Platform Coring in the Browser Domain – An Exploratory Study

Completed Research Paper

Benedict Bender
Chair of Business Informatics,
esp. Processes and Systems
University of Potsdam
August-Bebel-Str. 89,
14482 Potsdam, Germany
benedict.bender@wi.uni-potsdam.de

Christof Thim
Chair of Business Informatics,
esp. Processes and Systems
University of Potsdam
August-Bebel-Str. 89,
14482 Potsdam, Germany
christof.thim@wi.uni-potsdam.de

Felix Linke
University of Potsdam
August-Bebel-Str. 89,
14482 Potsdam, Germany
linke.felix@gmx.de

Abstract

Modern browsers are digital software platforms, as they allow third parties to extend functionality by providing extensions. In a highly competitive environment, differentiation through provided functionality is a key factor for browser platforms. As the development of browsers progress, new functions are constantly being released. Browsers could thus enter complementary markets by adding functionality previously provided by third-party extensions, which is referred to as 'platform coring'. Previous studies have missed the perspective of the parties involved. To address this gap, we conducted interviews with third-party and core developers in the security and privacy domain from Firefox and Chrome. This study provides three contributions. First, insights into stakeholder-specific issues concerning coring. Second, measures to prevent coring. Third, strategical guidance for developers and owners. Third-party vendors experienced and core developers confirmed that coring occurs on browser platforms. While developers with extrinsic motivations assess coring negatively, developers with intrinsic motivations perceive coring positively.

Keywords: Platform Coring, Browser Platforms, Platform Innovation, Firefox, Chrome

Introduction

Digital software platforms are able to provide users with a more diverse range of features than a single entity could provide on its own (Eisenmann et al. 2011). Through contributions, third-party developers extend functionality beyond the platform owner´s core functionality (Tiwana et al. 2010). External contributions constitute an important part of the platform's functionality (Allen 2012). Third-party developers contribute functionality in form of packaged code fragments, also referred to as 'extensions' or 'add-ons'. While the platform core usually provides functionality relevant for the general audience, third-party extensions focus on specialized functionality (Olleros 2008). Modern web-browsers can be considered as digital software platforms. The browser (e.g. Google Chrome, Mozilla Firefox) itself offers core functionality that allows users to browse the web, which can be extended by third-party extensions like script blocking (e.g. NoScript), password management (e.g. LastPass), which are offered on the browser's marketplace (e.g. Chrome Web Store, Firefox Marketplace).

Digital platforms are known to progress over time (de Reuver et al. 2017; Eaton et al. 2015). Updates introduce new and extend established features of the platform core. Innovation (i.e. adding features) on the owner's side is two-fold. On one hand, continuous platform innovation is considered a success factor for digital platforms (Kankanhalli et al. 2015; Toppenberg et al. 2016). On the other hand, new features can replace external contributions. The provision of functionality through the platform core, which was previously provided by third-party extensions, is referred to as 'coring'.

Coring has been assessed in different phases of platform lifecycles. While some studies addressed coring during the emergence of platforms in terms of creating a platform core (Gawer and Cusumano 2008; Saarikko 2016), other studies focused on the refinement and development phase of an existing platform (Bender and Gronau 2017; Toppenberg et al. 2016; Um and Yoo 2016). For this contribution we focus on coring during platform evolution. Following Bender and Gronau (2017), coring is specified as the *integration of several functionalities provided by third-party applications into the platform core*. Coring therefore does not address functionality completely new to the platform but refers to a shift in functionality from third-party extensions into the core.

Providing similar functionality through coring is attractive for platform owners as they enter complementary markets. Yet, there may be negative consequences for the further involvement of the developers (Gawer and Cusumano 2014; Kim et al. 2016). Coring may affect the use of extensions and success for third-party developers. Although continued involvement of third-parties as an external source of innovation is of primary importance (Boudreau 2010; Boudreau 2012), there is no understanding of the perception and assessment from the third-party perspective on coring. To ensure continued participation, platform owner must understand developer's *attitude towards coring*. So far, the developer perspective has only been discussed on a theoretical level with negative connotation (Bender and Gronau 2017). Coring can also be perceived positively as sign of innovation since functionality is added to the platform core. Platform owner are in a dilemma concerning coring. The abandonment from coring will leave room for external innovation, yet it will lead to a greater dependence on external innovations regarding to the platform's functionality. Increased coring activity, on the other hand, will increase the development and maintenance effort and may be considered as hostile by third-party developers.

The developer's perspective on the platform owner's coring activities is twofold. First, the developer assesses how *likely their contribution is to be cored* and whether it could still exist in the near future. Secondly, the assessment allows to shape the contributions attractiveness for coring. Depending on the developer's attitude, this evaluation is followed by different activities. If coring is considered a positive strategy, developers can promote their contribution for coring. If it is thought to have a negative impact, they may try to hinder related activities. Contrary to previous assumptions, third-party developers should not be considered as passive platform participants. Depending on the technology, they may have the possibility to take *countermeasures against coring*. However, due to the open modular software architecture and detailed development specifications, known measures from software development cannot be applied in a platform context. We therefore try to shed light on the conditions (likelihood, attitude, technological foundation) under which countermeasures can be implemented. Associated findings provide both platform owner and developer guidance concerning strategic development aspects.

Since the mixture of motivations, attitudes and potential countermeasures in software development for browser platforms has not yet been investigated in detail, we examine the above-mentioned issues regarding platform coring using a grounded theory approach. The rapidly evolving browser domain with different competing browser platforms, high prevalence and industry-wide challenges is well suited to investigate the coring phenomenon. To avoid platform-specific effects, we consider multiple browsers. We conduct semi-structured interviews with third-party developers from Mozilla Firefox and Google Chrome (we use Chrome and Chromium synonymously as the open-source equivalent of the Google product) and extend these findings by interviewing core developers about their thoughts on coring. The study develops an understanding of the perception of and reaction to platform coring. The next section presents literature related to platform coring. Section three describes the method. Section four presents the results, which are discussed in section five. Section six concludes the paper and highlights areas for further research.

Related Literature

This section explains the concepts used in this study and outlines prior research in the field of platform coring. 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 is referred to as a "[…] software subsystem that connects to the platform to add functionality to the platform" (Tiwana et al. 2010).

Usually, platforms coexist and compete with each other (Boudreau 2010; Eisenmann et al. 2011). Digital software platforms as two-sided markets depend on both sides (third-party developers and users) for their growth and competitive position (Boudreau 2012; Evans and Schmalensee 2010). For platform participants, functionality is a major concern when choosing between different platforms (Haile and Altmann 2016; Nikou et al. 2014). Therefore, a platform's functionality differentiates it from competing platforms. In order to create an attractive platform, owners are therefore interested in good core functionality as well as stimulating external contributions. For this, owners must strike a balance between features provided by the platform core and those offered by third-party developers. While core functionality allows for enhanced control, innovation may be limited compared to external innovators competing for the best features.

Many browsers fulfil the criteria of a platform in terms of architecture and extensibility. Previous studies considered browsers such as Google Chrome (van Angeren et al. 2016) and Mozilla Firefox (Song et al. 2018; Tiwana 2015) as digital platforms from a research perspective. In combination with their ecosystem, they constitute a multi-sided platform. Their modular architecture allows them to be extended by modules (Boudreau 2010), which are distributed via platform-specific marketplaces (e.g. Chrome WebStore) (Ghazawneh and Henfridsson 2015). The modules are provided by third parties and interact with the platform core via interfaces (e.g. Chrome API) as part of the boundary resources.

Platform coring is understood differently depending on the research focus. According to Gawer and Cusumano (2008), coring subsumes the activities related to the creation of a platform and the design of the connected platform core. The platform core is understood as the most central and fundamental system element (Gawer and Cusumano 2008). Gathering external innovations through acquisition is a type of coring considered by Toppenberg et al. (2016). Thereby, coring subsumes the acquisition and the steps to integrate related technologies. Considering software platforms in particular, Um and Yoo (2016) analyzed the evolution of the blogging software WordPress. As open-source software, WordPress has no central authority, which involves interesting aspects for coring. In terms of functional variety and growth, core-related Application Programming Interfaces (APIs) were found to be essential (Um and Yoo 2016).

Bender and Gronau (2017) understand platform coring as the integration of several functionalities provided by third-party applications into the platform core. The types of platform coring are systematized according to the amount of coring (whether some or all functionalities are cored) and the application maintenance (whether the module is available after the coring activity). The authors demonstrate coring using an example of the Apple iOS platform for mobile devices. In particular, similarities in the functional development of the messaging services WhatsApp (third-party application) and iMessage (part of the platform core) were shown that fulfil the criteria of platform coring. Finally, implications of coring were discussed. While coring constitutes a risk regarding functional differentiation for third-party developers, opportunities for a beneficial partnership incorporating coring activities were presented. Due to their differentiated approach with various types of coring activity during platform evolution, the definition by Bender and Gronau (2017) is considered most suitable for this contribution in order to explain the attitude formation and strategic decisions of third-party developers.

While previous studies have shown the existence of platform coring for mobile devices (Bender and Gronau 2017), technological (Toppenberg et al. 2016), and service platforms (Saarikko 2016), only case-studies were used to explain the phenomenon. The extension of research to a purely digital software platform increases the generalizability of previous findings. In addition, interviews allow a detailed analysis of stakeholder perspectives.

Previous research has identified differences between open-source and proprietary platforms and contributions, yet only proprietary platforms have been investigated. From a structural point of view, open-source platforms are considered more open than proprietary ones (Eisenmann et al. 2009). From a development perspective, it is assumed that open-source platforms require less development effort for extensions due to code-sharing possibilities (Holzer and Ondrus 2011). Open-source and proprietary platforms often coexist. Competition between them is subject to different dynamics than between platforms of the same type (Economides and Katsamakas 2006). The contributions themselves can also be open-source or proprietary. Previous studies have highlighted negative effects due to the loss of functional differentiation for third-party modules as a result of coring (Bender and Gronau 2017). Assuming that loosing differentiation will result in fewer users, as they are likely to use the default option, commercial third-party developers will suffer from declining revenues. This in turn will lead to less motivation to develop for the platform and maintain the application. It is unclear whether this still applies to open-source platforms. Unlike proprietary contributors, open-source ones do not expect a direct return on their investment (Allen 2012). They are more driven by a sense of community, by developing for a greater good or by gaining reputation. Thus, the type of development paradigm may be closely connected to the motivation of the developer and the perception of coring. By considering multiple proprietary and open-source browsers, we can examine differences related to platform openness, developer motivation and governance mechanisms. This closes the gap in the perception of coring by third-party developers with regard to their initial motivation to contribute:

RQ1: How does the motivation to contribute to a platform influence the attitude towards coring?

From a platform dynamics and evolution standpoint, it is of great interest to know which contributions are cored. While the decision on coring is the prerogative of the platform owner, third-party reactions (e.g. establishing countermeasures) depend on their perceived likelihood of being cored. In general, core functionality is designed to be relevant for a general audience, while extensions enrich the platform experience with specialized functionality. The specificity of extensions' functionality varies among the available extensions. It is assumed that extensions that are relevant to a larger audience are more likely to become part of the platform core than domain-specific functionality. More generally, we aim to identify:

RQ2: Which factors influence the perceived likelihood that an extension will be cored?

Depending on the individual assessment, the developers either welcome or oppose coring. In combination with the likelihood that an extension will be cored, developers are likely to take measures to prevent coring. During development, they can adjust the ease of coring or complicate coring activities by implementing countermeasures. However, traditional code protection mechanisms are difficult to implement given the highly modular software architecture and detailed development specifications of software platforms. So far, no research has been conducted on potential third-party protection mechanisms facing potential coring. Taking the developer perspective, it is necessary to investigate which countermeasures exist and are promising under which conditions:

RQ3: Which countermeasures can be used in the platform context to protect extensions from being cored?

Similar to the contributors, there is hardly any knowledge from the point of view of the platform owner. It is of great interest to assess whether the owners conduct coring systematically to improve and strengthen the platform. In terms of functionality, providing value is known to be important for users to adopt a platform (Haile and Altmann 2016). Owners may therefore be interested in differentiating themselves from competing platforms through the functionality they provide. External contributions constitute an attractive source of innovation for platform owners. Platform statistics offer insights regarding popularity, which allows to systematically identify interesting contributions. Owners are thereby able to reduce uncertainty concerning the implementation of new platform features. As such, we explore:

RQ4: Do platform owners systematically core extensions from the platform?

Research Method

Platform coring has mostly been investigated using publicly accessible data. No in-depth investigation on the perception of coring, its antecedents and impact exists, reconstructing the connection of the perception of coring and potential countermeasures. Existing quantitative measures of attitude and choices in software development (Feldt et al. 2010) only assess specific aspects in isolation. Furthermore, they do not capture the reflections of the people involved in the decisions and explain the underlying sentiments and dominant mental concepts of third-party developers.

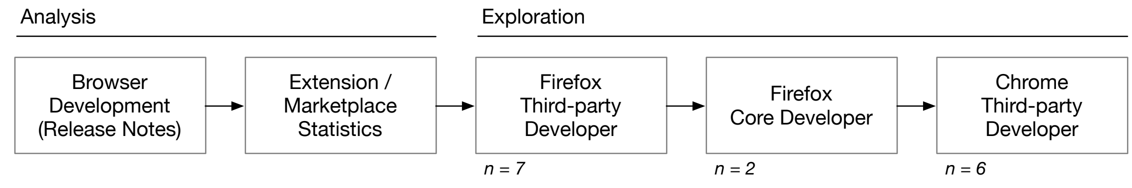

Figure 1. Data Collection

This study therefore uses an exploratory approach to fill this gap and provide insights into possible strategic reactions to coring that were previously unknown. In order to extract an explanatory model, we used a grounded theory approach that triangulates multiple data sources and enables a multi-dimensional analysis (Bryant and Charmaz 2007; Glaser and Strauss 2017; Urquhart 2012).

We have iteratively enriched the following empirical material as illustrated in Figure 1. The release note analysis provided insights into the functional development of browsers and their respective coring domains and thus the generation of narrow concepts and the definition of the sampling strategy. Browser statistics were used to identify prevalent browsers and market shares. Extension statistics and marketplace information allowed to identify suitable extensions for interviews. A total of 15 interviews were conducted during June 2018 and February 2019 to examine and refine the antecedents, attitude formation and strategies. One additional interview was conducted in August 2019.

In order to derive our insights from the material, we followed the three conceptualization phases laid out by Glaser and Strauss (2017). At first, statements from the interviews were categorized. Narrow concepts from previous studies and the release note analysis were used as initial categorization schemes. The categories were successively enriched and remodeled with each new interview and each contrasting statement. In the second phase, the resulting categories were condensed into key categories or themes and linked within each interview, creating concepts to explain the coring phenomenon. Additionally, the relationships between concepts as well as developer and platform specifics were deduced. This resulted in fractured data representing all different facets. This was followed by a phase of theoretical coding, in which associations and patterns of the concepts across all cases with the core concepts of motivation, attitude, coring likelihood and countermeasures were condensed. The generated models around the core concepts are still bound to the specific domain. However, they still provide a substantive focus for further research on coring in browser platforms (Urquhart et al. 2010). In order to create a sufficiently dense picture in this domain, a specific sampling strategy was used. Since coring is not equally present in all extension categories, a selective sample was generated to extract the relevant stakeholders.

Sampling Strategy

Sampling followed the grounded theory approach, in which multiple perspectives are regarded as data slices to describe the phenomenon correctly (Urquhart et al. 2010). In our sampling strategy, we have considered three perspectives: a platform-specific, a functional, and a stakeholder-specific perspective.

The *platform-specific perspective* addresses differences in attitude formation and technical possibilities of browser platforms. Browser are available for desktop computers and mobile devices. We focus on desktop applications, since mobile browsers do not offer comparable options to install extensions. To identify the most promising environments, the following requirements were defined:

1. The platform allows for functional extension following Tiwana et al. (2010). Since incorporating external functionality into the platform's core is the key of platform coring, this is mandatory.
2. With regard to the consolidating browser market, the platform should have a good perspective. This addresses the prospects of developers who affect their strategic decisions. Developers will restrain to invest in discontinued platforms. Similarly, owners will no longer make strategic decisions.
3. The platform should have a significant user base, which is necessary to ensure the momentum that platform and extensions can gain by being successful. Without a significant user base, a platform is not attractive to developers and decisions related to coring will be less strategic (Kim et al. 2016).

Currently, the browser market consists of the six players: Google's Chrome, Mozilla Firefox, Microsoft Edge browser, Microsoft Internet Explorer, Apple's Safari and Opera. Although all allow for extended functionality, not all of the corresponding marketplaces offer a reasonable amount of extensions to be considered active. Safari only provides 67 extensions directly in their store. Similarly, both Microsoft browsers only have few extensions available in their stores. They allow to install extensions via other sources. This is not only unfavorable for sampling purposes, but also for owners who are unable to monitor the use and success of extensions. Since they cannot directly access extensions, it is much harder and unlikely that the function will be cored. Safari, Edge and Internet Explorer were therefore removed from the sample. The *second* requirement filters both Microsoft browsers. They are discontinued or under heavy refactoring. While the Internet Explorer was discontinued in 2015, Microsoft announced in 2018 that the Edge browser will not be developed further with its own engine. The *third* requirement distinguishes the three remaining browsers. Chrome, the leading browser with around 70% market share, Firefox with 9.8% and Opera with only 2.3% (StatCounter). Opera was therefore excluded. Chrome (C) and Firefox (F) were addressed for further sampling. This platform sample also allows to differentiate between an open-source-dominated environment in Firefox and a proprietary environment in Chrome with its governance mechanisms, web store and monetization options (Ghazawneh and Henfridsson 2015; van Angeren et al. 2016).

Both browser environments offer various extensions. However, not every functionality is exposed to coring. The *functional perspective* selects a domain in which coring is likely to occur. Analysis of release notes revealed that the security and privacy domain is a potential area of coring. Exemplarily, Firefox implemented a pop-up blocker (2002), deletion of private data and password manager (2005), anti-virus notifications (2008), private browsing (2009), mixed content blocking (2013) or malware protection (2014). Recent versions (63 in 2018) introduce content-blocking features that block unwanted advertisement, pop-ups and scripting, which were previously provided by extensions. Similar developments can be shown for Chrome. Furthermore, both browsers name security in their mission statement. Given the sensitive nature, we believe attitude formation to be strong in this domain. Other domains are not considered that vital. We therefore focus on the security and privacy domain.

Table 1. Interview Partner Overview

#	Code	Extension Category	Platform	Focus	Stakeholder
1	PB	Privacy	F&C	Firefox	3rd party dev
2	PU	Popup Blocking	F	Firefox	3rd party dev
3	NS	Script Blocking	F	Firefox	3rd party dev
4	AG	Ad Blocking	F&C	Firefox	3rd party dev
5	AB	Ad Blocking	F&C	Firefox	3rd party dev
6	AP	Ad Blocking	F&C	Firefox	3rd party dev
7	AN	Proxy	F&C	Firefox	3rd party dev
8	EM	-	F	Firefox	Core Team
9	MZ	-	F	Firefox	Core Team
10	SP	Password	F&C	Chrome	3rd party dev
11	PR	Privacy	C	Chrome	3rd party dev
12	TS	Privacy	F&C	Chrome	3rd party dev
13	SG	Password	C	Chrome	3rd party dev
14	GI	Secure Transmission	C	Chrome	3rd party dev
15	PM	Password	F&C	Chrome	3rd party dev

The third perspective addresses the *motivation of different platform stakeholders*. Since coring is primarily relevant for the relationship between owner and third-parties, the group of users is neglected. A broad community is developing for Mozilla Firefox and parts of the Chrome browser (Chromium). Yet, strategic decisions for Firefox are made by the core development team and Mozilla´s product managers. At Google Chrome, decisions are made by the respective teams at Google. In addition to strategic decisions, Google and Mozilla operate related marketplaces that provide access to extensions, which is why they are considered the *platform owners*. As the entity taking strategic coring decision, owners should be included in the sample. *Third-party developers* provide a different perspective on coring. Their motivation varies from business aspirations to intrinsic motivations such as their own needs. Extensions can be published following the open-source idea or as closed-source components. In some cases, cross-platform development can also be observed. Concerning coring, third-party developers are affected the most. Their attitudes towards this practice is thought to lead to different strategies regarding their extensions. The sample therefore reflects both sides of the platform.

The sample consists of 15 interviews, which according to Marshall et al. (2013), are sufficient for a limited-range study to develop a substantial yet informal model. Two product managers (both Mozilla and ex-Mozilla) reflect the perspective of the platform operator. Respective persons at Google were not available for interviews. The thirteen third-party interviews are unevenly distributed between Mozilla (7) and Chrome (6). Eight extensions are provided on both platforms. All extensions belong to the security and privacy domain, yet in different sub-categories. Developer were recruited via direct contact through the web store. Owners were contacted through the development community, especially through GitHub. Table 1 summarizes the attributes of the sample and the subsequent codes for the interviews.

Interview Structure and Data Collection

The data was collected through qualitative semi-structured interviews which provide an open structure to explore and follow up on thoughts and associations. Following the guidelines laid out by Myers and Newman (2007) on conceptualizing qualitative interviews, the interview began by situating the interviewer into his research interest in order to reduce social dissonance. The interviewees were then asked to introduce themselves and their projects. Their narration was used to generate follow-up questions. Interview topics were semi-structured, allowing the interviewer to maintain a natural flow throughout the conversation. Two stakeholder-specific guidelines were used to capture important topics, covering awareness and the assessment of coring, as well as decisive factors in this assessment and their effects and implications on extensions and the development process. The interviews with the platform owner focused on platform strategy. Questions regarding the platform ecosystem, the review process, the browser development strategy and the need for coring were asked. For third-party developers, the questions mainly addressed motivational aspects, which mostly focused on extrinsic (business model) and intrinsic (skill enhancement, missing functionality) motivation. Since most browser ecosystems are community-driven, we also included altruistic motivation into our question set. Besides the antecedents, we were interested in possible protective mechanisms for developers.

The interviews were conducted by three different researchers in order to reduce the interviewer bias. During the data collection phase, the interviews focus shifted as usual in the grounded theory approach. While the first interviews were mainly concerned with Firefox and explored related issues (e.g. the switch of boundary resources), the later interviews focused on motivational aspects and cross-validation of Firefox findings in the Chrome environment. Hence, the weight put on the different topics in the guideline shifted during data collection, reducing the confirmation bias. Interviews were conducted using various media. Skype and other voice-services were used most, while fewer interviews were conducted via e-mail (King and Horrocks 2010). The oral interviews were recorded and transcribed using aggregating transcription rules, omitting pauses and insertions.

Data Analysis

The data analysis is based on the collection of interview transcripts. According to Lacity and Janson (1994) different possibilities for interview analysis exist. Since the goal is to reconstruct intentions, we applied intentional analysis for coding and condensing the raw data (Sanders 1982). In a four-step process, shared “facts” are separated from their individual interpretations or sentiments. The individual perspective is condensed into common themes. As part of building the explanatory model, the themes are then abstracted into “essences” which are interpretations of the themes. This approach allows both a generalization of our findings within the sample and a differentiated view of the individual positions. Our analysis is thereby rooted in the data and connected to a theoretical foundation.

In the initial step, the transcriptions were *individually coded*, marking statements related to the research questions. In order to focus the analysis on the research questions, we used thematic coding with a closed initial coding scheme (Gibbs 2008). The initial codes were generated along with the interview guidelines (motivation to develop; coring experience; coring likelihood; attitude towards coring; effects and implications of coring; strategies and measures against coring). They also served as core concepts during the theoretical coding phase. In order to take into account different interpretations of the same "fact", e.g. motivation for development, each code was extended with a sentiment from the statement, so that cases could later be compared and contrasted. The initial codes were enriched during the coding and more detailed categories were captured, e.g. different motivational aspects. Coding was conducted individually by two researchers to *avoid an initial bias*. Resulting codes were discussed and unified to find a common interpretation. Thereby, the initial coding scheme was modified to capture the specifics of the statements. The resulting coding scheme was then applied to all interviews. In a third step, categories were grouped into *concepts or common themes* linking sentiments and statements. Different concepts within each interview were then compared across the interviews. Similar and contradictory statements in one topic area were identified and recorded. For contrasting and common themes, an "essence" was formulated, capturing the basic characteristics of the core concepts. The analysis section explains related concepts within and across categories. Interview quotes are marked with the interviewee code in Table 1 and the corresponding page of the transcript.

Analysis

The analysis section is structured according to our research questions. Table 2 provides an overview of the identified themes and concepts and their association to the research questions.

Table 2. Interview Findings Concepts and Themes related to Research Questions

Research Question	Concept	Themes
RQ1: Attitude	See Figure 2	Intrinsic Motivations Extrinsic Motivations Mediation Possibilities
RQ2: Likelihood	Positive Influences	Popularity
	Negative Influences	Specifity
		Infrastructure
		Complexity / Usability
RQ3: Countermeasures	Technical Measures	Service Capsulation
		Code Encryption
	Strategical Measures	Differentiation
RQ4: Systematic Coring	Lean Core	Specialized and Advanced (Systematic)
		Conduct Coring Consciously

Developer explicitly recognized coring. We separated the experience related to coring in general from that specific to functionalities similar to the provided extensions. Coring in general was frequently recognized by the developers surveyed. Five developers recognized coring for their extension in specific (PU, GI, SG, PM). In one case, the extension was provided by default in a specialized browser version (NS).

> *"Google released an addon similar to my addon to filter spam sites. [...] my project got stopped after some years." (PU:1)*

> *"NS is shipped as a built-in component in the Tor Browser, which is currently the most important tool for security, privacy and anonymity on the web" (NS:1)*

Motivation and Attitude towards Coring

Developers named different motivations for providing browser extensions. Using prior literature, motivations were grouped into overarching categories (intrinsic, extrinsic) and related to found concepts.

Intrinsic Motivations

Missing functionality was mentioned by the developers as a reason to develop corresponding extensions (SP, GI, SG). Typically, the provided extensions encompass specialized domain functionality.

> *"The reason why I started them […] was that there was no good way to do email-encryption and I, as a user, wanted to have it available." (SP:2)*

Related to functionality are *specific requirements* such as security (SP), usability and simplicity (SP, SG, AG), as well as individual needs (SP, SG) were mentioned:

> *"And using that was not convenient for website day-to-day usage so I just built the Chrome extension […], in designing user interfaces with minimum interactions" (SG:2)*

Developers mentioned the aspect of fun as a motivation to develop browser extensions (SP, PR, TS).

> *"It is always fun learn like it was the first time I worked on a Chrome extension […]." (TS:3)*

The aspect of social good (altruistic motivations) was pointed out by a couple of developers (PR, TS, AB).

> *"I think in general we wanted to do a project that had some social good or some social initiative behind it." (PR:2)*

Developing extensions is also recognized as an opportunity to develop new skills (SG, TS, PR).

> *"I build them primarily for myself, both for my usage and for my experience and learning exercise." (SG:2)*

A shared vision with the platform is a motivation for some interviewees to develop extensions (NS, PB).

> *"We [Firefox and developer] both pursue the common goal of a safer web, where users have means to enforce their rights to privacy and security." (NS:3)*

Extensions are also used to raise awareness for specific issues. Especially security and privacy were mentioned frequently, which is not surprising as being a focus of Chrome and Firefox (PR, SG, TS, PB).

> *"We just wanted to sort of bring awareness to the issue, you know, that privacy policies are misunderstood and they're often misused by companies." (PR:5)*

Extrinsic Motivations

Besides the acquisition of new skills, extensions are used to present developers' skills. Providing an active extension is also considered to be relevant for potential job qualifications (PR, SG).

> *"Companies will look for developers that have actually shipped real products. So, having a big project like this […], is very valuable for interviews and job searches." (PR:3)*

Motivational business-related aspects include building a business model upon an extension, using extensions as a showcase (GI), as well as seeing major platforms as important markets. Business-related and monetization aspects were named by multiple developers (GI, AN, PR, PU, PM).

> *"So what I did is to do my own start up, worked on this and I provided the solutions […]. Recently it seems like the first state is finished. It is a product that is working […]" (GI:4)*

The considered browser platforms constitute an attractive and important market, which is a development motivation and a reason to choose a specific platform. The aspect of a platform having many users is equally relevant for extrinsic motivations in terms of market potential (AG, AB), as well as intrinsic motivations such as dissemination and reaching users (GI, PR, SG).

> *"Firefox is a popular browser so we just cannot ignore it. [...] nothing attracts better than the overall browser popularity." (AG:3)*

While the majority of interviewed developers' motivations are distinctively intrinsic or extrinsic. We also found shifting motivations, such as starting a free extension with the intention to monetize it later (GI) or extending an initially free service with paid-for options (PR).

During the interviews, we found motivation to be closely related to attitude formation. The intrinsically motivated developer showed a clearly positive attitude towards coring, albeit for different reasons. From the intrinsic hobbyist perspective originating in a missing functionality, closing the functional gap is sufficient and regarded positively (TS, SP):

> *"I wish they did. In this case I would be very much in favor of it. [...] So if the browser provides it that would be great [...] that is why I developed that and I wish that they add the capabilities to the browser" (SP:2)*

Developers take pride in making their solutions available for a broader audience, either from an awareness or altruistic perspective (SP; PR; SG):

> *"That's a fantastic step forward and with this the browser password manager covers around 70 % of SG' value. [...] Because I am doing it out of a hobby and a personal need I am super happy that this will now reach more users and will work better and more stable than an extension." (SG:5)*

This is also reflected in the missionary perspective, where coring is considered positive (PR, PB):

> *"It would mean that we won! It would be really good. We would love if PB got to the point where PB didn't have to exists or for us to integrate it." (PB:2)*

However, most developers take a divided stance and acknowledge that a negative attitude towards coring might exist if the developer has an extrinsic, commercial motivation (PR, SG, SP). Except one, statements related to a negative attitude are hypothetical, since only one interviewee had a primarily commercial focus. All see an ousting of the extension by the cored browser functionality, as this statement shows:

> *"I think if you are trying to make money from your extension that is bad for you. Because then users won't use your extension. They will use the browser without your extension." (SP:4)*

Interviewees make power imbalance a topic of discussion (PR, SG, PM), but also recognize the platform owner's interest in closing functional gaps regardless of the commercial interest of third-party developers:

> *"And it's not that the browser makes just destroys a particular market and pushes you out of business, it's that they're filling the gaps in their browsers which shouldn't have been gaps in the first place." (SG:7)*

Third-party developers understand that there is a coring necessity for the platform owner to stay user-focused and competitive (TS, PM) or to improve the performance and security of the browser (PR, PM):

> *"You [browser developer] want to look how you can make your browser better. You should. And then add features that seem to be good for the browser, sure." (SP:5)*

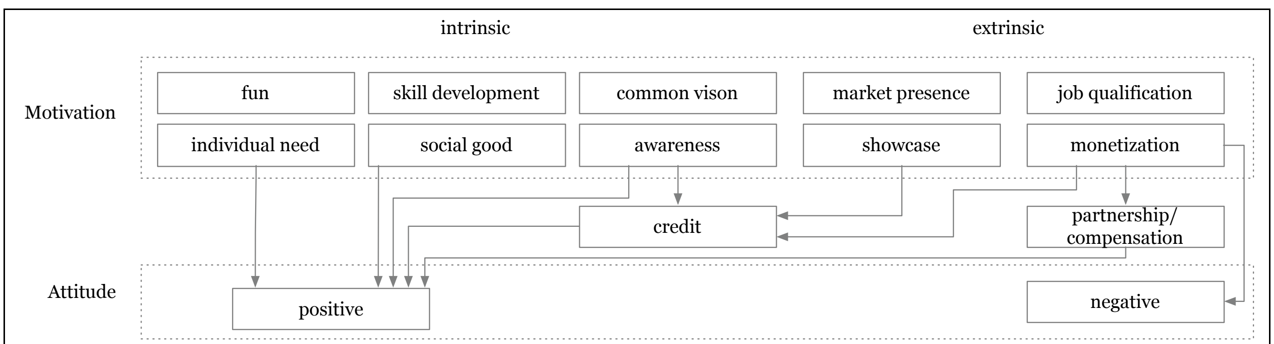

Figure 2. Core Concepts in the Relationship between Motivation and Attitude

Mediation Possibilities

Even though the overall attitude towards coring is positive, developers reflect on the conditions of coring. One condition is deeply connected to the understanding of the open-source community. Code used from other projects has to be attributed to the original author. Most open-source licenses contain this condition. Recognition and attribution are therefore also a present theme in the interviews, especially where awareness, showcasing and monetization are key (SP, GI, PR, TS, NS, PR).

> *"I don't need to necessarily be paid for it. I would like to be paid for it, but I don't have to. But if they'd credit me or something like that, that would be good." (PR:6)*

An open dialog about coring is therefore essential when coring occurs (AB). Where a monetizing option is present or the extension is part of a broader business model, different compensation options were discussed. One would be a partnership between the platform owner and the developer (GI). However, this only works if the third-party business model can be extended beyond the extension functionality, i.e. by providing special purpose features or integrating other products into the cored functionality. This is done, for example, in a partnership between protonvpn and Mozilla (Mozilla 2018), which combine revenue streams that both parties can participate from, and users take advantage of the extended functionality.

A buy-out might also be a way to fairly take over the commercially provided functionality (PR):

> *"I feel like if Google were to buy the extension or pay for the rights to use the functionality to the developer – it could be a good thing. It could be path to monetizing something that was previously unmonetized." (GI:8)*

The overall attitude towards coring in the security and privacy domain is positive, since most developers are not pursuing commercial interests. Their attitude is primarily driven by their intrinsic motivation or missionary approach. However, this does not mean that coring attempts are always welcomed. Even with an intrinsic motivation, developers demand a fair coring process with recognition of their work. If the developers are planning to monetize their work, they demand for some kind of compensation, either a partnership or a one-time payment. Otherwise their attitude towards coring is negative. Figure 2 summarizes the categories and concepts found in our investigation.

Coring Likelihood

We asked developers to assess which factors influence the likelihood that an extension will be cored. Depending on whether or not coring is aspired by the developers, their assessment might determine the use of countermeasures. *Application popularity* was frequently mentioned as factor influencing coring probability (PR, TS, AG, PM). Developers assume popular extensions are more likely to be cored.

> *"Google would just steal it when it becomes popular. [...] Browser companies would take the top couple of extensions every month or so and incorporate them into their browser [...]" (PR:4,6)*

Related to popularity, but more concrete, was the aspect of *specificity*. Developers assume generic extensions as more likely to be cored than more specialized extensions for specific domains (TS, AN).

> *"Being sort of generic so not as a feature, not specific [...] the [generic] is more likely to be cored than the [specific] one about food." (TS:6)*

The infrastructure required to provide a functionality is considered as relevant. A developer with multi-country proxy service allowing to switch IP addresses to access country-specific services mentioned this.

> *"Well, you don't operate it for free. So, the infrastructure behind it. Mozilla has strong capital behind it [...]. But I don't think that would be practicable for all Firefox users in the long run." (AN:2)*

The aspect of extension complexity in terms of usage along with usability was mentioned. Core functionality should be easy to use, which is why it is assumed that platform owners will refrain to core complex features that require extensive configuration efforts before employed effectively (GI, PB, AP)

> *"[Firefox] have been looking at which privacy features they could add that improve user's privacy without breaking too much stuff." (PB:2)*

Developers were in accordance that platform owners are capable of coring extension functionality.

> *"Google could integrate it at any moment with the Chrome browser […]. When I started the implementation, I kept it secret from anyone around me and did not answer any questions because I was scared of Google developing it before I do." (GI:5,7)*

Perceived Effects and Implications of Coring

Besides the motivation of the individual developer, attitudes and the resulting development decisions are governed by the perceived effects of coring. As such, perceived effects are important to understand related actions and the use of countermeasures. These effects can be related to either individual development efforts and motivation, or the entire ecosystem and platform. Individual effects mostly originate in the perceived decline in active users, since the built-in feature would dominate with its functionality (AB):

> *"There's tons of business cases where being the default is exactly what makes you the winner, right? […] So, if it's in the platform natively then extensions don't really have a chance to compete." (PR:5)*

One interviewee even experienced this impact on another platform, when Opera introduced and marketed its own ad-blocking function:

> *"Overall, this made a difference, and the users number stopped increasing and even slightly decreased over a 2-year period." (AG:2)*

For a commercial developer, this results in the loss of the business model and monetization prospects:

> *"And I understand how it can be painful in some cases […] and there will be some interesting court cases, […] But yes, the commercial prospects of extensions could be worsened by that." (SG:7)*

Coring might also lead to decreased development efforts if the initial motivation was to fill a functional gap. Even if the cored functionality does not offer all the functions the extension did, developers may consider ceasing to work on the extension and stop using it themselves:

> *"I would definitely suggest using that instead and that's one of the reasons why I didn't put considerable development into it lately. I am also myself considering to move away from SG, so there's that." (SG:3)*

Anticipated negative effects on the user base will lead to the omission of the specific extension (GI, TS, PU). However, since the platform might only perform a partial coring, there might still be a chance for differentiation (GI, AP):

> *"Unless I could think of a differentiator that I could compete on effectively." (PR:5)*

While a business model is a strong motivator to differentiate, some developers follow altruistic or even missionary aspects. For them, coring can be viewed from the angle of dissemination and user adoption. Some interviewees address this development when coring occurs (GI, PR, SG):

> *"I think we would probably be happy if this specific functionality got cored into Google Chrome because I think it would allow a wider audience to reach it." (PR:5)*

However, interviewees also see the downside of coring. The individual demotivation could hurt the entire ecosystem (PM). The fairness aspect is closely related to the demotivation effect (PR:4):

> *"I think it would be bad for developers that are looking to innovate. They wouldn't be as incentivized to build extensions anymore if they're all cored." (PR:6)*

This would also lead to less innovation in the ecosystem, since in the extension market different developers are competing with their solutions, while a cored function would have a monopoly and development would only be fueled by the platform owner's interests.

Also, there is a technical risk in monopolizing some security features, such as encryption algorithms. Besides the positive effect of being available to a broad audience, the cored functionality would offer the potential for manipulation (SP). This also leads to a political question in how far the platform owner can be trusted with centralized security functions like encryption and anti-tracking (PB).

Perceived effects are balanced between benefits and risks of coring. Even though developers like to see their functionality accessed by a larger audience, they see the risk of losing control and of being exploited by the platform owner. Since most of the developers are focused on their own needs or community needs, they are concerned about a less innovative platforms with the potential to abuse its central position.

Anti-Coring Measures

Third-party contributors can implement measures that hinder coring or result in advanced efforts to core an extension. Given the modular software structure, traditional protection mechanisms cannot be applied. As such, we aimed to explore platform-specific possibilities to protect extensions from being cored. The interviews revealed that developers include mechanisms or architectural components to hinder coring.

> *"[...] having the extension be the frontend and having most of the logic on the backend. That could be a sort of differentiator in terms of not being as easy to copy. Especially when doing something proprietary or something unique on the backend. (PR:5)"*

Developers provide their functionality using a combination of a capsuled server backend and browser extensions as the frontend (PR, GI, TS, PM). With this method, internal details, algorithms and logics in the backend are protected from the platform owner as well as other developers. We refer to this approach as '*service encapsulation*'. *Code encryption* is another approach to hide the details of an extension. While code encryption is rarely possible in the platform context, one developer used this approach to prevent competitive developers from gathering details about an extension (PU). Since providing details for decryption is required during the extension approval process, this approach is not suitable to protect it from the owner and thus from platform coring. Nonetheless, preventing other developers from copying extensions and releasing them under a different name was mentioned by multiple developers (PU, GI).

In contrast to static architectural aspects, *differentiation* is considered a dynamic technique to provide additional value to the platform. However, both times the corresponding developer acknowledged that if core functionality is good enough, they are likely to drop the extension (GI, PR).

> *"Unless I could think of a differentiator that I could compete on effectively" (PR:5)*

Currently, most developers do not systematically protect their extensions. The interview data reveals that developers who employ protection measures have an external motivation associated with their extensions.

Core Developer Analysis

We interviewed core developers to investigate their perspective on coring. First, we asked for the *role of extensions* for browser platforms. Besides extending browsers' functionality, the interviews revealed extensions to provide more specialized and advanced functionality than the platform core (MZ, EM).

> *"[Extensions] make the kind of changes that might not be acceptable in a mainstream browser. [...] That's something specialized functionality that probably not worth putting in the browser that everybody uses but people who work in fields for they need a lot different scientific literature maybe find that very helpful." (MZ:1)*

In that sense, extensions are viewed as an opportunity to provide domain-related functionality that is not relevant for the mass market, but is quite useful for specific user groups (MZ, EM). Extensions are also used to provide more advanced functionality than the browser itself. In that aspect, both third-party and core developers have a similar understanding of the role of extensions for browser platforms, and both acknowledged the advantages of an extensible platform.

> *"Security provided by extensions seems to be far greater than the core functions of the browser, however. Even with recent effort in private mode by the core Firefox teams to add more" (EM:1)*

The access to core features varies among extension providers. Mozilla extensions have superior access to core features, while third-party extensions are limited to the possibilities of WebExtensions (EM).

We were interested whether platform owners *consciously conduct coring* activities. While there is a chance that core and extension developers might simultaneously work on a functionality or might not know about corresponding functionalities, the interviews revealed that platform owners do actively conduct coring.

> *"Q: Did Firefox perform coring of functions which have been formerly provided exclusively by extensions? A: Yes, recently they added to private browsing mode some extra features" (EM:1)*
>
> *"Q: Coring happens in the Firefox ecosystem. I could answer that with a "yes" [...]? A: Yeah!" (MZ:2)*

Similar to contributors, we asked the core developers which factors influence the *likelihood of an extension being cored*. The core developers predominately named user experience and complexity.

> *"I still think the biggest blocker for coring is worry of breaking websites. The average user is not a power user and cannot bother to figure out how to fix broken websites." (EM:2)*
>
> *"Well there is always a trade-off between site-breakage and privacy [...]. So, user [...] can install that extension and they end up taking on more work and more inconvenience, they get more privacy. That's the kind of trade-off that you could make in the mainstream browser." (MZ:2)*

Core developers are focused on providing solutions that work from scratch. Websites shall be shown as intended (user experience). Major configuration effort is only acceptable for more advanced users aiming to achieve a higher level of privacy (complexity) (MZ). For them, extensions offer a valuable opportunity to do so. Core interviews revealed cost not to be important concerning coring likelihood (EM).

Core representatives acknowledged to be in *competition with other platforms* for the developers.

> *"A lot of the extension developers work on both Firefox and Chrome. So, I think there is a little bit of competition to give people a better development experience" (MZ:6)*

Both core developers named providing better development resources as a key factor in platform competition. Mozilla provides interfaces beyond the WebExtension standard, which allows for a more convenient development, but limits the possibilities to release extensions on other platforms.

Discussion

The results extend the existing literature by various *theoretical contributions* in the context of digital platforms. *First*, the study expands on prior coring studies by demonstrating the existence of coring in the browser domain and not to be restricted to hardware-related platforms (Bender and Gronau 2017; Toppenberg et al. 2016). The occurrence of coring is confirmed by third-parties and core developers who acknowledged that they actively conduct coring. Furthermore, the results provide insights into coring from the platform owner's perspective. With no claim to generalizability, the examples suggest that owners conduct coring systematically, as exemplified by NoScript as part of the Tor browser. With respect to RQ4, the results suggest that coring is being used systematically by platform owners. *Second*, the detailed assessment of coring attitudes extends previous literature by providing a more differentiated view of coring, as older studies predominately hypothesized that coring would be considered negatively by developers (Bender and Gronau 2017). Regarding RQ1, we found that the motivations of developers are a moderating factor in the formation of coring attitudes. Where intrinsically motivated developers see coring positively, they consistently assume negative attitudes when extrinsic motivations are present. While coring runs contrary to extrinsic motivations, this is not necessarily the case with intrinsic motivations, which could explain the moderating effect of the motivation type. *Third*, the concept of perceived coring likelihood was introduced. Various factors, such as the popularity or generosity of an extension, influence the developers' assessment of the coring likelihood of an extension (RQ2). Interviews with core developers confirmed that the usability and complexity is important for coring candidates. In this context, differences were noted between the aspects perceived as relevant by developers and the actual aspects considered by core developers, however common elements link the two. *Fourth*, the study confirmed that established software protection measures are hardly applicable in the platform context. The study identified architectural, development-specific, and strategic measures, to be actually used by developers (RQ3). Architectural countermeasures go hand in hand with negative coring attitudes. While architectural measures in particular are not necessarily the result of the desire to prevent coring, similarities can be found in the data. Similarly, the developers' assessment of the capability and likelihood of coring had a positive effect on the use of measures that hinder coring, especially in combination with a negative attitude. For developers, measures are not necessary if coring is unlikely but desired. Conversely, if coring is likely yet undesired, their implementation becomes reasonable. *Finally*, the study proves many aspects of earlier studies. For example, the assumption that the quality of development resources provided is important in platform competition (Kim et al. 2016) has been confirmed by both third-party developers and platform owners. Furthermore, the provision of publicly available software enables developers to present skills that are relevant for potential jobs (Holzer and Ondrus 2011).

The study has various *practical implications*. Previous studies assessed coring negatively due to the power imbalance between platform owners and developers. While the owner can freely choose to implement features, developers are faced with little chances to tackle the risks of their contribution being cored. However, this study showed that there are more options for both parties and that different dynamics can apply, leading to various strategies to cope with coring. *Developers* may welcome coring under specific circumstances, whereby motivation to contribute is key to attitude formation. Therefore, developers should be aware of their motivation and goals in developing an extension. With *intrinsic* or altruistic *motivation*, coring is a welcomed opportunity to gain visibility and contribute to a larger goal. For developers with *monetizing interests*, coring is seen as a threat. It has been shown that this risk can be addressed in several ways. One option are technical measures, i.e. one's own backend infrastructure or service. Another possibility is to extend the development scope to multiple extensions and other competing platforms. With this diversification, developers are less vulnerable to coring in the short term. In the long term, however, platform competition will lead to a functional convergence, putting third-party extensions at risk on other platforms too. Highly specialized extensions are another possible way to cope with coring. For platform owners, only functionality with a significant user base and general use cases is of interest. So, even if the general function is cored, the developer will still be able to survive in a specialized niche. This comes with the price of a smaller initial target audience, however. In all cases, third-party developers must make informed decisions about their business model and invest their efforts accordingly. For *platform owners*, coring is not as simple as earlier studies have suggested. Owners need to keep in mind the ecosystem around their platforms. The interviews showed that developers are sensitive to fair coring procedures. Owners must therefore be aware of the implicit rules in their community and actively engage with developers when coring functionality. Different models of recognition were extracted from the interviews, ranging from attribution to partnerships. Furthermore, coring must strike a functional balance between platform performance, flexibility, and the functional scope of integration. Owners must consider these aspects for two reasons. On the technological side, keeping the core lean to maintain its performance and adaptability to remain attractive to users and developers (Olleros 2008). On the ecosystem side, leaving functional niches for specialized extensions will encourage the development community and broaden the user base, particularly for highly specialized users who can add functionality by installing extensions.

This study is subject to several *limitations*. The *number of interviews* is relatively small. Given the complexity of coring and the different points of view uncovered, additional interviews help to better reflect the respective views. The *population studied is highly specific*. Only browser platforms were investigated. Although this is a new area for coring research, browser platforms differ significantly from other platforms; in particular the combination with hardware devices, but also the monetization possibilities differ from other platforms. In addition, the security domain has its own developer population. Even though the domain selection was a conscious choice, as outlined in the sampling strategy, it is unclear whether the same combination of motivations can be found in a less politically aware domain. The choice of domain and platform also limits the *selection of interviewees*. Most of them come from an open-source community with specific values regarding software development. This is amplified by the choice of browsers, most of which are also open-source. Therefore, the commercial perspective is likely to be underrepresented. Most statements about the risks of coring for commercial developers are hypothetical. Given the professional experience of the interviewees, these statements are not far-fetched, yet a distinctive, successful, commercial perspective is missing. Regarding the *sample structure*, the perspective of another platform owner is missing. The core interviews were conducted exclusively with Mozilla, as even with immense effort the Chrome/Chromium development remained unresponsive. Regarding the data analysis, the known *shortcomings of qualitative investigations* should be mentioned. The validity of the results is significantly influenced by the researcher's initial point of view in terms of conducting, coding and interpreting the interview. Even though the researchers attempted to objectify their individual interpretations during coding and concept aggregation, subjective bias might not have been entirely eliminated. Considering the above limitations, the contribution is not generalizable for all platform types and domains; but provides valuable insights into the dynamics between platform and third-party developers that should be followed up on in a larger quantitative study.

Conclusion

Modern web browsers can be considered a digital software platform as they allow third parties to extend the functionality of the platform through extensions. Given the intense competition differentiation is important for browser platforms. For software platforms, the functionality provided is a key factor for differentiation. As browsers progress over time, they constantly release new features. This allows browsers to enter complementary markets by adding functionality previously provided by third-party extensions, which is referred to as ‘platform coring’. While previous studies considered different aspects and stages of platform coring, they did not examine the perception of third-party developers as the party mainly affected by coring activities. To address this gap, we conducted semi-structured interviews with third-party and core developers in the security and privacy domain from Firefox and Chrome.

Essentially, this study provides three contributions. First, insights into the perspective of third-party developers concerning coring are revealed. Second, it identifies measures that developers can use to prevent coring. Third, practical implications for developers as well as platform owners in dealing with coring and balancing the interests of both are outlined. The results reveal that coring occurs on browser platforms. In addition to general recognition, some developers have personally gained experienced coring for their respective browser extensions. Core developers confirmed that they systematically core third-party extensions. The assessment of coring varies depending on the motivation of developers to contribute their extensions. While developers with extrinsic motivations assess coring negatively, developers with intrinsic motivations perceive coring positively. The effects of coring are systematized and discussed.

The study highlights many aspects for further research. In the interviews, the desire for recognition, partnership and compensation revealed important aspects that influence the assessment of a coring process. Therefore, future studies may focus on designing a fair coring process that is accepted by both developers and platform owners. Due to the limitations of study design, the generalizability of the results is limited. Increasing the sample size through additional interviews would increase the validity of the results. Additional platforms and developer with a commercial focus are of particular interest. Future studies in the sense of a mixed-method approach could supplement the insights gathered using a quantitative survey in order to assess the dynamics in a structured and reliable way.

References

Allen, J. P. 2012. "Democratizing Business Software: Small Business Ecosystems for Open Source Applications," Communications of the Association for Information Systems (30:1).

Bender, B., and Gronau, N. 2017. "Coring on Digital Platforms – Fundamentals and Examples from the Mobile Device Sector," in: International Conference on Information Systems (ICIS). Seoul, South Korea.

Boudreau, K. J. 2010. "Open Platform Strategies and Innovation: Granting Access Vs. Devolving Control," Management Science (56:10), pp. 1849-1872.

Boudreau, K. J. 2012. "Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation," Organization Science (23:5), pp. 1409-1427.

Bryant, A., and Charmaz, K. 2007. The Sage Handbook of Grounded Theory. Sage.

de Reuver, M., Sørensen, C., and Basole, R. C. 2017. "The Digital Platform: A Research Agenda," Journal of Information Technology).

Eaton, B., Elaluf-Calderwood, S., and Sorensen, C. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's Ios Service System," Mis Quarterly (39:1), pp. 217-243.

Economides, N., and Katsamakas, E. 2006. "Two-Sided Competition of Proprietary Vs. Open Source Technology Platforms and the Implications for the Software Industry," Management Science (52:7), pp. 1057-1071.

Eisenmann, T. R., Parker, G., and Van Alstyne, M. 2009. "Opening Platforms: How, When and Why?," Platforms, markets and innovation), pp. 131-162.

Eisenmann, T. R., Parker, G., and Van Alstyne, M. 2011. "Platform Envelopment," Strategic Management Journal (32:12), pp. 1270-1285.

Evans, D. S., and Schmalensee, R. 2010. "Failure to Launch: Critical Mass in Platform Businesses," Review of Network Economics (9:4).

Feldt, R., Angelis, L., Torkar, R., and Samuelsson, M. 2010. "Links between the Personalities, Views and Attitudes of Software Engineers," Information and Software Technology (52:6), pp. 611-624.

Gawer, A., and Cusumano, M. A. 2008. "How Companies Become Platform Leaders," MIT Sloan Management Review (49:2), pp. 28-35.

Gawer, A., and Cusumano, M. A. 2014. "Industry Platforms and Ecosystem Innovation," Journal of Product Innovation Management (31:3), pp. 417-433.

Ghazawneh, A., and Henfridsson, O. 2015. "A Paradigmatic Analysis of Digital Application Marketplaces," Journal of Information Technology (30:3), pp. 198-208.

Gibbs, G. R. 2008. Analysing Qualitative Data. Sage.

Glaser, B. G., and Strauss, A. L. 2017. Discovery of Grounded Theory: Strategies for Qualitative Research. Routledge.

Haile, N., and Altmann, J. 2016. "Structural Analysis of Value Creation in Software Service Platforms," Electronic Markets (26:2), pp. 129-142.

Holzer, A., and Ondrus, J. 2011. "Mobile Application Market: A Developer's Perspective," Telematics and Informatics (28:1), pp. 22-31.

Kankanhalli, A., Ye, H. J., and Teo, H. H. 2015. "Comparing Potential and Actual Innovators: An Empirical Study of Mobile Data Services Innovation," Mis Quarterly (39:3), pp. 667-682.

Kim, H. J., Kim, I., and Lee, H. 2016. "Third-Party Mobile App Developers' Continued Participation in Platform-Centric Ecosystems: An Empirical Investigation of Two Different Mechanisms," International Journal of Information Management (36:1), pp. 44-59.

King, N., and Horrocks, C. 2010. Interviews in Qualitative Research. London: Sage.

Lacity, M. C., and Janson, M. A. 1994. "Understanding Qualitative Data: A Framework of Text Analysis Methods," Journal of Management Information Systems (11:2), pp. 137-155.

Marshall, B., Cardon, P., Poddar, A., and Fontenot, R. 2013. "Does Sample Size Matter in Qualitative Research?: A Review of Qualitative Interviews in Is Research," Journal of Computer Information Systems (54:1), pp. 11-22.

Mozilla. 2018. "Testing New Ways to Keep You Safe Online." from https://blog.mozilla.org/futurereleases/2018/10/22/testing-new-ways-to-keep-you-safe-online/

Myers, M. D., and Newman, M. 2007. "The Qualitative Interview in Is Research: Examining the Craft," Information and organization (17:1), pp. 2-26.

Nikou, S., Bouwman, H., and de Reuver, M. 2014. "A Consumer Perspective on Mobile Service Platforms: A Conjoint Analysis Approach," Communications of the Association for Information Systems (34:1).

Olleros, X. 2008. "The Lean Core in Digital Platforms," Technovation (28:5), pp. 266-276.

Saarikko, T. 2016. "Platform Provider by Accident a Case Study of Digital Platform Coring," Business & Information Systems Engineering (58:3), pp. 177-191.

Sanders, P. 1982. "Phenomenology: A New Way of Viewing Organizational Research," Academy of management review (7:3), pp. 353-360.

Song, P. J., Xue, L., Rai, A., and Zhang, C. 2018. "The Ecosystem of Software Platform: A Study of Asymmetric Cross-Side Network Effects and Platform Governance," Mis Quarterly (42:1), pp. 121-142.

Tiwana, A. 2015. "Evolutionary Competition in Platform Ecosystems," Information Systems Research (26:2), pp. 266-281.

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

Toppenberg, G., Henningsson, S., and Eaton, B. 2016. "Reinventing the Platform Core through Acquisition: A Case Study," in: Hawaii International Conference on System Sciences (HICSS). pp. 4634-4643.

Um, S., and Yoo, Y. 2016. "The Co-Evolution of Digital Ecosystems," in: International Conference on Information Systems (ICIS). Dublin, Ireland.

Urquhart, C. 2012. Grounded Theory for Qualitative Research: A Practical Guide. Sage.

Urquhart, C., Lehmann, H., and Myers, M. D. 2010. "Putting the 'Theory'back into Grounded Theory: Guidelines for Grounded Theory Studies in Information Systems," Information systems journal (20:4), pp. 357-381.

van Angeren, J., Alves, C., and Jansen, S. 2016. "Can We Ask You to Collaborate? Analyzing App Developer Relationships in Commercial Platform Ecosystems," Journal of Systems and Software (113), pp. 430-445.