Wirtschaftsinformatik Proceedings 2015 Wirtschaftsinformatik

3-5-2015

How Informal Control Modes affect Developers' Trust in a Platform Vendor and Platform Stickiness

Tobias Goldbach

Alexander Benlian

How Informal Control Modes affect Developers' Trust in a Platform Vendor and Platform Stickiness

Tobias Goldbach, Alexander Benlian

Technische Universität Darmstadt, Chair of Information Systems and E-Services,
Darmstadt, Germany
{goldbach, benlian}@ise.tu-darmstadt.de

Abstract.
Although IS literature is quite rich on analyzing control modes in inter-organizational relationships, there exists quite sparse empirical work on the effects of control modes on the relationship between a platform vendor and third-party developers in software-based platform ecosystems. Drawing on IS control literature and trust theory, we conducted an online survey with 125 game developers of the online gaming platform Steam, in which we examined the perception of clan and self-control mechanisms and how they influence developers' trust in the platform vendor and their intention to stick with the platform. We demonstrate that clan control positively impacts both trust and stickiness, while self-control only affects the latter. Further, our results show that developers' trust in the platform vendor serves as a mediator explaining the mechanisms of why clan control positively affects platform stickiness.

Keywords: Software Platforms, Informal Control Modes, Trust, Stickiness

1 Introductions

By empowering third-party developers to complement functionalities and applications to a core software product, software-based platforms have become a promising and growing approach in the software industry to develop and distribute software [1]. The ecosystem of such platforms includes the platform vendor, numerous third-party developers and the platform users. Although such platforms exist since the early days of personal computing, where developers provide software and add-ons for operation systems and system software, like Microsoft Windows and its office-suite [2, 3], new and innovative software-platforms have emerged in the mobile and online context [1].

Apple for example, a platform where developers can develop and sell mobile apps, has experienced a massive overall growth in recent years. This is reflected in six million registered app-developers, more than one million active apps and revenues of 10 billion dollars from their App Store sales in 2013 [4, 5]. The game distribution and social network platform Steam is on its way to replace the old game retail selling system, while competing platforms struggle to grow in the same market [6]. These mobile and online platforms and their ecosystems have already been acknowledged several times as a significant novel research opportunity for IS research (e.g., [1, 7]).

12th International Conference on Wirtschaftsinformatik,
March 4-6 2015, Osnabrück, Germany

Goldbach, T.; Benlian, A. (2015): How Informal Control Modes affect Developers' Trust in a Platform Vendor and Platform Stickiness, in: Thomas. O.; Teuteberg, F. (Hrsg.): Proceedings der 12. Internationalen Tagung Wirtschaftsinformatik (WI 2015), Osnabrück, S. 1328-1342

Control theory [8] has often been invoked in order to analyze and explain the coordination and alignment between two parties, and control modes are an integral part of platform governance [1]. While formal control (i.e., process and outcome control) relies on written contracts and regular observations, informal control (i.e., clan and self-control) is exercised with social or people strategies by cultivating shared interests and values and by encouraging self-regulation. Informal control is of primary interest to our study, given that the relationship between a platform vendor and third-party developers is less hierarchical and more dependent compared to traditional software development contexts, which makes the implementation of informal control more promising than formal control [9-11]. There are two main reasons for this:

First, the relationship of a platform vendor and third-party developers is less hierarchical, because interests and goals, like increasing the platforms' customer base and generating revenues, are not necessarily incongruent. Therefore, the intention to engage in opportunistic behaviors is less prominent and the need of strict supervision and alignment from one side less necessary [11]. Further, software platforms are more open [12] and the myriad of third-party developers would make it prohibitively costly and time consuming to command and supervise each single software project.

Second, both parties rely heavily on each other in order to perform successfully [13]. On the one hand, a platform provides resources and tools for development and distribution of third-party software, and offers access to the platform's customer base [14]. Developers hence have to trust the platform vendor that these crucial resources will continuously be provided. On the other hand, the success of a platform is prominently determined by having enough developers offering applications on one side to attract customers on the other and vice versa [15]. Consequently, platform vendors depend on developers' intention to become loyal members of the platform ecosystem. Given that trust and loyalty are essential parts of relationship quality [16] and an enduring and cooperative relationship between a platform vendor and developers is crucial for a healthy ecosystem [17], we will further focus on developers' trust in a platform vendor and on stickiness of a platform.

Two major research gaps on control modes and trust are particular noteworthy. Previous studies have widely examined control modes within organizations (e.g., [18, 19]) and between organizations (e.g., [20, 21]) and have largely focused on formal control modes (e.g., [20, 22]). However, there is still little understanding about the effects of informal control modes in online platform contexts. Further, previous work on the interplay of control modes–either formal or informal–and trust have found conflicting results about whether the constructs are negatively or positively connected (e.g., [23, 24]). Studies have, for example, analyzed the interplay of control, trust and risk in strategic alliances [23], the effects of ideology tenets on team effectiveness through trust in open source development [25] and the coevolution of trust, control and learning in joint ventures [26]. Despite significant research in this area, effects of informal control modes on trust in platform settings has remained underexplored [1].

The purpose of this paper is therefore to understand how informal control modes, i.e. clan control and self-control, affect third-party developers' trust in the platform vendor and stickiness of a platform. In the next section, we lay out the theoretical background of this paper, followed by our research hypotheses. We describe our research methodology in section 4 and present the results of an online survey with 125 game developers of the Steam platform in section 5. In section 6, we discuss main findings, implications and limitations, and conclude the paper with final remarks.

2 Theoretical Background

2.1 Trust in a Platform Vendor

Despite widespread agreement of the importance of trust in human interactions, previous research has defined and conceptualized trust in a variety of ways [27, 28]. In IS literature, a commonly used definition is the willingness of one party (the trustor) to be vulnerable to the actions of another party (the trustee), based on the expectation that the other party will perform a particular important action [29]. While most conceptualizations aim at trust in interpersonal relations, trust in online contexts is often based on trust in organizations [30]. For instance, customers may trust an online shopping website for purchasing products, or third-party developers place their trust in a platform vendor or in a software platform as a collective actor.

Three dimensions are constantly used in e-commerce literature for characterizing trust, i.e. the beliefs in the competence, integrity and benevolence of the trustee [28, 29]. In a platform context, competence refers to the developer's perception of the ability and knowledge of a platform vendor to perform needed actions and to show expected behaviors. Developers for example expect from a platform vendor to have knowledge about providing a technical infrastructure, to have the ability to attract customers and to promote the platform with its products. Integrity is the perception that a platform vendor will act ethically and adhere to announced or expected principles and promises. Developers for example believe that a platform vendor provides promised resources and development tools and places products in the platform's store equitably. Finally, benevolence refers to the beliefs that a platform vendor will generally *do good* and will not behave merely opportunistic. A platform vendor may, for example, demonstrate benevolence by showing empathy towards developers' concerns and needs, which in turn may reduce uncertainty on the developers' side.

2.2 Platform Stickiness

As a crucial platform-related outcome, third-party developers' loyalty towards a platform, i.e. their willingness to constantly participate in and stick with a platform over time, has been shown to maintain a healthy platform ecosystem [17]. Accordingly, platform stickiness is a crucial performance factor that indicates a persistent and loyal relationship between a platform and its developers. Based on Zott et al. [31], we define platform stickiness as the ability of a platform to retain third-party developers on their platform, which is reflected in developers' intention to continue contributing to a platform ecosystem. When developers continue to develop and update apps for the platform and to engage in the platform community, they usually contribute to the platform's productivity, robustness and innovative capacity [15, 17]. By contrast, when developers switch to a rival platform, they are likely to destabilize the abandoned platform ecosystem through the drain of knowledge and expertise. Even worse, rival platforms and their customer bases may benefit from the influx of developers and their apps, which may ultimately tip network effects in favor of rival platforms.

2.3 Platform Control Modes

According to control literature (e.g., [8, 32]), control refers to a controller's attempts by which he or she influences an individual or an individual group (the controlee) to act in accordance with the objectives of the controller. Control modes can be subdivided into formal control and informal control. Formal control, can be further split into output and process (or also called behavior) control. With output control, controllers pre-specify output requirements and performance targets as objectives, which are then monitored, evaluated and accordingly rewarded. By contrast, under process control, specific procedures and methodologies are pre-determined, while characteristics of outcomes are free to be chosen by the controlee.

Informal control is categorized into clan control and self-control [8, 32]. Through clan control, controllers are able to minimize the differences between controlee's goals and their own strategies, by promulgating common values, beliefs, and behaviors. Members of a group of controlees therefore usually commit themselves to these mutual values and beliefs and thus often adopt comparable processes with similar performance outcomes. Clan control may lead to a sense of cohesiveness among clan members, which in turn facilitates self-regulatory mechanisms on a group level (e.g., monitoring, evaluating and correcting each other) in accordance with shared values and beliefs. In terms of self-control, controllers encourage controlees to exercise self-management and self-regulation by granting autonomy, providing tools for self-management and structuring the work environment appropriately. Controlees can organize themselves, mostly make their own decisions and evaluate themselves based on their own performance. Thus, control lies in the hands of each controlee.

Currently, a variety of informal control mechanisms are observed on software-based platforms [1]. As an example for clan control, Apple positions its products, primarily through its promotional campaigns and branding strategies, as being innovative, well designed and for people with a trendy lifestyle, which is reflected in the design, usability and user experience of Apple products. Customers and third-party developers mainly share these norms and values, which in turn manifest themselves in the high-quality designs of third-party apps. Conversely, design regulations are mostly absent in Google's Android ecosystem and therefore developers individually decide on the look-and-feel of their apps, which is an example for self-control [11].

3 Hypotheses Development

We first develop the hypotheses related to the effects of clan and self-control on trust in the platform vendor (H1) and platform stickiness (H2), followed by hypotheses related to the mediating role of trust (H3). The research model is shown in Figure 1.

3.1 The Effects of Informal Control Modes on Trust in Platform Vendor

By definition, clans rely heavily on trust [33]. With clan control mechanisms, controllers are able to minimize the differences between controlee's goals and their own strategies, by promulgating common values, beliefs, and behaviors. Carefully selecting and socializing members usually serves in identifying with shared beliefs and in adopting comparable goals and behaviors, which in turn often results in a group of individuals sharing these commonalities [8]. Strong clan cultures are likely to facilitate cohesiveness which in turn strengthens the willingness of controlees to trust the other party's benevolence [33]. Further, being certain that a partner will pursue shared goals instead of opportunistic interest usually leads to a higher confidence in the partner and the partners integrity [23]. Applied to the platform context, nurturing common platform values and cultivating shared goals might in turn also cultivate developers' trust in the platform vendor.

Self-control is based on self-management and self-regulating mechanisms. In order to implement self-control, controllers are granting autonomy to controlees without applying any other forms of regulation [8]. In view of the absence of regulation, granting autonomy requires a certain degree of trust in the controlees' intention to behave accordingly to the controller's interests and goals [21]. Given that trust is seen as a reciprocal concept, more trust by one party usually results in more trust by the other party [23]. Therefore, controllers being trusted to regulate themselves will increase their intention to trust the controlee. Against this backdrop, granting autonomy to developers in a platform setting may also lead to more trust in the platform vendor.

These arguments suggest that third-party developers will place more trust in a platform vendor, if higher levels of both clan and self-control are perceived. Accordingly, we hypothesize that: ***H1a:*** *Clan Control is positively related to Trust in Platform Vendor.* ***H1b:*** *Self Control is positively related to Trust in Platform Vendor.*

3.2 The Effects of Informal Control Modes on Platform Stickiness

Clan control is exercised to minimize differences between two parties in terms of their interests and goals by declaring common values, beliefs and behaviors [8]. On the one hand, clan control is therefore facilitating a sense of belonging and togetherness. Working for and with people with similar interests and mindsets and with common goals may lead to a homelike feeling along with loyalty and commitment to the group or company [23, 33]. On the other hand, shared visions and values are reducing the possibility of incongruent preferences and goals. Therefore, if a controlee agrees with the controllers' goals and beliefs and acts in similar ways, the risk of producing variances in outcomes of importance to the controlee is minimized [23, 33]. This lower risk and higher sense of belonging is in turn likely to translate into a higher intention to stay with the group. Applied to the platform context, if developers share similar values and goals with a platform, they may perceive lower risks and higher cohesiveness which in turn may result in a higher intention to keep contributing to the platform

As mentioned above, self-control is based on self-regulation through granting autonomy to controlees. According to self-determination theory [34], human beings' perceptions of autonomy are related to higher intrinsic motivation, more creativity, higher self-esteem, less tension and a higher work satisfaction. Previous empirical research has shown that employees feel more satisfied with their working place in an organization, when they are not under permanent scrutiny, but feel more independent in pursuing their personal goals [35]. This higher autonomy under self-control in platform settings is also likely to translate into higher loyalty with and stickiness to the platform, given that developers feel more self-determined, satisfied and more committed to a platform. In contrast, reducing developers' autonomy may lead to lower intrinsic motivation, less satisfaction and thus in a lower intention to stick with the platform. Consequently, we suggest that third-party developers will have a higher intention to stick with a platform under a higher perception of clan control and self-control. ***H2a:*** *Clan Control is positively related to Platform Stickiness.* ***H2b:*** *Self-Control is positively related to Platform Stickiness.*

3.3 The Mediating Effect of Trust in Platform Vendor

Given the high importance and prevalence of trust for enduring relationships in platform ecosystems, we suggest that developers' trust in a platform vendor is one essential mediation mechanism explaining the effects of clan and self-control on developers' intention to stick with a platform. As studies have shown, trusting beliefs are transformed into perceptions that a trustee has attributes that can be relied on and that are beneficial for the trustor [27]. Trust frees the trustor from worrying about potential risks of doing business with the trustee and about the anxiety of being exploited. The resulting sense of secureness is likely translated into a comfortable feeling with and loyalty towards the other party along with an enjoyment in conducting business together [36]. A reduced perception of risks, caused by a higher level of trust to a business partner, strengthens the loyalty towards this partner [23]. If a business partner is trusted, additional exhaustive monitoring of changes in rules or procedures becomes unnecessary and therefore saves costs, time and energy [36] which also enhance the intention to keep cooperating with a partner. Moreover, according to e-commerce literature [37], trust in an online-seller and the resulting satisfaction with the seller usually leads to repurchase intentions. Applied to the platform context, if developers place trust towards a platform and believe in the platform's reliability, they may also perceive less risk, feel more secure and therefore enjoy working for and with the platform, which in turn may lead to a higher intention to contribute to the platform.

These arguments and previous empirical evidence suggest that a higher level of trust in the platform vendor will lead to a higher intention to stick with the platform. Further, given that fostering shared norms and values as well as encouraging self-management may stimulate trust, we suggest that trust is a major explanatory factory of why clan control as well as self-control leads to higher platform stickiness. In other words, clan control and self-control will lead to a higher stickiness with the platform *because* of higher trust in the platform vendor. ***H3a:*** *Effects of Clan Control on Platform Stickiness are mediated by Trust in Platform Vendor.* ***H3b:*** *Effects of Self-Control on Platform Stickiness are mediated by Trust in Platform Vendor.*

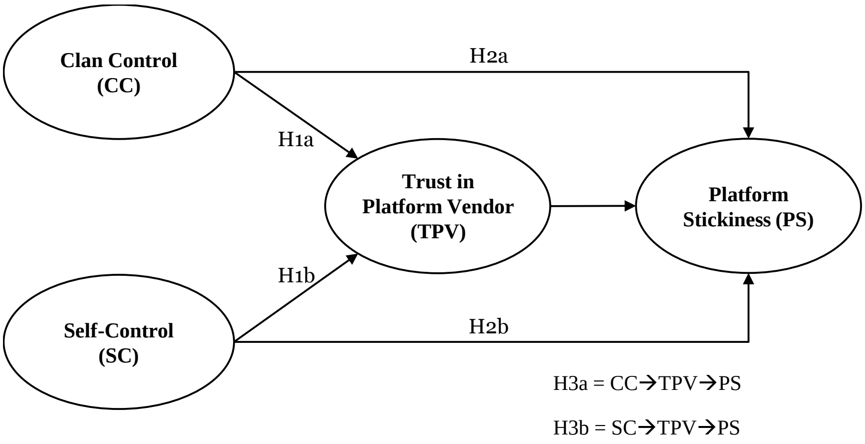

Fig. 1. Research Model and Hypotheses

4 Research Methodology

4.1 Data Collection and Sample Description

In order to verify our hypotheses, we created an online survey instrument to collect empirical data with developers of the above mentioned online gaming platform Steam. Steam[1] is a typical two-sided software platform operated by Valve Corporation. The platform has more than 75 million registered users worldwide [6] and more than 1.000 developers (including professional and freelance game developers) who develop, publish and distribute over 3.300 games and even more game modifications (called mods) via Steam. Users are provided with installation and automatic updating of games on multiple computers, and community features such as friends lists, cloud saving, and chat functionality. Steam has introduced an initiative to support developers (called Steam Greenlight) and tools to simplify the developing and distribution process (called Steam Works). With the freely available application programming interface (API), developers can integrate many of Steam's functions, including networking and matchmaking, in-game achievements, micro-transactions, and support for user-created content. In order to develop, publish and distribute games via Steam, game developers have to follow specific procedures and adhere to specific rules and conditions that serve as typical control mechanisms by the platform.

To obtain the contact details of Steam developers, we collected the names of all third-party developers published on the Steam platform. After deleting duplicate names and no longer existing developers, we gathered e-mail addresses or identified contact forms and Facebook pages by visiting developers' websites. Via these channels, a link to the online survey questionnaire was sent to 870 developer studios,

[1] We ran a web crawler on http://store.steampowered.com in July 2014 to get the latest developer and game statistics.

along with a letter outlining the purpose of our research, ensuring confidentiality and anonymity of the responses, and asking to forward the survey to the lead developer who could serve as key informant to provide answers to our survey questions [38]. As a reward for participation, participants were entered into a lottery for a tablet, an e-book reader or Amazon gift cards. Furthermore, participants received a management report with the core results of the study.

After two follow-up reminders via e-mail, and after deleting responses because of missing and inconsistent values, we received 125 valid responses, resulting in a response rate of 14.4%. All participating developers were lead developers and represented a distinct development company. Nonresponse bias was assessed by verifying that early and late respondents were not significantly different [39]. Both samples were compared based on their sociodemographics and responses to principal constructs in the study, and t-tests between the means of the early (first 50) and late (last 50) respondents showed no significant differences ($p > 0.05$). Overall, these findings indicate that nonresponse bias is unlikely to be a major issue in this study.

The majority of developers in our sample were freelance developers (73.4%) and were located in the US (34.5%), Canada (10.3%) and the UK (5.2%). Most developer companies had been founded after 2007 (69.9%) and employed between 10 and 49 people (70.4%). Most games published by the developing companies in our sample were indie games (70.4%), action (54.8%) and adventure (40%). The prices vary mostly between $5US and $20US. Almost 70% of the developers in our sample had more than 5 years of game development experience, while 67% had more than 1 year of experience with developing for Steam. Descriptive statistics are shown in Table 1.

Table 1. Sample Descriptives (N = 125)

Number of employees in company:		Employment of developers:	
< 10	0.0 %	Professional game developer	17.4 %
10 – 49	70.4 %	Freelance developer	73.4 %
50 – 250	24.3 %	Hobbyist developer	7.0 %
> 250	5.3 %	Other	2.6 %
Price range of Steam games:		Years of experience in game development:	
< $5	10.4	less than 1 year	1.7 %
$5 – $10	40.0	1 to 3 years	11.3 %
$11 – $20	38.3	3 to 5 years	17.4 %
> $21	11.3	5 to 7 years	17.4 %
Game genres (multi-selection):		7 to 9 years	14.8 %
Action	54.8 %	More than 9 years	37.4 %
Adventure	40.0 %	Years of experience of developing for Steam:	
Strategy	38.3 %	less than 1year	33.0 %
Roleplaying	25.2 %	1 to 3 years	36.5 %
Indie	70.4 %	3 to 5 years	19.1 %
Casual	34.8 %	5 to 7 years	8.7 %
Simulation	23.5 %	7 to 9 years	1.7 %
		More than 9 years	0.9 %

4.2 Measurement of Variables and Measurement Model Assessment

All constructs in the survey instrument were based on established measures from existing scales in IS literature and adapted to the study's context with minor wording changes. Measures for clan control and self-control were adopted from Tiwana and Keil [21], based on control measures of Kirsch et al. [19]. We focused the clan control measurement on the understanding of the Steam platform's goals, values, norms and rules in order to capture developers' perception of clan control execution. During our qualitative pre-test phase, and because of the specific characteristics of the Steam platform compared to organizational clan control, we found this measurement more suitable for the Steam context. Measures for platform stickiness were adopted from the website stickiness construct of Dahui et al. [40] and adjusted to the general platform context. Given that the measurement model of trust has often been misspecified in IS research [41, 42], we modeled trust as a second-order formative construct, following the guidelines provided by Jarvis et al. [43]. We based our trust conceptualization on the trusting belief construct of McKnight et al. [28]. We included one item for each subdimension of trust (i.e. competence, integrity and benevolence), which was found to be appropriate [41]. All the questionnaire items were measured using a 7-point-Likert scale, anchored at (1) = strongly disagree and (7) = strongly agree. All of the items of the constructs and their sources are shown in Table 2 in the Appendix.

For analyzing measurement properties of a model with second-order factors in PLS, we followed the recommendations of Wang and Benbasat [44], by first analyzing the measurement properties of all reflective constructs. Next, we replaced our first-order constructs with their latent variable scores provided by SmartPLS, which then allows us to analyze the structural paths of the model.

The psychometric properties of the reflective measurement models were assessed by examining content validity, convergent validity, and discriminant validity of all latent constructs. In order to ensure content validity, we developed questions for indicators from previous research and performed a qualitative pre-test to check for ambiguities. Convergent validity was evaluated using three criteria recommended by Fornell and Larcker [45]: (1) all measurement factor loadings must be significant and above the threshold value of .70; (2) the composite reliabilities should exceed .80; and (3) the average variance extracted (AVE) by each construct must exceed the variance due to measurement error for that construct (i.e., AVE should exceed .50).

All loadings of the reflective measurement items on their respective factors were significant ($p < .001$) and above .827. Composite reliabilities of the constructs were all above .937 and the values for AVEs for each construct ranged from .833 to .893. Therefore, all of our constructs fulfill these three criteria and met the norms for convergent validity. In addition, for satisfactory discriminant validity, the square root of the AVE from the construct should be greater than the variance shared between a construct and other constructs in the model [45]. Inter-correlation values between the latent variables are ranging between .303 and .601 and are therefore lower than the square root of the AVEs, providing strong evidence of discriminant validity. Hence, the constructs in our study represent concepts that are both theoretically and empirically distinguishable. In order to establish construct validity for our formative indicators, we relied on the approach of Ringle et al. [46] by ensuring that the indicator weights for formative constructs are roughly equal and all have significant t-values, which we were able to determine in SmartPLS for our formative items.

5 Data Analysis and Results

We used structural equation modeling using partial least squares (PLS) to test our research model in general, and our mediation hypotheses in particular. PLS allows for simultaneous testing of the measurement model (the psychometric properties of the measurement scales) and the estimation of the structural model (the strength and direction of the relationship between the variables). PLS has an added advantage over covariance-based methods (e.g., LISREL) in that (1) it maximizes the explained variance of endogenous variables in the structural model [47], which enables us to understand the amount of variance explained in the constructs, such as trust in platform vendor or platform stickiness, and (2) PLS does not make distributional assumptions for the data [47]. We used the software SmartPLS 2.0 M3 [48] and a two-step procedure to examine our hypotheses. In step 1, we analyzed the direct effects of clan control and self-control on trust in a platform vendor and platform stickiness. In step 2, we introduced trust in platform vendor as a mediating mechanism into the structural model (by adding the path from trust to stickiness), following the recommendations by Preacher and Hayes [49]. The final results of step 2 are shown in Figure 2.

The structural model in step 1 successfully explained a considerable portion of variance in trust in platform vendor ($R^2 = .279$) and platform stickiness ($R^2 = .393$). Concerning the path coefficients, we found positive and significant effects of clan control on trust in platform vendor ($\beta = .520$; $p < .001$) and on platform stickiness ($\beta = .510$; $p < .001$), which is in support of H1a and H2a. Further, self-control similarly had a positive and significant effect on platform stickiness ($\beta = .201$; $p < .05$), while there was no significant effect on trust in platform vendor ($\beta = .003$; $p > .05$). Therefore, H1b could be supported while H2b had to be rejected.

In step 2 of the PLS-SEM analysis, we assessed the results of the full path model including the impact of the mediator trust on platform stickiness. The variance explained in platform stickiness increased substantially ($R^2 = .450$) compared to the analysis in step 1, due to the significant positive impact of trust in platform vendor on platform stickiness ($\beta = .282$; $p < .05$). Clan control had a significant indirect effect on platform stickiness via trust in platform vendor ($\beta = .144$, $p < .01$), while the direct effect decreased but remained significant ($\beta = .358$, $p < .01$). Therefore, trust in platform vendor partially (by 29.8 %) mediated the relationship between clan control and platform stickiness. Although self-control had a significant direct effect on platform stickiness ($\beta = .192$; $p < .05$), we could not find a significant mediation effect on platform stickiness through trust, given that self-control had no significant impact on trust in platform vendor ($\beta = .003$; $p > .05$). Therefore, we had to reject H2a.

We also analyzed a full model including several control variables (i.e., developer employment, development experience, number of employees in company), but did not find any significant impact on trust or platform stickiness ($p > 0.5$).

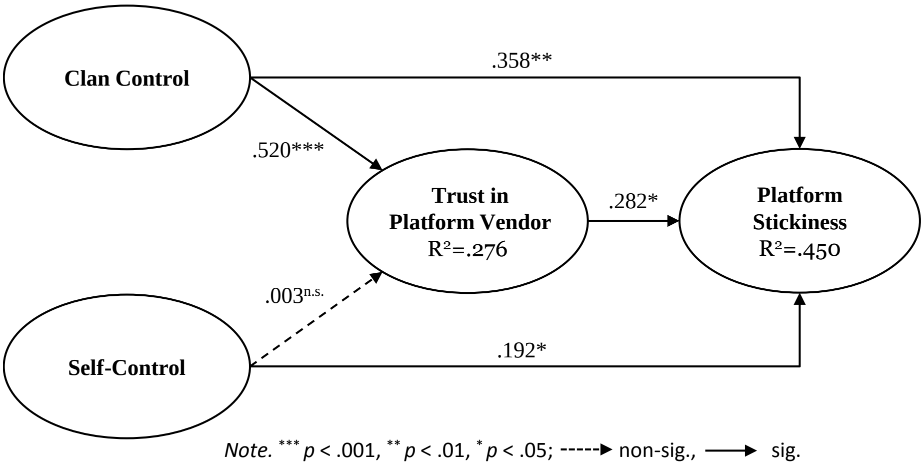
Clan Control
.358**
.520***
Trust in Platform Vendor R²=.276
.282*
Platform Stickiness R²=.450
.003n.s.
Self-Control
.192*
Note. *** p < .001, ** p < .01, * p < .05; -----► non-sig., ——► sig.

Fig. 2. Results of SEM Analysis

6 Discussion and Conclusion

Two key findings can be derived from this study. First, our findings show that clan control positively impacts both trust in platform vendor and platform stickiness, while self-control only affects the latter. Working autonomously may therefore not sufficiently flourish developer's trust towards a platform vendor. Second, our results show that developers' trust in a platform vendor serves as a mediator between clan control and platform stickiness. Given that trust and loyalty are essential parts of relationship quality [16] and that an enduring cooperative relationship between a platform vendor and third-party developers is crucial for a healthy successful platform [13, 17], our findings demonstrate the high importance of informal control in platform setting.

6.1 Theoretical and practical implications

From a theoretical standpoint, our study offers a deeper understanding of the effects of informal control modes on third-party developers' trust in a platform vendor and platform stickiness in software-platform settings. We show in our study that clan and self-control positively affects developers' intention to stick with a platform, and that clan control leads to higher developer's trust in the platform vendor. The latter is in line with related findings in an open source setting [25]. Further, trust positively affects developers' intention to stick with the platform and therefore serves as an explanation of why clan control positively affects platform stickiness. Our study establishes the importance of informal control as a driver for third-party developers' trust and loyalty in platform settings. It thus contributes to IS control literature by studying informal control in a yet underexplored context and advances platform governance literature by highlighting the importance of informal control as central governance devices to shape trust-building and loyalty on platforms.

Our results have also important implications for practice. First, platform vendors may increasingly focus on cultivating trust in their relationship with third-party developers by keeping in mind that satisfying developers' needs and expectations about the platform might strengthens their trust towards the platform. Platform vendors might, for example, continuously provide and expand prior agreed upon resources for development and distribution, along with solving technical difficulties and promoting the platform products. Second, implementing clan control mechanisms specifically affects developer's trust towards the platform. Platform vendors may promulgate and reinforce common values, beliefs and behaviors by publishing regular company newsletters and best practice solutions, and by arranging socialization processes and ceremonies, like developer-centric events and conferences (e.g., Apple's yearly Developers Conference in San Francisco). Further, encouraging self-management by granting more autonomy and structuring the work environment appropriately, platform vendors may additionally encourage developers to exert self-control which additionally contributes to developer's stickiness with the platform.

6.2 Limitations and Further Research

As a concluding reflection, we point out the main limitations of our study. First, the results of our study are based on a platform in an online gaming context. Such platforms may differ from mobile or social networking platforms, regarding implemented informal control modes and the interactions between platform vendors and developers. Additionally, because the sample consisted mainly of freelance developers with indie games, our results may not be representative for professional game development studios. Hence, issues of generalizability may exist. Future studies should focus on additional software-based platforms, like Apple's App Store or Google's Android ecosystem, with a more representative mix of hobbyist and professional developers to strengthen external validity. Second, we based our measurement of clan control mainly on the scope of understanding shared values, goals and rules. Future research could examine a broader view of clan control and community behaviors as well as explicitly examining the antecedents and impacts of sub-dimensions of trust [28] in order to provide a richer understanding of the role of trust and clan control in platform contexts. Third, future work may extend our research model by including other factors. We focused our study on informal control modes, while the effects of formal control modes (process and output) [8] may also lead to promising results. Likewise, other factors affecting cooperative relationships (e.g., empathy, security or autonomy) and other developer and platform-related outcomes (e.g., application quality, development costs or user count) may also enhance our research model.

6.3 Conclusion

How can platform vendors create an enduring and cooperative relationship with third-party developers in order to work towards a successful platform ecosystem? By integrating control literature and trust theory, results of our survey study suggest implementing clan and self-control on platforms cultivates developers' trust in a platform vendor and stickiness with the platform. We believe that examining control modes as well as trust in platform settings is a rich avenue for future research, especially given that hitherto under-researched informal control modes are gaining in importance.

References

1. Tiwana, A., Konsynski, B., Bush, A.A.: Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics (Research Commentary). Information Systems Research 21, 675-687 (2010)
2. Boudreau, K.J.: Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. Organization Science 23, 1409-1427 (2012)
3. Eisenmann, T., Parker, G., Van Alstyne, M.: Platform Envelopment. Strategic Management Journal 32, 1270-1285 (2011)
4. Apple, www.apple.com/pr/library/2014/01/07App-Store-Sales-Top-10-Billion-in-2013.
5. Macstories, www.macstories.net/news/the-numbers-from-apples-wwdc-2013-keynote/.
6. Techradar, http://www.techradar.com/news/gaming/steam-rises-again-to-75-million-active-users-1215529
7. Wareham, J., Fox, P.B., Cano Giner, J.L.: Technology Ecosystem Governance. Organization Science Published online in Articles in Advance 13 Mar 2014, (2014)
8. Kirsch, L.S.: Portfolios of Control Modes and IS Project Management. Information Systems Research 8, 215-239 (1997)
9. Goldbach, T., Kemper, V., Benlian, A.: Mobile Application Quality and Platform Stickiness under Formal vs. Self-Control — Evidence from an Experimental Study. Thirty Fifth International Conference on Information Systems, Auckland, New Zealand (2014)
10. Kirsch, L.J., Ko, D.-G., Haney, M.H.: Investigating the Antecedents of Team-Based Clan Control: Adding Social Capital as a Predictor. Organization Science 21, 469-489 (2010)
11. Tiwana, A.: Platform Ecosystems: Aligning Architecture, Governance, and Strategy. Morgan Kaufmann, Burlington, Massachusetts (2014)
12. Benlian, A., Hilkert, D., Hess, T.: How open is this platform? The meaning and measurement of platform openness from the complementors’ perspective. Journal of Information Technology (forthcoming), (2015)
13. Gao, L.S., Iyer, B.: Analyzing Complementarities Using Software Stacks for Software Industry Acquisitions. Journal of Management Information Systems 23, 119-147 (2006)
14. Ceccagnoli, M., Forman, C., Huang, P., Wu, D.J.: Cocreation Of Value in a Platform Ecosystem: The Case of Enterprise Software. MIS Quarterly 36, 263-290 (2012)
15. Economides, N., Katsamakas, E.: Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. Management Science 52, 1057-1071 (2006)
16. Grover, V., Cheon, M.J., Teng, J.T.C.: The effect of service quality and partnership on the outsourcing of information systems functions. Journal of Management Information Systems 12, 89-116 (1996)
17. Iansiti, M., Levien, R.: Strategy as Ecology. Harvard Business Review 82, 68-81 (2004)
18. Henderson, J.C., Lee, S.: Managing I/S Design Teams: A Control Theories Perspective. (Information System). Management Science 38, 757-777 (1992)
19. Kirsch, L.J., Sambamurthy, V., Dong-Gil, K., Purvis, R.L.: Controlling Information Systems Development Projects: The View from the Client. Management Science 48, 484-498 (2002)

20. Gregory, R.W., Beck, R., Keil, M.: Control Balancing In Information Systems Development Offshoring Projects. MIS Quarterly 37, 1211-A1214 (2013)
21. Tiwana, A., Keil, M.: Control in Internal and Outsourced Software Projects. Journal of Management Information Systems 26, 9-44 (2009)
22. Cardinal, L.B.: Technological Innovation in the Pharmaceutical Industry: The Use of Organizational Control in Managing Research and Development. Organization Science 12, 19-36 (2001)
23. Das, T.K., Teng, B.-S.: Trust, Control, and Risk in Strategic Alliances: An Integrated Framework. Organization Studies 22, 251-283 (2001)
24. Gallivan, M.J., Depledge, G.: Trust, control and the role of interorganizational systems in electronic partnerships. Information Systems Journal 13, 159-190 (2003)
25. Stewart, K.J., Gosaim, S.: The Impact of Ideology on Effectiveness in Open Source Software Developement Teams. MIS Quarterly 30, 291-314 (2006)
26. Inkpen, A.C., Currall, S.C.: The Coevolution of Trust, Control, and Learning in Joint Ventures. Organization Science 15, 586-599 (2004)
27. Gefen, D., Karahanna, E., Straub, D.W.: Trust and TAM in Online Shopping: An Integrated Model. MIS Quarterly 27, 51-90 (2003)
28. McKnight, D.H., Choudhury, V., Kacmar, C.: Developing and Validating Trust Measures for e-Commerce: An Integrative Typology. ISR 13, 334-359 (2002)
29. Mayer, R.C., David, J.H., Schoorman, F.D.: An Integrative Model of Organizational Trust. Academy of Management Journal 20, 709-734 (1995)
30. Janowicz, M., Noorderhaven, N.: Levels of Inter-Organizational Trust: Conceptualization and Measurement. In: Reinhard, B., Zaheer, A. (eds.) Handbook of Trust Research, vol. 1st ed. Edward Elgar Publishing, Inc., Cheltenham, UK (2006)
31. Zott, C., Amit, R., Donlevy, J.: Strategies for Value Creation in E-Commerce: Best Practice in Europe. European Management Journal 18, 463-475 (2000)
32. Ouchi, W.G.: A Conceptual Framework for the Design of Organizational Control Mechanisms. Management Science 25, 833-848 (1979)
33. Ouchi, W.G.: Markets, Bureaucracies, and Clans. Administrative Science Quarterly 25, 129-141 (1980)
34. Deci, E.L., Ryan, R.M.: The support of autonomy and the control of behavior. Journal of Personality and Social Psychology 53, 1024-1037 (1987)
35. Hackman, J.R., Oldham, G.R.: Motivation Through the Design of Work: Test of a Theory. Organizational Behavior & Human Performance 16, 250-279 (1976)
36. Sun, H.: Sellers' Trust and Continued Use of Online Marketplaces. Journal of the Association for Information Systems 11, 182-211 (2010)
37. Garbarino, E., Johnson, M.S.: The Different Roles of Satisfaction, Trust and Commitment in Customer Relationships. Journal of Marketing 63, 70-87 (1999)
38. Kumar, N., Stern, L.W., J.C., A.: Conducting interorganizational research using key informations. The Academy of Management Journal 36, 1633-1651 (1993)
39. Armstrong, J.S., Overton, T.S.: Estimating Nonresponse Bias in Mail Surveys. Journal of Marketing Research 14, 396-402 (1977)
40. Li, D., Browne, G., Wetherbe, J.: Why Do Internet Users Stick with a Specific Web Site? A Relationship Perspective. International Journal of Electronic Commerce 10, 105-141 (2006)
41. Söllner, M., Hoffmann, A., Maximilian Hirdes, E., Rudakova, L., Leimeister, S., Leimeister, J.M.: Towards a Formative Measurement Model for Trust. BLED Proceedings (2010)

42. Söllner, M., Leimeister, J.M.: What We Really Know About Antecedents of Trust: A Critical Review of the Empirical Information Systems Literature on Trust. Psychology of Trust: New Research, D. Gefen, Verlag/Publisher: Nova Science Publishers, 2013 (2013)
43. Jarvis, C.B., MacKenzie, S.B., Podsakoff, P.M.: A Critical Review of Construct Indicators and Measurement Model Misspecification in Marketing and Consumer Research. Journal of Consumer Research 30, 199-218 (2003)
44. Wang, W., Benbasat, I.: Attributions of trust in decision support technologies: A study of recommendation agents for e-commerce. Journal of Management Information Systems 24, 249-273 (2005)
45. Fornell, C., Larcker, D.F.: Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. Journal of Marketing Research 18, 39-50 (1981)
46. Ringle, C.M., Sarstedt, M., Straub, D.W.: Editor's comments: A critical look at the use of PLS-SEM in MIS quarterly. MIS Quarterly 36, iii-xiv (2012)
47. Chin, W.W.: The Partial Least Squares Approach for Structural Equation Modelling. Lawrence Erlbaum Associates, Hillsdale, NJ (1998)
48. Ringle, C.M., Wende, S., Will, A.: Smartpls 2.0 (M3) Beta. Hamburg Hamburg (2005)
49. Preacher, K.J., Hayes, A.F.: Asymptotic and Resampling Strategies for Assessing and Comparing Indirect Effects in Multiple Mediator Models. Behavior Research Methods 40, 879-891 (2008)

Appendix

Table 2. Measurement Items

Constructs	Indicators	Source
Clan Control (CC)	Do you clearly understand …	Based on [19, 21]
	... the goals of the Steam platform.	
	... the values of the Steam platform.	
	... the norms and rules of the Steam platform.	
Self-Control (SC)	Do you autonomously ...	[19, 21]
	... set specific goals without the platform owner's involvement.	
	... set release schedules for your games without the platform owner's involvement.	
	... define development and publishing procedures without the platform owner's involvement.	
Trust in Platform Vendor (TBV)	Regarding the platform vendor …	[28]
	… I believe that Steam is capable in solving technical problems.	
	… I believe Steam would keep its agreements.	
	… Steam would do its best to market and sell my games.	
Platform Stickiness (PS)	I am planning to keep using the Steam platform for game development and distribution in the future.	[40]
	I would not hesitate to use the Steam platform for game development and distribution in the future.	
	I am very likely to use the Steam platform for game development and distribution in the future.	