a relational and temporal perspective on ecosystems. For these reasons, we performed an in-depth, qualitative field study of the Philips Hue connected consumer lighting platform ecosystem, following the state-of-the-art principles for case study research (Goffin et al., 2019).

Research Setting

The Philips Hue platform ecosystem offers a suitable setting for our research aims as it provides a revelatory case (Yin, 2009) in which the phenomenon of interest—the

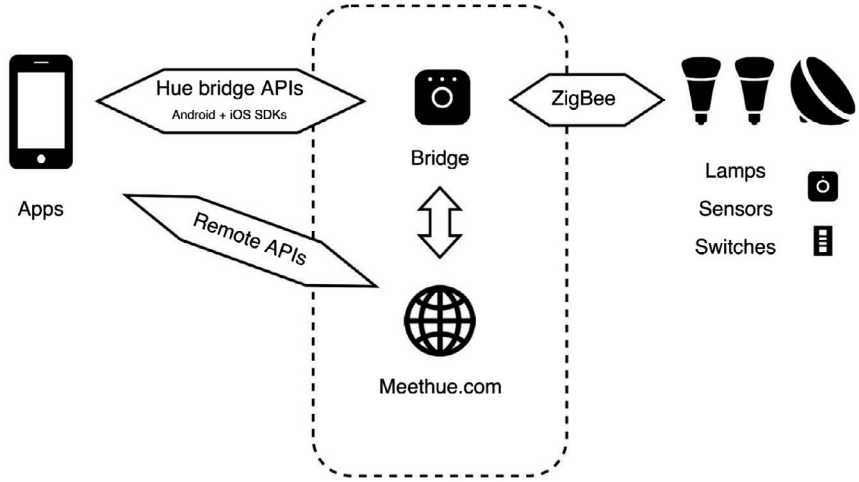

Figure 1. Philips Hue System Architecture

dynamics of complement quality in a digital platform ecosystem—was highly present (Pettigrew, 1990). The following reasons make this setting particularly appropriate for theory development in this domain.

First, Philips Hue exemplifies the digital transformation that many traditional product development firms aspire to, as it shifted its innovation model from selling products (e.g., traditional light bulbs) to open digital platforms (e.g., around smart light solutions). This shift was facilitated by novel digital technologies and increasing connectivity and enabled Philips Hue to become the market leader in smart consumer lights (Signify, 2018). The Philips Hue platform consists of LED light bulbs with a chip that can receive commands via a central "bridge" (a hub connected to the local Wi-Fi network) using the Zigbee communication protocol (see Figure 1). Commands can be sent to the bridge via a smartphone app, and its architecture offers endless possibilities for connections with other smart home appliances.

Second, a platform ecosystem emerged around Philips Hue, with many independent developers creating a variety of apps as complementary products. Philips Hue released a designated developer portal with official API documentation and software development kit (SDK) for developers in March 2013, which won an award for the world's "Best API in the Internet of Things" (Digital Accelerator Awards by Apigee). These tools supported developers to create hundreds of iOS and Android apps and several for Windows, Ubuntu, Chrome, and Mac.

Third, our theoretical interest in complement quality in platform ecosystems was aligned with Philips' main concern because their vision for the Hue platform is to provide an easy-to-use, high-quality consumer product. Because part of the user experience is provided through apps by independent third-party developers, ensuring the quality of complements over time was a pressing concern for Philips. The importance of complements' additional functionality was exemplified in the "more apps for Hue" recommendation list in the native Hue app. Moreover, concerning customer expectations on sustained integrity of the platform and complementary products, the manager of the Philips Hue developer program commented: "If you do not deliver on that, it will be very difficult to turn smart homes and smart lighting into a success."

Fourth, the Hue platform ecosystem exemplifies the relational and dynamic aspects of digital technologies, as it continuously evolved over time by expanding both its hardware and software. Since the Hue platform relies on embedded digital technology, the core platform elements are malleable and can be upgraded by Philips: "We did the first software update of a light bulb in 136 years since they were first invented. [This] means that our app, cloud, bridge, and bulbs are not static. […] Every few months, we release a software update, which brings consumers new features and functionality." (George Yianni, Hue inventor, February 2014).

Data Collection

We collected data about the dynamics of the platform ecosystem and its complementary products as embedded units of analysis. These complementary products

Table 1. Overview of Data Sources

Data Source	Method/Type of Data	Purpose
Field research at Philips Hue	Observations of weekly and ad-hoc meetings (primarily of the Hue partnership team), unstructured observations including chats at the coffee machine and lunches, and attending events such as a hackathon. Daylong field visits took place once a week, for a duration of one year.	• Better understanding of the day-to-day management of the Hue platform ecosystem. • Gained technical knowledge about the platform core and interface that served as important input for interviews and analysis of the app data.
Hue team members	14 formal (semi-structured) interviews (verbatim transcripts) and numerous informal interviews during field visits, which were captured in field notes that were written within 24 hours.	• The interviews allowed us to go back in time and inquire about past events and decisions and obtain an understanding of the considerations that informed earlier decisions.
Documents on Hue platform	Secondary (publicly available) data, including press releases and data from the developer portal (such as API changelog, release notes, and announcements by Philips Hue).	• Secondary data were used to create an overview of the development of the Hue platform over time. • Factual data from documents were also used as input for the interviews with the Hue team and third-party app developers.
App and app updates	Quantitative and qualitative data extracted from the AppAnnie database and App stores (Apple and Google), including release dates, release notes, app descriptions, and reviews.	• Quantitative data were used to map the landscape of third-party app and updates over time. • Qualitative data from the release notes were used to inductively derive insights on the different updating actions performed by third-party developers. • App data were also used as input for interviews with third-party app developers to develop questions about the individual app trajectories over time.
Third-party app developers	21 semi-structured interviews with 20 third-party app developers.	• The interviews allowed us to probe developers about the story behind a particular app and update. • The interviews revealed interactions between third-party app developers and other ecosystems actors (platform owner and users).

(apps) were the locus of complement quality, whereas the broader ecosystem dynamics were needed to explain how and why complement quality was sustained over time. Following recommendations for case study research (Yin, 2009), we collected data from multiple sources to incorporate the diverse perspectives of different ecosystem actors and triangulate findings across sources: (1) observations at Philips Hue during field visits; (2) interviews with Hue team members; (3) documents on the Hue platform development over time; (4) quantitative and qualitative data on app updates; and (5) semi-structured interviews with app developers (see Table 1 for an overview). Combining these data sources with their distinct and complementary nature enhanced the comprehensiveness and validity of interpretations (Gibbert and Ruigrok, 2010). For example, while the app data offered precise information on what happened when, the interview data complemented detailed information about the reasons behind certain events. The observation notes and interviews with the Hue team elucidated the choices made with regard to managing the platform ecosystem over time.

Field research at Philips Hue. To better understand the Hue platform ecosystem, its history, and the internal organization, the first author spent, on average, one day per week at the Philips Hue office from November 2015 until December 2016. Following the principles of engaged scholarship (Van de Ven, 2007),

Table 2. Informants from the Philips Hue Team

	Informants	Number of (Formal) Interviews
1	Developer program manager 1 (former)	1
2	Developer program manager 2 (former)	1
3	Product owner 1	1
4	Program lead (connected lighting)	1[a]
5	Head of Technology (inventor of Hue)	1[a]
6	Developer evangelist 1	2[a]
7	Director partnerships	2[a]
8	System architect	2
9	Researcher 1	1[a]
10	Researcher 2	1[a]
11	Program manager (analytics)	1
12	Product owner 2	1
13	Developer evangelist 2	1
14	Director standardization	1
15	VP Global Growth	1
	Total number of interviews	14

[a]Interviews held together with another informant.

we interacted closely with key informants in the Philips Hue team. In the pilot phase, exploratory interviews with key informants alerted us to core challenges faced by the Hue team, which were used to sharpen our research focus and data collection instruments. The first author observed recurring and ad hoc meetings of the team responsible for managing partnerships and the community of third-party developers. Furthermore, the first author observed department meetings and events like a 24-hours developer hackathon. During each day in the field, impressions and insights were captured for analysis in digital field notes.

Interviews with Philips Hue team members. During the field visits, we interviewed all the Hue members involved in managing external developers to obtain a deep understanding of the challenges of managing complement quality and gain an overview of the decisions and actions taken over time. We also interviewed a variety of other Hue members, including the original Hue inventors, internal developers and researchers, and senior management. In total, we performed 14 formal, semi-structured interviews with 15 informants; during four interviews, we spoke with two people at the same time, and some informants were interviewed twice (see Table 2 for an overview). Interviews lasted, on average, one hour. Interviews were voice recorded and transcribed verbatim, except for three interviews where the setup did not allow for recording; there, we relied on extensive notes. Furthermore, the first author had numerous informal ad hoc interviews to clarify what had been observed and stay up to date with ongoing events.

Documentation on Hue platform development. We also collected documents on the Hue platform. Specifically, we collected all release notes of various versions of the Hue bridge firmware and the API changelog to understand the changes in the platform core. We complemented these documents with press releases (e.g., about product releases) and Philips' official statements (e.g., published on the developer forum) to identify significant events in platform evolution. This documentation was also used as input for the interviews.

App and update data. Because complementary products (the Hue apps) were our embedded unit of analysis, we collected archival data about third-party Hue apps and how they changed over time. First, we created an overview of all apps for Hue by searching various app stores and cross-referencing these with internal Philips lists, resulting in a database of 115 iOS and 90 Android (Google Play) apps. We then collected data for all of these apps via the AppAnnie database, including reports of the app versions with release dates, version numbers, and release notes that provided qualitative insights into the developer's intentions with the app update (about 70% of the updates included release notes).

Interviews with third-party app developers. Finally, we conducted 21 semi-structured interviews with 20 independent app developers about the development and maintenance of their apps (see Table 3 for an overview). We sampled interviewees based on the apps that they developed. We aimed for maximum variety on the following dimensions: (1) popular and less well-known Hue apps; (2) apps with few and many updates; (3) single app or multiple Hue apps by the same developer; (4) Hue apps for one or multiple mobile operating systems. The interview protocol (see Appendix) focused on the developers' "innovation journey" (Van de Ven, Polley, Garud, and Venkataraman, 1999) and involved questions about the developers' initial ideas for the app, the challenges faced during app development and maintenance—specifically to understand the process preceding an app

Table 3. Overview Third-Party App Developer Interviews

Developer	Country	Duration Interview (Min)	Platform	Number of Apps (Total)	Number of Hue apps	Type of App	Release (First) Hue App	Last Update (Hue)
A	Austria	59	Android	2	1	Utilities	Jan 16, 2016	Feb 23, 2016
B	US	50	Apple	5	1	Utilities	Mar 24, 2014	Jun 07, 2016
C	Netherlands	52	Apple & Android	15	10	Music	Dec 14, 2012	May 20, 2016
D	Denmark	56	Apple	19	1	Lifestyle	May 09, 2013	Oct 30, 2014
E	US	59	Apple & Android	5	5	Utilities, books	Apr 08, 2014	Jul 01, 2016
F	Germany	79	Apple	6	2	Utilities	Apr 29, 2013	Jun 25, 2016
G	US	43	Apple	5	1	Lifestyle	Jul 10, 2014	no update
H	US	71 + 54	Apple	2	1	Lifestyle	Sep 14, 2013	Jul 01, 2016
I	US	61	Apple	2	1	Lifestyle	Dec 02, 2013	Jun 15, 2016
J	Sweden	53	Android	4	2	Utilities	Jul 03, 2014	Jul 01, 2016
K	Germany	53	Ubuntu	21	1	Utilities	Jan 27, 2014	Dec 13, 2015
L	Czech Republic	68	Android	41	2	Lifestyle	Oct 25, 2014	Jul 04, 2016
M	US	43	Apple	5	4	Music	Aug 31, 2015	Jul 10, 2016
N	Slovak Republic	96	Apple	2	2	Lifestyle	NA	Jul 06, 2016
O	Netherlands	64	Apple & Android	33	22	Games	Feb 04, 2015	May 31, 2016
P	Hong Kong	89	Apple	3	2	Utilities	Mar 03, 2014	Jun 20, 2016
Q	Netherlands	59	Windows	3	1	Utilities	NA	NA
R	Netherlands	51	Android	8	1	Tools	Nov 05, 2013	Jan 12, 2016
S	US	64	Apple & Android	3	3	Utilities	Apr 11, 13	Mar 14, 2016
T	Germany	61	Apple & Android	5	4	Utilities	Oct 12, 14	Jul 5, 2016
Total (average)		1285 (62)		189	67			

update—and their interactions with and dependencies on other actors in the ecosystem. In preparing for the interviews, we created timelines per app (Poole, Van de Ven, Dooley, and Holmes, 2000), which were used to help developers recall specific events and relate their experiences to factual data, to counter retrospective bias (Huber and Power, 1985). All interviews were voice recorded and transcribed, and mostly took place through videoconferencing tools because the app developers were located worldwide.

Data Analysis

Our aim was to develop a process model and uncover the generative mechanisms that explain how and why things happen (Van de Ven, 2007, p. 145). Thus, we sought to find how complement quality comes about through key events, instead of predicting quality through a variance model (Langley, 1999; Mohr, 1982). Because complement quality pertains to third-party apps in our setting, we analyzed apps as embedded units of analysis in the overall case setting of the Philips Hue platform ecosystem (see Yin, 2009, p. 46). As recommended for qualitative research, initial data analysis efforts happened parallel to data collection and informed our decisions to stop data collection when our overall analysis was saturated.

Step 1: Based on the archival data on third-party apps, we created an overview of the number of apps and updates released per month and visualized this data in Figure 2. The prevalence of updates compared to the release of novel apps supports the key tenet of our study that the sustenance of quality is important indeed, not just the introduction of a large quantity of apps. We also calculated the number of updates per month in relation to the number of available apps (see Figure 3). Through this analysis, we identified variations in update activity in the ecosystem at specific points in time, which informed further analysis.

Step 2: We compiled a database detailing the complete revision history for each app, as well as a timeline of critical events in the development of the

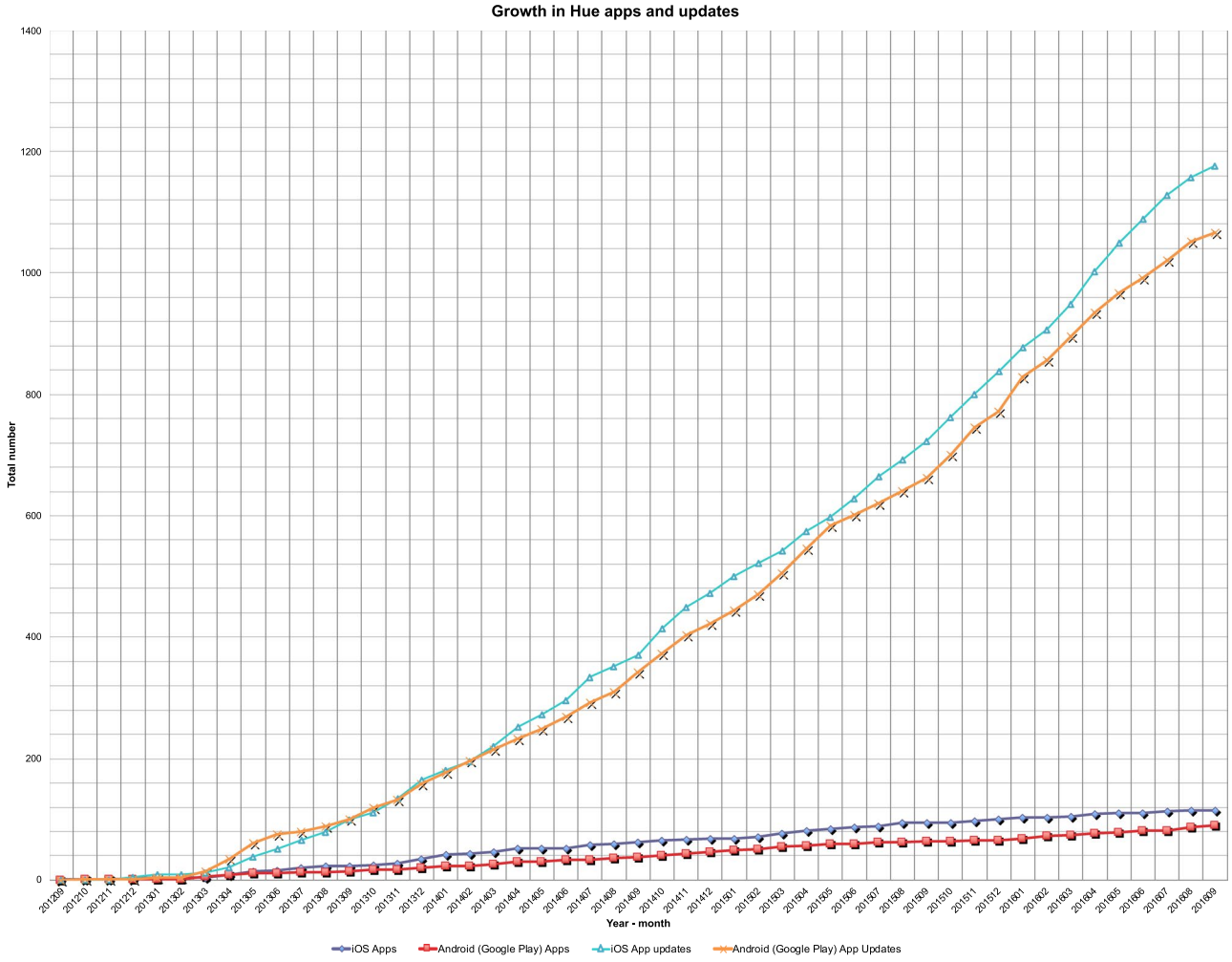

Figure 2. Number of Hue Apps and Updates Over Time

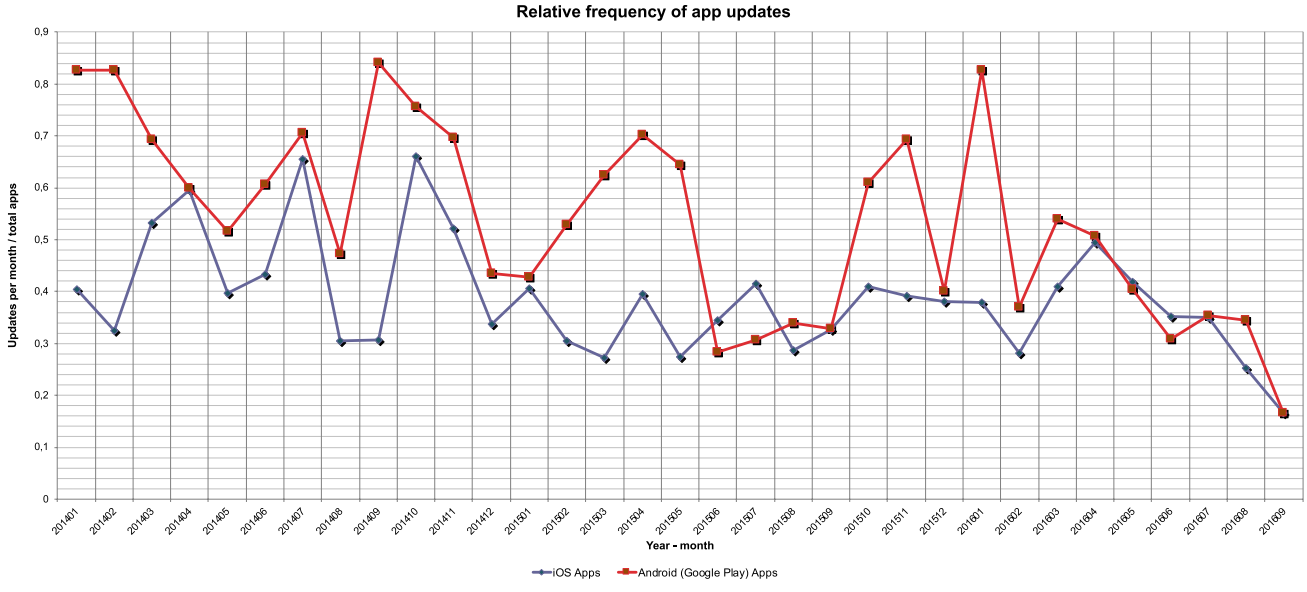

Figure 3. Relative Frequency of App Updates

platform ecosystem. Specifically, we represented each app's development as a sequence of events consisting of the initial release and a series of updates. Each event was marked with a timestamp that allowed for cross-referencing with important ecosystem dynamics (e.g., the release of a new Philips Hue API, hardware, or changes in other platforms such as iOS). The majority of updates (70%) were accompanied by release notes providing information about the type of changes and reasons for making the updates. For a subset of 67 apps, we complemented this secondary data with interview transcripts in which developers explained their motivations for updates and provided additional insights into the development of their apps.

Step 3: To analyze the sustained development of apps, we inductively coded all release notes and developer interview transcripts. We first coded for complementor actions occurring in the apps' revision history. By constantly comparing the coded segments (Miles and Huberman, 1994), we discerned types of complementor actions: updates directed towards *maintaining integrity* (entailing bug fixing and ensuring compatibility) and two towards *enhancing functionality* (entailing improving user experience and adding new features) (see Table 4 for supplementary evidence). Based on the different complementor actions, we categorized the development of all apps by the time that our data collection had ended, distinguishing between apps that had been updated to *enhance functionality*; updated only to *maintain integrity*; or apps that had not been updated in the past year or were removed: *abandoned* apps. All apps were coded by the first author. To ensure the reliability of our coding process, a second author coded a random sample of 10% of all apps, which yielded a Cohen's Kappa of 0.91, suggesting high interrater reliability.

Step 4: Next, we coded interviews with developers and Hue members as well as documents with the focus on developing a process explanation of complementor actions. Interviews and documents were specifically insightful to understand how updating efforts were related to changes in the ecosystem. We coded specific events in either the app revision history or platform ecosystem timeline that triggered complementor actions. For example, such an event was the removal of an API feature by Philips, which caused third-party apps to break and required developers to take action. This analysis yielded three categories of ecosystem dynamics (*platform core changes*, *changes to other complements,* and *idiosyncratic connections*). Subsequently, we analyzed these ecosystem events in relation to the app revision history to identify mechanisms affecting complements and the sustained development process. This step was essential for moving from condensing data into patterns towards theorizing (Eisenhardt, 1989). Specifically, we found that ecosystem dynamics resulted in *expanding affordances*, *materializing glitches*, and *emerging obsolescence*, to which complementors, platform owners, and users responded with different actions. Finally, we integrated different pathways created by the ecosystem dynamics, mechanism, and responses by the ecosystem actors into a process model.

Throughout these coding steps, the co-authors who were less involved in data collection critically questioned the emerging interpretations by taking an outsider-perspective to the patterns identified by the first author who had rich knowledge of the field study as a measure to reduce researcher bias and increase the internal validity of the findings (Evered and Louis, 1981). We returned to our data sources when more information was needed (e.g., from an app developer or Philips Hue informants) and decided not to collect additional data when our findings were replicated across apps (Yin, 2009). Moreover, we solicited feedback on the accuracy of our interpretations from informants to ensure the internal validity of our findings—also during the revision process (Gibbert and Ruigrok, 2010). The first author presented emerging insights twice to the Philips Hue team managing the developer community and a third time at the end of the research project to a wider audience (including higher management) at Philips Hue. Furthermore, when all analyses had been completed, a senior Hue member checked the findings by reading drafts of this paper; this resulted in minor changes only.

Findings

We found that sustaining complement quality was an important challenge in the Hue platform ecosystem. The efforts to sustain quality became apparent in frequent updating by developers after the launch of their apps. From our analysis of the third-party Hue apps, we found that updating was a substantial part of their development process (see Figures 2 and 3). On average, the number of updates steadily increased over time and was about ten times the number of active apps, with some peaks of even higher intensity.

Table 4. Complementor Actions and Representative Quotes

Complementor Actions	Representative Quotes from Release Notes and Developer Interviews
Enhancing functionality	
Improving user experience	*Release notes* "UI tweaks" "Added Group to x percent command for more detailed brightness control" "Great UI updates plus simplified Nav Drawer" "Added German, Danish, Dutch and more!" "Scene overview added for simple and fast selection of scenes." *Developer interview quotes* "I had a Russian user who said: 'I love [your app], but I wish it was in Russian, I will translate it for free.' So he did that and then I got some Russian sales, and then I had an Italian user offering the same thing, and then I had a German user offering the same thing. It sort of bootstrapped up to the point where I had 10 languages, and then I got addicted to that." "the user experience is by far the biggest thing that I focus on, so I just draw up little diagrams on the UI (user interface) to figure out a clean UI. But also, the most important part is getting feedback from friends and family on: what do you think of this layout? What do you think should change? Is it confusing? Is the flow weird? You know—stuff like that. Getting feedback on the UI was probably the most powerful thing in doing this development."
Adding new features	*Release notes* "Added new devices compatibility (Mojio, Ecobee, Nest family (protector and thermostat), Easy bulb lights)" "New feature: Change bulb state on status bar notifications—missed calls, missed SMS, missed facebook/whatsapp/skype messages etc" "Complete support of Philips Hue Dimmer. All 4 buttons can be programmed; either manually or with the rule wizard." "NEW: Experience the iconic Times Square Ball Drop live at home!" "Added Lava Lamp and Music widgets!" "Get your Philips Hue & LIFX bulbs ready for Halloween with the new scenes in this update!" *Developer interview quotes* "[I built in] voice control […] because the sales went down" [and my] "app was not spectacular in any way." "taking in [users] feedback and prioritizing new features based on their feedback is so very, very important." "taking in user feedback, like, 'oh this other app does this feature really well, why cannot do that in [the developers' own App]?'. That's a good idea, why not. Taking that experience and make it my own, and say, 'hey, what about this app is dealing really well, how can I make that better in my own?' And so it's really cool taking in other people's ideas and seeing a different perspective on the app, and so it's, it is just creative thinking and taking in a different perspective on ideas."
Maintaining integrity	
Bug fixing	*Release notes* "Some bugs squashed" "Crash in section "Groups and Rooms" fixed" "Random bug fixing" "Fixed issue with displaying brightness values in color picker" "Critical bug fix" *Developer interview quotes* "[Bug fixing] can get tedious at points […] but, in the end, helping the person is the most important part, seeing it work for them is really satisfying for me." "When I find a bug that needs to be fixed, I will do it, I try." "there's this weird issue that I have been trying to fix lately where my app does react in an odd way, where it just doesn't turn off the light in a specific case, that a fault of my app."
Ensuring compatibility	*Release notes* "Prepared for an upcoming firmware update for Hue" "All friends of HUE are supported now" "Support for newest bridges/firmware" "Tested on Android Marshmallow" "Added new Bridge Connect code to comply with latest API changes" *Developer interview quotes* "sometimes for you, everything works perfectly, and then for the person next door, one or two things don't work, so you always have to keep trying to understand why something is not working for A, for B, and trying to fix that." "Because I am on four platforms […] yeah, in terms of release timing and that sort of thing… But luckily, Philips communicates API changes and has some degree of backward compatibility. And also, a lot of users update to the latest firmware, so it's a lot better than it could be! But it's always though working with different hardware and firmware at the same time."

Table 5. Overview of Apps Categorized According to Complementor Actions

	Number of Apps		
Complementor Actions	Google Play (Android)	Apple App Store (iOS)	Percentage
Enhancing functionality[a]	33	52	42.3%
Maintaining integrity	29	25	26.9%
Abandoned: Updates in the beginning only	13	23	17.9%
Abandoned: Never updated	5	11	8%
N/A (too recent)	8	2	5%

[a]Updates enhancing functionality may also include maintaining integrity as secondary complementor action.

Yet, the quality of some apps was sustained more thoroughly than others: the number of updates ranged from zero to 50 throughout an app's existence. Not only did the update frequency vary considerably, so did the content of the updates. We identified distinct complementor actions aimed at *enhancing functionality* or *maintaining integrity* of an app; some developers *abandoned* their app so that it received no further updates (see Table 5 for an overview).

Enhancing Functionality

Many developers in the Hue platform ecosystem kept enhancing their app's functionality over time to ensure that it remained relevant in the continuously evolving ecosystem. At the time of study, 85 out of the 201 apps in our database had been updated to enhance functionality (42%), which involved updates aimed at: (1) improving user experience and (2) adding new features. Many developers made sustained efforts at improving the user interface to make their app aesthetically more pleasant, and also to improve its usability and provide an overall better user experience. In addition, developers aimed to enhance their app's quality by adding completely new features over time. Such new features may not only involve new functionality of the app itself, but also involve the addition of new content (e.g., premade lighting setups) or the integration with other complements within or beyond the Hue platform ecosystem (e.g., support for LIFX, a competitor light bulb, or other smart home devices).

Maintaining Integrity

Our analysis of the Hue platform ecosystem showed that some developers maintained the integrity of their app simply to keep it "alive." That is, they performed the necessary updates to ensure the app remained functioning over time. 54 out of 201 apps were updated by their developers only to maintain integrity (27%). *Maintaining integrity* entailed two types of updates: (1) bug fixing and (2) ensuring compatibility with the Hue platform ecosystem. Bug fixing involved the correction of mistakes or errors in the app code. These may go unnoticed when writing the code and, despite testing, only surface when the app is used in combination with other elements of the platform ecosystem. However, even without programming mistakes, an app may stop functioning due to compatibility issues caused by changes in either the core Hue platform or other complementary products and services in the broader platform ecosystem.

Abandoning

Finally, some complementary products may not receive any updates whatsoever; they are abandoned by their developer, and their quality is lost over time as the platform ecosystem continues to evolve. 52 out of the 201 apps in our sample were abandoned (26%). This number includes 16 apps that were never updated and 26 apps that were updated initially, but, over time, their developers became inactive. App quality deteriorated when its developer had abandoned it; the lack of updates implied that apps were often left "broken" on the platform. As a consequence, the app may not only have lost its complementary value (e.g., existing users were not able to properly use the app anymore), but the complementary value for the platform may have even become negative (e.g., if an outdated app caused other complements of users to fail). While many developers simply left their broken apps on the platform, some developers deliberately removed them from the app store. While removal may void its complementary

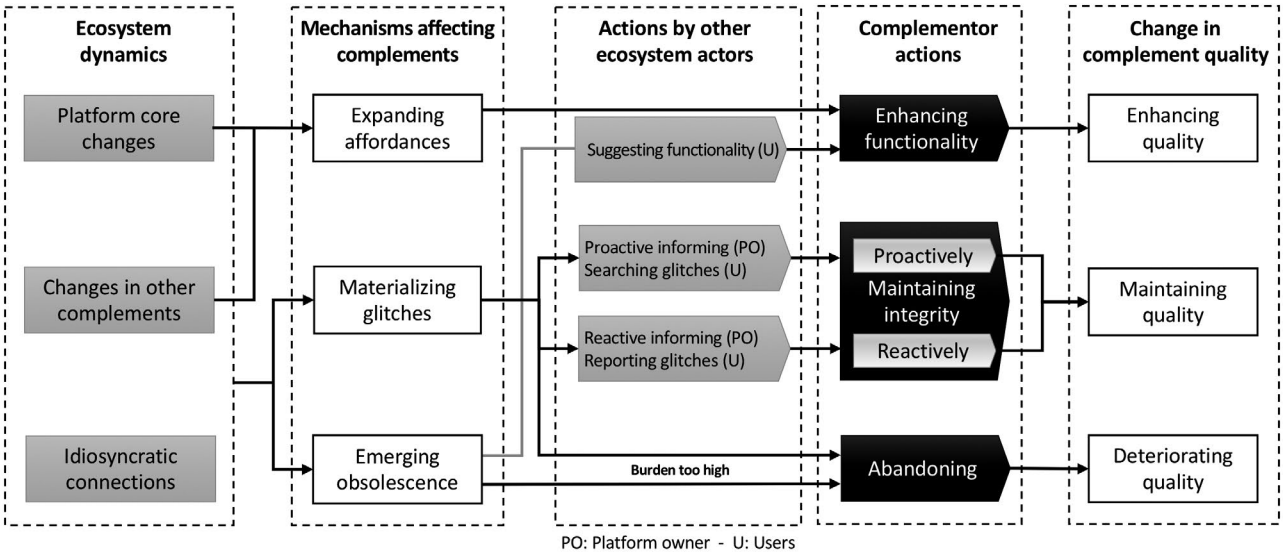

Figure 4. Process of Sustained Development

value for the platform, it reduces the risk of users installing an outdated app.

Ecosystem Dynamics Explaining the Sustained Development Process

So far, these findings indicate *that* developers are engaged in updating to sustain the quality of their apps. In the remainder of the findings, we explain *why* and *how* complement quality is sustained over time. Our explanation is summarized in Figure 4. We introduce this inductively derived process model here to show the structure of the findings that we will unpack later. The model shows that complementor actions (enhancing functionality or maintaining integrity), or the absence thereof (abandoning), are explained by how developers, platform owners, and users respond to mechanisms triggered by *dynamics* in the Hue platform ecosystem. These ecosystem dynamics include platform core changes, changes in other complements, and idiosyncratic connections created by users. For any existing complement, such changes may result in (a) *expanding affordances* (changed platform properties that enable new possible uses of complements), (b) *materializing glitches* (technical problems that cause a complement to malfunction), and (c) *emerging obsolescence* (functionality or look and feel of a complement do not keep up with growing standards). The same changes may impact various complements in different ways: changes may provide opportunities for some developers but may burden others. In combination with supportive actions of the platform owner and users, this explains developers' actions to sustain development to maintain or enhance complement quality. In Table 6, we provide empirical examples of the sustained development process of ten representative apps. Next, we will explain how different dynamics in the platform ecosystem triggered complementor actions.

Expanding affordances: Enhancing functionality. In many instances, platform dynamics like changes in the platform core and other elements can bring about opportunities to enhance complement quality. Specifically, such dynamics can result in expanding affordances (Volkoff and Strong, 2013), that is, properties of digital platforms that enable new possible uses in complements (Autio, Nambisan, Thomas, and Wright, 2018). When complementors embrace these opportunities and continuously implement new features, they can enhance the functionality of their complementary product over time and increase complement quality.

Many app developers in the Hue platform ecosystem made such an effort to deliver updates adding new features to their apps over time (see Table 5). For example, Apple's App Store, a key element of the platform ecosystem, enabled making in-app purchases, which afforded developers potential new ways to enhance functionality for users willing to pay for new features and additional content. Adding new scenes (i.e., pre-made lighting setups) turned

Table 6. Sustained Development Process of Ten Representative Apps

Pattern	App[a]	Ecosystem Dynamics	Mechanisms Affecting Complements	Actions by Complementors and Other Ecosystem Actors	Change in Complement Quality
Enhancing functionality	Hues	**Platform change:** Philips Hue released an official icon set for developers.	**Expanding affordances:** The icons allowed app developers to improve the usability of their apps.	**Enhancing functionality:** The developer of Hues adopted the new icon set to improve the user interface of his app: "Use of official Philips Hue Product Icons (…)" (Release notes, version 2.50)	**Enhancing quality:** With the help of the new tools, the developer improved the user experience, contributing to enhanced quality.
	iHue	**Platform change:** Philips Hue released the groups API feature as part of the introduction of the Hue Beyond lamp (hardware).	**Expanding affordances:** A new documented API feature allowed to group Hue light bulbs, which provided the developer of iHue with new possibilities for the app.	**Enhancing functionality:** The developer adopted the new API to improve his app: "Group together any combination of lights regardless of manufacturer (except for WeMo sockets and switches, which do not support grouping at all) (…)" (Release notes, 2.30)	**Enhancing quality:** By adopting the groups feature, the developer optimized control of the lights in the iHue app; the app's enhanced functionality resulted in enhanced complement quality.
	Control for Hue	**Platform change; changes in other complements:** Philips Hue introduced the Hue Tap (hardware, switch device). Apple introduced iCloud.	**Expanding affordances; materializing glitches; emerging obsolescence:** The new Hue Tap provided new opportunities for controlling the lights. At the same time, the increasing number of parameters to be configured created demand for backup and syncing functionality, which called for integration with cloud services like iCloud. The increasing complexity of the app gave rise to small glitches.	**Enhancing functionality; maintaining integrity; reporting glitches:** In addition to fixing small bugs that users had reported over time, the developer added two new features: "Hue Tap support! Configure your Hue Tap in more ways—use multiple actions per button and define your own fade times! iCloud support! Your presets and most of your settings are stored to iCloud! You never need to worry any longer about your presets on devices. They will sync if they have the same iCloud account. Of course, also several small bug fixes. Thank you for your support!" (Release notes, 1.8.1)	**Enhancing quality:** The complement quality was enhanced by two new features. Control for Hue was one of the first to offer Hue Tap support, thereby not only enhancing its own functionality, but also increasing the value and functionality of the new Hue Tap that depended on innovative 3rd party apps. Moreover, the iCloud integration further provided new complementarities with the iCloud platform. Also, the continuous maintenance of integrity ensured that the app's original features continued to function with integrity.
	Hueppy	**Changes in other complements:** Apple introduced a new generation of iOS (iOS 7) with a different look and feel ("flat interface").	**Emerging obsolescence:** The user interface of an app is expected to match the operating system, that is, "the iOS app should look like an iOS app and the Android app should look like an Android app."	**Enhancing functionality:** The release of iOS7 triggered the developer of Hueppy to do a major update on the user experience of the app: "We've rewritten our app from the ground to leverage the new iOS 7 design and to make it lightning fast"	**Enhancing quality:** Because the developer of Hueppy made use of the introduction of a new version of the iOS platform (in which this third-party app is embedded), the app's functionality was improved through an enhanced user experience.

(Continues)

Table 6. Continued

Pattern	App[a]	Ecosystem Dynamics	Mechanisms Affecting Complements	Actions by Complementors and Other Ecosystem Actors	Change in Complement Quality
Maintaining integrity	Sunrise	**Platform change (app):** Philips Hue restructured schedules in the native app; this change, which was not openly communicated by Philips Hue, became only apparent when Sunrise users pointed out problems in their feedback.	**Materializing glitches:** Because the Sunrise app relied on schedules set in the native Philips Hue app, the change caused a glitch: the app's functionality broke down because the schedules did not match the structure expected by Sunrise.	**Maintaining integrity, reporting glitches:** In response to users reporting issues, the developer implemented a fix: "Modifications to accommodate hue manufacturer's software data restructuring in July 2014." (Release notes 1.1.2)	**Maintaining quality:** After the app's intermittent breakdown, the update provided by the developer as a reaction restored the Sunrise app's full functionality.
	Smarter Hue	**Platform change (interface):** Philips Hue introduced major changes to its API (changes to whitelisting). These changes were announced well in advance.	**Materializing glitches:** The API change potentially could lead to glitches; for apps that use "a custom whitelist username" it implied that "the bridge authentication will start failing" (announcement on Hue developer portal). Since the change was announced in advance, Philips Hue and the user community expected developers to update accordingly.	**Maintaining integrity; proactive informing:** The developer of Smarter Hue changed his app to "use the randomly generated bridge username," as recommended by Philips Hue. He incorporated this change into a new version of the app, which was released before the actual change on the API side was implemented: "This version includes the new Philips Hue SDK to make sure the app is able to connect after the API changes (…)" (Release notes 1.1.1)	**Maintaining quality:** Because the developer anticipated and responded to the upcoming changes, users of B2 did not experience any issues. The app functioned with integrity before and after the API change, and thus, maintained its quality.
	Rainbow	**Idiosyncratic connections:** Users like to combine the Hue lights with other smart bulbs: "A lot of users add in Osram bulbs to their bridge" (developer). For example, because they are cheaper or provide other form factors.	**Materializing glitches; emerging obsolescence:** As more smart bulbs conforming to the Zigbee standard came to the market, users expected the various products to work together. However, the different light systems did not always work smoothly together, in this case creating a "weird issue (…) where my app does react in an odd way, where it just doesn't turn off the light in a specific case, that's a fault of my app."	**Maintaining integrity, reporting glitches:** The developer of Rainbow responded to the user feedback that the Osram bulbs did not turn off, and had to engage with problem-solving to fix the issue.	**Maintaining quality:** The update restored functionality for those users that are on different smart light platforms.

(Continues)

Table 6. Continued

Pattern	App[a]	Ecosystem Dynamics	Mechanisms Affecting Complements	Actions by Complementors and Other Ecosystem Actors	Change in Complement Quality
	Huerray	**Platform change; changes in other complements:** Philips Hue released new hardware: the bridge version 2.0, which included a new chip that allowed for Apple HomeKit integration.	**Materializing glitches; emerging obsolescence:** Users expected Huerray to work on all hardware; however, because the new bridge 2.0 came with changes to the bridge configuration API, the release of this new hardware could cause glitches for apps if not updated.	**Maintaining integrity; proactive informing:** Shortly after the new bridge was introduced (October 2015), the developer updated his app following Hue's API documentation to ensure compatibility with the new version of the bridge firmware: "Updated to the latest Hue API—Works on the new bridges" (Release notes, 1.20)	**Maintaining quality:** While functional integrity for users of the original bridge was not at stake, the update was important for Huerray users switching to the new bridge to get HomeKit functionality. Maintaining the app's quality was also crucial as the bridge 2.0 was included in all Philips Hue starter kits, and thus, all new users were on the new hardware and firmware.
Abandoning	Home Light	**Platform change:** Philips Hue started offering voice control in their native app, for example, through Siri. Home Light was affected by this change, as voice control was one of the unique features of this third-party app.	**Emerging obsolescence:** Because the official Philips Hue took over functionality that made this third-party app unique, the app became obsolete unless the developer would further enhance its functionality by adding other new features.	**Abandoning:** The developer felt that his app "is not that important anymore" because Philips "implemented many of the things from my app in the Philips app now." The burden to continue to enhance the app's functionality was too high. Therefore, he lost interest in his app and discontinued development.	**Deteriorating quality:** Existing users suffered from abandoning Home Light because they had to replace their preferred third-party app over time as the app developer did not provide the necessary updates to ensure continued functionality. On the other hand, new users were hardly affected (typically, users start with the native Hue app and then, expand the range of apps they download when they search for more tailored uses).
	More for Hue	**Changes in other complements; idiosyncratic connections:** Smartphones are important complements in the Philips Hue ecosystem. Regular updates of the Android OS represent changes in those complements. Besides, various new devices are introduced every year, and users are creating idiosyncratic connections between devices, their OS, and other apps they may use.	**Materializing glitches:** Because the developer of More for Hue stopped maintaining his app after some time, the app started to "crash" and "fail to connect to the bridge" as new device/OS combinations emerged. For example, one user wrote: "it used to work great! I have used this app for years, but recently I doesn't work on my GS7 edge" (user review)	**Abandoning; reporting glitches:** The burden to continuously maintain the app's integrity was too high. The developer did not update More for Hue, despite user requests on the app store to offer a fix to make it operable again.	**Deteriorating quality:** Both existing and new users were negatively affected when the developer discontinued his development, leaving the app broken on the platform. Users left one-star reviews, asking the developer, "please fix."

[a]App names are pseudonyms.

out to be a popular strategy to continually add new features, which frequently became visible in the release notes, for example, "added three new dynamic scenes, please check them out: 'summer breeze', 'vanilla sky', and 'pearl'." Also, often *new hardware* elements that were added to the platform ecosystem expanded the affordances developers could use to improve their apps. One developer explained that he was inspired by new Hue products, like switches and sensors, to add new features: "I am actually eagerly waiting for those things. Philips had shown new devices like last year, motion sensor, and temperature sensors. […] I already have ideas on how to include them into the system." This developer was the first to incorporate support for the Hue Tap, a device that was introduced by Philips Hue so that users could use a physical switch to turn their lights on and off. Being the first and only app supporting this new device gave him a competitive advantage over other apps.

While the new hardware was mainly targeted at the user-side of the platform, changes to the underlying software also resulted in expanding affordances, in many cases even beyond the additional functionalities that had been anticipated by the platform owner. For example, when Philips introduced "Beyond Light," a lamp that features multiple bulbs that work in unison, they also extended the API to enable the grouping of lights. The "groups API" opened up new opportunities for third-party developers. For example, one developer used it to change all lights with just one command. According to one developer, this helped overcome a "limitation that had been holding me back." He had tried with his app to change multiple light bulbs with music, but sending commands to individual bulbs resulted in a "huge latency issue." When it became possible to use the "groups API," he was delighted because addressing multiple lights as group avoided the latency problems and "made it a lot easier to give a nice experience."

Being the platform owner, Philips had a privileged position that offered more possibilities to extend the functionality of their own Philips Hue app over third-party apps. One major difference was that the native Philips Hue app had access to the "remote API," which allowed for cloud control, whereas third-party apps could only access the local API for control from within the home network. Most importantly, the remote API allowed users to control their lights when they were out of the home. Over time, the Philips Hue team decided to also open up the remote API to selected developers allowing them to offer extended functionality. Developers had to fill in a form on the developer's portal, explaining their envisioned use case. It was well received by the developer community as it opened up new possibilities for adding new features, contributing to enhanced complement quality over time (see Table 6 for additional examples: Hues, iHue, and Control for Hue).

Materializing glitches: Abandoning or maintaining integrity. Glitches are technical problems that occur when developers of different interdependent components or products are unaware of the implications their development actions may have on others (Hoopes and Postrel, 1999). While glitches mostly have been studied in the initial development of systemic products (e.g., Tiwana, 2008), in the case of digital platform ecosystems, glitches may materialize even between products that used to interoperate very smoothly. Because of the reprogrammability of the platform and other elements in the ecosystem, a complement may lose its compatibility unless it is updated too. Similarly, a complement may suddenly exhibit bugs when, enabled by the connectivity of digital technologies, users and other ecosystem actors create idiosyncratic connections with other complements that have never been anticipated. To maintain complement quality, complementors need to engage in bug fixing and ensuring compatibility to address such glitches and maintain the complement's integrity.

Indeed, the integrity of many apps for Hue depended on other elements in the platform ecosystem that continued to evolve over time. Because third-party apps are built on the Hue platform and interact with its interfaces, changes to the platform (e.g., a new version of the API) often resulted in glitches that caused apps to break. Thus, complement quality deteriorated when developers failed to maintain the integrity of their apps by fixing these glitches. For example, after Philips had released a firmware update for the Hue bridge, the "Speedy Hue" app failed to connect to the bridge with the latest firmware, thereby becoming dysfunctional and negatively impacting the functioning of the entire Hue system for its users. The following user review illustrates such negative consequences: "I don't expect the world from a Hue app, but I do expect it to connect to my up-to-date bridge,

just as all my other Hue apps do. This one, does not." ("Speedy Hue," July 26, 2016), leaving a one-star rating. In cases like this, when developers abandoned rather than maintained an app's integrity, the result of platform changes was deteriorating quality of a complement that previously may have worked well (see additional example for abandoning in Table 6: More for Hue).

In many cases, though, developers responded to platform changes and provided updates ensuring compatibility even if this sometimes proved to be rather complex. A Hue API change in November 2015, for instance, encumbered a developer to maintain his app's integrity. His app featured a popular "police pre-set," which "strobes all your lights, and three light bulbs alternate red and blue, so it looks like an American police car." This feature required the undocumented API key "points symbol," which was removed after the API change and caused the app to malfunction, in turn causing negative user ratings and feedback: "I got a lot of support requests saying: 'I got the new lights, strobing doesn't work: one star!' […] And then [the users] said: 'but it worked last year?' And I have to explain what a firmware update is […]. It is a waste of time that I would rather spend doing other things." Despite being a significant burden, the developer addressed the glitch and implemented the police pre-set with the new API (albeit without strobing), restoring the app's compatibility with the platform.

Besides changes in the platform core itself, changes in other complements, such as smartphones, operating systems (e.g., Apple iOS, Android), and hardware components (e.g., Zigbee compatible light bulbs or switches), triggered glitches that required developers to update their apps. As a consequence, developers had to keep up with a host of other interdependent products and services as well. For example, after Apple had released a new iOS version, one developer had to update his app to fix its voice control feature. He explained that: "the update in '14 was because […] Apple changed a lot of things regarding access to microphone and so on. And also, the sensitivity was different on the different iOS units." The developer explained how an update addressed the glitch regarding proper microphone use: "so instead of having a certain level in the app to detect voice […], I had to implement a slider to adjust it through the phone, so the app could recognize the voice."

In other instances, glitches materialized as users tried out the app in their idiosyncratic contexts. Such glitches often were the result of bugs, that is, mistakes or errors in the app code. Despite testing, bugs may go unnoticed as developers can only test apps using their own bridge and set of lights. Users have different setups entailing various combinations of lamps and bulbs, different smartphones and operating systems, and older or newer Hue bridge firmware versions. Accordingly, many glitches caused by bugs materialized only later due to specific (atypical) user interactions or their idiosyncratic setup. For example, one developer was surprised to find people using the app to connect with smart lights from other manufacturers such as Osram. Although Osram lights should work in principle, it compromised the app's integrity: "there's this weird issue that I have been trying to fix lately where [my Hue app] does react in an odd way, where it just doesn't turn off the light in a specific case, that's a fault of my app." Indeed, shortly thereafter, the developer released an update offering "Better Osram bulbs support" (release notes, December 2016). These and other examples (see Table 6: Sunrise, Smarter Hue, Rainbow, and Hurray) show how complementors maintained quality by ensuring compatibility with the platform ecosystem as well as the app's integrity over time.

Emerging obsolescence: Abandoning or enhancing functionality. In addition to materializing glitches, dynamics in the platform ecosystem affect complement quality by raising the bar for the functionality, predominantly in terms of the user experience expected from a "good" complement. Rapidly changing technological environments, such as digital platform ecosystems, can drive a product into technological obsolescence (Bstieler, 2005). If complementors do not keep up with this development, their products will become outdated and other elements in the ecosystem may take over some of its functionality. Complementors can respond to this emerging obsolescence by abandoning their app or continuously enhancing a complement's user experience, which may contribute to an enhanced complement quality over time.

We observed such dynamics in the Hue platform ecosystem. Various developers explained how they benchmarked their apps against the official Hue app, trying to top that user experience in order to deliver complement quality beyond what the official Hue app offered. Sometimes, major updates of other elements of the platform ecosystem gave rise

to new design standards. For example, when iOS7 introduced a "flat" user interface design, many developers followed as evidenced in their release notes ("Ready for iOS7 with a fresh new look"). Similarly, developers adopted new design standards for Android: "Switch to Android Material Design (Android 5.0+)." Keeping up with the most recent trends implied extensive additional work, particularly for older apps as they grew more complex over time. Yet, existing customers expected to get such updates for free. As one developer mentioned, "people are expecting too much for too few euros," which discouraged developers to continuously develop their app. While developers were generally committed to updating their app after initial launch, we observed that for several developers, the burden became too high over time, which is exemplified in the 26% of apps that were abandoned.

Developers were also pushed to enhance their app if competing apps or other elements in the Philips Hue ecosystem introduced similar or better functionality. For example, updates in the native Philips Hue app made one developer's app irrelevant even though it had been highly popular in the beginning. The developer reflected: "it is not that important anymore, my app. Perhaps I will take it down from the AppStore. [Philips] implemented many of the things from my app in the Philips app now." In a similar case, a previously unique app became largely obsolete. The developer was caught off guard when, only two weeks after releasing his unique feature to automatically switch on the lights upon arriving home, such a geofencing feature was also introduced in the official Hue app. The developer was "disappointed" and struggled: "[My] app […] competes with the official Hue app, and this is quite a complicated thing for me because, [when the Hue app] gets better […] I have to try to separate myself featurewise and have totally different features." Yet, he made a significant effort to keep up with the platform: "I am putting my full energy into it right now." To enhance complement quality, the developer had to develop yet another update to enhance its functionality by adding new features. Others, however, gave up and abandoned their app in similar situations (see Table 6: Home Light).

In addition, every time Philips extended its product portfolio with new lights, users expected third-party apps to support these too. One developer explained: "I continue to receive requests like 'hey, why doesn't it work with this light?' There was a guy [who] was constantly getting back to me: 'why don't you support all these products? […] I have the Philips Bloom light: why isn't it working?'" These increasing expectations, threatening to make an app obsolete over time, extended well beyond the Philips Hue product range, including lights by other manufacturers and even other digital product platforms. At the same time, reacting to such suggestions for functionality was a valuable opportunity to improve the user experience. As one developer noted: "Several people are purchasing the Hue bridge because they want to control multiple things." Enhancing an app's functionality by supporting an ever-expanding ecosystem implied an improved complement quality for these users, that is, they were able to control their continuously growing smart lighting system with just one app (see additional examples for enhancing functionality in Table 6: Control for Hue and Hueppy).

Roles of Ecosystem Actors in the Process of Sustained Development

So far, we have explained how complement quality is maintained over time through a process of sustained development. While our findings emphasize the central role of complementors in this process, other ecosystem actors play important roles, too, notably the platform owner and users. While complementors are responsible for maintaining the integrity and enhancing the functionality of their complements, the platform owner and users can alleviate the burden experienced by complementors. In addition, users also help to see new opportunities for sustained development.

Role of complementors. As evident from our case analysis so far, complementors play a key role in ensuring complement quality through maintaining integrity and enhancing functionality over time. Developers often described that fixing bugs and compatibility issues caused by ecosystem dynamics was "a lot of work and very time-consuming." This *burden* was further exacerbated as developers had to keep up with changes beyond the Hue platform, for example, changes in platforms such as Android or Apple iOS that were part of the broader ecosystem. To illustrate, one developer voiced his frustration about this process: "Every time Apple is making a major update, it is always very time consuming, and

sometimes you [try to] find a solution for [an issue] and then one month later it turns out to be Apple's fault."

Some developers who enjoyed the tinkering when creating their app found it challenging to stay motivated to maintain and update, which is a mundane task and not considered as "fun" and rewarding as initial development. Some professional developers who had several apps constantly had to decide how to divide their time and resources, sometimes at the cost of their Hue app. As one developer admitted: "I don't have a lot of attention to the app at the moment because I am so busy with other things. […] I have to make jobs that I know will pay well […] I have two very good customers that are taking up much of my time, and I can't let them down." As a result of these constraints, many developers tend to neglect the maintenance of their apps over time, and some eventually abandoned them (see additional examples in Table 6: Home Light and More for Hue).

However, developing an app update could also represent an *opportunity* for developers—particularly in instances where development efforts were triggered by new affordances emerging from changes in the platform ecosystem. App updates often resulted in new features and better user experience, which enhanced the quality of apps (see additional examples in Table 6: Hues, iHue, Control for Hue and Hueppy). Some apps were highly successful and exceeded the developer's expectations, which further motivated them to engage in a sustained development process. As one developer explained: "I have been working on [my app] pretty regularly since launch, but I am really surprised how it has done [laugh]. I had no idea how well it was going to take off. Because of that, I put more effort in than originally planned." Accordingly, some developers who initially saw their app as a hobby project decided to invest more time in their apps after realizing that they could generate their income from app sales.

Role of the platform owner. Philips Hue's actions as a platform owner influenced the sustained development of complements by reducing the burden associated with glitches. Specifically, Philips Hue helped developers by informing them about platform changes. Although all platform changes should ideally be backward compatible, in practice, this was sometimes impossible. In such cases, the way a platform change is introduced to the developers matters. Over time, we observed that Philips Hue shifted from a *reactive* to an increasingly *proactive* approach to deal with the interdependency between its platform and complementary apps. To illustrate, when Philips released a new "white" light (that did not support colors), they did not inform developers upfront about how to send commands to this particular light—it had "gone right into […] the shops." The Philips' developer program manager recalled that this new hardware caused third-party "apps to behave oddly" because these apps could not recognize the white lights. As evident in the spike in updates after August 2014 (see Figure 3), the changes introduced by Philips necessitated updates in many third-party apps. This frustrated developers and users, who could not have anticipated the effort needed to keep their apps and home setup functioning (see also Table 6: Sunrise).

Over time, Philips became more aware of how their continuous platform development affected developers. For example, in April 2016, when Philips released a security update that could potentially break a large share of the third-party apps, the Hue team informed developers seven months in advance. They repeatedly announced the upcoming changes to spur developers to proactively update their apps to comply with the new way of pairing with the Hue bridge. To minimize any negative consequences, Philips also undertook a significant effort to pretest all third-party apps and tracked down and contacted developers who had ignored prior calls. This "total head ups" was positively received by developers, as illustrated by one developer who enthusiastically remarked, "that was a nicely managed roll out!"

The effectiveness of Philips' proactive informing approach is reflected in the pattern of updates before and after the change (see Figure 3). During the time between communicating the security update to developers and its actual implementation, developers updated their apps more frequently than usual, suggesting they were proactively keeping up the quality of their app. The coordinated effort by developers and Philips had prevented app quality from being at stake (see additional example in Table 6: Smarter Hue).

The platform owner also helped complementors addressing glitches resulting from idiosyncratic connections. First, as part of reactive informing, Philips provided detailed documentation and developer resources that helped developers understand problems and how to fix them. Second, as part of proactive informing, Philips Hue established a beta-testing community that brought together various worldwide

users and developers testing platform changes (e.g., updated bridge firmware, updated app, new bulbs or accessories) before these were officially launched. This beta-testing community could test new elements in the context of their specific complementary apps and setups to find out "what goes wrong, what do we break." The feedback from beta-testers enabled the Hue team to fix issues before the official release or identify remaining problems in particular user setups, allowing developers to address the glitch in a timely manner.

Role of users. Our findings also revealed an important role for the users in assisting developers by *searching* and *reporting glitches* as well as *suggesting functionality* for an improvement of the complement. First, users reached out to developers when apps did not work properly (e.g., if an app failed to connect with the Hue bridge, rendering the app useless). For example, one developer recalled how he received "tons of messages" the day after a new release, signaling that "something was wrong." The developer then tried to figure out what was causing the problem. While much of these inputs allowed for reactive maintenance, users could also proactively reduce the burden for developers by participating in the Hue beta-testing community. Together with the platform owner and complementors, users performed a critical role in the sustained development process by proactively searching for glitches.

Second, users provided direct feedback that inspired developers to improve the user experience. This could happen, for example, through questions from confused users that did not understand how the app was supposed to work. One developer explained that he had been receiving "ten emails a day" and that he read all of them because it helped in his ongoing development process: "it takes a lot of time to get through them, but I think that the user feedback and the great support is the most important process in this app, because making sure the users are happy, that is the best way to make the app grow." Furthermore, user's expectations also gave developers *suggestions for enhancing functionality*. For example, one developer explained that he received "all kinds of ideas" that were instrumental input for his app development. He reflected that "some (ideas) are very good and make total sense. I can give you many examples! […] because people always said: 'Hey is there a way to backup? Because every time I install an app I have to enter everything again.' So, I included this feature where you can backup and restore using iCloud."

Besides responding to user feedback, some developers went one step further by actively involving the users *during* the development process. For example, when deciding which features to add in the next update, some developers polled users' opinions and even shared prototypes to test if users were interested at all. One developer reflected: "because of the feedback on my proof of concept video, I decided to implement it in the app, [even though] personally, I don't like voice control! [laughs]." Users sometimes also played a role in improving an app's user experience by working directly with developers. For instance, one important way to improve an app's user experience is to add additional language support. To accomplish this, one developer relied on Russian, Italian, and German users who volunteered to translate the app's user interface.

Discussion

Our study responds to calls for innovation management research to investigate the consequences of digital transformation for organizing innovation (Nambisan et al., 2017). We extend insight into the process of sustained development needed to ensure the quality of complementary products in digital platform ecosystems. We found that the process of sustained development extends beyond postlaunch activities, such as monitoring of market performance (e.g., Cooper, 2008). Also, it is more complicated than incremental product innovation because it concerns updating products that are in use and connected in myriad ways with other ecosystem elements. Our process analysis of the Hue platform ecosystem over four years revealed that, without additional action, platform dynamics may lead to *materializing glitches* and *emerging obsolescence*, and ultimately deteriorating quality. Concerted activity by all ecosystem actors—complementors, platform owner, and users—is needed to mitigate glitches and obsolescence to maintain complement quality and grasp *expanding affordances* to enhance it. Below we discuss implications for theory on platform ecosystems.

Theoretical Implications

Our findings offer a *relational* and *dynamic* perspective on complement quality in platform ecosystems by considering both technical and social aspects. Whereas technology management and information

systems scholars have mainly studied the technical architecture of platforms, and strategic management scholars have studied the interactions and roles of platform ecosystem actors, we follow calls to consider such technical and social aspects in tandem to better understand platform dynamics (Gawer, 2014; McIntyre and Srinivasan, 2017). We do so using a relational perspective that considers connections between products as well as actors and sees these connections as dynamic (Garud et al., 2013; Yoo et al., 2012). Overall, we identified three mechanisms (expanding affordances, materializing glitches, emerging obsolescence) affecting complements, which are triggered by dynamics in platform ecosystems, and point to the roles of ecosystem actors (complementors, platform owners, and users) in jointly acting to enhance or maintain complement quality. Our findings have implications for both literature streams on platform ecosystems.

First, we enrich technology management research on platform ecosystems by explaining that complement quality extends beyond the *bilateral* relation between platform core and complement and is dynamic in nature. Although scholars have acknowledged that complementary value is inherently relational—as it refers to the value of products together (Baldwin, 2018; Jacobides et al., 2018)—the few studies that have explicitly considered relational aspects of complement quality have focused exclusively on the bilateral relation between platform core and complement (Cennamo et al., 2018; Saadatmand, Lindgren, and Schultze, 2019; Zhu and Iansiti, 2012), even though digital products allow manifold and reprogrammable connections (Henfridsson et al., 2018; Yoo et al., 2010). Our findings unveiled that complement quality depends on the multiple relations with other products, which could interact in unexpected and dynamic ways. The apps that we studied not only interacted with the platform core (i.e., the bridge and API), but also with a wide variety of other ecosystem elements, such as various smart lights by Hue and other brands, connected home products, other home automation apps, smartphones and associated platforms such as iOS.

Our process model explains how the multitude of connections in a platform ecosystem and associated dynamics influence complement quality after launch. Specifically, we identified three underlying theoretical mechanisms triggered by dynamics in the platform ecosystem: *expanding affordances*, *materializing glitches*, and *emerging obsolescence*. Informed by literatures that have used these concepts to understand the relational and dynamic nature of phenomena, we explain the importance of these mechanisms for complement quality in platform ecosystems. Glitches and obsolescence are known as side-effects of other development efforts (Bstieler, 2005; Hoopes and Postrel, 1999). Our study shows how these originate from various interdependent products, adding to the difficulty to foresee them. Quality may deteriorate through glitches that materialize after complementary products have been introduced, as an unintended consequence of changes in the platform core and other ecosystem elements developed by many different ecosystem actors. Also, obsolescence may occur unexpectedly in rapidly changing environments, such as digital ecosystems (Bstieler, 2005). As a core platform component or another complement introduces similar functionality at a better user experience, obsolescence may negatively impact complement quality. Similarly, fueled by the reprogrammability and connectivity of digital technologies (Autio et al., 2018; Volkoff and Strong, 2013), affordances often expanded unintentionally, as a by-product of changes in the platform or other ecosystem.

Thus, whereas prior research took complement quality for granted or considered it as stable over time, our findings imply that complement quality is not fixed once complements are in use but may deteriorate due to the materializing glitches and emerging obsolescence, yet, increase when complementors act upon expanding affordances. This was exemplified in user ratings of the same complement that changed considerably over time. Our findings suggest that a relational and dynamic conceptualization of complement quality is particularly important for studying digital product ecosystems. Prior research that operationalized complement quality through single measurements of user or expert ratings (Binken and Stremersch, 2009; Cennamo et al., 2018; Gretz et al., 2019) was primarily focused on earlier generations of video game consoles and games. These were less subject to dynamics of complement quality as these platforms and complements were only updated through discrete next-generation of the ecosystem and did not allow for continuous updating.

Second, our findings also contribute to understanding the relations and roles of ecosystem actors, which has been the focus of the strategy literature on platform dynamics. Our process model explains how

quality depends on different actors' responses to affordances, glitches, and obsolescence, demonstrating that technical and social relations need to be considered in tandem. We show how managing complement quality in platform ecosystems involves a distributed innovation process in which ecosystem actors become increasingly interdependent, yet, remain formally autonomous. Different from traditional perspectives on collaborative relations (e.g., Dyer and Singh, 1998), these ecosystem collaborations are more diverse and at arm's length, without strong bonds such as in alliances, or even without contractual arrangements at all. Sustaining quality over time becomes an ongoing concern for all ecosystem actors (complementors, users, and platform owners), who increasingly depend on each other in the process of sustained development. To date, literature on the governance of platform ecosystems has primarily focused on the dyadic relation between platform owners and complementors (e.g., Boudreau, 2017; Saadatmand et al., 2019; Wareham et al., 2014). Our findings, however, emphasize the importance of considering the triadic relation between platform owner, complementors, and users.

More specifically, we unpack the distinctive role of *complementors* after launch. Existing literature explained why and how complementors initially affiliate with a platform to develop complementary products or services (e.g., Boudreau and Jeppesen, 2015) or leave a platform (Tiwana, 2015b). In contrast, our process model of sustained development (Figure 4) identifies the different pathways of engagement over time. That is, the Hue case shows the need for *sustained* development to maintain complement quality, because the value that complements contribute to a platform may otherwise deteriorate as a result of changes in the platform ecosystem. We found that some complementors are only motivated to create a product but not to maintain it, as this involves different, more mundane tasks and a different set of motivations. The burden of maintaining integrity imposed on complementors after initial development may result in abandoning the platform ecosystem. In contrast, other complementors remain motivated to solve problems after glitches materialize or even proactively avoid potential glitches, for instance, by inviting users to participate in beta-testing before officially releasing a new version.

Regarding the role of the *platform owners*, our findings point to pathways for ensuring quality over time and help explain why they are restricted in their ability to fully control quality. Prior literature has predominantly focused on the role that platform owners play by setting the initial quality requirements for complements that can be enforced through selective admission (e.g., Ghazawneh and Henfridsson, 2013; Tiwana, 2013) or selective promotion of complements (Rietveld et al., 2019). We find that a platform owner—Philips Hue in our setting—is severely constrained in its ability to control complement quality. Because users independently select and recombine complements with other ecosystem elements, like their smartphones and unique lighting setup, platform owners and complementors cannot anticipate—let alone test—all possible recombinations, which will continue to evolve over time. Moreover, because platform ecosystems become increasingly dependent on other platforms (Hilbolling, Berends, Deken, and Tuertscher, 2020)—i.e., Philips Hue depends on the Apple and Android App Stores—their ability to act as "gatekeeper" (e.g., Ghazawneh and Henfridsson, 2013) is limited. Instead, platform owners have to rely on indirect approaches: they can proactively inform developers about upcoming platform changes so that developers can anticipate these; establish a beta testing community to reveal possible glitches in a large variety of user setups; and offer high-quality documentation to developers; and spur them to exploit expanding affordances.

With regard to the role of the *users,* we uncover the critical role they play in the sustained development of platform ecosystems, which has been mostly overlooked in the literature. Expanding on the insight that digital innovations are built on the recombination of digital resources *in use* (Henfridsson et al., 2018), the user's role changes from a mere consumer to an actor that shapes platform ecosystems. To date, most studies have focused on the onboarding of complementors, which regards situations where complementors do not yet have an active user base (e.g., Boudreau, 2010). The Hue case revealed that users of third-party apps contribute to the sustained development in platform ecosystems by providing qualitative feedback to complementors, searching for and reporting of glitches, and suggesting ways to enhance a complement's functionality. Even complementors with a small user base can benefit from user feedback: our findings suggest that not the amount of feedback or quantitative ratings mattered, but the qualitative input derived from direct messages and contact with users. These interactions between users and developers are mostly hidden for platform owners, yet, vital for the health of their platform ecosystem.

Third, our findings offer new insights into the fundamental concept of network effects. Prior research juxtaposed *platform* quality against (indirect) network effects as separate competitive forces (e.g., McIntyre, 2011; Tellis, Yin, and Niraj, 2009). The few studies on *complement* quality and network effects only studied complement quality as a driver for platform growth (e.g., Binken and Stremersch, 2009; Gretz et al., 2019). Our study suggests that quality might also have a negative influence on network effects: quality problems of complements can make them less attractive for users. Because such negative experiences are rooted in the interdependencies between the ecosystem actors, it may backfire on those actors, even if this is not deserved. For instance, users may blame the platform owner for malfunctioning, even though it does not control complements. Vice versa, users could blame app developers for app failures, although these might have been triggered by platform owner actions. This implies that platform owners also need to educate users on these roles and dependencies in the ecosystem.

Moreover, we also offer an additional mechanism for how indirect network effects operate, rooted in users' contribution to quality. More users imply that more people consider quality aspects (e.g., finding bugs and suggesting new features). When users inform complementors about bugs and new feature ideas, they can use such comments to improve the quality of their complements, thereby generating more value for the user base in return. Thus, the positive forces of network effects not only depend on the quantity of complements—as has been the core focus of prior platform research (Boudreau, 2012; Katz and Shapiro, 1986)—but also on their quality and users' attention for quality.

Limitations, Future Research, and Conclusions

Our study has a number of limitations that need to be considered. First, we have mainly considered the role of the Philips Hue platform ecosystem. However, third party developers also depend on other platforms, such as smartphone operating systems. Future research is needed to take such interdependencies also into consideration. Second, we primarily considered app updates as an indicator of sustained development to ensure the quality of complements. An interesting opportunity for future research is to use app reviews and ratings as a measure for subjective quality perception by users to test our findings on a larger scale. Yet, it should be acknowledged that users may use negative app ratings to blame developers for things unrelated to their app (e.g., to voice frustration or get attention), making it an indicator of the actual quality that needs to be handled with care.

Notwithstanding these limitations, the Philips Hue case sheds light on the transformation process that traditional product-centric firms go through when moving from developing, producing, and selling physical products to the orchestration of an actors around a digital platform ecosystem that heavily relies on the quality and availability of complementary products. This paper points out the importance and complexity of sustained development to ensure the quality of complements over time, a relevant topic for a growing number of firms who engage with digital technologies in their product development processes in the context of platform ecosystems. We provide practical insight into the ways in which platform owners can alleviate the burden of sustained development for complementors, notably through taking a proactive role in organizing platform updates to avoid negative consequences such as glitches.

References

Adner, R., P. Puranam, and F. Zhu. 2019. What is different about digital strategy? From quantitative to qualitative change. *Strategy Science* 4 (4): 253–61.

Autio, E., S. Nambisan, L. D. Thomas, and M. Wright. 2018. Digital affordances, spatial affordances, and the genesis of entrepreneurial ecosystems. *Strategic Entrepreneurship Journal* 12 (1): 72–95.

Baldwin, C. Y. 2018. *Design rules (Volume 2): How technology shapes organizations. Chapter 5: Complementarity*. Harvard Business School Working Paper 19-036.

Baldwin, C. Y., and C. J. Woodard. 2009. The architecture of platforms: A unified view. In *Platforms, markets and innovation*, ed. A. Gawer, 19–44. Cheltenham: Edward Elgar.

Binken, J. L. G., and S. Stremersch. 2009. The effect of superstar software on hardware sales in system markets. *Journal of Marketing* 73 (2): 88–104.

Boudreau, K. J. 2010. Open platform strategies and innovation: Granting access vs. devolving control. *Management Science* 56 (10): 1849–72.

Boudreau, K. J. 2012. Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. *Organization Science* 23 (5): 1409–27.

Boudreau, K. J., and L. B. Jeppesen. 2015. Unpaid crowd complementors: The platform network effect mirage. *Strategic Management Journal* 36 (12): 1761–77.

Boudreau, K. J. 2017. Platform boundary choices & governance: Opening-up while still coordinating and orchestrating. In *Entrepreneurship, innovation, and platforms*, ed. J. Furman, A. Gawer, B. S. Silverman and S. Stern, 227–297. Emerald: Bingley.

Bstieler, L. 2005. The moderating effect of environmental uncertainty on new product development and time efficiency. *Journal of Product Innovation Management* 22 (3): 267–84.

Cennamo, C. 2018. Building the value of next-generation platforms: The paradox of diminishing returns. *Journal of Management* 44 (8): 3038–69.

Cennamo, C., and J. Santalo. 2013. Platform competition: Strategic trade-offs in platform markets. *Strategic Management Journal* 34 (11): 1331–50.

Cennamo, C., and J. Santalo. 2019. Generativity tension and value creation in platform ecosystems. *Organization Science* 30 (3): 617–41.

Cennamo, C., H. Ozalp, and T. Kretschmer. 2018. Platform architecture and quality tradeoffs of multihoming complements. *Information Systems Research* 9 (2): 461–78.

Claussen, J., T. Kretschmer, and P. Mayrhofer. 2013. The effects of rewarding user engagement: The case of Facebook apps. *Information Systems Research* 24 (1): 186–200.

Clements, M. T., and H. Ohashi. 2005. Indirect network effects and the product cycle: Video games in the US, 1994–2002. *The Journal of Industrial Economics* 53 (4): 515–42.

Cooper, R. 2008. The stage-gate idea-to-launch process: Update, what's new and NexGen systems. *Journal of Product Innovation Management* 25 (2): 213–32.

Dyer, J. H., and H. Singh. 1998. The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy of Management Review* 23 (4): 660–79.

Eaton, B., S. Elaluf-Calderwood, C. Sørensen, and Y. Yoo. 2015. Distributed tuning of boundary resources: The case of Apple's iOS service system. *MIS Quarterly* 39 (1): 217–44.

Eisenhardt, K. M. 1989. Building theories from case study research. *Academy of Management Review* 14 (4): 532–50.

Eisenmann, T., G. Parker, and M. Van Alstyne. 2006. Strategies for two-sided markets. *Harvard Business Review* 84 (10): 92–101.

Evans, D., A. Hagiu, and R. Schmalensee. 2006. *Invisible engines: How software platforms drive innovation and transform industries*. Cambridge: MIT Press.

Evered, R., and M. R. Louis. 1981. Alternative perspectives in the organizational sciences: "Inquiry from the inside" and "inquiry from the outside". *Academy of Management Review* 6 (3): 385–95.

Faulkner, P., and J. Runde. 2012. On sociomateriality. In *Materiality and organizing: Social interaction in a technological world*, ed. P. M. Leonardi, B. A. Nardi, and J. Kallinikos, 49–66, Oxford: Oxford University Press.

Fisher, G., and H. Aguinis. 2017. Using theory elaboration to make theoretical advancements. *Organizational Research Methods* 20 (3): 438–64.

Garud, R., S. Jain, and P. Tuertscher. 2008. Incomplete by design and designing for incompleteness. *Organization Studies* 29 (3): 351–71.

Garud, R., and A. Kumaraswamy. 1995. Technological and organizational designs for realizing economies of substitution. *Strategic Management Journal* 16: 93–109.

Garud, R., P. Tuertscher, and A. H. Van de Ven. 2013. Perspectives on innovation processes. *Academy of Management Annals* 7 (1): 775–819.

Gawer, A. 2014. Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research Policy* 43 (7): 1239–49.

Gawer, A., and M. A. Cusumano. 2014. Industry platforms and ecosystem innovation. *Journal of Product Innovation Management* 31 (3): 417–33.

Ghazawneh, A., and O. Henfridsson. 2013. Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal* 23 (2): 173–92.

Gibbert, M., and W. Ruigrok. 2010. The "what" and "how" of case study rigor: Three strategies based on published work. *Organizational Research Methods* 13 (4): 710–37.

Goffin, K., P. Åhlström, M. Bianchi, and A. Richtnér. 2019. Perspective: State-of-the-art: The quality of case study research in innovation management. *Journal of Product Innovation Management* 36 (5): 586–615.

Gretz, R. T., A. Malshe, C. Bauer, and S. Basuroy. 2019. The impact of superstar and nonsuperstar software on hardware sales: The moderating role of hardware lifecycle. *Journal of the Academy of Marketing Science* 47 (3): 1–23.

Hagiu, A.2011. Quantity vs. quality: Exclusion by platforms with network effects. Harvard Business School Working Paper 11-125.

Henfridsson, O., J. Nandhakumar, H. Scarbrough, and N. Panourgias. 2018. Recombination in the open-ended value landscape of digital innovation. *Information and Organization* 28 (2): 89–100.

Hilbolling, S., H. Berends, F. Deken, and P. Tuertscher. 2020. Complementors as connectors: Managing open innovation around digital product platforms. *R&D Management* 50 (1): 18–30.

Hoopes, D. G., and S. Postrel. 1999. Shared knowledge, "glitches", and product development performance. *Strategic Management Journal* 20 (9): 837–65.

Huber, G. P., and D. J. Power. 1985. Retrospective reports of strategic-level managers: Guidelines for increasing their accuracy. *Strategic Management Journal* 6: 171–80.

Jacobides, M. G., C. Cennamo, and A. Gawer. 2018. Towards a theory of ecosystems. *Strategic Management Journal* 39 (8): 2255–76.

Katz, M. L., and C. Shapiro. 1986. Technology adoption in the presence of network externalities. *Journal of Political Economy* 94 (4): 822–41.

Langley, A. 1999. Strategies for theorizing from process data. *Academy of Management Review* 24 (4): 691–710.

Mäkinen, S. J., M. Seppänen, and J. R. Ortt. 2014. Introduction to the special issue: Platforms, contingencies and new product development. *Journal of Product Innovation Management* 31 (3): 412–6.

McIntyre, D. P. 2011. In a network industry, does product quality matter? *Journal of Product Innovation Management* 28 (1): 99–108.

McIntyre, D. P., and A. Srinivasan. 2017. Networks, platforms, and strategy: Emerging views and next steps. *Strategic Management Journal* 38 (1): 141–60.

McIntyre, D., A. Srinivasan, A. Afuah, A. Gawer, and T. Kretschmer. 2020. Multi-sided platforms as new organizational forms. *Academy of Management Perspectives*. Advance online publication. https://doi.org/10.5465/amp.2018.0018.

Miles, M. B., and A. M. Huberman. 1994. *Qualitative data analysis: An expanded sourcebook* (2nd ed.). London: Sage Publications.

Mohr, L. B. 1982. *Explaining organizational behavior*. San Francisco, CA: Jossey-Bass.

Nambisan, S., K. Lyytinen, and A. Majchrzak. 2017. Digital innovation management: Reinventing innovation management research in a digital world. *MIS Quarterly* 41 (1): 223–38.

Panico, C., and C. Cennamo. 2020. User preferences and strategic interactions in platform ecosystems. *Strategic Management Journal*. Advance online publication. https://doi.org/10.1002/smj.3149.

Pettigrew, A. 1990. Longitudinal field research on change: Theory and practice. *Organization Science* 1 (3): 267–92.

Poole, M. S., A. H. Van de Ven, K. J. Dooley, and M. E. Holmes. 2000. *Organizational change and innovation processes*. Oxford: Oxford University Press.

Rietveld, J., M. A. Schilling, and C. Bellavitis. 2019. Platform strategy: Managing ecosystem value through selective promotion of complements. *Organization Science* 30 (6): 1232–51.

Rochet, J.-C., and J. Tirole. 2003. Platform competition in two-sided markets. *Journal of the European Economic Association* 1 (4): 990–1029.

Saadatmand, F., R. Lindgren, and U. Schultze. 2019. Configurations of platform organizations: Implications for complementor engagement. *Research Policy* 48 (8): 103770.

Sayer, R. A. 1992. *Method in social science: A realist approach* (2nd ed.). London: Routledge.

Schilling, M. A. 2002. Technology success and failure in winner-take-all markets: The impact of learning orientation, timing, and network externalities. *Academy of Management Journal* 45 (2): 387–98.

Selander, L., O. Henfridsson, and F. Svahn. 2013. Capability search and redeem across digital ecosystems. *Journal of Information Technology* 28 (3): 183–97.

Signify. 2018. Annual report 2018. Available at https://www.signify.com/static/2018/signify-annual-report-2018.pdf.

Song, M., M. E. Parry, and T. Kawakami. 2009. Incorporating network externalities into the technology acceptance model. *Journal of Product Innovation Management* 26 (3): 291–307.

Stremersch, S., G. J. Tellis, P. H. Franses, and J. L. Binken. 2007. Indirect network effects in new product growth. *Journal of Marketing* 71 (3): 52–74.

Tellis, G. J., E. Yin, and R. Niraj. 2009. Does quality win? Network effects versus quality in high-tech markets. *Journal of Marketing Research* 46 (2): 135–49.

Tiwana, A. 2008. Does interfirm modularity complement ignorance? A field study of software outsourcing alliances. *Strategic Management Journal* 29 (11): 1241–52.

Tiwana, A., B. Konsynski, and A. A. Bush. 2010. Platform Evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research* 21 (4): 675–87.

Tiwana, A. 2013. *Platform ecosystems: Aligning architecture, governance, and strategy*, Waltham: Morgan Kaufmann.

Tiwana, A. 2015a. Evolutionary competition in platform ecosystems. *Information Systems Research* 26 (2): 266–81.

Tiwana, A. 2015b. Platform desertion by app developers. *Journal of Management Information Systems* 32 (4): 40–77.

Tsoukas, H. 1989. The validity of idiographic research explanations. *Academy of Management Review* 14 (4): 551–61.

Van de Ven, A. H. 2007. *Engaged scholarship: A guide for organizational and social research*. Oxford: Oxford University Press.

Van de Ven, A. H., D. Polley, R. Garud, and S. Venkataraman. 1999. *The innovation journey*. Oxford: Oxford University Press.

Volkoff, O., and D. M. Strong. 2013. Critical realism and affordances: Theorizing IT-associated organizational change processes. *MIS Quarterly* 37 (3): 819–34.

Wareham, J., P. B. Fox, and J. L. Cano Giner. 2014. Technology ecosystem governance. *Organization Science* 25 (4): 1195–215.

Yin, R. K. 2009. *Case study research, design and methods* (5th ed.). London: Sage.

Yoo, Y., O. Henfridsson, and K. Lyytinen. 2010. The new organizing logic of digital innovation: An agenda for information systems research. *Information Systems Research* 21 (4): 724–35.

Yoo, Y., R. J. Boland, K. Lyytinen, and A. Majchrzak. 2012. Organizing for innovation in the digitized world. *Organization Science* 23 (5): 1398–408.

Wang, R. D., and C. D. Miller. 2020. Complementors' engagement in an ecosystem: A study of publishers' e-book offerings on Amazon Kindle. *Strategic Management Journal* 41 (1): 3–26.

Zhu, F., and M. Iansiti. 2012. Entry into platform-based markets. *Strategic Management Journal* 33 (1): 88–106.

Zittrain, J. 2006. The generative Internet. *Harvard Law Review* 119: 1974–2040.

Appendix: General Interview Guide Third-Party App Developers

Developer Background

How long have you been developing apps/software?
Did you get any formal training in programming/software/IT?
Are you a professional developer and/or do you develop apps in your spare time?
Was this Hue app your first app or did you develop apps before this one? *(what kind of app?)*
For which platform(s) do you develop apps (iOS, Android, other)?

About the Hue, Hue App and Development Process

When and where did you first hear about Hue? *(what was your first impression?)*
What Hue products do you own? *(which products, how many)*
Can you briefly explain what your Hue app is about?
Where and when did you get the idea for the app?
Do you earn money with the app? (*if applicable, how important is earning money with the app?)*
What attracted you to develop for Hue?
When did you decide to start with developing an app for Hue? *(what was the first step?)*
How much time did you spend on developing the app *before* initial launch and *after*? *(what were the major tasks involved?)*
What do you think about the API and documentation provided by Philips Hue? (*what do you like/dislike, can you give an example?)*
What challenges did you face during the development process? *(how did you solve these?)*
Once it was on the market, how much work is maintaining/updating it? *(what were the major activities?)*
Added: specific questions about updates depending on the developer's app(s) release notes

Interaction with Users

Do you actively promote your app? *(if applicable, how and where?)*
Are you satisfied with the number of downloads? *(what were your expectations? do download matter to you?)*
What kind of feedback do you get from your users?
How do you interact with your users? *(what channels do you use?)*

Interaction with Other Developers

Do you know other people that develop for Hue?
Did you get help during the process from other developers or did you develop it entirely by yourself? *(what kind of help did you seek, and where)*
Did you help other developers during their development process? *(if yes, how and with what?)*
Do you look at other developer's apps? *(how does it affect your own development?)*
Do you attend hackathons or other events/meetups? (*if applicable, can you give an example? what do you like about them?)*

Interaction with Philips Hue

During the development of your app, did you contact Philips? *(if yes, when and for what reasons)*
How do you communicate with Philips? *(about what do you communicate, what channels do you use?)*
If and how do you keep track of what Philips is doing concerning the Hue?
Did recent developments in software/hardware affected your app? *(if so, how?)*
What do you (dis)like about the Philips Hue system in general and developing for Hue in particular?

Beyond Hue App

Besides the app, did you develop code/libraries for Hue?
Do you also develop software/apps for other smart home/lighting systems?
Are you involved in other developer communities?
How do you compare developing the hue versus other apps you have developed?
Do you consider developing more apps for Hue in the future?