Bus Inf Syst Eng 64(2):223–237 (2022)
https://doi.org/10.1007/s12599-021-00706-y

RESEARCH PAPER

The Impact of Strategic Core-Component Reuse on Product Life Cycles

The Case of Location-Based Games

Cristina Mihale-Wilson · Patrick Felka · Oliver Hinz · Martin Spann

Received: 17 July 2020 / Accepted: 11 May 2021 / Published online: 23 June 2021

Abstract The mobile games business is an ever-increasing sub-sector of the entertainment industry. Due to its high profitability but also high risk and competitive atmosphere, game publishers need to develop strategies that allow them to release new products at a high rate, but without compromising the already short lifespan of the firms' existing games. Successful game publishers must enlarge their user base by continually releasing new and entertaining games, while simultaneously motivating the current user base of existing games to remain active for more extended periods. Since the core-component reuse strategy has proven successful in other software products, this study investigates the advantages and drawbacks of this strategy in mobile games. Drawing on the widely accepted Product Life Cycle concept, the study investigates whether the introduction of a new mobile game built with core-components of an existing mobile game curtails the incumbent's product life cycle. Based on real and granular data on the gaming activity of a popular mobile game, the authors find that by promoting multi-homing (i.e., by smartly interlinking the incumbent and new product with each other so that users start consuming both games in parallel), the core-component reuse strategy can prolong the lifespan of the incumbent game.

Accepted after one revision by Óscar Pastor.

C. Mihale-Wilson (✉) · P. Felka
Chair of Information Systems and Information Management, Goethe University, Theodor-W.-Adorno-Platz 4, 60323 Frankfurt am Main, Germany
e-mail: mihale-wilson@wiwi.uni-frankfurt.de

M. Spann
Institute of Electronic Commerce and Digital Markets, Ludwig-Maximilians-Universität, Munich, Germany

Keywords Mobile games · Location-based games · Core-component reuse · Product life cycle

1 Introduction

The prevalence of smartphones and the mobile Internet changed the entertainment industry fundamentally by contributing to some of its sub-sectors' unexpected rise or fall. The mobile gaming sector, for instance, has benefited from this development and turned into a fast-growing, very profitable, but also "intensively competitive" (Merikivi et al. 2017) business. Once a niche pursuit, playing games have become "one of the most popular leisure activities globally" (Boyle et al. 2012). With players growing more comfortable with in-game purchases (Khalaf 2017), the mobile games business records high revenues, e.g., 61.3 billion US $ in 2018 (SuperData 2019, p. 7), which in turn attract new competitors to the market. However, as competition is growing fiercer, mobile game publishers face an increased risk of their games not generating enough profits before users substitute them with new ones.

From an economic point of view, the development of entertainment products such as mobile games is very costly and time-consuming (Clement et al. 2006; Folmer 2007; Engelstätter and Ward 2018). However, at the same time, entertainment products such as games have a relatively short product life cycle (Calantone et al. 2010; Zhu and Zhang 2010) and suffer from monotonically decreasing popularity over time (Yi et al. 2019). Consequently, publishers who wish to thrive in the highly dynamic market (Yi et al. 2019) are forced to continuously develop and release new games (Engelstätter and Ward 2018) while minimizing their production costs and failure risk.

Traditionally, companies that operate in markets with a fast innovation pace but a simultaneous risk of failure routinely use their existing successful products (Faircloth and Richard 1995; Mihale-Wilson et al. 2021) to develop new ones. More specifically, companies incorporate the functionalities and core-components of successful existing products into new products, reducing their time to market, and their risk of failure substantially. Although this strategy is commonly successful in developing software applications, mobile game publishers can have reservations about it due to technical and economic considerations (Folmer 2007). In general, software and core-component reuse hinge on technical and non-technical concerns (Kim and Stohr 1998).

From a technical point of view, integrating the various existing components can be difficult, and increase the complexity of the game architecture significantly, such that it is more challenging to manage (Folmer 2007). From an economic perspective, however, the core-component reuse might also have adverse side-effects. Because new products (and especially mobile games free of charge) frequently do not only galvanize users' attention away from the products of the competing companies but also from existing products of the same firm, the continuous release of new products can accelerate user attrition. User attrition is not only a central issue for all sorts of digital products but even more so for free games where users' interest in this type of game usually vanishes rapidly (Nieborg 2016, p. 35). Considering the cost of production and the short life span in which mobile games can create revenues, any new products that put a strain on the popularity of the existing games are in effect curtailing the already short product life cycles of these games (Yi et al. 2019).

Although there is extensive research on the product life cycle (PLC) concept (Golder and Tellis 2004) and also on games in general, the PLC of mobile games is not sufficiently understood (Yi et al. 2019). Thus, prior literature has not yet identified a set of components that are promising to reuse. Due to this gap in the literature, it is currently difficult to accurately predict whether the core-component reuse benefits will ultimately exceed its drawbacks or vice versa.

Drawing on the current body of literature and rich observational data on two games that have been developed with the same core-components, in this research we pursue two goals. First, we intend to extend the currently under-explored research of mobile games' product life cycle (Yi et al. 2019). Second, we seek to provide valuable insights that can assist mobile game publishers' strategic decision to either adopt or avoid core-component reuse when developing new products. Thus, in this study, we investigate whether introducing a new location-based mobile game built with core-components of an already successful existing game, curtails or benefits the product life cycle of the existing game.

The remainder of this study is organized as follows: after introducing the theoretical background and research setting of this study, we discuss the similarities and differences between the two focal games. Then, after introducing the data and the methodological approach, we present the estimation results, their implications for practice, and conclude with potential avenues for future research.

2 Theory and Related Work

The current body of literature presents a plethora of studies investigating various aspects of PLC, for different products (e.g., Rink and Swan 1979; Day 1981; Calantone et al. 2010). Similarly, there is a rich body of literature studying various aspects of different games (e.g., video games or multiplayer- online games). Some studies, for instance, focus on determinants of success (e.g., Hsu and Lu 2004; Lin and Bhattacherjee 2010; Park et al. 2014; Merikivi et al. 2017), while others investigate the in-game behavior of players (e.g., Chesney et al. 2009; Hsiao and Chen 2016). Similarly, some studies explore individuals' choices and engagement in games (e.g., Boyle et al. 2012; Wu et al. 2013). Again others discuss new game design and development strategies (e.g., Stacey and Nandhakumar 2009; Nah et al. 2014). However, despite the rich body of literature on PLC and mobile games, research examining the PLC of mobile games are rare. Similarly, although we can currently draw on a plethora of research exploring software reuse from various perspectives, no prior work has investigated the benefits and drawbacks of software reuse through the lens of PLC.

Aiming to extend the current state of the literature on software reuse and PLC for mobile games, in this work we combine various streams of prior literature to investigate whether the benefits of reuse of core-components outweigh their drawbacks in relation to the PLC of existing products.

2.1 Product Life Cycle Theory

At its core, the PLC concept is similar to the life cycle in nature (Day 1981). It consists of four stages (Day 1981; Golder and Tellis 2004), and each stage requires a different managerial and marketing strategy. According to Golder and Tellis (2004), for instance, the time frame from a products' release until takeoff (i.e., the introduction stage) hinge on managerial decisions such as the optimal release date or the suitable initial marketing campaign. In contrast, the second stage (i.e., the growth stage) refers to the period in which product adoption increases until a slowdown of adoption occurs. During this stage, managers must meet the demand for the product, maybe expand their marketing strategy to reach peak demand, without surpassing it, as the third stage (i.e., maturity) sets in. Finally, the fourth stage (i.e., the decline stage) starts once the slowdown of the maturity turns into a steady downturn in revenue and ends with the eventual withdrawal of the product from the market.

Due to its "intuitive appeal" (Golder and Tellis 2004, p. 207), the PLC concept has garnered considerable attention from academia and practice. Hence, extant literature on this topic is rich and presents various streams of research. One stream of research, for instance, focusses on forecasting the PLC turning points and the appropriate strategies that go with each PLC stage (e.g., Day 1981; Kurawarwala and Matsuo 1998; Golder and Tellis 2004; Lukas et al. 2017). Another stream investigates the de role of diffusion and precisely the optimal release date for the success of new products (e.g., Elberse and Eliashberg 2003; Clement et al. 2006; Engelstätter and Ward 2018; Yi et al. 2019). Again another, recently evolving stream of the literature concentrates on technology-based products, with rather short product life cycles (e.g., Calantone et al. 2010; Marchand 2016; Yi et al. 2019).

Compared to products with a longer life span (e.g., of many years, and even decades), products with a short PLC stay only briefly on the market (Nieborg 2016, p. 35). Products with a short PLC typically experience rapid growth, short maturity, and a fast decline (Kurawarwala and Matsuo 1998). Longer PLCs are usually observed for utilitarian products, while short PLC are rather common for hedonic products.

In contrast to utilitarian products (such as cars or appliances), which are instrumental and purchased based on a real need or practical use (Clement et al. 2006), hedonic products (such as music, films, books, games) are experience products consumed primarily for leisure and enjoyment purposes. Hedonic products have different life cycles and require different product management strategies than utilitarian goods (Clement et al. 2006). While the PLC of utilitarian products is often bell-shaped and can last years, the PLC of hedonic products is typically only a few weeks long, with rapidly declining sales (Clement et al. 2006). The study at hand focuses on hedonic products with short PLCs only. In particular, in this study we investigate a paramount example of hedonic products with short PLCs: i.e., games.

2.2 Product Life Cycle of Games

Industry reports suggest that mobile games can have very short PLCs of one week to one month (GameAnalytics 2019). According to Game Analytics (2019) – a player analysis platform – the retention rates for most games at less than 11% after day 7 and under 4% after day 28. Surprisingly, although games have been subject to numerous studies, besides practice-related reports on the life span of games, scholarly research on the PLC of games is very sparse (Yi et al. 2019). One notable exception is the work of Yi and colleagues (2019).

Aiming to understand the diffusion of mobile games at the brand level (rather than category level), Yi et al. (2019) explore how linkages to a mobile messenger, users' satisfaction and dissatisfaction with a game, or the ranking position of a game may affect the growth and decline of games. Therefore, the scholars conceptualize the growth and decline phase of mobile games in terms of *attracting* and *holding power*.

The *attracting power* of a game relates to its ability to defer the attention of users towards itself in the introduction phase of its PLC. In contrast, the *holding power* refers to the ability of a game to maintain the attention of users over time while still appealing to new users (Yi et al. 2019). In their study, Yi and colleagues (2019) conclude that various game characteristics affect the attracting and holding power of mobile games, thereby shaping their PLC evolution. Notably, Yi et al.'s (2019) conceptualization of the growth and decline phases of games demonstrates that their PLC stages are interlinked and identifiable based on user and usage-related metrics. Specifically, the current PLC stage of a game can be identified by investigating user-base attrition rates, the inflow of new users, and by monitoring the in-game activity levels. An increasing influx of new users, for instance, indicates that a game is currently in the growth phase. In contrast, a slowdown in the number of users joining the game suggests that a game has reached its maturity phase. Further, declining activity or daily active users suggest that the respective game is currently in the declining stage.

Attrition in the active user base, in-game activity and the subsequently observed adverse changes in its PLC can be due to various reasons and (mobile) games' characteristics. According to Yi and colleagues, for instance, mobile games can suffer increasingly shorter PLCs due to the fierce competition in the gaming market. Further, with most games free-to-play, consumers' investment in time and money to try the game is minimal. Due to such low resource investments on the consumers' side, mobile games can be downloaded and deleted at any time, thereby putting even more pressure on the PLC of such games (Yi et al. 2019). Similar to Yi et al. (2019), prior literature suggests that the interest of gamers in free-to-play games vanishes fairly rapidly (Liu et al. 2014). Thus, retaining active gamers and gaining new users becomes increasingly tricky (Nieborg 2016), and the survival of game publishers on the market is getting increasingly complicated (Yi et al. 2019).

Related literature on games concurs that to accommodate the highly complex and competitive gaming market conditions, successful game publishers must bring new games to the market, fast and with high overall game quality (Aleem et al. 2016). In this endeavor, software reuse – i.e., incorporating successful existing products' functionalities and core-components into new products – might be a promising approach to issue new games with reduced time to market while minimizing the risk of failure substantially.

2.3 Software Reuse

To date, we can draw on a rich and well-established body of literature on software reuse. In general, software reuse refers to creating new products or software systems based on previous ones (Mohagheghi and Conradi 2007; Jalender et al. 2011) rather than building them from scratch (Krueger 1992; Kim and Stohr 1998). Reuse is not limited to code reuse and can also be applied to each software development and maintenance life cycle phase (Swanson et al. 1991). Since reusable assets include any components considered part of a system's design, such assets can be requirements, architectures, implementations, program code, and data (Suri and Garg 2009).

Extant literature on software design concurs that successful asset reuse enhances productivity (Karimi 1990; Banker and Kauffman 1991), software quality, maintainability, and software development efficiency (Karimi 1990; Kim and Stohr 1998) while simultaneously decreasing costs (Sherif et al. 2006). However, prior work also acknowledges that successful software reuse requires a suitable reuse strategy, a high synergy between projects, and a suitable technology architecture (Bombonatti et al. 2017) with appropriately sized modular components. Suppose two products have a similar or common software architecture which consists of common or very similar elements. In that case, the reuse assets require none or only negligible modifications, which increases the effectiveness of component reuse (Mohagheghi and Conradi 2007).

In general, prior literature distinguishes between three modes of reuse: verbatim reuse, reuse with slight modifications and reuse with extensive alternations (Thomas et al. 1997). Since verbatim reuse requires no changes to the assets reused, it yields the most significant benefits (Thomas et al. 1997). Also, verbatim reuse is particularly suited for reusing core-components – i.e., for assets which due to their complexity and importance, have a longer life cycle than smaller and simpler components.

Besides the three mentioned modes of reuse, prior literature also suggests that assets can be reused in other products by following one of two reuse approaches: opportunistic or strategic software reuse (Fortune and Valerdi 2013). Opportunistic software reuse describes the unplanned reuse of assets, while strategic reuse refers to a disciplined, systematic, and upfront planned reuse of assets (Mohagheghi and Conradi 2007; Fortune and Valerdi 2013). Because opportunistic software reuse is done mostly ad-hoc, it also does not require upfront investments to generate, document, store, and manage reuse pre-engineered assets (Mohagheghi and Conradi 2007). On the downside, opportunistic software reuse can be a challenge with little payoff (Swanson et al. 1991). For instance, for undocumented and unstructured code, the costs incurred by preparing existing code for reuse can exceed any cost savings the reuse strategy could theoretically yield (Karimi 1990). In contrast, due to the disciplined and systematic reuse of assets with modular design and implementation, the strategic reuse approach has a higher likelihood of success (Fortune and Valerdi 2013). In this work, we concentrate on the verbatim and slight modifications strategic (planned) reuse of mobile games components.

2.4 Software Reuse in Games

As with any other software development process involving asset reuse, the underlying rationale behind asset reuse in games development is that developers can attain shorter development cycles and cost reductions (Folmer 2007; Neto et al. 2009). However, unlike other software products, games development combines the software engineering process with the creation and implementation of artistic assets (e.g., animations, characters, and storytelling). Thus, games' development process is also more complex than other traditional software development (Aleem et al. 2016).

To account for the complexity of games, often entail strategically designed layers with "fixed" core-components and variable elements (Neto et al. 2009). Fixed core-components are very complex and suitable for verbatim reuse, whereas the variable elements can be altered as necessary to build a variety of other games. Figure 1 illustrates a simplified view of the main building blocks for games (Anderson et al. 2008). It reveals three main groups of core-components.

The *game engine* represents the backbone of a game (Neto et al. 2009). From a developer perspective, the game engine can be described as a collection of tools that allow the development and compilation of a game for a target platform – e.g., mobile apps (Toftedahl and Engström 2019). The game engine can be self-developed, open-source, or commercially distributed (see Andrade 2015). Two of the most widely employed commercially available game engines are "Unity" and "Unreal". These two are used by almost 39% of the games available on Steam – one

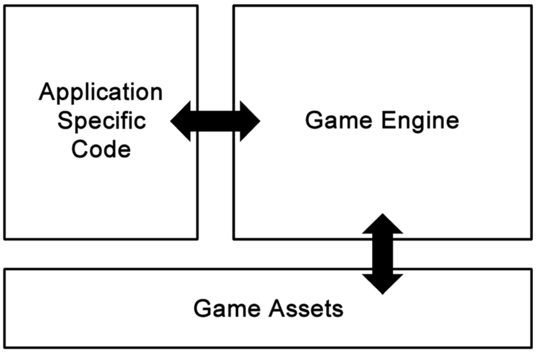


Fig. 1 Simplified perspective on game elements (Anderson et al. 2008; Neto et al. 2009)

of the largest Internet distribution platforms for computer games (Toftedahl and Engström 2019).

The game engine typically controls the game assets (also called game objects). It communicates with the other game elements as it is responsible for rendering (i.e., the generation of graphics), animations, the management of game assets, and many more tasks essential to gameplay. Due to its complexity and importance, the game engine represents a core-component of games. It is typically reusable for the development of various new games.

Game Assets (also called game objects) are another essential component of games. Game objects are usually 3D models and elements that are loaded into the game at runtime (Neto et al. 2009). Game assets interact with the user and the *application-specific code*, which forms the scenario or the environment in which game assets act. Further, application-specific code also contains components with the game's base functionalities, such as the game specific logic, AR functionalities, player inventory, or account management (Neto et al. 2009). The application-specific code also comprises the Graphic User Interface (GUI), which allows users to interact with the game objects. Notably, game assets are typically not suitable for reuse (Neto et al. 2009). In contrast, while some parts of the application-specific code can be reused (e.g., AR functionalities) others cannot (e.g., GUI) be reused to build new products.

2.5 Bringing PLC and Software Reuse in Games Together

As mentioned previously, all game components can be off-shelf assets, open-source, or self-developed. While off-shelf and open source components might be too generic or do not work well with other components, combining various elements from existing products into a new product can also pose multiple technical and non-technical challenges (Kim and Stohr 1998; Haefliger et al. 2008). Assuming that the technical challenges can be mastered, managers still need to address the non-technical code reuse issues. These non-technical issues can relate to the direct cost and benefits of the reuse strategy, or organizational efforts required to improve developers' commitment to reuse (Kim and Stohr 1998; Bombonatti et al. 2017). Furthermore, non-technical issues can also include indirect economic consequences that may result from a swifter development of new products. In this work, we focus on the latter and combine the knowledge of extant literature on PLC and software reuse to investigate whether the introduction of one product built with components of another product curtails or benefits the PLC of an incumbent product.

In general, the economic assessment of the market impact of new products depends on the extent to which the new products either crowd out or complement the consumption of existing ones (Gentzkow 2007). In this vein, products with similar attributes are likely to be viewed by consumers as substitutes (e.g., Deleersnyder et al. 2001; Smith and Telang 2009), while products with differential characteristics are considered to be complements.

In a scenario with two games, where the incumbent is currently in a declining stage, Fig. 2 visualizes the PLC evolution of the incumbent game after introducing a substitute (Fig. 2a) or complementary (Fig. 2b) new game.

Figure 2a shows that if users view the incumbent and new games as substitutes, the chances are high that the launch of the new game will defer users' attention away from the incumbent. Suppose the introduction of the new game motivates the users of the incumbent game to transition to the new game. In this case, the decline in the number of active users and the influx of new users is likely to speed up the decline in the popularity of the incumbents and curtail its PLC.

Analogously, Fig. 2b illustrates potential PLC changes if individuals view two products as complements. In general, if individuals view two products as complements, chances are very high that consumers start to multi-home. In our context, multi-homing relates to individuals who will play both games in parallel (Koukova et al. 2012; Xu et al. 2014). Assuming no significant change in the incumbent's user base activity, multi-homing (i.e., users of the incumbent game adopt and play the new game in addition to playing the incumbent game) should show no significant impact on the PLC of the incumbent. However, considering that the introduction of new games is accompanied by substantial marketing efforts (Aleem et al. 2016; Yi et al. 2019), it is also possible that individuals who became aware of the new game might also want to try out the predecessor – i.e., the incumbent. In that case, if individuals try out and like both games and multi-home, the increase in the new users adopting the incumbent might extend the PLC of the incumbent (see Fig. 2b).

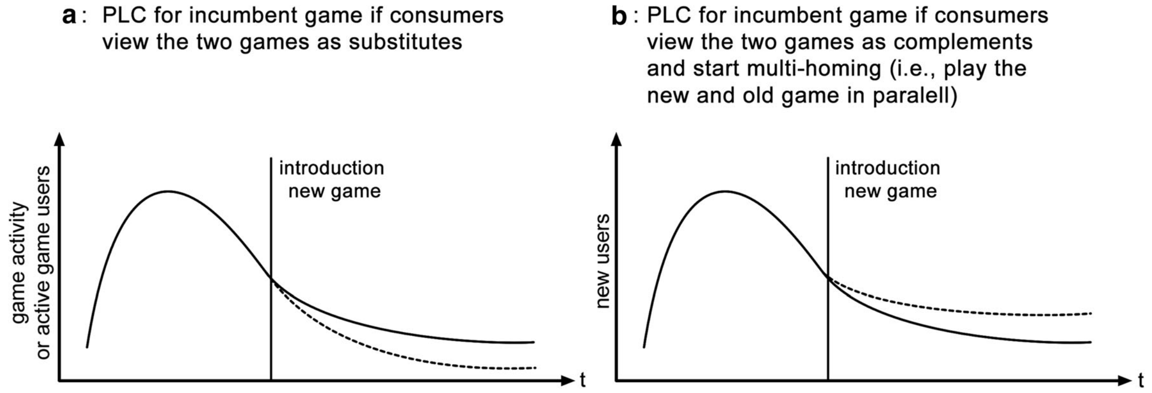


Fig. 2 PLC of incumbent for a substitute or complementary new game

In sum, we note that software reuse can cut both ways: Although software reuse brings several advantages and benefits, it can also negatively affect the PLC of existing products if the incumbents and new products are too similar.

Surprisingly, despite extensive prior research on software reuse, previous literature does not present structured knowledge on which combinations of reuse components yield products dissimilar enough to avoid the contraction of existing products' PLCs. Consequently, the question of whether the benefits outweigh the drawbacks of reusing core-components in mobile games is ultimately an empirical question that we investigate based on the use case of two popular location-based games.

3 Research Setting: Location-Based Games

By definition, location-based games are mobile games that use the position of the players as a core element in the gameplay. In such games, players are required to be mobile and change their current location to advance in the game.

Our study analyses two location-based games developed by the same company (Niantic): Game1 – Ingress and Game2 – Pokémon Go. Game1 was released in 2012. Game2 was launched mid-2016 and reused several of the core-components of Game1.

As mentioned previously, games often entail strategically designed layers with "fixed" core-components and variable elements (Neto et al. 2009). Typically, the three main building blocks for games are: game engine, application-specific code and game assets (Anderson et al. 2008). Table 1 gives an overview which core-components from Game1 were reused in Game2 and to which extent such core-components were adapted for Game2. It shows that Game2 reused various components from Game1: Firstly, both games are using the same game engine. More precisely, Game1 and Game2 use the "Unity" game engine. Thus, both games possess the same "backbone" which can typically be reused in the development of new games with no or very little effort (Neto et al. 2009). Second, Game2 reuses various parts of the application-specific code of Game1. Whereas the GUIs of both games are different, Game1 and Game2 both use a game-engine plugin called Niantic Native Plugin.

The Niantic plugin represents application-specific code developed by Niantic. It encompasses basic functionalities such as account management, the player inventory, AR functionalities and the in-game maps with real-world Game Interaction Points (Niantic 2020).

Real-life gaming locations[1] – or Gaming Interaction Points (GIP) represent special places of interest which can be explored by the users of Game1 and Game2. Many of the in-game artifacts and GIP in Game1 and Game2 are identical and located at the same locations in the real world (e.g., public buildings, monuments) (see Fig. 3).

Additionally, users can suggest new GIP in Game1, which are then mainly added to both games. With this strategy, Niantic enables users to play both games in parallel and multi-home. Once individuals travel to a GIP, they can perform in-game actions in both, Game1 and 2.

The Niantic plugin was used in Game1 and reused in Game2, but also in other of the publisher's mobile games. Examples encompass Harry Potter: Wizards Unite or CATAN – World Explorers (Hu and Wu 2019; Niantic 2020).

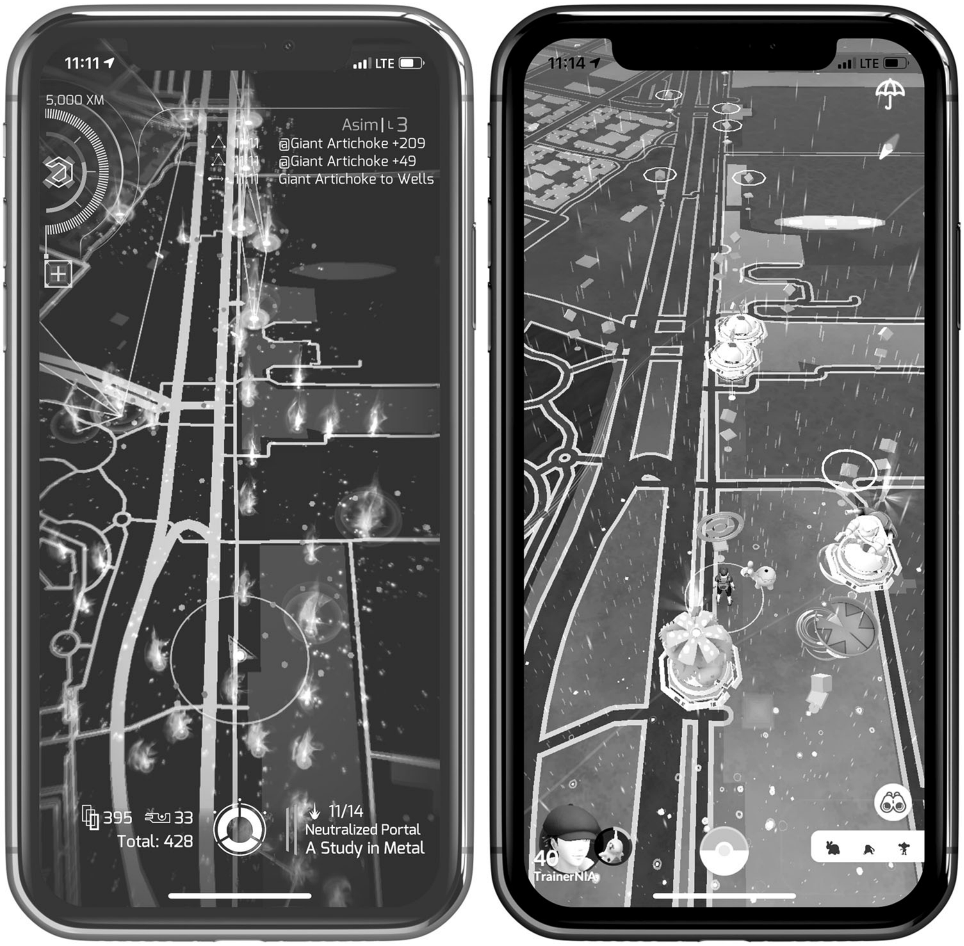
Besides the reuse of the game engine and parts of the application-specific code in Game2, the two games do not show any direct similarities with respect to their game assets such as 3D models and textures. Figure 3 illustrates that the 3D models in Game1 represent portals and weapons whereas the 3D models in Game2 show Pokémon and Pokéstops. Game1 has a science-fiction setting in which players are split in two factions which try to conquer GIPs

[1] Gaming interaction points are called "Portals" in Game1 and "Pokéstops" in Game2.

Table1 Comparison of application specific code, game engine and assets between Ingress and Pokémon Go

Component	Description		Reuse (from Game1 for Game 2)	Ingress (Game1)	Pokémon Go (Game2)
Game engine	Software-development environment		Yes	Unity	Unity
Application Specific Code (Elements of the Niantic Native Plugin)	Map Tile	Map of the area	Yes (modified)	Underlay vector map	Underlay vector map
	Geospatial Objects	Real-world GIP (game interaction points)	Yes (modified)	Portals	Pokéstops, Pokémon arenas
	Player Inventory	Items (limited number of total items per user)	Yes (modified)	Weapons, Resonators, Mods, etc	Pokéballs, Pokémon, etc
	Account MGMT	User status	Yes (modified)	User-level, Points, Badges	User-level, Points, Badges
		User Factions		2 Factions	3 Factions
	AR	Functions to layer location, map and geospatial objects	Yes (modified)	Rendering and stylization of real-word map information, as well as, geospatial objects	
Game assets	3D-Models	3D representation of different objects	No	3D-Models of Portals, Weapons, Shields, etc	3D-Models of Pokéstops, Pokémon arenas, Pokémon, etc

Fig. 3 Screenshots of Ingress Prime (left) and PokemonGo (right) (Hu and Wu 2019)



and occupy big fields for their faction. In contrast, Game2 is all about catching and training Pokémon.

Altogether, Game1 and 2 are particularly suited to empirically investigate whether the benefits of the strategic reuse of core-components outweigh its drawbacks vice versa for mainly two reasons: Firstly, Game2 reuses various of the core-components of Game1 (as shown in Table 1). Second, although the two games are intertwined with each other through the GIPs thereby inviting individuals to multi-home, the two games function independently from each other, on different server infrastructures. Thus, whenever the servers of Game1 are in maintenance or down, the availability of Game2 is not influenced by it. The same holds for any Game2 outages.

4 Empirical Analysis

In general, the classical PLC theory defines the stages in a product's life in terms of unit sales (Rink and Swan 1979) or revenues. However, in the absence of revenue data, and as mentioned previously, especially in the case of games, changes in the PLC stage can also be identified in terms of (1) changes in the currently active number of players or (2) the intensity with which the players engage in the game. In our analysis we rely on these metrics to investigate whether core-component reuse can curtail or benefit the PLC of an existing game.

4.1 Data

We collected the data on Game1 from the official website of ingress: www.ingress.com/intel. This website shows the current state of Game1 and visualizes the current worldwide game actions of other players. To store and collect this data, we developed a Python-based application that can process and store changes in the game state every second. These changes on the map are caused by in-game actions, which also have a precise geographical position. Notably, there is no official website by the game publishers for Game2. Thus, we gathered the data on Game2 from www.pokemonradar.de and www.pokemongomap.info. Both websites display game interaction points of Game2. Altogether, we collected the information on the activity levels in Game1, the number of active users, and new users joining Game1 daily. Additionally, we also collected the information on the special in-game artifacts, i.e., the specific gaming interaction points (GIP) in both games and server outage that occurred for Game2 during our observation period.

The activity levels in Game1 are measured based on highly granular data on in-game actions, including GPS positions and timestamps from 1 Jan 2016 to 31 Dec 2016, within a dedicated geographical region in a Western European country (i.e., the German state of Hesse). Our region of interest is comparable to the size of Vermont, USA and is approximately 25,000 square km. The area of focus entails densely populated urban regions, with many interaction points and less-populated rural spaces with no interaction points to play at all.

The time frame of analysis was chosen strategically, to include the introduction of Game2. Game2 was introduced to the public in mid-2016. By choosing a time frame which includes the launch of Game2, we are able to explore our research question based on the observational data in a quasi-natural experiment and with a differences in differences empirical analysis. Furthermore, it is notable that although the time frame assessed in this study spans the entire year of 2016, due to a five-day maintenance period of our data crawling program, the final data set comprises the information on Game1 activities from 361 days.

Besides data on Game1 and 2, we also collected additional relevant environment variables. For instance, as both games require their players to go outside, weather conditions are expected to impact the activity levels in such games (Felka et al. 2018; Mihale-Wilson et al. 2021). Similarly, public holidays represent additional leisure opportunities, leading to an increase in gaming activity. Additionally, we also gathered information on server outages of Game2 – i.e., the days on which a high number of players in Game2 brought down the servers of the Game2 for several hours. Because the server outages of Game2 represent shocks that impeded users to play Game2, we can use them to test for multi-homing behavior.

4.2 Identification Strategy

To explore whether the launch of Game2 curtails the PLC of Game1, we build on the relationship between game usage intensity – i.e., the number of actions in Game1, and the number of active users in Game1 – and the products' current life cycle status. Sharp drops in Game1 activity after the introduction of Game2 would indicate an acceleration of Game1's decline and a contraction in the PLC of Game1. Analogously, a statistically significant decline in the daily number of active users in Game1 would suggest a contraction of the games' PLC. In contrast, an increase in the number of new users joining Game1, or an increase in the number of active users would indicate a slowdown in the decline stage of Game1.

Striving to identify the relationship between the release of Game2 and the PLC of Game1, we first use Game2's release date to mark the pre- and post-Game2 period. Then, we monitor the changes in Game1 activity and user base after the release of Game2. Additionally, we also supervise the activity and user base levels in Game1 during the server outages in Game2. This way, we can test for the existence of multi-homing behavior[2] – a user behavior encouraged by the fact that Game1 and Game2 share the same game artifacts which are essential for advancing in the respective game.

Besides the mentioned identification strategies, the high granularity of our data can be harnessed to monitor the changes in Game1 activity after the launch of Game2 from three different perspectives (see Fig. 3).

First, the data allow for an *aggregate level analysis* (a) which focuses on the daily activity levels in Game1. This analysis exposes statistically significant changes in the total daily number of actions, active users, or new users in Game1 during the time frame before and after the release of Game2, and for the entire area examined. To perform this analysis, we estimated a model exposing the effects caused by the release of Game2 (Game2 Released), Game2 server outages, and other covariates – whose inclusion we discussed in the Data section of the paper – on the activity and user base levels in Game1 (DV).

Second, the data set allows for a more detailed analysis in which we investigate changes in gaming activity within dedicated "grids" – i.e., smaller-scaled areas. We split our entire focal area into 25,020 equally sized smaller sections called "grids " (see Fig. 4) for this analysis perspective. We later refer to this analysis perspective as *"grid" analysis* (b). Although all grids are the same size, they differ concerning the number of in-game interaction points available for Game1 and Game2, and thus in-game activity levels for both Games. In this regard, it is notable that grids entailing densely populated regions contain many GIPs for both games. Thus, they should exhibit higher levels of game activity than those grids that mainly coincide with uninhabited or rural regions. In this study, we exploit the variances in the activity levels between grids, to detect the influence of Game2 on the activity levels in Game1 in a more detailed manner.

Ultimately, the data allow for a *GIP analysis* (c). This even more granular analysis is meant to expose the changes in each GIP's daily activity levels individually (see crosses and stars in Fig. 4). This analysis perspective incorporates 49,110 of the GIPs in Game1 before and after the launch of Game2. Given that Game2 utilizes the GIP database and infrastructure built in Game1 for its gameplay, each Game1 GIP can have a similar purpose in Game2. In fact, approximately 88% of the analyzed GIP in Game1 also have a purpose in Game2. We exploit this circumstance to monitor (1) the activity levels of the Game1 GIP pre- and post-Game2's release, and (2) monitor differences between GIP that are relevant only in Game1 and those relevant in both games.

Notably, the dependent variables of interest (e.g., number of actions per day, number of active users per day) are stationary[3] but over-dispersed count variables. Hence, following Cameron and Trivedi (2001), we estimate Negative Binomial Models with bootstrap standard errors. Furthermore, to facilitate the interpretation and comparability of the results, coefficients are reported as incidence rates.

Formally, we estimate the following model specification:

$$\mathrm{DV_t} = \beta_0 + \beta_1 \cdot \mathrm{Trend_t} + \beta_1 \cdot \mathrm{Game2\ Released_t} + \beta_2 \cdot \mathrm{Server\ Down_t} + \sum_{n}^{N} \Delta_n \cdot \mathrm{Weather\ Conditions}_{tn} + \sum_{m}^{M} \gamma_m \cdot \mathrm{Time\ Controls_{tm}} + \varepsilon_\mathrm{t}$$

whereDVtotal daily number of actions in Game1; the daily number of active users in Game1; or the daily number of new users joining Game1.Weather cond.temperature, precipitations.Time controlsday of the week, public holidays.

4.3 Estimation Results

Table 2 presents the descriptive statistics for the pre- and post Game2 release period, corresponding to our three analysis perspectives. Section A exhibits that after the launch of Game2, all indicators of interest (e.g., the aggregate number of daily actions in Game1) declined considerably. More specifically, the average daily number of actions performed in Game1 decreased in the post-Game2 period by approximately 23% – i.e., from an average of 56,037 to 43,259 actions per day. Similarly, the daily mean of active users and new users joining Game1 declined in the post-Game2 period from 1,430 to 1,125, and from around 50 to 44, respectively. Results of Mann–Whitney U Tests corroborate that the discussed changes in gaming activity, number of active users, and users joining Game1 after the release of Game2 are all statistically significant ($p > \ = 0.000$).

Besides non-parametric tests of equal means of actions, active users, or new users in Game1 pre- and post-Game2 period, also conducted various estimations (see Table 3). The estimation results of the Models (1) to (3) in Table 3

[2] In the gaming context, gamers' decision to pursue multi-homing relates to them playing more than one game in parallel. Hence, in the case at hand, we expect that players visiting GIP relevant to both games (*i.e.*, real-world GIP that coincide in both games) are likely to play both games rather than one.

[3] Dickey-Fuller-Test statistics corroborate that data is stationary. The T-Statistic for the number of daily actions in Game1 = -6.329, the T-Statistic for the number of active users = -9.532, the T-Statistic for the number of daily new users joining Game1 = -3.451. In contrast, the interpolated Dickey-Fuller critical values at 1% = -3.986, 5% = -3.426 or 10% = -3.13.

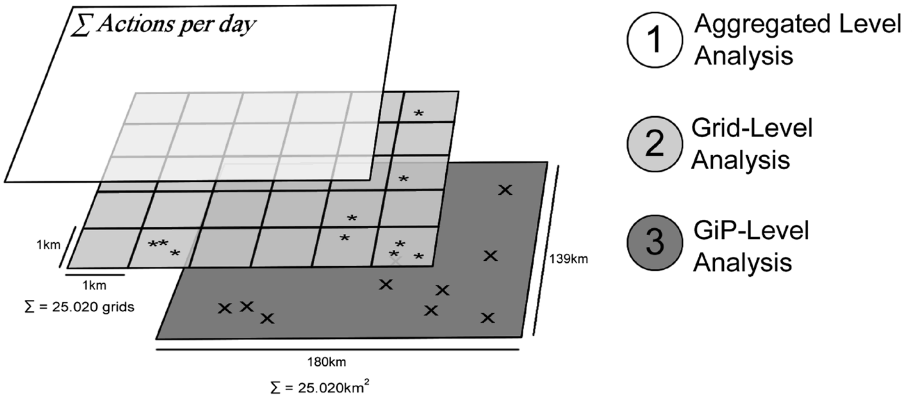


Fig. 4 Overview of the different levels during the analysis

expose the impact of Game2 on the number of actions in Game1 from our aggregate, grid-level, and GIP perspectives. Additionally, Model (4) reveals the impact of Game2 on the active users in Game1, while Model (5) illustrates the effect of Game2 on the number of new users joining Game1.

Altogether, the GIP analysis (Column 3) indicates that before the launch of Game2, Game1 experienced a minimal and statistically significant descending trend in the number of daily in-game actions. However, the estimated results also show that the release of Game2 (operationalized by the binary variable Game2 Released) had negative and positive effects on Game1.

On the one side, the introduction of Game2 had a statistically significant negative impact on the overall activity levels in Game1 – Models (1) to (3); incidence rate < 1. For instance, after the introduction of Game2, Game1 saw a statistically significant decline in total actions per day by 40% – see Table 3 Model (1).

Similarly, the introduction of Game2 decreased the number of actions in each grid, by 22%, while the number of actions pro Game1 GIP decreased after Game2's market release by 23%. Additionally, as the statistical significance and value of the interaction term of trend and Game2 released in Table 3 Model (3) shows, Game2's negative impact on the activity levels in Game1 remains consistent throughout the post-Game2 observation period. Accordingly, we can conclude that Game2's release has persistently undermined the intensity with which Game1 players played the game.

On the other side, our estimations coefficients also reveal that the launch of Game2 also benefited Game1. As Table 3 Models (4) and (5) show, Game1 suffers in general from a weak downward trend in the number of active players and the number of new players joining it. However, both – the number of active players and new players joining Game1 – increase after the release of Game2. More specifically, Game2's introduction relates to an increase in the number of active players in Game1 by 32%, while the number of new players joining Game1 multiplied 16-fold. Unfortunately, the positive effect of Game2 on Game1 is declining at a 2% rate for the number of active users and at a 13% rate in the case of new users joining Game1. Despite the transitional character of Game2's positive effect on Game1, the results presented in this section suggest that Game1 users did not entirely abandon the game after the release of Game2 release but rather decreased their intensity to play the game. At the same time, our estimations show that the introduction of Game2 contributed to expanding the (active) player base in Game1, by probably introducing it to individuals who have not heard of Game1 before.

Furthermore, observing a decrease in the intensity with which Game1 users play the game and a simultaneous increase in the number of active players, our results support the notion that both Game1 and Game2 players started playing both games in parallel (i.e., to pursue multi-homing).

Altogether, the notion that Game2 enticed players to multi-home is supported by various results. Firstly, as the estimates throughout the Models (2) to (5) show, whenever the Game2 servers were down so that users could not play Game2 for hours, the number of active players, new users, and actions in Game1 increased significantly. More specifically, when the Game2 servers were down, the one square km sections experienced, on average, 8.7% more actions per day – see Table 3 Model (2) – while GIP in Game1 saw, on average, 5.1% more activity.

Second, the statistically significant changes in the activity of individual portals corroborate the existence of multi-homing behavior even further. As Table 3, Model (3) shows, after the introduction of Game2, the number of actions per GIP diminished by about 23% (operationalized by the binary variable Game2 released). However, the estimation results also demonstrate that GIPs that serve as in-game interaction points in both games will experience,

Table 2 Activity levels in Game1, for various analysis perspectives

	Mean	SD	Min	Max	Obs
Section A: Aggregated Level Analysis (daily activity levels in the entire focal area)					
Actions Pre-Game2 release	56,037.25	7,119.49	32,849	74,982	194
Actions Post-Game2 release	43,259.62	10,492.71	15,123	76,838	167
Active Users, Pre-Game2 release	1,430.866	150.348	1013	1679	194
Active Users, Post-Game2 release	1,125.24	187.977	503	1733	167
New Users, Pre-Game2 release	50.840	17.152	25	166	194
New Users, Post-Game2 release	44.629	46.264	6	286	167
Section B: Grid Analysis (daily activity levels in Game1 per 1 square km grid)					
Average Actions per grid and day, Pre-Game2 release	7.301	52.102	0	5609	1,488,950
Average Actions per grid and day, Post-Game2 release	5.636	43.835	0	4953	1,281,725
Average number of Game1 GIP per grid	6.398	12.87	1	224	2,770,675
Average number of Game2 GIP per grid	5.614	11.159	0	224	2,770,675
Note: The grid analysis considers only 7675 of our total 25,020 grids, as only 7675 grids have interaction points in Game1. Accordingly, in-game activity for Game1 can be only observed in these 7675 grids					
Section C: GIP (daily activity levels in Game' GIP)					
Average Actions per GIP in Game1, Pre-Game2 release	1.141	3.702	0	1070	9,527,340
Average Actions per GIP in Game1, Post-Game2 release	.881	3.132	0	737	8,201,370

on average, 4%[4] less severe downturn than the GIP applies only to Game1.

Regarding the multi-homing behavior of players, it is also notable that, in principle, it is conceivable that both: Game2 users started to play Game1 and vice versa. However, the 16-fold increase in the number of new Game1 users after the introduction of Game2 strongly indicates that the observed multi-homing is ultimately driven by Game2 users joining Game1. A plausible explanation for why Game2 users are interested in Game1 and start playing it relates to Game 1's strategic decision to allow Game1 users to propose new Game1 portals, which are then also adopted in Game2. Considering that around the release of Game2, numerous articles, and blogs discussed how to use Game1 to populate regions with a weak Game2 in-game locations (GIP) network, it is conceivable that many Game2 users joined Game1 for precisely this reason. Given the discrepancies in urbanization across parts of the area of interest (see Fig. 4), it is possible that not only Game2 users living in more rural regions started to play Game1 with the ultimate goal of extending the number of GIP in their surroundings.

Another explanation for why Game2 users could show interest in Game1 is that Game2 related marketing efforts covered mostly both games, attracting widespread attention not only towards Game2 but also towards Game1—a game consumers might have not have not yet heard before.

5 Conclusion

As one of the fastest-growing, very profitable, but also "intensively competitive" (Merikivi et al. 2017) and risky sub-sector of the entertainment industry, the mobile game sub-sector forces game publishers to develop strategies which allow them to release new products at a high rate (Engelstätter and Ward 2018; Yi et al. 2019), but without compromising the already short product life cycle (PLC) of the brands' existing products. To be more specific, game publishers who wish to survive and thrive in the mobile games business must master a two-fold challenge: Firstly, game publishers must find ways to attract a sufficiently large user base right after the release of new mobile games. Second, publishers need to extend the PLC of existing games by enticing the current user base to remain active and engage with existing games for longer (Yi et al. 2019).

Starting from the PLC concept of mobile games, in this study we examined the core-component reuse strategy's value for the development of new mobile games. More specifically, we investigated whether the introduction of a new mobile game (Game2; in our case Pokémon Go) built by reusing core-components of an existing game (Game1; in our case Ingress), curtails the product life cycle of the incumbent (Game1; in our case Ingress) in a significant way.

Striving to understand the usefulness of core-component reuse strategy, we performed a variety of analyses and tests, with various key performance indicators and from various perspectives. Correspondingly, we investigated whether the release of Game2 relates to significant changes

[4] Computed based on the estimation Coefficient for GIP overlaps in both games (= .818) and the coefficient of Game2 Released (= .778).

Table 3 Estimation results

Variables	(1) Aggregated Analysis: **actions per day in the area of interest**		(2) Grid Analysis: **actions per day in a grid**		(3) GIP Analysis: **actions per day and GIP**		(4) Total number of **active users per day in Game1**		(5) Total number of **new users joining Game1 in a day**	
Trend	.999	(.000)	.999	(.000)	.999*	(.000)	.999***	(.000)	.999	(.000)
Game2 Released^{b}	.601*	(.118)	.781***	(.006)	.778***	(.005)	1.322***	(.130)	16.276***	(7.158)
Trend*Game2 Released^{i}	1.001	(.000)	1.000	(.000)	1.000**	(.000)	.998***	(.000)	.987***	(.001)
Game2 servers down^{b}	1.033	(.041)	1.087***	(.007)	1.052***	(.002)	1.052***	(.020)	1.328***	(.118)
Effect of Game1 GIP in grid	–	–	1.016***	(.005)	–	–	–	–	–	–
Effect of Game2 GIP in grid	–	–	1.010	(.006)	–	–	–	–	–	–
GIP overlaps in both games ^{b}	–	–	–	–	.818***	(.018)	–	–	–	–
GIP overlaps in both games * Game2 Released$^{b\ i}$	–	–	–	–	1.005	(.007)	–	–	–	–
Public Holiday^{b}	787***	(.070)	.825***	(.009)	.816***	(.002)	.784***	(.040)	1.053	(.074)
Temperature	1.022***	(.006)	1.009***	(.000)	1.014***	(.000)	1.003	(.002)	.999	(.007)
Precipitations	.987***	(.002)	.993***	(.000)	.989***	(.000)	.995***	(.000)	.990***	(.003)
Weekday Controls	Yes		Yes		Yes		Yes		Yes	
Constant	51,412.85***	(1895.77)	.103***	(.002)	.192***	(.004)	1,313.52	(19.128)	59.082***	(3.26)
N =	361		2,770,675		17,728,710		361		361	

Coefficients are reported as incidence rates (< 1: negative effect, > 1 positive effect); Bootstrap standard errors in parenthesis

*$p < 0.1$, **$p < 0.05$, ***$p < 0.01$

Variables with "b" in superscript are binary variables (0/1)

Variables with "i" in superscript refer to interaction terms between the variables of interest

in the number of active users or new users joining Game1. Likewise, we also estimated the effect of Game2 on the activity level in Game1 for the entire area of interest, for one square km sections (grids), or in designated real-world in-gaming locations (GIP) tied to Game1 or Game2.

All in all, our estimation results showed that although the introduction of Game2 attenuates the intensity with which the players play Game1, it also benefits Game1 by extending its player base. In this regard, the release of Game2 increased the number of new users joining Game1 and the number of active players in Game1. Furthermore, our estimations indicate that players like to take advantage of multi-homing in those GIP available to perform in-game actions in both games.

Accordingly, in contrast to anticipations that core-component reuse could adversely impact the PLC of existing products, in this study we show that the value of the core-component reuse strategy depends heavily on game publishers' additional strategic choices. In essence, our results indicate that the reuse of the core-components from Game1 to build Game2 rather exacerbates a decline in the number of actions performed in Game1 and, by extension, a decline in users' loyalty and in-game purchase intentions in Game1. Based on this relationship, our results suggest that a pure component usage that only makes two games similar but does not connect them to each other in any way would ultimately curtail the PLC of the existing products. Simultaneously, our results show that the extension of the core-component reuse by interlinking Game2 with Game1 in designated real-world locations (GIPs) can facilitate and encourage multi-homing. Multi-homing behavior can, in turn, lead to an expansion of the existing products' life cycle. In this sense, the results show that going beyond the core-component reuse strategy and interlinking games to each other can extend the PLC of existing games.

These insights have both practical and theoretical value. From a scholarly perspective, with this study we contribute to the existing literature in mainly three ways. First, we extend the current state of the literature to the PLC of mobile games. To date, the PLC concept has been studied in various contexts and for various products. Nonetheless, considering the economic importance of mobile games as an increasingly popular entertainment activity, it is surprising that research on the PLC of mobile games is still scarce (Yi et al. 2019). Second, by assessing the effects of the core-component reuse from a life cycle perspective, we contribute to the current knowledge of products with short PLCs and the determinants shaping it. Finally, we also present first empirical evidence that the core-component reuse strategy can extend the PLC of existing products.

On the practical side, our results present valuable insights that can support game publishers in tackling the challenges of the mobile game industry in an informed way and with suitable strategies. In a broader sense, our estimation results suggest that reusing components of existing products for building new products can be an efficient way to innovate faster and keep customers in the brands' product world and multi-home. Particularly for the location-based games, where the location of players is decisive for gameplay, reusing location-specific gaming artifacts in other games can strongly motivate players to start playing both games. As our estimations show, using core-component reuse to develop new products that are interlinked and thus creating multi-homing opportunities for users can benefit game publishers in more than one way. On the contrary, the pure component reuse, which can lead to the new products being too similar to the old one, can have a detrimental effect on the consumption of the existing products. On this account, as Niantic's use case has shown, combining the core-component reuse with further strategic decisions that encourage multi-homing can pay off.

Despite these practical and scholarly contributions, the study exhibits some limitations that constrain the presented results' generalizability. We acknowledge that we study two specific, albeit popular games. However, we believe that the main results should be generalizable to other games which are free of charge (e.g., video games, PC-based online games) given that we study games developed on a popular game engine ("Unity") with comparable components (i.e., the game engine, application specific code and game objects) applied in many other games. However, the replication of our analyses for different games relying on the same as well as different game engines appears to be a promising avenue for future research. In addition, another promising path for future research would be to investigate the applicability of the insights presented in this study to further entertainment products (e.g., music, movies). Given the wide variety of entertainment products that differ, inter alia, in terms of user engagement,[5] or mobility requirements,[6] future research needs to investigate the effects of the component reuse strategy for other types of entertainment products by accounting for all possible product type differences. Another limitation of this study is that we have no revenue and cost information to derive the profit impact of component reuse in location-based games.

Despite the mentioned limitations, this study presents valuable insights that future research can corroborate and extend by analyzing the effects of core-component reuse based on revenue data from other location-based AR games. After all, the mobile gaming market experiences a constant influx of new location-based games built with core-components from existing ones (e.g., Ingress Prime and "Harry Potter: Wizards Unite").

[5] , Some entertainment products require their user to be "active", some entertainment products can be consumed "leaned back" – *i.e.*, passively.

[6] AR-based games, for instance, require their user to be mobile and roam through the streets while PC Online games or TV usually require their user to be at home.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Aleem S, Capretz LF, Ahmed F (2016) Critical success factors to improve the game development process from a developer's perspective. J Comput Sci Technol 31:925–950

Anderson EF, Engel S, Comninos P, McLoughlin L (2008) The case for research in game engine architecture. In: Proceedings of the 2008 conference on future play: research, play, share. Toronto, pp 228–231

Andrade A (2015) Game engines: a survey. EAI Endorsed Trans Serious Games 2:e8

Banker RD, Kauffman RJ (1991) Reuse and productivity in integrated computer-aided software engineering: an empirical study. MIS Q 15:375–401. https://doi.org/10.2307/249649

Bombonatti D, Goulão M, Moreira A (2017) Synergies and tradeoffs in software reuse – a systematic mapping study. Softw Pract Exp 47:943–957

Boyle EA, Connolly TM, Hainey T, Boyle JM (2012) Engagement in digital entertainment games: a systematic review. Comput Hum Behav 28:771–780. https://doi.org/10.1016/j.chb.2011.11.020

Calantone RJ, Yeniyurt S, Townsend JD, Schmidt JB (2010) The effects of competition in short product life-cycle markets: the case of motion pictures. J Prod Innov Manag 27:349–361. https://doi.org/10.1111/j.1540-5885.2010.00721.x

Cameron AC, Trivedi PK (2001) Essentials of count data regression. In: Baltagi EH (ed) A companion to theoretical econometrics. Wiley, New Jersey, pp 331–348

Chesney T, Coyne I, Logan B, Madden N (2009) Griefing in virtual worlds: causes, casualties and coping strategies. Inf Syst J 19:525–548. https://doi.org/10.1111/j.1365-2575.2009.00330.x

Clement M, Fabel S, Schmidt-Stolting C (2006) Diffusion of hedonic goods: a literature review. Int J Media Manag 8:155–163. https://doi.org/10.1207/s14241250ijmm0804_1

Day GS (1981) The product life cycle: analysis and applications issues. J Market 45:60–67. https://doi.org/10.1177/002224298104500408

Deleersnyder B, Geyskens I, Gielens K, Dekimpe MG (2001) How cannibalistic is the internet channel? Int J Res Market 19:337–348. https://doi.org/10.1016/S0167-8116(02)00099-X

Elberse A, Eliashberg J (2003) Demand and supply dynamics for sequentially released products in international markets: the case of motion pictures. Market Sci 22:329–354. https://doi.org/10.1287/mksc.22.3.329.17740

Engelstätter B, Ward MR (2018) Strategic timing of entry: evidence from video games. J Cult Econ 42:1–22. https://doi.org/10.1007/s10824-016-9276-7

Faircloth JB, Richard MD (1995) A product extension model of consumer choice. J Market Theory Pract 3:99–108. https://doi.org/10.1080/10696679.1995.11501699

Felka P, Sterz A, Keller K, Freisleben B, Hinz O (2018) The context matters: predicting the number of in-game actions using traces of mobile augmented reality games. In: Proceedings of the 17th international conference on mobile and ubiquitous multimedia, Cairo, pp 25–35

Folmer E (2007) Component based game development–a solution to escalating costs and expanding deadlines? International symposium on component-based software engineering. Springer, Berlin, pp 66–73

Fortune J, Valerdi R (2013) A framework for reusing systems engineering products. Syst Eng 16:304–312

GameAnalytics (2019) Mobile gaming benchmarks In: Game analytics. https://pages.gameanalytics.com/rs/686-EPV-320/images/H1-2019-Mobile-Benchmarks-Report-GameAnalytics.pdf. Accessed 9 Mar 2021

Gentzkow M (2007) Valuing new goods in a model with complementarity: online newspapers. Am Econ Rev 97:713–744. https://doi.org/10.1257/aer.97.3.713

Golder PN, Tellis GJ (2004) Growing, growing, gone: cascades, diffusion, and turning points in the product life cycle. Market Sci 23:207–218. https://doi.org/10.1287/mksc.1040.0057

Haefliger S, Von Krogh G, Spaeth S (2008) Code reuse in open source software. Manag Sci 54:180–193. https://doi.org/10.1287/mnsc.1070.0748

Hsiao K-L, Chen C-C (2016) What drives in-app purchase intention for mobile games? An examination of perceived values and loyalty. Electron Commer Res Appl 16:18–29. https://doi.org/10.1016/j.elerap.2016.01.001

Hsu C-L, Lu H-P (2004) Why do people play on-line games? An extended TAM with social influences and flow experience. Inf Manag 41:853–868

Hu D, Wu E (2019) Designing a planet-scale real-world AR platform. In: Niantic. https://nianticlabs.com/en/blog/nrwp-update/. Accessed 9 Mar 2021

Jalender B, Govardhan A, Premchand P (2011) Breaking the boundaries for software component reuse technology. Int J Comput Appl 13:37–41. https://doi.org/10.5120/1782-2458

Karimi J (1990) An asset-based systems development approach to software reusability. MIS Q 14:179–198. https://doi.org/10.2307/248776

Khalaf S (2017) On their tenth anniversary, mobile apps start eating their own. In: Flurry analytics blog. https://flurrymobile.tumblr.com/post/155761509355/on-their-tenth-anniversary-mobile-apps-start. Accessed 17 Apr 2019

Kim Y, Stohr EA (1998) Software reuse: survey and research directions. J Manag Inf Syst 14:113–147. https://doi.org/10.1080/07421222.1998.11518188

Koukova NT, Kannan PK, Kirmani A (2012) Multiformat digital products: how design attributes interact with usage situations to determine choice. J Market Research 49:100–114. https://doi.org/10.1509/jmr.10.0058

Krueger CW (1992) Software reuse. ACM Comput Surv 24:131–183. https://doi.org/10.1145/130844.130856

Kurawarwala AA, Matsuo H (1998) Product growth models for medium-term forecasting of short life cycle products. Technol Forecast Soc Change 57:169–196. https://doi.org/10.1016/S0040-1625(97)00102-9

Lin C-P, Bhattacherjee A (2010) Extending technology usage models to interactive hedonic technologies: a theoretical model and empirical test. Inf Syst J 20:163–181. https://doi.org/10.1111/j.1365-2575.2007.00265.x

Liu CZ, Au YA, Choi HS (2014) Effects of freemium strategy in the mobile app market: an empirical study of google play. J Manag Inf Syst 31:326–354. https://doi.org/10.1080/07421222.2014.995564

Lukas E, Spengler TS, Kupfer S, Kieckhäfer K (2017) When and how much to invest? Investment and capacity choice under product life cycle uncertainty. Eur J Oper Res 260:1105–1114. https://doi.org/10.1016/j.ejor.2017.01.027

Marchand A (2016) The power of an installed base to combat lifecycle decline: the case of video games. Int J Res Market 33:140–154. https://doi.org/10.1016/j.ijresmar.2015.06.006

Merikivi J, Tuunainen V, Nguyen D (2017) What makes continued mobile gaming enjoyable? Comput Hum Behav 68:411–421. https://doi.org/10.1016/j.chb.2016.11.070

Mihale-Wilson C, Felka P, Hinz O, Spann M (2021) The influence of location-based games on traditional entertainment products. Decision support systems, forthcoming

Mohagheghi P, Conradi R (2007) Quality, productivity and economic benefits of software reuse: a review of industrial studies. Emp Softw Eng 12:471–516

Nah FF-H, Eschenbrenner B, Zeng Q et al (2014) Flow in gaming: literature synthesis and framework development. Int J Inf Syst Manag 1:83–124. https://doi.org/10.1504/IJISAM.2014.062288

Neto B, Fernandes L, Werner C, de Souza JM (2009) Reuse in digital game development. In: Proceedings of the 4th International conference on ubiquitous information technologies and applications. IEEE, Fukuoka, pp 1–6

Niantic (2020) New real-world adventures coming to the Niantic platform. In: Niantic. https://nianticlabs.com/en/blog/new-real-world-adventures/. Accessed 9 Mar 2021

Nieborg DB (2016) Free-to-play games and app advertising: the rise of the player commodity. Explorations in critical studies of advertising. Routledge, England, pp 38–51

Park E, Baek S, Ohm J, Chang HJ (2014) Determinants of player acceptance of mobile social network games: an application of extended technology acceptance model. Telematics Inform 31:3–15. https://doi.org/10.1016/j.tele.2013.07.001

Rink DR, Swan JE (1979) Product life cycle research: a literature review. J Bus Res 7:219–242. https://doi.org/10.1016/0148-2963(79)90030-4

Sherif K, Zmud RW, Browne GJ (2006) Managing peer-to-peer conflicts in disruptive information technology innovations: the case of software reuse. MIS Q 30:339–356. https://doi.org/10.2307/25148734

Smith MD, Telang R (2009) Competing with free: the impact of movie broadcasts on DVD sales and Internet piracy. MIS Q 33:321–338. https://doi.org/10.2307/20650294

Stacey P, Nandhakumar J (2009) A temporal perspective of the computer game development process. Inf Syst J 19:479–497. https://doi.org/10.1111/j.1365-2575.2007.00273.x

SuperData (2019) 2018 Digital games & interactive entertainment industry year in review. In: Market brief. https://strivesponsorship.com/wp-content/uploads/2018/02/SuperData-2017-year-in-review-digital-games-and-interactive-media.pdf. Accessed 17 Apr 2019

Suri PK, Garg N (2009) Software reuse metrics: measuring component independence and its applicability in software reuse. Int J Comput Sci Net Secur 9:237–248

Swanson K, McComb D, Smith J, McCubbrey D (1991) The application software factory: applying total quality techniques to systems development. MIS Q 15:567–579. https://doi.org/10.2307/249460

Thomas WM, Delis A, Basili VR (1997) An analysis of errors in a reuse-oriented development environment. J Syst Softw 38:211–224

Toftedahl M, Engström H (2019) A taxonomy of game engines and the tools that drive the industry. In: DiGRA 2019, The 12th digital games research association conference, Kyoto. Digital Games Research Association (DiGRA)

Wu H-K, Lee SW-Y, Chang H-Y, Liang J-C (2013) Current status, opportunities and challenges of augmented reality in education. Comput Educ 62:41–49. https://doi.org/10.1016/j.compedu.2012.10.024

Xu J, Forman C, Kim JB, Van Ittersum K (2014) News media channels: complements or substitutes? Evidence from mobile phone usage. J Market 78:97–112. https://doi.org/10.1509/jm.13.0198

Yi J, Lee Y, Kim S-H (2019) Determinants of growth and decline in mobile game diffusion. J Bus Res 99:363–372. https://doi.org/10.1016/j.jbusres.2017.09.045

Zhu F, Zhang X (2010) Impact of online consumer reviews on sales: the moderating role of product and consumer characteristics. J Market 74:133–148