The Impact of Integration on Application Success and Customer Satisfaction in Mobile Device Platforms

Benedict Bender

Abstract Digital software platforms allow third parties to develop applications and thus extend their functionality. Platform owners provide platform boundary resources that allow for application development. For developers, platform integration, understood as the employment of platform resources, helps to realize application functionality effectively. Simultaneously, it requires integration effort and increases dependencies. Developers are interested to know whether integration contributes to success in hyper-competitive platform settings. While aspects of platform participation have been studied, research on a comprehensive notion of integration and related implications are missing. By proposing a platform integration model, this study supports a better understanding of integration. Concerning dynamics related to integration, effects were tested using information from over 82,000 Apple AppStore applications. Regression model analysis reveals that application success and customer satisfaction is positively influenced by platform integration. To achieve superior results, developers should address multiple aspects of integration, such as devices, data, the operating system, the marketplace as well as other applications, and provide updates. Finally, the study highlights the importance for all platform participants and their possibilities to employ integration as a strategic instrument.

1 Introduction

Digital platforms have recently become a common phenomenon in the context of connected digital devices. Devices such as Smartphones, Tablets, Smart TVs and Connected Cars are equipped with access to the producer's digital platforms and are part of their ecosystems. Digital platforms allow external contributors to extend the functionality beyond the platform's core functionality provided by the platform owner (Tiwana et al. 2010). Thereby, digital platforms are able to provide users with a more diverse range of functionality than a single entity can realize alone (Eisenmann et al. 2011).

Typically, extended functionality is provided through capsuled code fragments, often referred to as applications or add-ons. These applications are distributed by means of the marketplace maintained by the platform owner. Platform customers utilize the marketplace to search and gather applications for their devices. In order to be distributed through the marketplace, third-party applications need to comply with platform design rules (Ghazawneh and Henfridsson 2015). Applications are deployed using specialized boundary resources provided by the platform owner. Platform-specific Software Development Kits (SDKs) allow third parties to effectively realize their application on the individual platform. Application Programming Interfaces (APIs), as part of the boundary resources, allow

B. Bender (✉)
Chair of Business Informatics, esp. Processes and Systems, University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany
e-mail: benedict.bender@wi.uni-potsdam.de

applications to access platform resources and thereby establish integration with the platform.

Integration is of utmost importance considering the highly modularized software architecture of digital platforms as well as the distributed development with various actors participating on a platform (Tiwana 2015a). Integration in this study needs to be seen as distinct from its simplest version, that is, participation. While *participation* of third-party developers refers to providing applications for the platform regardless of their interaction with the platform (Ceccagnoli et al. 2012), *integration* means the effective usage of platform boundary resources in order to interact with the platform core and exchange information, as well as to use the features provided. Integration can be considered a more intense form of participation. Whereas participation is present for every third-party application on the platform, the degree of integration varies widely between applications. While the use of some boundary resources is mandatory in order to realize applications (participation), developers are free to choose most of the boundary resources to be used in application development (integration) (Ghazawneh and Henfridsson 2013). For the platform owner, integration might be desirable as it increases control over platform activities and allows for providing a uniform user experience (Boudreau 2010; Eaton et al. 2015). For contributors, integration can be an effective way to realize applications and utilize advanced platform features; however, it requires an additional platform-specific integration effort and simultaneously increases dependencies (Hsieh and Hsieh 2013; Kim et al. 2016). Developers are therefore interested in whether the integration effort involved will pay off.

While positive effects have been found for platform participation, there is a lack of research on integration which is addressed in this study. In contrast to participation, understanding the role and implications of integration requires a more detailed view of platform architecture, especially concerning different types and levels of integration for applications on digital platforms. Since neither a systematic consideration of integration aspects nor a related model is available, we review related platform concepts to better understand integration possibilities prior to exploring the research question: *How does application integration on digital platforms impact application success and customer satisfaction?* By conceptualizing an application integration model based on established platform concepts, this study addresses a literature gap and provides a basis for subsequent analysis. The model distinguishes aspects of integration and provides means to measure platform integration. Subsequently, the model is operationalized and applied to the Apple iOS platform using linear regression to assess the impact of integration for over 82,000 applications. Application success and customer satisfaction are used since these are most prevalent among developer targets and for characterizing application performance (Lee and Raghu 2014; Siegfried et al. 2015). The analysis provides valuable insights for developers and platform owners. Related implications are discussed.

2 Related Literature

We refer to a software platform following Tiwana et al. (2010) as an "extensible codebase of a software-based system that provides core functionality shared by the modules that interoperate with it and the interfaces through which they interoperate". A module, in the mobile domain referred to as application, is a "[···] software subsystem that connects to the platform to add functionality to the platform" (Tiwana et al. 2010). Following Ghazawneh and Henfridsson (2015) we consider an application marketplace to be "a platform component that offers a venue for exchanging applications between developers and end-users [···]".

2.1 Participation on Digital Platforms

Considering the *effects of participation* as a predecessor of integration helps to assess similarities and differences to integration. For the platform owner and contributors, positive economic effects related to participation on platforms were found. For the platform owner, adopting a business model incorporating a digital platform could result in multiplied market value compared to work as a stand-alone company. It has been found that the ecosystem was the main reason for this (Antero and Bjørn-Andersen 2013). In a similar vein, financial effects for contributors include positive effects on sales and an increased likelihood of an initial public offering (Ceccagnoli et al. 2012). Regarding the contributions themselves, higher prices were found for commodities traded through a digital platform (Banker et al. 2011).

2.2 Platform Openness

Platform openness constitutes a prerequisite for integration and is a major design element of digital platforms. Gawer and Cusumano (2014) distinguish internal and external industry-wide platforms. Open platforms allow third parties to participate without restrictions, whereby each side can be considered individually (Eisenmann et al. 2009). While openness is less important for end-users (Nikou et al. 2014), it is highly important for contributors. Koch and Kerschbaum (2014) consider openness to be important for developers in terms of innovation and resource expenditure. Hilkert et al. (2011) identified openness as a

satisfaction factor for developers. Exemplarily, Google's Android platform is considered more open than the competing Apple platform (Ghazawneh and Henfridsson 2015; Parker et al. 2017).

Since third parties' abilities to integrate as well as related boundary resources are controlled by the platform owner, openness is considered a requirement for integration (Tilson et al. 2012) and has been given much attention (Thomas et al. 2014). Since platform resources change over time (Eaton et al. 2015; Baldwin and Woodard 2009), managing openness is an important task (Eisenmann et al. 2009) which induces effects concerning innovation dynamics and profits (Parker and Van Alstyne 2008). Opening a platform increases the innovation rate (Boudreau 2010) and more open platforms with more developers lead to higher profits (Parker et al. 2017). The elements, effects and measurement approaches of platform openness provide a framework to capture application integration (Ondrus et al. 2015; Benlian et al. 2015).

2.3 Boundary Resources as Integration Mechanism

For *boundary resources* as the primary element for technical integration, literature is in agreement regarding their importance (Eaton et al. 2015). Ghazawneh and Henfridsson (2013) define boundary resources as "the software tools and regulations that serve as the interface for the arm's-length relationship between the platform owner and the application developer". Platform owners use boundary resources as control mechanism. In the area of conflict between stimulating external contributions and maintaining control over the platform, precise decisions are required regarding the design of boundary resources and related governance mechanisms (Eaton et al. 2015; Ghazawneh and Henfridsson 2013). Ghazawneh and Henfridsson (2013) specify that boundary resources typically consist of a software development kit (SDK), e.g., iOS SDK, and a multitude of related APIs. For the *developer*, both quality and functionality of boundary resources are relevant selection criteria (Hsieh and Hsieh 2013). Designing boundary resources according to developers' needs is important (Kim et al. 2016), since efficiency in application realization is a main driver for them to participate in digital platforms (Hyrynsalmi et al. 2014).

Platform integration is distinct from bilateral integration efforts in the IT systems context. As platform resources are highly standardized, owners offer possibilities for integration through boundary resources. However, conducting integration is subject to the developer during application development. Therefore, third-party developers choose the degree of platform integration themselves, which we define as *the extent to which boundary resources are employed to utilize platform resources*. Platform integration is understood as *the use of boundary resources during application development*.

3 Hypotheses Development

We focus on the level of systems integration for digital platforms (Stavridou 1999). Thereby, integration is not considered to be a system property, but a relationship of system elements, which seems appropriate since applications are contributed by a diverse developer community (Boudreau 2012; Thomas and Nejmeh 1992).

The employment of platform resources to realize application functionality is considered as platform integration. Developers use interfaces as an integration mechanism to access platform functionality. The platform core subsumes basic functionality useful for applications realization regardless of their domain. The incremental evolution ensures them to be well-tested and demonstrate a good performance while being easy to use and robust with regard to changes (Ghazawneh and Henfridsson 2013; Eaton et al. 2015).

Employing platform core functionality is assumed to enhance *effectiveness*, since the developer is able to focus on core competencies, which is application realization to provide new functions. Developers typically contribute specialized functionalities that are not provided by the platform itself (Haile and Altmann 2016; Bosch and Bosch-Sijtsema 2010). Compared to developers who do not employ the provided resources, less effort is required of the first to realize a similar functionality (Harter et al. 2000; Bosch and Bosch-Sijtsema 2010). Conversely, assuming limited development time and budget, more time can be used to develop specialized functionality. Devoting more time is assumed to result in better functionality, which leads to superior success. Software development studies have proved that the devotion of additional time results in better software quality (Harter et al. 2000). Focusing on the contribution of specialized functionality is not a one-time aspect. Instead, this becomes increasingly relevant with the many updates typical for digital platforms, that impact functionality and competition between developers (Kapoor and Agarwal 2017). Since platform resources are maintained by the owner, developers can rely on their availability and focus on providing specialized functions. Developers thereby enhance effectiveness throughout the application lifecycle. Correspondingly, studies on product platforms have found integration to be associated with performance enhancements (Droge et al. 2004).

Concerning *efficiency*, employing platform resources allows developers to realize specialized functionality with less input required (Bosch and Bosch-Sijtsema 2010). Here two aspects are relevant. First, realizing necessary elements
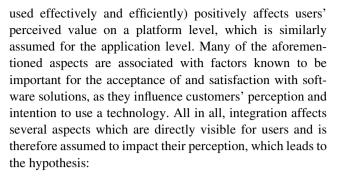
while requiring fewer resources makes it possible to devote resources to developing specialized functionality, which results in better functionality and therefore differentiates from competitors (Bosch and Bosch-Sijtsema 2010). Second, faster realization results in reduced time to market (Droge et al. 2004), which is especially relevant in hypercompetitive markets such as application marketplaces (Sangaralingam et al. 2012). Releasing applications faster allows for a better starting position compared to competitors, which allows to gain a substantial market share, especially if competition lags behind. In light of the dynamics associated with two-sided markets such as network effects (Eisenmann 2008), this creates a competitive advantage which is accelerated by the dynamics themselves and hard to catch up for others.

As platform integration allows for a more effective and efficient realization of applications, it makes it possible to gain competitive advantage. Consequently, more integrated applications are assumed to be more successful, which leads to the hypothesis:

Hypothesis A More integrated applications on digital platforms are more successful than less integrated ones.

In information system literature, integration is an antecedent of system quality which impacts user satisfaction (Wixom and Todd 2005). Even though typically considered on the system level, integration is likely to be relevant for system components as well and consequently impact customer satisfaction.

Integrated applications and functions have better availability and are therefore easier to access. Since platforms offer various access options for third-party functionality, greater accessibility is assumed to be favored by users. As accessibility influences the ease of use, which is relevant for customers' attitude towards system usage (Wixom and Todd 2005), a positive impact on satisfaction is assumed. Integration from a user-interface (UI) and design perspective is another important criterion. Platforms implement typical UI elements that are used across different platform contexts (Wasserman 2010). Through continued use, users become familiar with these platform-specific characteristics and are more likely to use them, as these design concepts are familiar to them (Andre and Wickens 1995; Morris et al. 2010). For users, integration thus reduces learning effort, which allows them to be productive faster which is assumed to be perceived positively (Nikou et al. 2014). Similarly, integration can reduce the setup effort, since integrated applications can access central settings, data and accounts, which is likely to be perceived positively (Park and Koo 2016). Considering central settings in individual applications allows to adapt solutions to the individual user's needs, without the necessity to specify them individually. Haile and Altmann (2016) found system usability (the extent to which a system can be used effectively and efficiently) positively affects users' perceived value on a platform level, which is similarly assumed for the application level. Many of the aforementioned aspects are associated with factors known to be important for the acceptance of and satisfaction with software solutions, as they influence customers' perception and intention to use a technology. All in all, integration affects several aspects which are directly visible for users and is therefore assumed to impact their perception, which leads to the hypothesis:

Hypothesis B Users perceive more integrated applications on digital platforms more positively than less integrated ones.

4 Model Development

Given the situation of a missing model for platform integration and for the use as measurement instrument, an integration model has to be conceptualized first. While aspects of integration have been discussed individually, an integrated model incorporating the different aspects in the platform context is missing. The platform context with its highly modular architecture, high level of standardization and uniqueness of boundary resources is different from typical system integration aspects which is why existing integration models lack the complexity of platform integration. Following a three-step process, a platform integration model is developed (see Table 1). *First*, concepts related to the integration on digital platforms are identified by reviewing existing literature (first column). Literature on digital platforms with a focus on, but not restricted to, integration, openness and boundary resources was considered. *Second*, relevant aspects and findings were used to identify additional related aspects relevant for integration. *Third*, identified concepts were grouped according to the integration aspect they refer to, which is subsumed in the model category (fourth column). Table 1 presents the identified concepts, a brief description and relevant findings as well as the associated model category. While options for integration are platform-specific, similarities exist concerning the platform architecture and related integration possibilities. The derived platform integration model constitutes a general version for digital software platforms.

4.1 Aspects of Platform Integration

4.1.1 Device Ecosystem Integration (DEI)

On mobile device platforms, integration can be established on the software and hardware level (Saarikko 2016). Platform access for mobile devices is bound to specific devices

Table 1 Concepts related to platform integration

Platform-related concepts		Description	Findings	Model category
Multi-homing in sub-ecosystems	Idu et al. (2011) and Mijsters et al. (2018)	Multiple sub-ecosystems within one platform/ecosystem (e.g., iPhone, iPad, Apple Watch)	Providing features for different sub-ecosystems provides a larger customer base and requires less effort than provision on another platform	Device Ecosystem Integration
Platform/ Ecosystem complexity	Tilson et al. (2013) and Kapoor and Agarwal (2017)	Platform complexity or variety describes the number of components in a platform (e.g., hardware configurations available for a platform)	The variety and platform complexity are important concerning dynamics and required adoptions during application development	
Platform resources/ Boundary resources	Ghazawneh and Henfridsson (2013) and Eaton et al. (2015)	Boundary resources provide access to the core modules and frameworks of the platform. Basis for application development	The design of boundary resources varies among platforms. They emerge over time and are used to control platform development	Operating System Integration
Platform openness	Eisenmann et al. (2009), Boudreau (2010), Koch and Kerschbaum (2014) and Benlian et al. (2015)	Platform openness can be assessed on various levels. Openness characterizes the degree to which external parties can participate and act on a platform	Platform openness is a prerequisite for integration and utilization of platform features. Openness varies among platforms	
Data integration and Governance	Gawer (2014), Schreieck et al. (2016) and Lee et al. (2018)	Data is a major resource on digital platforms. Data can be provided by the owner as well as complementors	Platforms facilitate the integration of data from complementors, e.g., by providing specialized platform frameworks (e.g., Apple Health, Game Center)	Data Integration
Marketplace as distribution mechanism	Sangaralingam et al. (2012), Lee and Raghu (2014) and Ghazawneh and Henfridsson (2015)	Marketplaces provide a catalogue, facilitate transactions, and offer an institutional infrastructure for application distribution	Marketplaces as a central distribution mechanism are important for contributors as well as customers. Similarities are identified across platforms	Marketplace Integration
Uncertainty reduction	Müller et al. (2011) and Song et al. (2013)	Application presentation on the marketplace allows to reduce uncertainty for potential users	Elaborate presentation is useful for uncertainty reduction, which is important for users during application selection	
Cross-module composability	Tiwana et al. (2010)	Composing modules refers to providing extended functionality without comprising existing functionality	Extended functionality can be provided by integration of different existing applications	Platform Application Integration
Product Integration	Kim et al. (2012)	Various possibilities for integration exist in the context of digital platforms	New services can be provided by integrating different products	
Provision of application updates	Sangaralingam et al. (2012), Lee and Raghu (2014) and Kapoor and Agarwal (2017)	Application updates provide new features, correct bugs, and ensure compatibility with updated platform versions	The frequent provision of updates for complementors' applications is associated with positive performance	Timeliness
Sustainment of platform position	Lee and Raghu (2014) and Kapoor and Agarwal (2017)	Sustaining a prominent position on a platform is important for contributors	Sustaining a position depends on platform and application characteristics	
Developer multi-homing	Hyrynsalmi et al. (2016), Landsman and Stremersch (2011), Evans (2003) and Boudreau (2008)	Multi-homing is the provision of applications for multiple platforms	Overall, multi-homing among developers is rare in the mobile domain. However, it is typical for superstar applications	Cross-Platform Availability
Agent multi-homing	Armstrong (2006), Rochet and Tirole (2003) and Choi (2010)	Multi-homing, i.e., participating on multiple platforms, can occur on both platform sides	Platform dependency and dynamics vary depending on if one or both parties multi-home	

or at least operating systems. While Google opened up the Android platform for other device manufacturers, access to the iOS platform is restricted to Apple devices (Kenney and Pon 2011). From a developer's perspective, iOS is considered to have a lower platform complexity regarding device-specific adaptations than Android (Kapoor and Agarwal 2017). Developers need to face platform complexity by implementing adoptions, which is considered an
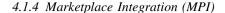
integration. While mobile devices platforms supported only smartphones in the early days, they have been opened for devices such as tablets, watches and the like (Tilson et al. 2013). As a result, device-related platforms typically inhibit sub-ecosystems for which applications are released individually (Idu et al. 2011). Providing applications for multiple sub-ecosystems requires adoptions which are considered as integration, since this allows additional users to access the application's functionality.

4.1.2 Operating System Integration (OSI)

For device-bound platforms, the operating system (OS) is a key component within the ecosystem infrastructure. The operating system provides basic functionalities and access to specialized third-party functionality. Since the OS mediates the interaction with the user, operating system integration is a viable element for enhanced interaction (e.g., speech assistants). Platforms offer multiple ways to access third-party complements, which is why integration can ease access (Wasserman 2010). Aside from direct interaction, the OS and related boundary resources typically offer frameworks with context functionality for developers (Eaton et al. 2015). This includes functions such as authentication (e.g., fingerprint verification) which developers can use for their applications. Developers are free to choose the extent to which they integrate applications with the operating system (Ghazawneh and Henfridsson 2013). Platforms progress over time and are usually released in cycles. Through updates, developers are regularly offered additional options for integration (Ghazawneh and Henfridsson 2013).

4.1.3 Data Integration (DI)

Data integration is of great importance on digital platforms (Schreieck et al. 2016). With its central position within the ecosystem, the platform is ideally suited for data integration (Gawer 2014). While integration of various data sources is one aspect, the integration of applications via shared data is another. A platform can act as central abstraction layer for data integration. Typically, platforms provide storage, retrieval and synchronizing features for data. For storing specialized information, platforms offer domain-related frameworks (e.g., health applications). Regarding data provision, platforms offer data access (e.g., sensor information) through interfaces. Data synchronization allows for data access across devices within the platform. Through the reuse of data and settings, data integration makes it possible to customize solutions to a user's needs.

4.1.4 Marketplace Integration (MPI)

Marketplaces are an important instrument for matching the two sides of the platform, i.e., developers who contribute applications and users who seek applications. Application marketplaces as primary distribution mechanisms play three central roles: they provide a catalogue of applications to match the supply and demand, they facilitate transactions, and offer institutional infrastructure for regulatory and legal aspects (Ghazawneh and Mansour 2015). Marketplaces' exclusiveness varies across platforms. While Apple strictly controls functionality distributed on iOS through the exclusive AppStore, Google is less restrictive and allows for additional marketplaces (Ghazawneh and Henfridsson 2015; Koch and Kerschbaum 2014).

For users, marketplaces facilitate gathering additional functionality. In the search phase, users seek an application that fits their needs (Müller et al. 2011). To support this, marketplaces allow for application presentation, e.g., by categorizing applications, providing a description, or screenshots. Typically, marketplaces also allow users to rate and review applications (Siegfried et al. 2015). Marketplace integration facilitates search process efficiency. A greater integration, i.e., using the features provided, reduces users' search cost (Müller et al. 2011). This is especially relevant if many alternatives are available, as for applications on mobile devices (Qiu et al. 2017; Avinadav et al. 2015). While developers can make choices regarding application presentation, transactions and regulatory aspects are standardized. Marketplace integration allows to reduce uncertainty and to stand out from competition.

4.1.5 Platform Application Integration (PAI)

Integration with other third-party applications on the platform can be established via the platform or through direct connections. Integration via the platform can be achieved using frameworks or data integration. Through direct application integration, also referred to as product integration, new services are created by combining existing ones (Kim et al. 2012). Integration with other applications allows access to resources other applications provide (Tiwana et al. 2010). For instance, social networks offer recommendation functions which are useful for enhancing awareness (Lis and Neßler 2014). Since PAI targets the platform ecosystem rather than the platform directly, it is categorized as ecosystem integration. Application integration allows for enhanced functionality.

4.1.6 Timeliness

Platform owners regularly release updates of their platforms which introduce new features (Dutta et al. 2017).

Integration with digital platforms is understood as a continuous task instead of a one-time action (Tiwana et al. 2010). Complying with platform standards and utilizing new features is important in order to be competitive (Tiwana 2015a). Examples have shown that platform updates shift existing application performance and rankings (Kapoor and Agarwal 2017). To remain competitive, developers need to adjust their application and release new versions. Continuously providing application updates is therefore considered as integration. Application updates are also used to introduce new features. Similar to competition on platform level, the introduction of new features is important for developers on the application level (Ghazawneh and Henfridsson 2013). Updates are important to provide continuous innovation and compatibility.

4.1.7 Cross-Platform Availability (CPA)

While the former categories focus on a single platform, cross-platform availability considers platforms beyond the focal one. Previous studies have found that developer multi-homing, that is, contributing to more than one platform, is not a common phenomenon in the mobile domain, except for top-ranked applications (Hyrynsalmi et al. 2016; Ali et al. 2017). Literature suggests developers not to rely on a single platform from an innovation and risk standpoint (Selander et al. 2013). Platforms change governance over time (Eaton et al. 2015), which poses a risk to third-party developers. Multi-homing makes it possible to reduce platform dependency (Hyrynsalmi et al. 2016). While platforms are well suited to establish integration within their context, developers might aspire to create their own application-specific ecosystem across platform boundaries.

A messaging application, for instance, has limited value if only available on a single platform, e.g., WhatsApp in contrast to iMessage allows for cross-platform messaging (Bender and Gronau 2017). The availability of a corresponding application on another platform is considered as integration, since the application works as a connector to the application ecosystem. Thereby, applications allow to bridge platform boundaries.

Figure 1 depicts the research model for platform integration in the mobile device sector.

5 Analysis

Together with Google's Android, iOS is one of the largest platforms when it comes to contributed third-party applications. From a research perspective, the platform offers various interesting governance and dynamics aspects, which is why iOS was involved in many studies already (Dutta et al. 2017; Eaton et al. 2015; Ghazawneh and Henfridsson 2013). Apple iOS with its single marketplace, AppStore, allows for coherent and exhaustive measurement of success and satisfaction metrics, which is important for this analysis and the reason why iOS was chosen.

While the conceptualized integration model is suitable for different device-bound platforms, it is not yet sufficient to measure integration for a specific platform. Operationalization is necessary to appropriately cover the integration characteristics for the iOS platform and surrounding ecosystem accordingly in order to provide meaningful analysis results.

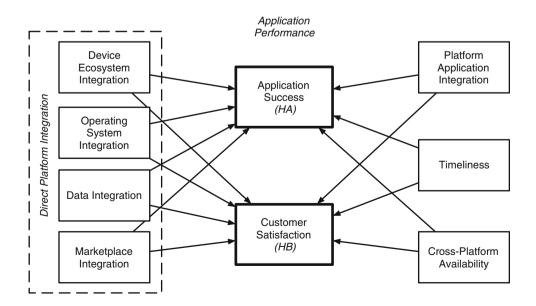**Fig. 1** Platform integration research model

Table 2 Integration model operationalization for the Apple iOS platform

Model category	Measure	Description
Device Ecosystem Integration (DEI)	Apple Watch*	Specifies whether the application runs on the Apple Watch
	CarPlay*	Specifies whether the application provides a specialized in-car frontend
	iPhone	Number of different iPhone models the application runs on
	iPad	Number of different iPad models the application runs on
	iPod	Number of different iPod models the application runs on
Operating system integration (OSI)*	Siri	Specifies whether application functionality can be accessed through Siri
	Touch ID	Specifies whether the application uses Touch ID
	Face ID	Specifies whether the application uses Face ID
	AirPrint	Specifies whether the application supports printing via AirPrint
	AirDrop	Specifies whether the application supports sharing files via AirDrop
	Widgets	Specifies whether the application provides a Widget for the notification and control center
	iMessage	Specifies whether the application provides an iMessage app
	Spotlight	Specifies whether the application content is searchable via Spotlight
	Push Notifications	Specifies whether the application supports push notifications
Data integration (DI)*	iCloud	Specifies whether the application supports iCloud
	Apple Health	Specifies whether the application integrates Apple Health
	Game Center	Specifies whether the application uses Game Center functionality
	Apple Wallet	Specifies whether the application supports Apple Wallet
	Apple Home	Specifies whether the application supports Apple Home
Marketplace integration (MPI)	Description	Length of the application description in the AppStore
	Release note	Length of the release notes in the AppStore
	Screenshots	Number of screenshots provided in the AppStore
	Company URL	Specifies whether a company URL is provided in the AppStore
	Categories	Number of categories the application is assigned to
	Market-adoptions	Number of country-specific applications adoptions available in the AppStore
Platform Application Integration (PAI)*	Facebook	Specifies whether the application supports interaction with the Facebook application
	Twitter	Specifies whether the application supports interaction with the Twitter application
	Instagram	Specifies whether the application supports interaction with the Instagram application
	YouTube	Specifies whether the application supports interaction with the YouTube application
	Dropbox	Specifies whether the application supports interaction with the Dropbox application
	ownCloud	Specifies whether the application supports interaction with the ownCloud application
	WhatsApp	Specifies whether the application supports interaction with the WhatsApp application
Timeliness	Updates	Number of application updates within six-month period on a weekly basis
Cross-platform availability (CPA)	Pendant	Specifies whether a corresponding application is available in the Google Play Store

*Indicators are deducted from other measures or identified based on textual descriptions

6 Model Operationalization

Model operationalization was conducted in a two-step procedure. First, the possibilities to measure platform integration were assessed. Preferably, application integration would be measured on the source code as ground truth. Since neither the proprietary platform nor the application code were available, approximations had to be used. Second, to measure integration of each application individually, a set of approximating indicators is needed. Indicators are identified based on the model categories of the generalized platform integration model. For each model category, based on the related concepts, appropriate measures were identified for the iOS case. Table 2 describes selected indicators for each model aspect.

6.1 Model Categories

Device Ecosystem Integration (DEI) consists of five indicators to describe the different *sub-ecosystems* which an

application is available for. In addition to iOS-based devices (iPhone, iPad, iPod), compatibility with the Apple Watch (watchOS) and Car Play is considered. *Platform complexity* is covered through the number of devices an application supports for iOS-based devices.

Operating System Integration (OSI) encompasses nine indicators to describe operating system integration. The category covers aspects that indicate the use of *platform frameworks/boundary resources* for authentication (Touch ID, Face ID), interaction (Siri, Widgets, Notifications, Spotlight), communication (iMessage) and data transmission (AirPrint, AirDrop).

Data Integration (DI) subsumes five indicators to characterize data integration. The iCloud framework allows for the storage of data for backup purposes, as well as for the synchronization of data across devices. The other indicators are *domain-related frameworks* for specialized purposes. Apple Health allows users to store and retrieve fitness-related information. The Game Center similarly combines functionality for games. Apple Wallet makes it possible to centrally store tickets, cards, and vouchers. Apple Home allows for managing smart home equipment.

Marketplace Integration (MPI) covers six indicators to describe AppStore integration. Marketplace integration is mandatory on iOS for *application distribution*. To describe the extent of *uncertainty reduction* the elaborateness of application presentation is used. Description and release notes were considered in their length. Assuming a more extensive description allows for a better comparison of different applications and their functionality. Providing screenshots or a website assists user's upfront assessment. Assigning applications to multiple categories facilitates their discovery. Market-specific adoptions, measured as the availability in country-specific application stores, also characterize marketplace integration.

Platform Application Integration (PAI) requires *cross-module composability* to realize *product integration*. Product integration is characterized by seven popular examples which were identified by analysing all available AppStore applications. Applications that were integrated with most often were chosen as examples to cover the aspect of composability as a requirement for integration. These include social networks (Facebook, Instagram), a blogging service (Twitter), a video sharing platform (YouTube), a messaging service (WhatsApp) and storage services (Dropbox, ownCloud).

Timeliness is measured by a single indicator which describes the *provision of updates*. Even though release number conventions exist, version numbers are not sufficiently standardized to be employed as a reliable indicator. Therefore, the number of updates throughout a half-year timeframe is considered.

To assess *Cross-Platform Availability (CPA)*, a similar platform is needed. Given the market structure for mobile device platforms, the Google PlayStore is used. Since the two platforms (iOS and Android) combined account for a market share of more than 95%, other platforms cannot be considered comparable (Dutta et al. 2017). Furthermore, the two are often considered to be direct competitors for mobile devices platforms (Tilson et al. 2012). CPA measures the *developer multi-homing* as binary variable indicating whether a corresponding application is available on the PlayStore.

6.2 Control Variables

Different *control variables* are considered to account for alternative explanations in the variation of the success and satisfaction measures. The *application category* is considered since the importance of integration aspects varies across domains. While the Game Center is highly relevant within the game's category, the overall influence for success is rarely significant. The applications belong to 23 different categories. Application categories have already been considered in prior studies (Sangaralingam et al. 2012). The *primary device category* controls for the device type (phone or tablet) an application was originally designed for. Studies found variations in dynamics between *free and paid* applications, e.g., the time for which an application remains successful (Lee and Raghu 2014; Koch and Guceri-Ucar 2017). Other studies found the combination of free and paid applications to be a strategic element (Baird et al. 2016; Voigt and Hinz 2016). The price variable accounts for this. Similar to earlier studies, we control for the *age* of an application, in days from the initial release, as applications need time to achieve a reasonable spread, reputation and position within the platform and marketplace (Comino et al. 2019; Zhou et al. 2018).

6.3 Application Success Measurement

Assessing *success* of applications on digital platforms is complex. Various factors could be considered, e.g., installed base, revenue, or time of usage. While revenue might be a useful measure for paid applications, it is not suited for the large proportion of free applications. The optimal measurement depends on the goals the individual developer aims to attain. For this study, a coherent measure is required to ensure comparability.

Application marketplaces often provide *application rankings*. Ranks are assigned for each 'top' (at least well-performing) application individually. The application rank is acknowledged to be important in application ecosystems and furthermore an established measure of success (Lee and Raghu 2014; Kapoor and Agarwal 2017).

Marketplaces actively promote the top-performing applications, which is an advantage for those in the hyper-competitive environment (Sangaralingam et al. 2012).

Depending on the application marketplace, rakings are provided as either category-specific or overall. Category-specific ranks, as provided by the AppStore, are highly appreciated, since the impact of integration factors is assumed to vary across application domains. Furthermore category-specific rankings are well-suited for comparisons within each segment and are simultaneously comparable on the overall level. While the AppStore allows for assigning an application to multiple categories, we focus on the ranking within the primary application category, since the position is assumed to best characterize the application success relative to competition.

While the exact calculation of AppStore ranks is not public (Lee and Raghu 2014; Garg and Telang 2013), studies have identified factors that impact the ranking, e.g., the number of downloads (Bergvall-Kåreborn and Howcroft 2011). The importance of application rankings is further supported by their impact on customer decisions (Ursu 2018).

For this study, the primary application rank is employed to measure application success. Within each category, applications are ranked from 1 to 1000.

6.4 Customer Satisfaction Measurement

Various aspects influence customer perception. For this study, a coherent measure is required to ensure comparability across applications. Typically, marketplaces encompass a review and rating functionality for customers to state their application experience (Ghazawneh and Henfridsson 2015). Application ratings in the PlayStore and AppStore are realized through a five-point star schema. Additionally, users can review applications by giving textual feedback. While reviews are useful for addressing specific points, they are not as an overall measurement. Sentiment analysis can be employed to gain insights but does not provide a coherent metric measurement as required (Qiu et al. 2017). Additionally, more ratings are given than reviews are written, which makes the former more reliable to reflect user opinions. Previous studies supported the ratings' relevance by highlighting their impact on buying decisions in e-commerce (Lin 2014; Bao and Chang 2014) and mobile device platforms (Siegfried et al. 2015).

To express customer satisfaction, the aggregated application rating is used. Through the combination of positive and negative ratings, a well-balanced average assessment is achieved. The star-based ratings from 1 to 5 are rounded to half-point values, resulting in nine different ratings. Two different ratings are provided on the AppStore: the rating of the current application version and the average of all ratings. Since both, applications and their integration, develop over time, we consider the current version's rating as an appropriate time-dependent measurement for satisfaction (Bergvall-Kåreborn and Howcroft 2011).

7 Results

7.1 Descriptive Analytics

Data was collected for applications from the Apple AppStore for iOS devices. Since the Apple US Store had the most applications, we chose this for our analysis. Data from the Apple iTunes Store was obtained for a six-month period from October 2017 to March 2018 using iTunes Affiliate Resources.

While integration information was available for all applications, data regarding their success and satisfaction were limited. We gathered ratings for the same applications that were ranked to ensure comparability.

For the analysis, data from 82,876 applications were used. Each application is assigned to a primary category out of the 23 categories. Separate rankings are maintained for free and paid applications, as well as for smartphones and tablets. Ranks are attributed from 1 to 1000, adding up to a theoretical maximum of 92,000 (23 categories × 2 price categories × 2 device categories × 1000 ranks) ranked applications. Since not all ranks in each category are assigned, the number of actual observations is lower. With the exception of the newspaper & magazine category, the categories contain between 2910 and 3938 observations (separately for price and device category), with a mean of 3063 per category, which is why the data is considered to be relatively equally distributed among categories.

The application *ranks* are almost equally distributed among the 1000 possible ranks. A very slight tendency towards a decreasing number is found for higher ranks. This is attributed to the aspect discussed before, namely that later ranks are not assigned in all categories. With a mean of 492, an equal distribution can be assumed.

Since a minimum number of *ratings* is required before the average rating for the current version is calculated, not all applications are assigned a rating. For 61,788 observations in the dataset, average ratings could be retrieved. The distribution of the ratings is right-modal with an average of 3.68 and a median of 4 stars. Most applications are assigned a 4.5-star rating, followed by a five-star ranking.

Rankings are listed separately for *free and paid* applications. The dataset contains 43,514 free (53%) and 39,362 (47%) paid applications. Overall, more free than paid applications are available, which represents the distribution. With regard to the primary *device category*, the phone

category with 41,978 (51%) contains slightly more applications than the 40,898 (49%) tablet category.

7.2 Regression Metrics

Although being explanatory models, the regression models are not intended for prediction but only for testing the hypothesized relationship. To test hypothesis A, application success measured by the category-specific application rank is used as dependent variable. For hypothesis B, application satisfaction measured by the average rating of the application's current version is used. Indicators of the operationalized integration model are used as independent variables. Linear regression (OLS) was used to test the influence of integration.

Table 3 shows the regression model summary for the success measure on the left and the satisfaction measure on the right. Both models were computed using the lm-function from the stats package with R 3.6.1. To test for multicollinearity, we employed the variance-inflation factor (VIF) for model indicators. The average VIF of the model variables is 1.3. Two indicators showed a value above three, with a maximum of 3.29. The aspect of multicollinearity does not violate the commonly applied threshold of 10 for VIF values (O'brien 2007). The correlation matrix for model indicators is shown in the Appendix (available online via http://link.springer.com). During model analysis, no relation between fitted values and residuals was found. The model was evaluated for heteroscedasticity, which revealed a non-equal distribution of residuals, which is why robust standard errors according to White (1980) were used and reported in Table 3 (Long and Ervin 2000).

7.3 Integration Success Model

Regression model A contains 23 indicators that are significant at least at the 0.05-level. Highly significant variables are found in any category. In ranking logic, lower values are preferred over higher ones. Regarding coefficients, negative coefficients support hypothesis A.

The hypothesis can be split in overall significance and assumed hypothesis direction. The overall significance clearly shows that integration impacts application success. Concerning the positive impact, 20 out of the 23 significant model indicator have a negative coefficient. Since higher indicators values represent more integration, negative coefficients decrease the rank number, which in turn is positive in the logic of rankings. Therefore, 87% of the significant indicators support hypothesis A, which is therefore considered to be confirmed.

The regression model explains a variance of 8.6% (adjusted R^2) for application rank. Thereby, the goal is not to explain a high variance of success, but rather to investigate the influence of applications' integration on their success. Considering that integration is only one of many aspects that influence success, the explained variance can be considered good, especially since important aspects such as functionality are neglected (Nikou et al. 2014).

7.4 Integration Satisfaction Model

The regression model for satisfaction as well includes 23 indicators that are significant at least at the 5%-level. Each integration aspect contains significant variables. Contrary to the first model, higher values express higher satisfaction (stars logic). Therefore, positive coefficients support hypothesis B.

The overall model is highly significant and thereby confirms the impact of integration on satisfaction. From 23 significant integration indicators, 16 have a positive coefficient. Thereby, 70% support the direction of hypothesis B which is considered to be supported.

The model explains a variance of 11.2% for customer satisfaction. Similar to the previous model, the idea was not to build a model with explanatory power but rather to test for hypothesis B. We employ linear regression to test hypothesis B. Linear regression requires the dependent variable to be metric. When assessing the applicability for our setting, the calculation procedure of ratings has to be considered. When a customer rates an application in the AppStore, one can assign between one and five stars. The customer is thereby given an ordinal set of choices in the form of stars. Once a reasonable number of ratings are assigned, the marketplace calculates an average rating. The aggregated mean rating builds an almost metric measurement. The mean rating, even though not made publicly available, is afterwards rounded to the half-star values. The derived ranking ranges from 1.0 to 5.0 stars in half-point steps, resulting in nine different values which are shown in the marketplace. Given the assumption that ratings are rounded subject to mathematical conventions, the rating values represent the mid-points of the given interval of 0.5. Mid-point regression is a common procedure and similar results have been shown in previous applications (Boukezzoula et al. 2011). Therefore, using it in this case is considered appropriate. Furthermore, typical related issues do not apply since only the influence is tested.

In summary, regression analysis confirms integration to impact application success and customer satisfaction. Moreover, model analysis reveals that greater integration goes along with more success and higher satisfaction and thereby confirms hypotheses A and B.

Table 3 Regression model A (application performance) and model B (customer satisfaction)

Model variables	Model A – Application success				Model B – Customer satisfaction			
	β	SE	t-value	p-value	β	SE	t-value	p-value
Constant	683.8***	13.98	48.902	0.000	2.765***	0.07	39.669	0.000
Device Ecosystem Integration								
DEI_AppleWatch	− 36.308***	5.546	− 6.547	0.000	0.094***	0.024	4.169	0.000
DEI_CarPlay	− 32.136	28.68	− 1.121	0.262	− 0.222	0.149	− 1.491	0.136
DEI_iPhone	− 3.214***	0.457	− 8.865	0.000	0.033***	0.002	13.217	0.000
DEI_iPad	− 8.856*	0.433	− 2.046	0.041	0.005**	0.002	2.643	0.008
DEI_iPod	30.348***	1.786	16.991	0.000	− 0.192***	0.008	− 23.398	0.000
Operating System Integration								
OSI_Siri	− 16.369	15.44	− 1.060	0.289	− 0.015	0.065	− 0.227	0.820
OSI_TouchID	− 39.959***	8.799	− 4.542	0.000	0.039	0.038	1.031	0.302
OSI_FaceID	− 45.635**	15.99	− 2.854	0.004	− 0.031	0.069	− 0.450	0.653
OSI_AirPrint	25.878**	8.905	2.906	0.004	− 0.099*	0.044	− 2.229	0.026
OSI_AirDrop	− 14.377	11.62	− 1.237	0.216	0.129**	0.049	2.620	0.009
OSI_Widgets	13.642	16.10	0.847	0.397	0.02	0.071	0.250	0.802
OSI_iMessage	− 12.669	23.89	− 0.530	0.596	0.158*	0.076	2.070	0.039
OSI_Spotlight	− 9.549	13.80	− 0.692	0.489	0.127*	0.056	2.256	0.024
OSI_Push-Notification	− 8.824**	3.542	− 2.491	0.001	− 0.006	0.017	− 0.388	0.698
Data Integration								
DI_iCloud	− 20.676***	5.319	− 3.887	0.000	0.019***	0.022	8.665	0.000
DI_Apple_Health	− 45.769***	11.23	− 4.049	0.000	0.134***	0.041	3.279	0.001
DI_Game_Center	24.387	12.53	1.946	0.052	0.304***	0.028	10.691	0.000
DI_Apple_Wallet	− 97.678***	25.42	− 4.049	0.000	− 0.726***	0.153	− 4.747	0.000
DI_Apple_Home	132.99*	56.62	1.946	0.019	− 1.342***	0.031	− 4.354	0.000
Marketplace Integration								
MPI_Description	− 0.024***	0.001	− 21.971	0.000	0.0001***	0.00001	19.775	0.000
MPI_Release-Note	− 0.027***	0.004	− 6.497	0.000	0.0002***	0.00002	12.520	0.000
MPI_Screenshots	4.614***	0.561	8.230	0.000	− 0.03***	0.003	− 10.637	0.000
MPI_Company-URL	− 6.579***	2.112	− 3.115	0.001	0.119***	0.011	11.077	0.000
MPI_Categories	− 8.012***	2.098	− 3.819	0.000	0.068***	0.01	6.983	0.000
MPI_Market-Adoptions	− 3.152***	0.152	− 20.797	0.000	0.008***	0.001	12.806	0.000
Platform Application Integration								
PAI_Facebook	3.317	4.086	0.812	0.417	− 0.012	0.018	− 0.701	0.483
PAI_Twitter	− 3.295	4.386	− 0.751	0.452	0.029	0.019	1.504	0.132
PAI_Instragram	− 49.085***	5.635	− 8.710	0.000	0.268***	0.023	11.468	0.000
PAI_YouTube	− 20.088**	6.265	− 3.334	0.001	− 0.124***	0.028	− 4.442	0.000
PAI_Dropbox	− 22.988***	5.572	− 4.126	0.000	0.037	0.024	1.520	0.129
PAI_ownCloud	20.731	44.68	0.464	0.642	0.083	0.159	0.521	0.602
PAI_WhatsApp	0.739	9.191	0.080	0.936	0.004	0.042	0.084	0.932
Timeliness (Updates)	− 18.340***	0.401	− 44.953	0.000	0.05***	0.002	30.042	0.000
Cross-Platform Availability (Pendant)	− 58.434***	2.614	− 22.359	0.000	− 0.06***	0.019	− 5.406	0.000
Control_Variables								
Device_category (Phone)	15.879***	1.980	8.021	0.000	− 0.012	0.001	− 1.268	0.204
Price (Paid)	− 70.308***	2.232	− 31.506	0.000	− 0.10***	0.011	− 9.702	0.000
Application age (Days)	− 0.009***	0.001	− 6.897	0.000	− 0.00001*	0.00001	− 1.996	0.046

Model A: Observations 82,876; R^2 0.087; Adjusted R^2 0.086; F Statistic 142.6*** (0.000)

Model B: Observations 61,788; R^2 0.113; Adjusted R^2 0.112; F Statistic 177.9*** (0.000); *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$

8 Discussion

8.1 Results Discussion

The two regression models use different scales for their dependent variables. While the rank (success) ranges from 1 to 1000, the maximum for ratings (satisfaction) is five. The integration indicators as independent variables were measured consistently to allow for comparability. The satisfaction model explains a slightly higher variance than the success model. Both models are highly significant and support corresponding hypotheses.

The consideration of the aspects in more detail reveals the importance of integration aspects. For *Device Ecosystem Integration*, the same indicators are significant, and their direction matches each hypothesis which confirms further findings in that better availability is positive for applications (Sangaralingam et al. 2012). The iPod indicator is an exception for both models. One explanation is that iPods constitute a relatively small device proportion considering the total ecosystem and the constantly growing proportion of cellular smartphones (iPhones) compared to WiFi-only iPods. Given the intense competition, iPod-specialized applications might not be able to achieve an outstanding result, and thus show a reverse-directed influence. As rankings are maintained separately for iPhones and iPads but not for iPods, this intensifies the competition between iPhone and iPod-specialized apps. For *OSI*, both models have four significant indicators, albeit different ones. The overall positive impact of OSI integration supports findings of studies with a similar focus (Dibia and Wagner 2015). For *Data Integration*, all indicators were significant for satisfaction, while Game Center was not for success. Overall, data integration greatly impacts both measures. This highlights the importance of domain-related frameworks for integration, whereby only integration with mature and widespread solutions has positive impacts, a finding which Apple Home exemplifies by its low support. Regarding *Marketplace Integration,* all indicators are highly significant in both models. The many positive influencing factors highlight the importance of marketplaces as a distribution mechanism and confirm former studies concerning application presentation in marketplaces (Dibia and Wagner 2015; Siegfried et al. 2015). For *Platform Application Integration*, both models involve YouTube and Instagram as significant indicators. While both were positive for success, integration with YouTube negatively impacts satisfaction. While simply measured by common examples, the importance of integration with other applications is supported. *Updates* are highly significant and support the hypotheses for both models, which highlights their importance on platform markets. In line with earlier studies, we confirm the importance of updates

for application success (Zhou et al. 2018; Comino et al. 2019). While *Cross-Platform Availability* positively impacts success, negative effects on satisfaction were found. While providing additional value, several issues are related to the provision across platforms. For instance, previous studies revealed functionality not always to be consistent across platforms (Ali et al. 2017), which might be perceived negatively. Since some platform owners – especially Apple – are known to create lock-in effects (Kenney and Pon 2011), customers might perceive it negatively if applications do not work seamlessly compared to what they are used to within the Apple ecosystem.

Considering significant indicators that point contrary to hypothesized direction reveals interesting results. With the exception of Apple Wallet, YouTube and application pendant, significant indicators with a reversed influence point in the same direction in both models. Overall, 19 of the 23 significant indicators overlap in both models. This proves the importance of similar integration aspects for both success and satisfaction. In conclusion, integration positively impacts application success (model A) as well as customer satisfaction (model B).

8.2 Theoretical Contributions

Regarding *theoretical contributions*, this study contributes to existing literature in various ways. An initial attempt to *theoretically model* and empirically test the concept of *platform integration* is made. The model includes different aspects of integration and provides a reasonable segmentation of integration possibilities in the context of digital platforms. The developed model may serve as a foundation to explore effects related to integration beyond the success metrics employed here.

Additionally, the study operationalizes the generic model for device-bound platforms for the prominent Apple iOS platform and thereby provides a measurement instrument for application integration. The study shows that platform integration is actively employed by application developer. Future studies might provide operationalized models for other platforms to explore related effects on similar platforms.

The study extends prior literature in that the notion of platform participation as a success factor is not comprehensive enough. With platform integration this study explores the relationship between platform and application in more detail and analyses the respective effect on success. While former studies highlighted the positive effects of platform participation (Ceccagnoli et al. 2012), this study extends prior results through the aspects of integration in specific that contribute to success.

The study confirms prior studies with regard to the *importance of boundary resources* (Karhu et al. 2018;

Ghazawneh and Henfridsson 2013). The study extends existing literature by a more detailed model of boundary resources as integration mechanism for mobile device platforms and uncovers effects related to integration. The study reveals platform integration to positively affect application success and customer satisfaction. In doing so the study contributes to the understanding of platform dynamics and modeling application's success. Developers use boundary resources for their applications, which contributes to their success that is especially important in hypercompetitive environments such as mobile device platforms. Simultaneously, using boundary resources creates dependencies. Developers rely on the boundary resources provided by the platform owner for their application to work. Boundary resources as a mechanism allow for control by the platform owner and are known to evolve and change (Eaton et al. 2015).

Therefore, *risks* are associated with integration. Enhanced integration results in a higher dependence on the platform or the owner respectively. Through the provision of specialized resources, digital platforms realize lock-in effects which is why developers are willing to tolerate changes regarding platform resources and governance mechanisms. Earlier studies proved termination costs to be important regarding the continued provision of applications on platforms (Kim et al. 2016), which is intensified with increased levels of integration. The developed integration model allows to explore aspects of lock-in effects in more detail which may provide strategical guidance for developer.

Application integration is also important in terms of multihoming. As applications rely on the platform-specific boundary resources of the focal platform, providing similar functionality on another platform requires more effort than if the functionality had been realized more autonomously with less boundary resources usage (less integration). Therefore, integration itself is part of the effort to switch to another platform (termination cost) (Kim et al. 2016). Closely related to that, integration runs contrary to platform multi-homing. Similar to switching a platform, multi-homing requires an application to run on multiple platforms. To achieve efficiency, developers are well-advised to realize functionality as independently from platform-specific resources as possible, to allow similar procedures to work on various platforms and thereby reduce the cost for multi-homing (Hyrynsalmi et al. 2016).

Additionally, enhanced usage of boundary resources for provision of functionality eases coring endeavors of the platform owner. As integrated applications already rely on platform resources to a great extent, they can be realized more easily by platform owners in a similar manner (Bender et al. 2019). Therefore, coring becomes more likely, which means the integration of third-party functionality into the platform core (Bender and Gronau 2017). Less integrated applications require more effort to be cored. Generally speaking, greater integration ties applications to the focal platform which makes them less competitive on their own. This can be seen critical in light of the power imbalance between contributors and owners.

Given the relative disadvantages of less integrated applications on digital platforms, it remains unclear whether an optimal degree of integration exists, which allows developers to balance advantages and downsides of integration.

The applicability of the implications to other platforms is of great interest. Similar results are expected for similar device-bound platforms (e.g., Google Android) as well as non-device related software platforms (e.g., WordPress). It remains questionable whether the results can be transferred to more specialized platforms such as data platforms. While data integration is only one aspect of platform integration, it is the key aspect for data platforms. Integration in the context of data platforms requires standardization. Once data is integrated and shared with the owner and other stakeholders, the risk to lose competitive advantage arises. Using the example of fitness and health data, we find specialized frameworks such as Apple health. For example, fitness trackers can contribute data to the health data framework. If they do so, users might tend to use the platform application for data analyses instead of their own, which results in fewer engagements with the fitness brand. In order not be reduced to a data provider, companies need to provide functionality of additional value. For instance, fitbit tried to establish an own ecosystem for fitness trackers across platforms (Schreieck et al. 2016) which offers additional value to customers who use devices of multiple platforms. The aspect of cross-platform integration is of great importance. While digital platforms are well-suited for integration within the platform context, they are not beyond the platform scope. Developers might aspire to create an own ecosystem independent of existing platforms. While the power of established platforms as well as the need to be present cannot be neglected by contributors, they aim to offer additional value. Therefore, the establishment of an own ecosystem across platforms can be a strategy to mitigate the risks of platform integration.

8.3 Practical Implications

Concerning *practical implications*, the results reveal that a higher level of integration positively impacts application success and customer satisfaction. First and foremost, this has implications for developers, who are free to choose the degree of integration for their applications. Integration allows them to achieve superior performance when they

integrate their applications well with the platform. In doing so, the different integration aspects are important to reveal the full potential of integration.

Integration is relevant for all participants, i.e., platform owners, developers and customers. Each party should make careful decisions regarding platform integration. The owner, as the entity maintaining and governing the platform, is responsible for providing integration opportunities. Since integration enhances efficiency and effectiveness of application development, integration possibilities are relevant in platform competition and demanded by developers. Owners can use integration as a twofold strategic component. While attracting developers with good integration possibilities, they can also be employed to achieve lock-in effects. Prior research showed that developers with more integrated applications are less likely to leave the platform (Tiwana 2015b), and that termination costs keep developers from leaving a platform (Kim et al. 2016). Since integrated applications rely on platform-specific resources, providing a similar functionality on another platform requires extensive development effort. Developers, therefore, might tolerate more governance-related changes as a result of lock-in effects. Platform owners use integration and related boundary resources to govern third-party development (Ghazawneh and Henfridsson 2010) and to enforce their interests (Tilson et al. 2012).

Third-party developers can use integration as competitive advantage. The results have revealed a positive impact on success and satisfaction. However, required efforts are platform-specific (Hyrynsalmi et al. 2016). Developers should consider both effort and use of integration carefully. Furthermore, contributors should consider multiple platforms for innovation (Selander et al. 2013) and risk mitigation purposes (Hyrynsalmi et al. 2016). However, this results in integration effort for each platform. Platform-specific contributions that are not relevant for other platforms are an exception, therefore integration should be employed to gain related advantages. Developers should consider the strategy for and value of multi-homing before deciding on integration. Developers might not always consider success as most important but focus on other aspects such as customer engagement during development.

For customers, the results suggest that more integrated applications are preferred to less integrated ones. Customers probably consider integration when choosing applications. Depending on the use case, integration itself, as desired functionality, can provide value.

Finally, the study shows integration not to be limited to single platform boundaries. Providing corresponding applications on competing platforms plays a crucial role for applications' functionality and success. Here, the system-of-systems idea applies within the broader mobile device ecosystem.

8.4 Limitations

Regarding the conceptualized integration model, the focus on mobile device platforms limits generalizability. While many aspects also apply to other platform types, some are specific for mobile device platforms. The conceptualized integration model does not claim to be exhaustive. During conceptualization, we focused on the most prominent examples of integration with other applications. However, an application's popularity is not necessarily related to its integration importance. While the model itself provides a reasonable segmentation of integration, the usefulness regarding the comprehensive expression of integration is relatively limited. While not necessary for regression analysis, those aspects support further model refinement. Finally, the model focuses on technical integration of applications, while non-technical integration aspects are out of scope.

For the regression analysis model, constraints apply regarding the model operationalization and the data used. We employed the category-specific application rank as a success measure. Even though it has been acknowledged as a measure for success, the ranking is a self-reinforcing mechanism and the exact calculation procedure is not made transparent. Regarding generalizability, our focus on the top applications, since ranks are only assigned for them, has to be noted.

Some indicators in the operationalized model were derived from other measures or by means of text-analysis procedures. While these approximations seem reasonable for this study, future studies may preferably consider the applications source code as basis, which is not available in the case of iOS.

Even though the integration model provides an approximation and first measurement instrument for application integration in digital platforms, it is not yet suitable to guide development efforts.

8.5 Future Research

Future research to improve the conceptualized integration model can add new aspects and related indicators to describe platform integration. Additionally a more detailed notion of integration, guidance on comprehensively measuring integration and indices formulation is desirable. Regarding the analysis, alternative success measures could be employed to overcome current shortcomings.

Comparing results with competing platforms, e.g., Android, would provide input with respect to similarities and differences. Even though similar results are expected, the role of integration might have more impact in more restrictive ecosystems such as iOS compared to less restrictive ones. Incorporating other platform types makes

it possible to evaluate and compare integration importance. Open-source platforms are favorable, since source code analysis allows for an in-depth analysis of integration and related implications. Comparing results for specific application domains may reveal more detailed insights which are useful for developers.

While this study only considers a single time-bound variable (number of updates), employing time-series analysis makes it possible to assess the impact of integration over time. Additionally, this allows to investigate the role of external events, such as major platform updates or the introduction of new integration opportunities.

Focusing on the platform owner's perspective enables insights into providing integration opportunities, as well as strategic considerations regarding integration. Insights from the developer's perspective provide valuable input regarding the design of integration opportunities. Assuming that integration is not a simple yes-or-no decision, a cost–benefit consideration could assist integration decisions.

While integration is here predominately found to be positive, studies could focus on the downsides of integration. In combination with this study, this would shed light on whether there is an optimal degree of integration.

9 Conclusion

Most of today's mobile digital devices are equipped with access to the manufacturer's digital platform by default. Platforms allow users to extend devices' basic operating system functionality through associated marketplaces. Platform owners provide resources that allow for application development by third-parties. During development, platform integration is important to effectively realize application functionality on a platform. While aspects of participation on digital platforms have been studied, research on a comprehensive notion of integration is still missing. Integration refers to the effective employment of platform boundary resources for application realization. Understanding implications of integration is of great importance for developers in order to make deliberate decisions when designing or refining their products.

By proposing a platform integration model based on existing concepts, this study takes a first step towards a better understanding of integration on digital platforms. Concerning dynamics related to integration, hypotheses regarding the impact of integration were developed and tested using data from over 82,000 applications of the Apple AppStore. The results reveal that application success and customer satisfaction are positively influenced by platform integration. To achieve superior results, developers should address multiple aspects of integration, such

as devices, data, the operating system, the marketplace, other applications, as well as provide updates and access to other platforms. Furthermore, the study highlights the importance of integration for all platform participants and their possibilities to employ integration as a strategic instrument.

References

Ali M, Joorabchi ME, Mesbah A (2017) Same app, different app stores: a comparative study. In: Proceedings of the 4th international conference on mobile software engineering and systems. IEEE Press, pp 79–90, Buenos Aires. https://doi.org/10.1109/mobilesoft.2017.3

Andre AD, Wickens CD (1995) When users want what's not best for them. Ergon Des 3(4):10–14. https://doi.org/10.1177/106480469500300403

Antero MC, Bjørn-Andersen N (2013) Why a partner ecosystem results in superior value: a comparative analysis of the business models of two ERP vendors. Inf Res Manag J 26(1):12–24. https://doi.org/10.4018/irmj.2013010102

Armstrong M (2006) Competition in two-sided markets. RAND J Econ 37(3):668–691. https://doi.org/10.1111/j.1756-2171.2006.tb00037.x

Avinadav T, Chernonog T, Perlman Y (2015) Consignment contract for mobile apps between a single retailer and competitive developers with different risk attitudes. Eur J Oper Res 246(3):949–957. https://doi.org/10.1016/j.ejor.2015.05.016

Baird A, Miller CJ, Raghu TS, Sinha RK (2016) Product line extension in consumer software markets in the presence of free alternatives. Inf Syst Res 27(2):282–301. https://doi.org/10.1287/isre.2016.0621

Baldwin CY, Woodard CJ (2009) The architecture of platforms: a unified view. In: Gawer A (ed) Platforms, markets and innovation. Edward Elgar, Worcester, pp 19–44

Banker R, Mitra S, Sambamurthy V (2011) The effects of digital trading platforms on commodity prices in agricultural supply chains. MIS Q 35(3):599–611

Bao T, Chang TLS (2014) Why Amazon uses both the New York Times best seller list and customer reviews: an empirical study of multiplier effects on product sales from multiple earned media. Dec Support Syst 67:1–8. https://doi.org/10.1016/j.dss.2014.07.004

Bender B, Gronau N (2017) Coring on digital platforms – fundamentals and examples from the mobile device sector. In: Proceedings of the international conference on information systems (ICIS), Seoul

Bender B, Thim C, Linke F (2019) Platform coring in the browser domain – an exploratory study. In: Proceedings of the international conference on information systems (ICIS), Munich

Benlian A, Hilkert D, Hess T (2015) How open is this platform? The meaning and measurement of platform openness from the complementors' perspective. J Inf Technol 30(3):209–228. https://doi.org/10.1057/jit.2015.6

Bergvall-Kåreborn B, Howcroft D (2011) Mobile applications development on Apple and Google platforms. Commun Assoc Inf Syst 29(1):565–580

Bosch J, Bosch-Sijtsema P (2010) From integration to composition: on the impact of software product lines, global development and ecosystems. J Syst Softw 83(1):67–76. https://doi.org/10.1016/j.jss.2009.06.051

Boudreau KJ (2008) Too many complementors? Evidence on software developers. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=943088. Accessed 30 Jan 2019. https://doi.org/10.2139/ssrn.943088

Boudreau KJ (2010) Open platform strategies and innovation: granting access vs. devolving control. Manag Sci 56(10):1849–1872. https://doi.org/10.1287/mnsc.1100.1215

Boudreau KJ (2012) Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. Organ Sci 23(5):1409–1427

Boukezzoula R, Galichet S, Bisserier A (2011) A midpoint–radius approach to regression with interval data. Int J Approx Reason 52(9):1257–1271. https://doi.org/10.1016/j.ijar.2011.07.002

Ceccagnoli M, Forman C, Huang P, Wu DJ (2012) Cocreation of value in a platform ecosystem: the case of enterprise software. MIS Q 36(1):263–290

Comino S, Manenti FM, Mariuzzo F (2019) Updates management in mobile applications: iTunes versus Google Play. J Econ Manag Strat 28(3):392–419. https://doi.org/10.1111/jems.12288

Choi JP (2010) Tying in two-sided markets with multi-homing. J Ind Econ 58(3):607–626. https://doi.org/10.1111/j.1467-6451.2010.00426.x

Dibia V, Wagner C (2015) Success within app distribution platforms: the contribution of app diversity and app cohesivity. In: Proceedings of the 48th Hawaii international conference on system sciences, Kauai. https://doi.org/10.1109/hicss.2015.515

Droge C, Jayaram J, Vickery SK (2004) The effects of internal versus external integration practices on time-based performance and overall firm performance. J Oper Manag 22(6):557–573. https://doi.org/10.1016/j.jom.2004.08.001

Dutta A, Puvvala A, Roy R, Seetharaman P (2017) Technology diffusion: shift happens – the case of iOS and Android handsets. Technol Forecast Soc Chang 118:28–43. https://doi.org/10.1016/j.techfore.2017.01.024

Eaton B, Elaluf-Calderwood S, Sorensen C (2015) Distributed tuning of boundary resources: the case of apple's iOS service system. MIS Q 39(1):217–243

Eisenmann TR (2008) Managing proprietary and shared platforms. Calif Manag Rev 50(4):31–53

Eisenmann TR, Parker G, Van Alstyne M (2009) Opening platforms: How, when and why? In: Gawer A (ed) Platforms, markets and innovation. Edward Elgar, Worcester, pp 131–162

Eisenmann TR, Parker G, Van Alstyne M (2011) Platform envelopment. Strateg Manag J 32(12):1270–1285. https://doi.org/10.1002/smj.935

Evans DS (2003) Some empirical aspects of multi-sided platform industries. Rev Netw Econ. https://doi.org/10.2202/1446-9022.1026

Garg R, Telang R (2013) Inferring app demand from publicly available data. MIS Q 37(4):1253–1264. https://doi.org/10.25300/misq/2013/37.4.12

Gawer A (2014) Bridging differing perspectives on technological platforms: toward an integrative framework. Res Policy 43(7):1239–1249

Gawer A, Cusumano MA (2014) Industry platforms and ecosystem innovation. J Prod Innov Manag 31(3):417–433. https://doi.org/10.1111/jpim.12105

Ghazawneh A, Henfridsson O (2010) Governing third-party development through platform boundary resources. In: Proceedings of the international conference on information systems (ICIS), Saint Louis

Ghazawneh A, Henfridsson O (2013) Balancing platform control and external contribution in third-party development: the boundary resources model. Inf Syst J 23(2):173–192. https://doi.org/10.1111/j.1365-2575.2012.00406.x

Ghazawneh A, Henfridsson O (2015) A paradigmatic analysis of digital application marketplaces. J Inf Technol 30(3):198–208. https://doi.org/10.1057/jit.2015.16

Ghazawneh A, Mansour O (2015) Value creation in digital application marketplaces: a developers' perspective. In: Proceedings of the international conference on information systems (ICIS), Fort Worth

Haile N, Altmann J (2016) Structural analysis of value creation in software service platforms. Electron Market 26(2):129–142. https://doi.org/10.1007/s12525-015-0208-8

Harter DE, Krishnan MS, Slaughter SA (2000) Effects of process maturity on quality, cycle time, and effort in software product development. Manag Sci 46(4):451–466. https://doi.org/10.1287/mnsc.46.4.451.12056

Hilkert D, Benlian A, Sarstedt M, Hess T (2011) Perceived software platform openness: the scale and its impact on developer satisfaction. In: Proceedings of the international conference on information systems (ICIS), Shanghai

Hsieh J-K, Hsieh Y-C (2013) Appealing to internet-based freelance developers in smartphone application marketplaces. Int J Inf Manag 33(2):308–317. https://doi.org/10.1016/j.ijinfomgt.2012.11.010

Hyrynsalmi S, Seppänen M, Suominen A (2014) Sources of value in application ecosystems. J Syst Softw 96:61–72. https://doi.org/10.1016/j.jss.2014.05.064

Hyrynsalmi S, Suominen A, Mäntymäki M (2016) The influence of developer multi-homing on competition between software ecosystems. J Syst Softw 111:119–127. https://doi.org/10.1016/j.jss.2015.08.053

Idu A, Zande Tvd, Jansen S (2011) Multi-homing in the Apple ecosystem: why and how developers target multiple Apple app stores. In: Proceedings of the international conference on management of emergent digital ecosystems, San Francisco. https://doi.org/10.1145/2077489.2077511

Kapoor R, Agarwal S (2017) Sustaining superior performance in business ecosystems: evidence from application software developers in the iOS and android smartphone ecosystems. Organ Sci 28(3):531–551. https://doi.org/10.1287/orsc.2017.1122

Karhu K, Gustafsson R, Lyytinen K (2018) Exploiting and defending open digital platforms with boundary resources: android's five platform forks. Inf Syst Res 29(2):479–497. https://doi.org/10.1287/isre.2018.0786

Kenney M, Pon B (2011) Structuring the smartphone industry: is the mobile internet OS platform the key? J Ind Compet Trade 11(3):239–261. https://doi.org/10.1007/s10842-011-0105-6

Kim J, Lee S, Geum Y, Park Y (2012) Patterns of innovation in digital content services: the case of app store applications. Innovation 14(4):540–556. https://doi.org/10.5172/impp.2012.14.4.540

Kim HJ, Kim I, Lee H (2016) Third-party mobile app developers' continued participation in platform-centric ecosystems: an empirical investigation of two different mechanisms. Int J Inf Manag 36(1):44–59. https://doi.org/10.1016/j.ijinfomgt.2015.09.002

Koch S, Guceri-Ucar G (2017) Motivations of application developers: innovation, business model choice, release policy, and success. J Organ Comput Electron Commer 27(3):218–238. https://doi.org/10.1080/10919392.2017.1331673

Koch S, Kerschbaum M (2014) Joining a smartphone ecosystem: application developers' motivations and decision criteria. Inf Softw Technol 56(11):1423–1435. https://doi.org/10.1016/j.infsof.2014.03.010

Landsman V, Stremersch S (2011) Multihoming in two-sided markets: an empirical inquiry in the video game console industry. J Mark 75(6):39–54. https://doi.org/10.1509/jm.09.0199

Lee G, Raghu TS (2014) Determinants of mobile apps' success: evidence from the app store market. J Manag Inf Syst 31(2):133–169. https://doi.org/10.2753/MIS0742-1222310206

Lee SU, Zhu L, Jeffery R (2018) A contingency-based approach to data governance design for platform ecosystems. Paper presented at the Pacific Asia conference on information systems, Yokohama

Lin ZJ (2014) An empirical investigation of user and system recommendations in e-commerce. Dec Support Syst 68:111–124. https://doi.org/10.1016/j.dss.2014.10.003

Lis B, Neßler C (2014) Electronic word of mouth. Bus Inf Syst Eng 6(1):63–65

Long JS, Ervin LH (2000) Using heteroscedasticity consistent standard errors in the linear regression model. Am Stat 54(3):217–224. https://doi.org/10.2307/2685594

Mijsters Y, Mustafa A, Mihai I, Jansen S (2018) On the nature of software sub-ecosystems and their health. Paper presented at the IEEE/ACM 1st International Workshop on Software Health (SoHeal), Gothenburg

Morris MR, Wobbrock JO, Wilson AD (2010) Understanding users' preferences for surface gestures. In: Proceedings of the graphics interface 2010, Ottawa. https://graphicsinterface.org/proceedings/gi2010/gi2010-35/. Accessed 16 Mar 2019

Müller RM, Kijl B, Martens JKJ (2011) A comparison of inter-organizational business models of mobile app stores: there is more than open vs. closed. J Theor Appl Electron Commer Res 6(2):63–76. https://doi.org/10.4067/s0718-18762011000200007

Nikou S, Bouwman H, de Reuver M (2014) A consumer perspective on mobile service platforms: a conjoint analysis approach. Commun Assoc Inf Syst 34(1):82

O'brien RM (2007) A caution regarding rules of thumb for variance inflation factors. Qual Quant 41(5):673–690. https://doi.org/10.1007/s11135-006-9018-6

Ondrus J, Gannamaneni A, Lyytinen K (2015) The impact of openness on the market potential of multi-sided platforms: a case study of mobile payment platforms. J Inf Technol 30(3):260–275. https://doi.org/10.1057/jit.2015.7

Park Y, Koo Y (2016) An empirical analysis of switching cost in the smartphone market in South Korea. Telecommun Policy 40(4):307–318. https://doi.org/10.1016/j.telpol.2016.01.004

Parker G, Van Alstyne M (2008) Managing platform ecosystems. In: Proceedings of the international conference on information systems (ICIS), Paris

Parker G, Van Alstyne M, Jiang XY (2017) Platform ecosystems: how developers invert the firm. MIS Q 41(1):255–266. https://doi.org/10.25300/misq/2017/41.1.13

Qiu YX, Gopal A, Hann IH (2017) Logic pluralism in mobile platform ecosystems: a study of Indie app developers on the iOS app store. Inf Syst Res 28(2):225–249. https://doi.org/10.1287/isre.2016.0664

Rochet JC, Tirole J (2003) Platform competition in two-sided markets. J Eur Econ Assoc 1(4):990–1029. https://doi.org/10.1162/154247603322493212

Saarikko T (2016) Platform provider by accident a case study of digital platform coring. Bus Inf Syst Eng 58(3):177–191. https://doi.org/10.1007/s12599-016-0426-4

Sangaralingam K, Pervin N, Ramasubbu N, Datta A, Dutta K (2012) Takeoff and sustained success of apps in hypercompetitive mobile platform ecosystems: an empirical analysis. In: Proceedings of the international conference on information systems (ICIS), Orlando

Schreieck M, Wiesche M, Krcmar H (2016) Design and governance of platform ecosystems-key concepts and issues for future research. In: Proceedings of the European conference on information systems (ECIS), Istanbul

Selander L, Henfridsson O, Svahn F (2013) Capability search and redeem across digital ecosystems. J Inf Technol 28(3):183–197. https://doi.org/10.1057/jit.2013.14

Siegfried N, Koch OF, Benlian A (2015) Drivers of app installation likelihood – a conjoint analysis of quality signals in mobile ecosystems. In: Proceedings of the international conference on information systems (ICIS), Fort Worth

Song C, Park K, Kim BC (2013) Impact of online reviews on mobile app sales: open versus closed platform comparison. Paper presented at the Pacific Asia conference on information systems (PACIS), Jeju Island

Stavridou V (1999) Integration in software intensive systems. J Syst Softw 48(2):91–104. https://doi.org/10.1016/S0164-1212(99)00049-7

Thomas I, Nejmeh BA (1992) Definitions of tool integration for environments. IEEE Softw 9(2):29–35

Thomas L, Autio E, Gann D (2014) Architectural leverage: putting platforms in context. Acad Manag Perspect 28(2):198–219. https://doi.org/10.5465/amp.2011.0105

Tilson D, Sørensen C, Lyytinen K (2012) Change and control paradoxes in mobile infrastructure innovation: the Android and iOS mobile operating systems cases. In: Proceedings of the Hawaii international conference on system sciences (HICSS), Maui. https://doi.org/10.1109/hicss.2012.149

Tilson D, Sørensen C, Lyytinen K (2013) Platform complexity: lessons from the music industry. In: Proceedings of the Hawaii international conference on system sciences (HICSS), Maui. https://doi.org/10.1109/hicss.2013.449

Tiwana A (2015a) Evolutionary competition in platform ecosystems. Inf Syst Res 26(2):266–281. https://doi.org/10.1287/isre.2015.0573

Tiwana A (2015b) Platform desertion by app developers. J Manag Inf Syst 32(4):40–77. https://doi.org/10.1080/07421222.2015.1138365

Tiwana A, Konsynski B, Bush AA (2010) Platform evolution: coevolution of platform architecture, governance, and environmental dynamics. Inf Syst Res 21(4):675–687. https://doi.org/10.1287/isre.1100.0323

Ursu RM (2018) The power of rankings: quantifying the effect of rankings on online consumer search and purchase decisions. Market Sci 37(4):530–552

Voigt S, Hinz O (2016) Making digital freemium business models a success: predicting customers' lifetime value via initial purchase information. Bus Inf Syst Eng 58(2):107–118. https://doi.org/10.1007/s12599-015-0395-z

Wasserman AI (2010) Software engineering issues for mobile application development. In: Proceedings of the FSE/SDP workshop on Future of software engineering research, Santa Fe. https://doi.org/10.1145/1882362.1882443

White H (1980) A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. Econometrica 48(4):817–838. https://doi.org/10.2307/1912934

Wixom BH, Todd PA (2005) A theoretical integration of user satisfaction and technology acceptance. Inf Syst Res 16(1):85–102

Zhou G, Song P, Wang Q (2018) Survival of the fittest: understanding the effectiveness of update speed in the ecosystem of software platforms. J Organ Comput Electron Commer 28(3):234–251. https://doi.org/10.1080/10919392.2018.1482602