

# Chapter 2: Control Mechanisms on Software Platforms

**Title:** Kontrollmechanismen auf Softwareplattformen

**Authors:** Goldbach, Tobias, Technische Universität Darmstadt, Germany  
Benlian, Alexander, Technische Universität Darmstadt, Germany

**Published in:** HMD - Praxis der Wirtschaftsinformatik, 52 (3), 347-357

## Abstract

Software platforms, like the Apple App Store or Google Play, depend heavily on external app-developers who regularly develop and update apps for the platform. As part of platform governance, control theory can be invoked in order to describe and analyze the coordination between a platform vendor and app-developers. In this article we describe and categorize existing control mechanisms on software platforms and point out that Apple and Google largely exercise informal control (i.e., self- and clan control) and less formal control (i.e., input, process and output control). Additionally, in a lab experiment, we found evidence that self-control has more positive effects on the quality of developed apps and on developers' loyalty to the platform compared to formal control. The article demonstrates and suggests that platform vendors should increasingly focus on implementing informal control mechanisms.

**Keywords** Software Platforms, Platform Governance, Control Mechanisms, Formal and Informal Control

**Zusammenfassung**

Software-Plattformen, wie Apple App Store oder Google Play, sind stark von externen App-Entwicklern abhängig, die für Nutzer der Plattformen regelmäßig Apps erstellen und weiterentwickeln. Um die Koordination zwischen Plattformbetreiber und App-Entwickler zu beschreiben und zu analysieren, kann die Kontrolltheorie als Teil der Plattform-Governance herangezogen werden. In diesem Artikel werden eingesetzte Kontrollmechanismen auf Software-Plattformen zusammengetragen und klassifiziert. Wir legen dar, dass Apple und Google verstärkt auf informale Kontrollmechanismen (d. h. Selbst- und Klankontrolle) setzen und weniger auf formale Mechanismen (d. h. Zugangs-, Prozess- und Ergebniskontrolle). In einem Laborexperiment konnten wir außerdem aufzeigen, dass Selbstkontrolle im Vergleich zu formaler Kontrolle in Bezug auf die Qualität der entwickelten Apps und die Loyalität der Entwickler gegenüber der Plattform einen starken positiven Effekt ausüben kann. Der Artikel zeigt, dass Plattformbetreiber vermehrt informale Kontrollmechanismen einsetzen und einsetzen sollten.

**Schlüsselwörter:** Software-Plattformen, Plattform-Governance, Kontrollmechanismen, Formale und informale Kontrolle

## 2.1 Software-Plattform Ökosysteme

Software-Plattformen im Onlinekontext haben in den letzten Jahren die Software-Industrie bedeutend verändert. Unternehmen öffnen ihre Unternehmensgrenzen und ermöglichen externen Entwicklern den Software-Kern einer Unternehmensplattform um Funktionalitäten, Module oder Applikationen (Apps) zu erweitern (Tiwana et al. 2010). Die Unternehmen setzen dabei auf Wissen, Fähigkeiten und neue Ideen der externen Ressourcen, um schnellere Innovationszyklen zu generieren und um durch größeres Wachstum in den dynamischen Märkten konkurrenzfähig zu bleiben (Boudreau 2012). Apples App Store und Googles Android-Plattform werden durch externe Entwickler um Apps erweitert, Mozilla Firefox und Google Chrome um Browsermodule und auch kleinere Plattformen im TV-Sektor bieten externe Apps an. Um solche Plattformen herum entsteht ein eigenes Software-Ökosystem mit verschiedenen Akteuren, namentlich dem Plattformbetreiber, den externen Entwicklern und den Nutzern der Plattform (vgl. Figure 2-1) (Tiwana et al. 2010). Die verschiedenen Akteure interagieren miteinander und bringen eigene Vorstellungen und Ziele in das Ökosystem ein. Durch die offenen Strukturen entstehen neue Dynamiken und Chancen für Plattformbetreiber, aber auch neue Risiken und Herausforderungen.

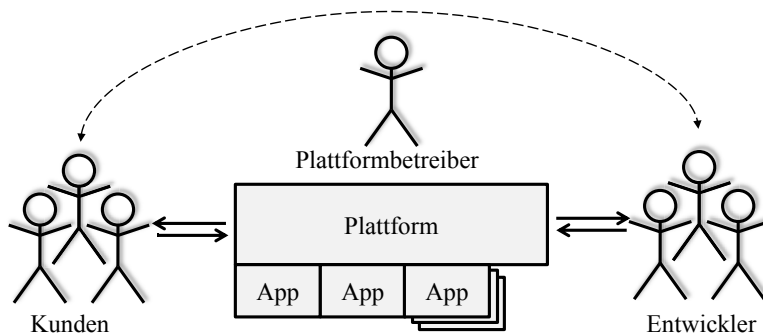


Figure 2-1: Akteure eines Software-Plattform Ökosystems

In der traditionellen Softwareentwicklung stellen Auftraggeber meist spezifische Vorgaben an Entwicklungsergebnisse und -prozesse, welche sie basierend auf Zwischenergebnissen und Protokollen überwachen und beeinflussen. Auf Software-Plattformen ist durch externe Entwickler und direkten Kundenkontakt die Softwareentwicklung allerdings anderen Dynamiken ausgesetzt (Tiwana 2014): Zunächst ist das Verhältnis zwischen Plattformbetreiber und Entwicklern weniger hierarchisch und weniger verbindlich. Entwickler sind nicht fest im Unternehmen des Plattformbetreibers eingebunden und können überwiegend eigene Entscheidungen treffen. Interessen und Ziele von Plattformbetreiber und

externen Entwicklern sind nicht in allen Bereichen unterschiedlich. Meist zielen beide Akteure darauf ab, Kunden zufriedenzustellen und Umsätze zu erwirtschaften, z. B. durch innovative und hochwertige Apps. Letztlich ist es für Plattformbetreiber äußerst kosten- und zeitaufwendig jedes einzelne externe Projekt gewissenhaft zu überwachen und zu beeinflussen.

Für Plattformbetreiber ist es dennoch bedeutsam, die Dynamiken in Plattform-Ökosystemen zu steuern und zu kontrollieren sowie die unterschiedliche Zielsetzungen zu harmonisieren, um bestmögliche Ergebnisse aus Sicht des Betreibers und des Marktes zu erzielen. Für Unternehmen, die sich durch Öffnen ihrer Produkte für externe Entwickler zu einer Software-Plattform entwickeln möchten, stellt sich die Frage, welche Vorgaben spezifiziert werden sollten und welche Entscheidungen den externen Entwicklern überlassen werden können. Das Thema Plattform-Governance behandelt diese Fragestellung und wird im nächsten Kapitel näher betrachtet.

## 2.2 Plattform-Governance und Kontrolltheorie

Aus Sicht der Plattform-Governance ist eine zentrale Herausforderung für Plattformbetreiber eine Balance zu finden zwischen genügend Freiheiten für externe Entwickler und ausreichend eigener Kontrolle und Steuerung über Aktivitäten auf der Plattform. Ersteres fördert Innovationen und Wachstum während letzteres die Integrität sowie Werte und Ziele der Plattform bewahrt. Plattform-Governance kann aus drei Perspektiven betrachtet werden (Tiwana et al. 2010): Die erste bezieht sich auf *Entscheidungsrechte* und wie diese unter den Akteuren der Plattform aufgeteilt werden. Wer ist autorisiert Entscheidungen zu treffen und übernimmt dafür Verantwortung? Entscheidungen fallen z. B. an in Bezug auf Design und Implementierung, Monetarisierung und Vertrieb sowie Funktionalitäten und Schnittstellen der Plattform und der Apps. Die zweite Governance-Perspektive behandelt das *Eigentumsrecht*. Gehört die Plattform allen Beteiligten oder lediglich einem einzelnen Unternehmen? Bleiben veröffentlichte Apps im Besitz der App-Entwickler oder geht das Eigentums- und Nutzungsrecht an den Plattformbetreiber über? Als dritte Perspektive nennen (Tiwana et al. 2010) die *Kontrolle auf Plattformen*. Kontrolle (Control) wird definiert als das Bestreben, Individuen zu motivieren, um gewünschte Ergebnisse zu erzielen (Ouchi 1979). Hierbei wird unterschieden zwischen einer Person bzw. einem Unternehmen, das Kontrolle ausübt (dem Controller) und einer Person bzw. Personengruppe, die der Kontrolle ausgesetzt ist (dem Controllee). In der Kontrolltheorie haben sich zwei Kategorien von Kontrollformen herauskristallisiert, namentlich formale Kontrolle und informale Kontrolle (z.B., Kirsch et al. 2002):

Formale Kontrolle basiert auf vertraglich festgehaltenen Regeln, einer regelmäßigen Überwachung der Einhaltung dieser Regeln und einer entsprechenden Belohnung bzw. Sanktionierung. Formale Kontrolle wird weiter unterteilt in Zugangskontrolle, Prozesskontrolle und Ergebniskontrolle. *Zugangskontrolle* (Input Control) regelt, wer oder was Zugang zu einem Unternehmen oder einem Projekt erhält. Eingesetzt wird Zugangskontrolle häufig in der Personalverwaltung, in der Kriterien zur Auswahl von Mitarbeitern festgelegt werden. Zur Zugangskontrolle gehört auch die Verteilung von finanziellen Mitteln oder sonstigen Ressourcen, die einem Projekt zur Verfügung gestellt werden und dadurch den weiteren Verlauf des Projektes beeinflussen. *Prozesskontrolle* (Process oder Behavior Control) dient der Steuerung von Aktivitäten und Handlungen der Kontrollierten. Hierzu gehören neben Verhaltensregeln oder Prozessstandards auch vorgeschriebene Entwicklungsmethoden. Nach der Spezifikation von Prozessvorgaben werden die Prozesse in der Regel auf Basis von Berichten und Protokollen überwacht und Mitarbeiter bezogen auf die Einhaltung dieser Vorgaben belohnt bzw. sanktioniert. *Ergebniskontrolle* (Output Control) kontrolliert die Zielerreichung eines Projektes. Dies umfasst z. B. technische, funktionale und visuelle Spezifikationen des Endproduktes sowie Kosten- und Zeitvorgaben. Ergebnisse werden anhand vordefinierter Metriken analysiert und ebenfalls entsprechend belohnt bzw. sanktioniert. Ergebniskontrolle spezifiziert nicht, mit welchen Methoden ein Ergebnis erreicht werden soll und umgekehrt werden unter Prozesskontrolle keine Anforderungen zur Erreichung der Endziele gestellt (Cardinal 2001; Kirsch et al. 2002).

Informale Kontrolle basiert auf sozialen und (zwischen-) menschlichen Strategien und ist unterteilt in Selbst- und Klankontrolle. Informale Kontrollmechanismen werden vor allem eingesetzt, wenn Prozesse und Ergebnisse nur schwierig oder sehr kostspielig überprüft und überwacht werden können sowie wenn gewünschte Prozesse und Ergebnisse nicht oder nur ungenügend bekannt sind. Mit *Selbstkontrolle* (Self-Control) ermöglicht ein Kontrolleur den Kontrollierten sich selbst zu organisieren, indem er z. B. das dafür notwendige Wissen und Methoden schult und entsprechende Hilfsmittel zur Verfügung stellt. Individuen werden dazu ermutigt, sich eigene Ziele zu setzen und eigene Methoden und Prozesse zu etablieren. Eine Belohnung bzw. Sanktionierung findet nach eigenem Ermessen statt. Hierdurch können vor allem neue Prozesse und Ergebnisse entstehen, die später als Standards übernommen werden. Mit *Klankontrolle* (Clan Control) schaffen Kontrolleure gemeinsame Werte und Normen sowie gemeinsame Ziele und Visionen. Hieraus entstehen Gruppen von Individuen, die gemeinsame Werte und Normen teilen und sich für gemeinsame Ziele einsetzen. Solch eine Gruppe tendiert zur Umsetzung ähnlicher Prozesse und Ergebnisse. Belohnungen und Sanktionen finden direkt oder indirekt durch die Gruppe und durch Gruppendynamiken statt. Eine direkte Überwachung ist durch den Kontrolleur nicht notwendig (Kirsch et al. 2002).

## 2.3 Kontrollmechanismen auf Apples App Store Plattform

Die Kontrolltheorie ist bereits weit in der Wirtschaftsinformatik-Literatur verbreitet, allerdings existieren bisher nur wenige Studien zu Kontrollformen auf Software-Plattformen (Tiwana 2014). Um ein Verständnis dafür zu bekommen, welche Kontrollmechanismen auf Software-Plattformen derzeit verwendet werden, betrachten wir im Folgenden den Apple App Store als Fallbeispiel. Die Firma Apple Inc. ist in den letzten Jahren insbesondere durch das iPhone und das iPad international bekannt und erfolgreich geworden. Nachdem 2007 das iPhone vorgestellt wurde, folgte 2008 der Apple App Store, welcher nur wenige Monate später für externe Entwickler geöffnet wurde. Über ein Software Development Kit (SDK) können Drittanbieter eigene Apps für das iPhone und iPad entwickeln und vertreiben. Apple berichtete Anfang dieses Jahres über 6 Mio. registrierte App-Entwickler und über 1,4 Mio. angebotene Apps (Apple 2015). Apple setzt an verschiedenen Stellen unterschiedliche Kontrollmechanismen ein, um die Prozesse und Ergebnisse der App-Entwickler zu beeinflussen. Nachfolgend werden Kontrollmechanismen beispielhaft aufgeführt (vgl. Table 2-1). Als Quellen dienen Analysen des Apple App Stores und des Apple Developer Portals (<https://developer.apple.com/>) sowie (Bergvall-Kåreborn and Howcroft 2011).

Table 2-1: Kontrollmechanismen auf Software-Plattformen

	Kontrollform	Kurzbeschreibung	Beispiel Kontrollmechanismen: Apple App Store
Formale	Zugangskontrolle	Auswahl und Zulassung von (Personal-) Ressourcen	Eintrittsbarrieren für Entwickler: kostenpflichtige Registrierung und Bindung an Apple Hard- und Software
	Prozesskontrolle	Steuerung und Überwachung von Verhalten und Aktivitäten	Bereitstellung von SDKs und Frameworks zur Programmierung und Veröffentlichung von Apps, Empfehlungen für Best-Practice Vorgehensweisen
	Ergebniskontrolle	Spezifikation und Überprüfung von (End-) Ergebnissen	Veröffentlichung von Review und Interface Guidelines. Überprüfung und ggf. Ablehnung von eingereichten Apps. Nutzerbasierte Bewertungen und Bestenlisten von Apps
Informale	Selbstkontrolle	Ermöglichung und Förderung von Selbstorganisation	Freie Budget-, Zeit- und Personalplanung der Entwickler. Freie Auswahl und Umsetzung von App-Funktionalitäten
	Klankontrolle	Schaffung und Förderung gemeinsamer Werte, Normen und Ziele	Kommunikation von Werten, Normen und Zielen durch Produkte, Marketingkampagnen, Corporate Identity und die allgemeine Firmenpolitik

*Zugangskontrolle* Apple rekrutiert neue Entwickler nicht selbst, sondern hält seine Plattform für alle Entwickler offen, stellt hierbei allerdings einige Eintrittsbarrieren auf. Das Apple-Developer-Programm berechtigt zum Zugang zu bereitgestellten Entwicklungsressourcen und ermöglicht die Veröffentlichung von Apps im App Store. Eine Registrierung kostet \$ 99 US-

Dollar und muss jährlich erneuert werden, veröffentlichte Apps werden sonst wieder aus dem App Store entfernt. Weiter stellt Apple für die Entwicklung von Apps Entwicklungswerkzeuge bereit. Diese können allerdings nur mit Mac OS X Geräten von Apple betrieben werden. Der Einstieg in die Apple App-Entwicklung setzt somit eine kostenpflichtige Registrierung und den Besitz von Apple Geräten voraus. Mit diesen Barrieren, bzw. beschränkter Plattformoffenheit (Benlian et al. 2015), stellt Apple u. a. sicher, dass Entwickler ernsthaft an der Entwicklung und Veröffentlichung von gewinnbringenden Apps interessiert sind.

*Prozesskontrolle* Die von Apple bereitgestellten Entwicklungswerkzeuge haben einen direkten und indirekten Einfluss auf die Entwicklungsprozesse der App-Entwickler. Mit der integrierten Entwicklungsumgebung XCode stellt Apple ein zentrales Werkzeug zur Verfügung, mit dem die Programmierung, das Testen und die Veröffentlichung von Apps durchgeführt werden kann. Daneben unterstützen das veröffentlichte SDK und weitere Frameworks sowohl bei der technischen als auch bei der visuellen Integration der Apps in die Plattform. Über das Developer Portal informiert Apple zusätzlich über Best-Practice-Vorgehensweisen im Implementierungsprozess und Lösungen zu typischen Problemen bei der App-Entwicklung. Die Verwendung dieser Entwicklerwerkzeuge und Frameworks ist zwar größtenteils freiwillig, dennoch mangelt es derzeit an gleichwertigen Alternativen. Durch die bereitgestellten Ressourcen beeinflusst Apple maßgeblich, auf welche Weise Apps entwickelt, getestet und veröffentlicht werden.

*Ergebniskontrolle* Apple setzt sowohl eine direkt als auch indirekte Ergebniskontrolle um. Über Allgemeine Geschäftsbedingungen behält sich Apple das Recht vor, unangebrachten und ungeeigneten Apps den Zugang zum App Store zu verweigern. Regelungen für den Zugang zum App Store sind in den Review Guidelines und den iOS Human Interface Guidelines hinterlegt. Dort befinden sich Hinweise zur technischen Umsetzung, zu Sicherheits- und Privatsphäre-Konfigurationen sowie zu Design und Interfaces der Apps. Im Falle einer Ablehnung nennt Apple allerdings keine Gründe und der Ablehnungsprozess bleibt folglich intransparent. Ebenfalls ist nicht bekannt, ob diese Kontrolle stichpunktartig oder für alle eingereichten Apps stattfindet. Neben der direkten Ergebniskontrolle etabliert Apple einige indirekte Mechanismen, um hochwertige und möglichst fehlerfreie Apps zu fördern. Hierzu gehören das nutzerbasierte Bewertungssystem und die Platzierung von beliebten Apps in Bestenlisten. App Bewertungen und Kommentare durch Benutzer beeinflussen die Qualitätssignale einer App sowie die Sichtbarkeit im App Store. Beliebte und erfolgreiche Apps können zudem von Apple für die Startseite ausgewählt und beworben werden. Durch diese Mechanismen entstehen für Entwickler zusätzliche Anreize, hochwertige

und für die Benutzer zufriedenstellende Apps zu entwickeln. Eine Belohnung bzw. Sanktionierung findet in diesem Falle indirekt über den Markt statt.

*Selbstkontrolle* Trotz der zahlreichen Vorgaben, Richtlinien und Restriktionen sind externe Entwickler der Apple Plattform in vielen Punkten sich selbst überlassen. Im Vergleich zu Kontrollmechanismen in traditioneller Softwareentwicklung wird z. B. Budget-, Zeit- und Personalplanung durch die App-Entwickler selbstständig vorgenommen. Die Einhaltung von Veröffentlichungsterminen und Kostenstrukturen wird nicht von Apple überwacht und liegt in der Verantwortung der Entwickler. Ebenso haben Entwickler freie Hand bei der Auswahl und Umsetzung der funktionalen Implementierung ihrer Apps. Apple setzt hier bewusst auf die Ideen und Innovationen der externen Entwickler und fördert somit neue Projekte und das Wachstum der Plattform.

*Klankontrolle* Apples Werte und Normen werden vor allem in der allgemeinen Firmenpolitik und in den Produkten des Unternehmens deutlich. Apple Produkte wie iPhone und iPad haben gewisse Eigenschaft gemeinsam, welche die Werte des Unternehmens widerspiegeln: Hohe Innovation, ein modernes Design, eingängige Nutzbarkeit, für manche Nutzer ein Statussymbol sowie eine einfache Integration der Produkte in das Apple-Ökosystem. Viele dieser Werte, Normen und Ziele von Apple werden durch Marketingkampagnen, die Corporate Identity und die Produkte nach außen getragen und von den Apple Kunden und App-Entwicklern verstanden, anerkannt und übernommen. Entwickler reproduzieren diese Eigenschaften in den Funktionalitäten und Designs ihrer Apps und schaffen einen Wiedererkennungswert des Apple-Ökosystems. Apple beeinflusst somit durch die Kommunikation der Unternehmenswerte die Prozesse und Ergebnisse der externen App-Entwickler.

*Vergleich zur Android App-Entwicklung* An dieser Stelle findet ergänzend ein Vergleich zu Kontrollmechanismen auf Googles Android Plattform statt. Die Android App-Entwicklung ist weniger restriktiv und setzt an vielen Punkten verstärkt auf Selbstkontrolle der Entwickler. Für Android-Entwickler fällt lediglich eine einmalige Registrierungsgebühr von \$ 25 US-Dollar an. Dies stellt eine niedrigere Eintrittsbarriere im Vergleich zum Apple-Developer-Programm dar und erklärt möglicherweise die höhere Anzahl an Entwicklern und angebotenen Apps im Google Playstore. Durch das offenere und freiere System sind eine Vielzahl von Entwicklungs-, Design und Testumgebungen in der Android Community verfügbar und somit werden auch eine Vielzahl verschiedener Entwicklungsprozesse umgesetzt. Eine direkte Ergebniskontrolle findet nicht statt, da alle Apps im Google Playstore zugelassen werden. Lediglich nach Beschwerden von Kunden werden nachträglich unangemessene Apps entfernt, die etwa wichtigen Sicherheits- und Privatsphäre-Aspekten nicht gerecht werden. Bewertungssysteme sind ebenso vorhanden wie Best-Practice-



Implementierungen und Lösungen für gängige Probleme. Die Werte, Normen und Ziele der offenen Android-Plattformen spiegeln sich folglich in den eingesetzten Kontrollmechanismen wieder.

## 2.4 Laborexperiment zur Bedeutung von Selbstkontrolle

Um Auswirkungen verschiedener Kontrollmechanismen auf Software-Plattformen zu untersuchen, haben wir ein Laborexperiment in diesem Kontext vorbereitet und durchgeführt (Goldbach et al. 2014). Ziel war es zu untersuchen, ob gerade die auf Software-Plattformen weit verbreiteten Selbstkontrollmechanismen im Vergleich zu formaler Kontrolle zu hochwertigeren Apps führt und ob selbstorganisierte Entwickler eine Plattform dadurch eher weiterverwenden würden. Die Teilnehmer des Experiments nahmen die Rolle von App-Entwicklern einer fiktiven Software-Plattform ein. Der fiktive Plattformbetreiber gab vor, einen Mockup-Designer und neue Plattform-Vorgaben analysieren zu wollen. Die Teilnehmer hatten die Aufgabe eine App für eine ebenfalls fiktive deutsche Hotelkette mit Hilfe des Mockup-Designers (vgl. Figure 2-2) zu entwerfen.

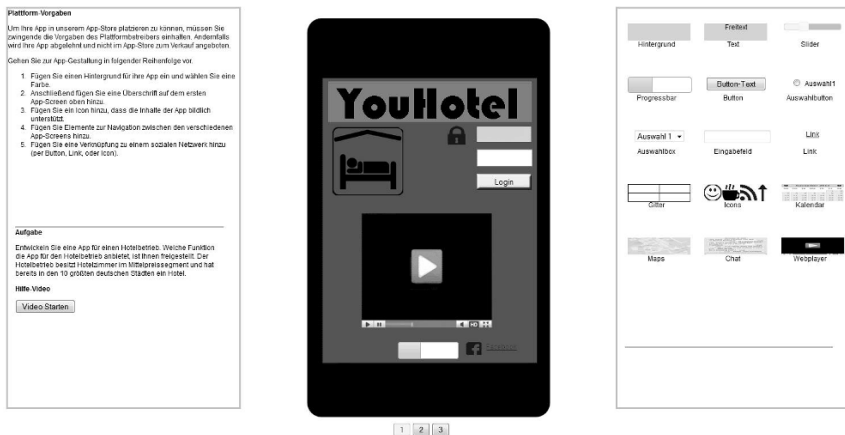


Figure 2-2: Screenshot des Mockup-Designers

Die genaue Funktionalität und das Design der App waren hierbei frei bestimmbar. Die Teilnehmer wurden in drei Gruppen eingeteilt, die jeweils unterschiedlichen Kontrollformen ausgesetzt waren. Diese waren für formale Kontrolle die Gruppen Prozesskontrolle und Ergebniskontrolle und für informale Kontrolle die dritte Gruppe Selbstkontrolle. Die Kontrollmechanismen wurden in Form von Vorgaben des Plattformbetreibers umgesetzt. Die Teilnehmer wurden aufgefordert, diese Vorgaben einzuhalten und umzusetzen, da ihre entworfene App sonst nicht für den App Store zugelassen werden würde. Als zusätzliche

Motivation konnten die Teilnehmer ein Tablet gewinnen, sofern ihre App den Vorgaben entsprach. Die Gruppe Prozesskontrolle bekam Vorgaben zum Vorgehen bei der App-Entwicklung. Die Teilnehmer sollten beispielsweise zuerst einen Hintergrund auswählen, dann den Titel der App hinzufügen sowie anschließend ein Icon zur bildlichen Beschreibung auswählen. Zum genauen Design der App wurden allerdings keine Vorgaben angegeben. Die Gruppe Ergebniskontrolle bekam Vorgaben zum Erscheinungsbild der App. Beispielsweise sollte die App ein Icon in einer bestimmten Größe an einem bestimmten Platz, einen bläulichen Hintergrund sowie eine Navigationsleiste im unteren Bereich der App aufzeigen. In welcher Reihenfolge die Teilnehmer vorgehen, wurde dieser Gruppe freigestellt. Die dritte Gruppe Selbstkontrolle wurde aufgefordert, sich selbstständig zu organisieren, sich eigene Ziele zu setzen und nach eigenen Vorstellungen vorzugehen.

Die 138 Teilnehmer des Experiments waren Studierende einer großen deutschen Universität, überwiegend aus technischen Studiengängen mit Erfahrung in der Software- und App-Entwicklung und wurden mit 5 € pro Teilnehmer vergütet. An einem Rechner im PC-Labor wurden die Teilnehmer zunächst in den vorliegenden Plattformkontext eingewiesen und zufallsbasiert einer der drei Gruppen zugewiesen. Dann wurde den Teilnehmern der Mockup-Designer mit einem Video erklärt, mit dem verschiedene Design-Elemente per Drag & Drop auf die App-Bildschirme hinzugefügt, ausgerichtet und formatiert werden konnten. Zu den Elementen zählten z. B. Hintergrundfarben, Texte und Eingabefelder sowie eine Reihe von Icons und Bildern, außerdem App-spezifische Elemente wie Kalender, Chat und Videoplayer. Nach Ablauf der Entwurfszeit folgte ein Fragebogen bezogen auf die Absicht der Teilnehmer, die Software-Plattform auch in Zukunft weiter für App-Entwicklung zu verwenden und bezogen auf ihre wahrgenommene Autonomie während der Entwurfszeit. Beide Variablen wurden jeweils mit drei Fragen auf einer 5-Punkt-Likertskala erhoben. Im Anschluss an das Experiment wurden die entworfenen Apps in einer Expertenrunde bezüglich Ihrer Qualität, basierend auf den Merkmalen Funktionalität, Nützlichkeit, Bedienkomfort und Design, auf einer 6-Punkt-Likertskala bewertet.

Die Studie kam zu folgenden Ergebnissen (vgl. Figure 2-3): In der Gruppe Selbstkontrolle war die Absicht der App-Entwickler, die Software-Plattform auch in Zukunft für ihre App-Entwicklung zu nutzen, und damit die Loyalität der Entwickler, signifikant höher (d. h. um 15,7%) als in den Gruppen Prozesskontrolle und Ergebniskontrolle (zusammengefasst in der Kategorie „Formale Kontrolle“). Ähnlich war die Qualität der entworfenen Apps in der Gruppe Selbstkontrolle signifikant höher (d. h. um 16,4 %) als in den beiden anderen Gruppen. Aus theoretischer Sicht erklären wir diese Ergebnisse anhand der Theorie für Selbstbestimmung nach Deci and Ryan (2002).

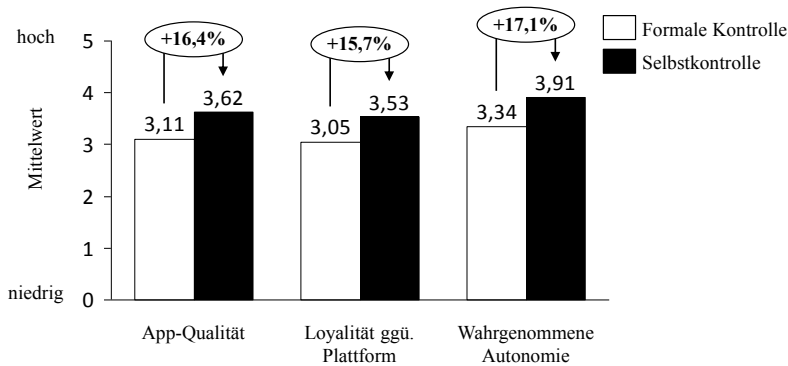


Figure 2-3: Ergebnisse des Experiments

Teilnehmer unter formalen Kontrollmechanismen sind Restriktionen ausgesetzt und müssen ihre Ideen und Ziele sowie ihr Vorgehen anpassen. Sie sind dadurch höherem Druck ausgesetzt, können weniger frei arbeiten und sind insgesamt weniger zufrieden mit ihrer Tätigkeit. Dies verringert auch die Absicht, weiterhin unter solchen Bedingungen zu arbeiten. Dies spiegelte sich in unserem Experiment darin wieder, dass Teilnehmer der Gruppe Selbstkontrolle ein signifikant höheres (d. h. um 17,1 %) Autonomiegefühl wahrgenommen haben. Die daraus folgende höhere intrinsische Motivation und höhere Identifikation mit der Tätigkeit und der App haben schließlich zu einem besseren Ergebnis geführt. Dies zeigte sich in unserer Studie in der signifikant höheren Qualität der entworfenen Apps. Zusammengefasst kann festgehalten werden, dass Selbstkontrolle nicht nur zu Entwicklern führt, die sich auf der Plattform wohler fühlen und die Plattform weiter nutzen möchten, sondern ebenfalls zu hochwertigeren besseren Ergebnissen dieser Entwickler. Der verstärkte Einsatz von Selbstkontrolle auf Software-Plattformen kann also einen wichtigen Beitrag leisten, Innovation und Wachstum zu fördern. Als Limitation der Studie ist die eher künstliche Umgebung des Experiments zu nennen, dessen Ergebnisse in zukünftigen Feldstudien weiter zu untersuchen sind. Erste Umfragen mit App-Entwicklern verifizieren jedoch die Experimentergebnisse bereits.

## 2.5 Erkenntnisse und Empfehlungen

Obwohl Software-Plattform Ökosysteme im Onlinekontext schon seit einigen Jahren erfolgreich im Markt etabliert sind und der Software-Industrie neue erfolgsversprechende Wege für die Entwicklung und Distribution von Software ermöglichen, ist die Literatur im Bereich der Kontrolle und Steuerung auf Software-Plattformen noch in den Anfängen. Wir konnten in diesem Artikel aufzeigen, dass auf Apples App Store Plattform und auch auf

Googles Android Plattform weniger strikte formale Kontrollmechanismen eingesetzt werden als in traditioneller Softwareentwicklung, dafür vermehrt informale Kontrollmechanismen. Bei den eingesetzten formalen Kontrollmechanismen wurde zudem deutlich, dass diese nicht in ihrer Reinform umgesetzt werden. Für die Prozesskontrolle bei Apple sind zwar einige Werkzeuge und Spezifikationen vorhanden, die den Entwicklungsprozess beeinflussen, diese müssen aber nicht zwingend eingesetzt werden. Ebenfalls ist bei der Ergebniskontrolle nicht ersichtlich, wie detailliert Apps vor der Freigabe untersucht werden. Weniger hierarchische Beziehungen und eine große Masse an Entwicklern und Projekten erschweren zudem eine strikte Ausführung formaler Kontrolle. Apple hat vor allem eine starke und präsente Klankontrolle, die sich in den eigenen Produkten und auch den Apps der externen Entwickler widerspiegelt. Zudem konnten wir in unserem Laborexperiment zeigen, dass Selbstkontrolle im Vergleich zu formaler Kontrolle zu loyaleren Entwicklern und hochwertigeren Apps führen kann.

Plattformbetreiber und Unternehmen, die sich hin zu einem Plattform-Ökosystem entwickeln wollen, sollten sich explizit darüber Gedanken machen, in welchen Kontexten sie Vorgaben und Richtlinien spezifizieren möchten. Während einige kritische Themen wie Sicherheit und Privatsphäre-Einstellungen von Apps gewiss einer formalen Kontrolle bedürfen, um Nutzer zu schützen, zeigt unser Artikel, dass verstärkt informale Kontrollmechanismen eingesetzt werden können und sollten. Vor allem die selbstständige Organisation durch Selbstkontrolle kann Verhalten und Ergebnisse der Entwickler auf Software-Plattformen positiv beeinflussen. Ziel ist die Schaffung von Werten und Zielen, die gemeinsam verfolgt werden und weniger starker Kontrolle und Überwachung unterliegen. Möglich ist dies beispielsweise durch Trainings, Konferenzen und Zertifizierungen für Entwickler sowie Schulungen zur selbstständigen Organisation. Es bleibt zu beobachten, wie sich die Entwicklung und der Vertrieb von Software weiter durch das Internet verändern werden. Wir glauben, dass durch die Öffnung der Unternehmensgrenzen und den Einsatz von nur wenigen formalen Kontrollmechanismen externe Potentiale und Innovationen bestmöglich genutzt werden können, um in den dynamischen Märkten konkurrenzfähig zu bleiben.

Control Modes on Mobile Software Platforms

Empirical Studies on the Importance of Informal Control

Goldbach, T.

2016, XIX, 110 p. 12 illus., Softcover

ISBN: 978-3-658-14892-8