Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Open at the Core: Moving from Proprietary Technology to Building a Product on Open Source Software

Jérémie Haese, Christian Peukert

To cite this article:
Jérémie Haese, Christian Peukert (2026) Open at the Core: Moving from Proprietary Technology to Building a Product on Open Source Software. Management Science 72(5):4173-4199. https://doi.org/10.1287/mnsc.2023.02886

informs
https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 72, No. 5, May 2026, pp. 4173–4199
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Open at the Core: Moving from Proprietary Technology to Building a Product on Open Source Software

Jérémie Haese,[a] **Christian Peukert**[a,*]

[a] Faculty of Business and Economics (HEC), University of Lausanne, Quartier Chamberonne, 1015 Lausanne, Switzerland
*Corresponding author

Contact: jeremie.haese@unil.ch, https://orcid.org/0009-0009-5171-8411 (JH); christian.peukert@unil.ch, https://orcid.org/0000-0003-3997-8850 (CP)

Received: September 8, 2023
Revised: August 30, 2024; October 30, 2024
Accepted: November 6, 2024
Published Online in Articles in Advance: September 5, 2025

https://doi.org/10.1287/mnsc.2023.02886

Abstract. Firms are increasingly moving away from proprietary technology to building commercial products on top of open source software. However, it is unclear how such commercialization of open source affects contributions from the community, as well as product quality and firm performance. Web browser technology provides a unique setting to study such questions. The largest open source project is Chromium, led by Google, which serves as the core of various web browsers. Unexpectedly, Microsoft announced a drastic change in strategy in 2018 and adopted Chromium in a complete redesign of the web browser Edge. Unique data lets us compare Chromium to other open source technologies, and Chromium-based web browsers to other web browsers. We find that overall development activity increases after Microsoft adopts Chromium, predominantly because Microsoft starts to contribute to the project. We also see a modest increase in contributions from external developers. We further document an increase in scrutiny, evident from an increase in the number of individuals performing code reviews and a surge in security vulnerability reporting. We also find positive effects for Microsoft. Edge makes a giant leap in functionality, moving it on par with the market leader Chrome. With the adoption of Chromium, Microsoft fixes more bugs, accelerates release cycles, and increases the market share of Edge at the expense of other less popular Chromium browsers. We discuss a number of general implications for managers and policy.

History: Accepted by D. J. Wu, information systems.

Funding: This work was supported by Schweizerischer Nationalfonds zur Förderung der Wissenschaftlichen Forschung [Grant 100013_197807].
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mnsc.2023.02886.

Keywords: open source • web browser • Microsoft Edge • chromium • cybersecurity • GitHub • crowding out

1. Introduction

Digital transformation has spurred a significant shift in corporate strategies, with an increasing number of firms gravitating toward open source software (OSS). In contrast to proprietary software, OSS is code that is freely available, modifiable, and typically redistributable. While firms initially only incorporated OSS in their technology stack and reused open source code for internal purposes (Fosfuri et al. 2008, Nagle 2018), more and more firms now actively contribute to OSS communities (Stam 2009, Nagle 2019), making code publicly available on software development platforms.[1] This corporate shift toward OSS has predominantly occurred over the last five to ten years (Chesbrough 2023). The increasing engagement of firms in OSS has also raised concerns, such as the potential crowding out of individual contributors, potential underinvestments in cybersecurity, and threats to the essence of OSS as reflected in the chosen licensing agreements (Lifshitz-Assaf and Nagle 2021). Despite its increasing prevalence, the transition of incumbent firms toward OSS has not been extensively studied (Altman et al. 2022). In this paper, we explore how the OSS community reacts when a private company switches from proprietary technology to an open source core for a commercial product, as well as how that affects cybersecurity, product quality, and firm performance.

Our empirical setting is the market for web browsers, a tremendously understudied "killer application" that is used by half of the world population on a daily basis. Web browser technology is perhaps also the most prominent example of the commercialization of OSS. In the last decade, the bulk of the market has been split between several private firms that use OSS components inside a proprietary shell. The leading web browsers in the market are Google Chrome which is built on the open source project Chromium, Apple's Safari which is based on the open source technology WebKit, and Mozilla Firefox, which is built on the open source project Gecko. Core web browser technologies are among the largest OSS projects, with a combined number of over 2.4 million individual contributions on the development platform GitHub. In contrast, Microsoft's browser technology has historically always been proprietary. It came as a surprise to many industry observers when Microsoft decided to adopt the open source technology Chromium for their Edge browser in December 2018. In this context, we study how Microsoft's switch to Chromium affects the open source community behind it. Moreover, we study how Microsoft's decision to adopt Chromium affects cybersecurity, product functionality, and Microsoft's market share in web browsers, using a wide range of novel and unique data. We gather evidence on development activity from both project stewards and external contributors using data from GitHub. Further, we extract security patching activity from release notes. We also compile information on the number of supported functionalities and conduct a series of tests to estimate page load time across browser versions. Finally, we examine firm performance by collecting data on web browsers' market shares from different sources. For most of our analyses, we use a difference-in-differences approach, comparing Chromium to non-Chromium projects/browsers.

Our results show that trends in contributions to Chromium did not significantly differ from competitor projects and remained stable, except for a brief surge in contributions during the period when Microsoft developed its first version of Chromium-based Edge. Inside the Chromium project, we document an increase (+6%) in contributions of external developers (neither from Google nor Microsoft) after Microsoft adopts Chromium. We show that increased activity from external developers is in part driven by Microsoft's open source community, that is, individuals who have contributed to Microsoft's open source projects before. We also document a 271% increase in the number of contributors that report security vulnerabilities of Edge, which can benefit every Chromium user. We further highlight substantial churn in the pool of developers contributing to Chromium and non-Chromium projects. Around 66% of the contributors in the period following Microsoft's announcement did not contribute in the twelve months before. At the same time, we show that Microsoft benefited from joining the Chromium project. Regarding product functionality (assessed in the number of features supported), Microsoft reached the technological frontier, but at the cost of becoming similar to its competitors. Microsoft engineers gave up most of the functionalities they had developed before. The adoption of Chromium also disrupted the development process at Microsoft with the adoption of fast release cycles, moving from 203 to 53 days between two releases. However, we also show that switching to Chromium made Edge slower in terms of average page load time. Still, with the new Chromium-based Edge, Microsoft was able to increase the market share of Edge by 4 percentage points. Microsoft Edge did not steal market share from the dominant browser Chrome, but from other less popular Chromium-based browsers and from users finally switching to Edge from the discontinued Microsoft Internet Explorer.

Our findings have important managerial and policy implications. Microsoft's move to Chromium is part of a broader trend where companies integrate open source technology into their commercial products. The phenomenon extends beyond browsers to operating systems like Android and MacOS, as well as commercial Linux distributions. In other sectors, like the news industry, open source is replacing proprietary tech for website design and content management. Additionally, key artificial intelligence (AI) technologies, including Machine Learning libraries like TensorFlow and PyTorch, now have many popular commercial applications, such as OpenAI's ChatGPT. A key concern, therefore, is whether the commercialization of open source crowds out the development community. With increased publicly visible contributions of the commercial player, other contributors may substitute away from contributing to an OSS project (Burtch et al. 2013). In our setting, we do not find evidence of a large exodus of external contributors (i.e., developers neither from Google nor Microsoft) following Microsoft's adoption of Chromium. In addition, we show that the effects we find in the case of Chromium also hold at a larger scale. Looking at 13,740 open source repositories for AI, we show that contributions from the external community increase when a commercial player enters the project. Further, our study highlights dynamics that warrant managerial attention. We find that Microsoft engineers gave up the majority of the functionalities they developed previously, which can reduce incentives (Katz and Allen 1982). Finally, commercial entry into existing open source projects leads to increased concentration, which may result in market inertia and concerns about innovation over the long term. On the other hand, the reduction in the number of technologies can provide more standardization and compatibility between products.

We contribute large-scale empirical evidence to a largely theoretical literature on OSS. In particular, we add to a work that has focused on explaining why firms invest in open source (e.g., Iansiti and Richard 2006, August et al. 2020), on the performance implications of engaging with open source (e.g., Nagle 2018, 2019; Lin and Maruping 2022; Conti et al. 2025), and most directly on how the entry of for-profit firms can affect open source contribution (e.g., Reisinger et al. 2014). Our work is also complementary to papers highlighting the competitive tensions between proprietary and open source technologies (Casadesus-Masanell and Ghemawat 2006, Economides and Katsamakas 2006, Wang et al. 2020). Perhaps closest to our paper, Nagaraj and Piezunka (2024) show that competition from a for-profit firm can have intricate effects on open source contributions. In the context of OpenStreetMap, the entry of Google Maps led new contributors to decrease their efforts while pre-existing contributors have increased their efforts. Similar evidence shows that acquisitions of OSS technologies by external firms can have a positive impact on contributions to other OSS projects (Chen et al. 2021). Further, Yue and Nagle (2024) show that switching the mode of governance in an OSS project by changing the steward from a commercial entity to a nonprofit organization can spur more contributions from commercial players.

More broadly, by adding empirical evidence on dynamics in the web browser market (Yin and Bresnahan 2005, Bresnahan and Yin 2006, Tiwana 2015, Song et al. 2018), we contribute to the understanding of technology strategy in a market that affects every Internet user. Finally, Microsoft's decision to join Chromium is also related to the literature on platform entry strategies. For example, Karhu and Ritala (2021) discuss strategies that allow the entrant platform to capture value using the incumbent ecosystem's resources by copying parts of the resources (exploitation), by following the development cycle of key boundary resources (pacing), or by placing itself inside the platform (injection), all of which seem to apply in our setting. In that sense, we add to a literature that has studied the effects of platform owner entry on complementors (Foerderer et al. 2018, Zhu and Liu 2018, Wen and Zhu 2019).

2. Background Discussion

2.1. Related Literature

Early research on crowdsourced public goods, such as OSS, has focused on motivations for individuals and organizations to invest time and effort into open source (Lerner and Tirole 2002, von Hippel and von Krogh 2003). Incentives to contribute have been characterized as mostly intrinsic, while some are more intrinsic (enjoyment, identification with the ideology) than others (reputation, signaling of capabilities to the job market). More extrinsic motivations often contaminate the intrinsic motivations (Jeppesen and Frederiksen 2006). When motivations to contribute to crowdsourced public goods such as OSS are mostly intrinsic, this means that participants often self-select into projects. The decision to contribute can then depend on the licensing choices of project owners (Belenzon and Schankerman 2015), as well as individual contributors' choice of intellectual property protection (Yilmaz et al. 2023).

When it comes to profit-oriented firms, scholars have highlighted that the selling of complementary services can act as incentives for profit-maximizing organizations to contribute to crowdsourced projects (Iansiti and Richard 2006; August et al. 2013, 2017, 2020). Similarly, complementary assets, such as intellectual property rights in adjacent markets, can provide a channel through which profit-oriented firms can benefit from OSS (Fosfuri et al. 2008). Firm involvement in open source projects also connects to a discussion on the impact of digitization on the boundaries of the firm (Nagle et al. 2024). For instance, the collective provision of an open source input, an open vertical strategy, can be optimal for competing firms on downstream markets (Gambardella and von Hippel 2019). Finally, open source projects originated and led by firms such as Chromium share certain characteristics with other collaborative modes of organizing R&D and production between competitors such as research joint ventures and R&D networks (Kamien et al. 1992, Caloghirou et al. 2003, Hanaki et al. 2010). R&D spillovers are here fully shared but the absence of strong property rights and the presence of a community make the open source a unique mode of organization.

Another set of studies concerns the organization of OSS development and the corresponding strategic interactions between firms and the open source community. Changes in incumbent firms' OSS strategies have effects on individuals' incentives to contribute to OSS. For instance, IBM's announcement not to enforce their patents against OSS developers in the mid-2000s led to more entry of OSS-based products by startup companies (Wen et al. 2016). The opposite is true with stronger IP enforcement of incumbents. Evidence shows that OSS development activities decreased substantially after widely publicized lawsuits arose around copyright issues in OSS (Wen et al. 2013). In addition, scholars have studied how licensing and corporate sponsorship affect development activity and project success, showing that it is more difficult for for-profit organizations to engage the community than for nonprofit organizations (Stewart et al. 2006, Spaeth et al. 2014, Yue and Nagle 2024). Competition from a for-profit firm can have intricate effects on open source contributions. In the context of OpenStreetMap, the entry of Google Maps led new contributors to decrease their efforts while pre-existing contributors increased their efforts (Nagaraj and Piezunka 2024). Finally, theoretical work suggests that firms' investment in OSS can lead their competitors to invest in OSS as well (Reisinger et al. 2014). By empirically measuring how a large private company's entry into an existing OSS project affects contribution dynamics from existing contributors as well as newcomers, we extend the knowledge base of how firm participation affects OSS communities.

The absence of direct monetary incentives has raised the question of whether OSS can have a similar quality to proprietary software. In-depth case studies show that OSS code is modular and general libraries are often reused (Haefliger et al. 2007), perhaps because more modular projects are better at motivating developers to contribute (Baldwin and Clark 2006, MacCormack et al. 2006). An early hypothesis was that OSS would lead to improved software quality (Raymond 1999), yet empirical evidence found no significant difference in quality between proprietary and open source software (Spinellis 2008, 2011). However, beyond pure code quality, theoretical work has highlighted that product quality of OSS, when measured according to user utility, can be higher due to complementary products and indirect network effects (Casadesus-Masanell and Ghemawat 2006, Economides and Katsamakas 2006, Wang et al. 2020). Our study contributes to this literature stream by documenting the effects of moving from a proprietary to an open source technology on functionality as an indicator of product quality. Our work is also complementary to prior research as we study competition between proprietary shells of different open source projects instead of highlighting competitive tensions between proprietary and open source technologies.

In addition to software quality or product quality, the question of how open source can create value more broadly has been approached from various angles. Scholars have measured the contribution of OSS to the gross domestic product (Greenstein and Nagle 2014), established a link between OSS and entrepreneurship (Wright et al. 2023), as well as shown how OSS is connected to startups reaching funding milestones (Lin and Maruping 2022, Conti et al. 2025). In the context of Linux, evidence shows that using OSS can increase productivity when firms have complementary capabilities (Nagle 2019) or actively contribute to OSS (Nagle 2018). In our setting, we are able to quantify how adopting a leading open source technology contributes to firm performance measured by product market shares.

2.2. Empirical Context

The following sections provide more context on the empirical setting, guiding our exploratory study. This section, but also the empirical analysis below, benefited from conversations with senior developers in the open source community, including core members of the Chromium project.

2.2.1. Chromium: Emergence and Governance. Chromium emerged as an open source project led by Google, with the first release of Google's web browser Chrome in 2008. Before that, the web browser market experienced competition between proprietary and open source technologies.[2] In the late 1990s, perhaps as a response to Microsoft's dominance with its proprietary web browser Internet Explorer, the open source community started to develop rendering engines – the core component of a web browser that converts raw code (e.g., HTML and CSS) into a web page that is displayed to the user. Table 1 summarizes the core technologies used by different browsers and which company has taken the lead in maintaining the core technologies. Over time, Chromium garnered the collective efforts of a vast community of developers and contributors, evolving into a pivotal player in the realm of web browser technologies. Chromium is now the leading web browser engine, which makes it the de facto standard for new features.[3] Today, desktop devices are no longer the sole and primary way to access the World Wide Web, but the core technologies behind web browsers for smartphones, tablets, smart TVs, etc. remain Gecko, WebKit, and Chromium for the vast majority of devices, making up for a combined market share of more than 85%.

Large open source projects typically employ several governance mechanisms that provide a structure for contributions, define control rights, and shape incentives (von Krogh et al. 2012). The governance structure of the Chromium project is designed to foster collaboration, transparency, and effective decision-making among its diverse contributors. All code changes submitted by contributors undergo thorough code reviews.

Table 1. Prominent Rendering Engines and Web Browsers

Steward	Engine	Open Source	Browser
Microsoft	Mosaic, Trident	No	Internet Explorer until V10
Microsoft	EdgeHTML	No	Edge until V18, Internet Explorer from V11
Mozilla	Gecko	Yes	Firefox, Camino, Netscape V6-7, etc.
Apple	WebKit (forked from KHTML)	Yes	Safari, Chrome until V28, all browsers on iOS, etc.
Google	Blink (forked from WebKit and key component of Chromium)	Yes	Chrome, Opera, Edge from V79, etc.

Source. https://en.wikipedia.org/wiki/Comparison_of_browser_engines and https://en.wikipedia.org/wiki/List_of_web_browsers.

Contributions to Chromium need to be validated by at least two reviewers, where one of them needs to be a *code owner*, who plays a similar role as senior editors at academic journals. A meritocracy-based model, which also applies to the project steward's employees, governs how one becomes a reviewer and a code owner.[4] This process involves scrutiny by senior developers and maintainers who assess the code's quality, security, and adherence to project standards. Code contributions are made by a wide range of contributors, including individuals employed by Google and other firms, and the broader open source community. Major decisions are often discussed openly within the community, and consensus is sought through technical discussions, mailing lists, and collaborative forums. Chromium's development process, in general, is highly transparent, with design discussions, code changes, and decision-making discussions conducted openly in public forums.

The Chromium project uses a 3-clause BSD license.[5] This permissive licensing model offers much flexibility and room for customization as the only constraints are to maintain copyright notices and the license's disclaimers of warranty, and not use the names of contributors for endorsement of a derived work without specific permission. This allows developers and organizations to include essential core technologies within proprietary shells, enabling them to create tailored browser experiences that align with specific needs and preferences. For example, the Brave browser is positioned as privacy-focused, blocking advertisements and trackers by default. Moreover, the flexibility extends to decisions regarding release cycles, granting Chromium users the autonomy to adapt their strategies in response to evolving market dynamics and user demands.

2.2.2. Microsoft's Drastic Change in Strategy. With the Internet and key open source technologies becoming ubiquitous, Microsoft's strategy has changed, culminating in the decision to abandon their proprietary rendering engine in exchange for joining the set of browsers built on Chromium technology. The change in strategy can perhaps be traced back to 2015, when Microsoft initiated a transformation of its software release and update strategy with the introduction of Windows 10. The objective was to deliver updates to users automatically, streamlining the company's development efforts toward the latest versions. This change resulted in the end of the development of Internet Explorer and the release of a new web browser, Edge, which was based on an entirely new proprietary rendering engine called EdgeHTML.[6] However, the launch of Edge failed to produce the desired results, perhaps due to the unfavorable reputation of Internet Explorer, compatibility issues, the rise of mobile Internet, and the already strong market position of Chrome.[7] The Chromium project played a crucial role in the development of new web standards, influencing their specifications and reducing the implementation burden in subsequent Chromium releases. Compliance with standards is a key focus of all open source engines. Apple Safari's significant market share, driven by iOS device adoption, incentivized web developers to adapt to its WebKit engine, creating a second de facto standard alongside Chromium and diminishing the appeal for developers to accommodate a third engine like Microsoft's EdgeHTML.

In this broader context, Microsoft announced on December 6, 2018, that they would adopt Chromium as the underlying technology for a new Edge browser.[8] The announcement came as a major surprise to market participants and observers.[9] For instance, Mozilla voiced concerns that Microsoft's decision to abandon EdgeHTML would decrease competition and shift power toward Google.[10] Microsoft seemed to immediately get to work on the new project, and in August 2019, they reached out to the community, launching a bounty program for Chromium-based Edge.[11] The first version of Chromium-based Edge (version 79, codenamed "Project Anaheim") was finally rolled out on January 15, 2020.

As any public observer, Microsoft could know that Chromium-based browsers had a broader range of functionalities than their own web browser. Therefore, it seems likely that one of the underlying reasons for Microsoft to join Chromium was to leapfrog the competition or at least be on par with the technological frontier. This comes against the backdrop of the failed attempt to substantially ramp up relative product quality by developing the EdgeHTML engine as a replacement for Internet Explorer's Trident engine.[12] Another reason behind Microsoft's decision to adopt Chromium may have been to increase the market share of Edge to be able to promote better complementary products such as its productivity software suite or its search engine.[13]

Microsoft's embrace of open source is a significant departure from its historical position and this marked a cultural change within the company.[14] For instance, in the early 2000s Microsoft's CEO, Steve Ballmer, thought of the open source operating system Linux as "communism" and a "cancer".[15] The shift from seeing OSS as a competitive threat toward an opportunity to collaborate is reflected in Microsoft's increased investment in open source over the past decade. The adoption of Chromium for Edge is significant given the immense size of the web browser market, but the broader switch in strategy is perhaps best illustrated by the acquisition of the open source development platform GitHub for USD 7.5b in 2018. Microsoft has also made efforts to enhance compatibility between its products and OSS, consistently following Google as the second largest private contributor to GitHub (see Figure A.2).

Anecdotal evidence suggests negative community reactions to Microsoft's actions in the OSS space. For example, developers left GitHub to switch to GitLab after Microsoft acquired the OSS development platform in 2018.[16] In 2022, the Software Freedom Conservancy, a prominent OSS organization, called for developers to leave GitHub because of intellectual property issues in Microsoft's development of generative AI products.[17] Yet, despite these critics, GitHub has increased its user base over the period.

2.2.3. The Growing Significance of Web Browsers in the Digital Economy. Web browsers are an essential tool for accessing and navigating the Internet. The web browser market shares many characteristics with other digital products. As often in digital markets, web browsers are priced at zero, moving the competition to other product characteristics. This also implies that web browsers use other means to monetize and recover costs. Like many digital platforms, they do so by leveraging data on user web browsing activity. Web browsers can observe the full web browsing history for each of their users as well as some user characteristics (IP address, location, etc.).[18] These data are particularly valuable for companies in the advertising business, such as Google.

Modern web browsers can be considered to be multisided platforms. They enable interactions between users, websites and applications developers, browser extension developers, and search engines that seek to be set as default in the browser's search bar. In a few decades, browsers went from initially only supporting plugins and email applications to various highly used applications. For instance, Google released Docs Editors in March 2006, which competes directly with standalone software Microsoft Office and Apple iWork. Various popular software-as-a-service applications run entirely in the browser, for example, ChatGPT. Furthermore, cloud gaming applications such as Google Stadia or Nvidia Geforce Now show that web browsers can today be seen as substitutes for operating systems or video game consoles. One of the competitive advantages of web browser applications lies in cross-platform capabilities, as many web browsers run on different operating systems, both on desktop and mobile. As multisided platforms, web browsers are therefore characterized by indirect network effects. The value of a browser derived by users is, first and foremost, dependent on the number of websites that can be properly displayed by the browser. This leads to concentration dynamics, where a single browser can become dominant in the market. Finally, web browsers are embedded in larger ecosystems and serve as distribution channels for complementary products such as search engines or web applications. Reciprocally, complementary products are also used to bundle web browsers. For instance, Google is using its search engine or other applications to promote Chrome, Apple is bundling Safari with hardware devices. By leveraging its dominant position in the operating system market, Microsoft can strategically promote Edge as the default web browser on desktop computers.[19] A major criticism of previous versions of Edge was that it was only available in Windows 10, which changed when versions of Chromium-based Edge were released for a larger set of operating systems, including MacOS.[20] This distribution strategy can introduce Edge to a larger user base, potentially increasing its visibility and adoption rates (Dubé et al. 2010, Boudreau 2021). Bundling effects can create a strong initial user base for Edge, but sustained usage and market share gains will depend on Microsoft's ongoing efforts to provide value and maintain a competitive edge in the web browser landscape (Tellis et al. 2009, Zhu and Iansiti 2012).

3. Research Objectives, Data, and Methods

Taking the insights of the related literature and empirical context into consideration, we now specify our research objectives and describe the various datasets we use for our exploratory analysis. We collect a wide range of novel and unique data to study how Microsoft's decision to join Chromium affects the open source community, product quality, and firm performance. Table 2 provides an overview of all variables, data sources, as well as the time we cover in the observation period. Depending on the research focus and the data set, the definitions of the groups we compare differ. Table A.1 provides an overview. In what follows, we will first discuss the objectives of the analysis, then describe what data we use to provide answers, and finally describe the estimation approach we take.

3.1. Open Source Community

3.1.1. Development Activity. First, we examine the impact of Microsoft's transition from proprietary technology to OSS on the dynamics of the OSS community. This is important because such a shift can influence the motivations of contributors—both intrinsic and extrinsic—potentially expanding the developer base (Boudreau and Jeppesen 2015) and increasing engagement due to larger size and heightened visibility of the OSS project (Zhang and Zhu 2011, Mollick 2016, Loh and Kretschmer 2023), as well as career opportunities (Bitzer and Geishecker 2010, Hann et al. 2013, Xu et al. 2019). However, there is also a risk that Microsoft's involvement could crowd out individual contributors or alter the dynamics of corporate participation (Burtch et al. 2013, Spaeth et al. 2014), affecting the overall health and sustainability of the OSS community. Understanding which effect dominates is crucial

Table 2. Variable Definitions

Name	Definition	Source	Coverage
Dependent variables			
Development activity			
Commits	Daily sum of commits to repositories	GitHub	01/2016–12/2021
Code Change	Number of lines in a commit	GitHub	01/2016–12/2021
Reviewers	Monthly count of code reviewers	GitHub	07/2017–12/2021
Commits/Reviewer	Monthly sum of commits divided by monthly count of code reviewers	GitHub	07/2017–12/2021
Security and patching			
Reporters	Quarterly sum of vulnerability reporters	Patch Release Notes	04/2016–12/2021
Vulnerabilities Fixed	Quarterly sum of fixed vulnerabilities	Patch Release Notes	07/2015–12/2021
Product quality			
Functionalities	Number of functionalities available in a release version	Caniuse	07/2015–12/2021
Release Cycles	Days since release of prior version	Caniuse	01/2016–12/2021
Loading Time	Combined loading time of craiglist.org and wikipedia.org webpages (in milliseconds)	BrowserStack simulations	03/2017–04/2022
Market performance			
Market Share	Monthly average market share of a browser in a country	Statcounter	01/2016–12/2021
Market Share Desktop	Monthly average market share of a browser's desktop version in a country	Statcounter	01/2016–12/2021
Independent variables			
Groups (see Table A.1 for exact definitions)			
Chromium (0/1)	One if in Chromium repo / related to Chromium browsers		
Microsoft (0/1)	One if related to Microsoft employee / browser		
Google (0/1)	One if related to Google employee / browser		
Periods			
Announcement (0/1)	One between December 6, 2018 and January 15, 2020		
PostRelease (0/1)	One after January 15, 2020		

because it informs how firms and communities can best collaborate to ensure the long-term health and innovation of the OSS ecosystem.

To study the effects on the OSS community, we access fine-grained data from GitHub. These data enable us to precisely quantify the level of effort invested over time by individuals and organizations toward the development of open source components for web browsers. We focus on three open source rendering engines, namely Chromium (Google), Gecko (Mozilla), and WebKit (Apple). Chromium, Gecko, and WebKit are among the most active repositories on GitHub in terms of the number of commits. Chromium has over 1.3 million commits, Gecko more than 850,000, and WebKit over 260,000 commits in 2023. For comparison, the Linux repository, one of the most significant open source projects in history, has around 1.2 million commits on GitHub. We obtain the universe of activities in the respective repositories. This lets us track contributions accepted and implemented ("commits") from the companies that provide stewardship to the respective project, and from external participants.[21] We define project stewards in the Chromium project as users whose accounts are linked to @google.com and @chromium.org email addresses, and also separately distinguish external participants and accounts with @microsoft.com email addresses.[22]

3.1.2. Security and Patching.

A security vulnerability in software refers to a weakness or flaw in the code that can be exploited by attackers to gain unauthorized access to sensitive information or cause harm to the system. These vulnerabilities are reported by security researchers, independent testers, and sometimes by users. They can be reported directly to software vendors, via institutional bug bounty programs, or through private intermediaries. The common aspect of these solutions is that vulnerability discovery has been increasingly outsourced to the crowd (Malladi and Subramanian 2020, Zrahia et al. 2024). Once reported, the software vendor will investigate and verify the vulnerability and then create a patch or update to fix the issue. This update is then made available for users to download and install to protect their systems from exploitation. The ability of software vendors to attract reporters and fix vulnerabilities in a timely manner is therefore part of the quality dimension of software.

The integration of major commercial firms into an open source project, and their use of the OSS core in a proprietary shell, can have significant implications for cybersecurity as it can introduce new dynamics in the way security vulnerabilities are reported and addressed. This is particularly relevant as companies increasingly rely on bug bounty programs to incentivize external reporters to discover and responsibly disclose security flaws (Maillart et al. 2017). In the case of Chromium, Microsoft's entry is further notable because it introduced its own bug bounty program in addition to the one already operated by Google. This dual incentive structure is likely to attract a higher volume of vulnerability reports, potentially leading to more robust patching activity and improved security for Chromium. However, this situation also raises important questions about the broader effects on the open source ecosystem. Specifically, there is a need to explore whether the increased focus on Chromium might divert attention and resources away from other open source projects, leading to negative spillovers in the overall cybersecurity landscape.

We collected details on security vulnerabilities for Microsoft Edge, Google Chrome, Mozilla Firefox, and Apple Safari from the security update release notes published by the companies on their websites.[23] We started by collecting a list of the URLs of all security release notes for all major browsers. We then parsed the HTML pages in a semiautomated fashion to collect information. For each vulnerability, we are able to identify the release date for the patch, its severity, and a description. For the four web browsers, the release notes enable us to identify reporters' names when applicable. Over our observation period, web browser vendors fixed 4,338 vulnerabilities. Each vulnerability is usually identified by a unique identifier (CVE code) as part of the Common Vulnerabilities and Exposures system managed by the MITRE Corporation, a US nonprofit organization.

3.2. Product Quality and Firm Performance

A critical question arising from Microsoft's decision to join the Chromium open source project is whether this move was strategically beneficial for the company. The answer is not immediately obvious, as it involves weighing several complex factors.

On one hand, joining an existing open source project like Chromium offers Microsoft significant advantages. It can be more cost-effective than developing software entirely in-house or creating a new open source community from scratch. Additionally, contributing to an established open source project allows Microsoft to align its products with commonly accepted standards, potentially correcting past misalignments and ensuring compatibility with industry norms (Leone and Reichstein 2012, Simcoe and Watson 2019). By actively participating in Chromium's development, Microsoft also gains influence over future standards, which can be strategically important for maintaining a competitive edge in the market (Gross 2020, Baron and Kanevskaia 2023).

However, joining an open source project also requires Microsoft to relinquish some control over the direction of the technology, as open source development is inherently collaborative and often involves multiple stakeholders with varying interests (Folta 1998, Rothaermel et al. 2006). This loss of control could lead to scenarios where features developed by Microsoft are commercially exploited by others, creating potential appropriability issues (Grimpe and Kaiser 2010, Laursen and Salter 2014). Moreover, the transition to open source might affect the company's ability to differentiate its products, as shared open source technology could lead to increased similarities with competitors, potentially reducing users' incentives to switch to or remain loyal to Microsoft's products.

Despite these challenges, open source participation can also bring positive outcomes for Microsoft's engineering team, fostering higher quality contributions due to the transparency and collaboration inherent in open source projects (Kochhar et al. 2019). Additionally, the widespread adoption of a familiar open source technology like Chromium could lower switching costs for users, enhancing the overall market performance of Microsoft's products.

Overall, this background leads us to study product functionality and how it changed when Microsoft joined Chromium. Further, we assess how Microsoft's firm performance has changed to better understand whether joining Chromium was beneficial for the company.

3.2.1. Functionalities.

Our first measure of product quality concerns the functionalities that a web browser offers. This is of primary importance for web developers, but indirectly also for the browsing experience of users.

Information on functionalities supported by specific versions of major browsers comes from *caniuse.com*, which is led by a community of web developers. A functionality relates to the capability of a web browser to render elements of a web page (e.g., reading MPEG-4 videos), to display them in a specific way (e.g., adapted to the screen size) but also to provide services to the website (e.g., get user geolocation). Hence, functionalities are key elements of web browser architecture that can (un)lock new designs and innovations for web developers and users.[24] Choices made by web browser vendors can have a major impact on the downstream ecosystem of websites. Notable examples include the end of the support for Adobe Flash technology by Apple Safari (Horton and Tambe 2025) or the recent discussion on the scheduled end of support for third-party cookies.[25] It is important to note that while the rich data from *caniuse.com* lets us measure web-based functionalities, that is, functionalities that web developers can exploit, we cannot observe UI/UX functionalities of the respective browser, for example, Multitab browsing, toolbars, etc.

At the time we collected the data, *caniuse.com* covered 528 functionalities for 17 web browsers and their historical versions. The unit of observation is at the browser-month level. New functionalities are added based on votes following suggestions on the corresponding GitHub page. The website is widely used in the web developer community to ensure compatibility and implement new features on the websites they maintain.[26]

3.2.2. Page Load Time. Our second measure of product quality concerns end-users more directly. The speed at which a web browser loads a web page has become an important aspect of user experience. However, common speed tests are often designed by web browser vendors and can be optimized to favor their own designs, which raises questions about the validity of the results. To address this issue, we create a benchmark that is independent of any particular web browser vendor using BrowserStack, a service lets us quickly access various browsers and in different versions on a virtual machine running Windows 10. Testing old versions of browsers presents a challenge though, as older browser versions may not support modern website technologies. To address this potential issue, we limit ourselves to Wikipedia.org and Craiglist.org, as these websites have remained relatively unchanged over the past decade and are simple enough to be displayed by older web browser versions without compatibility issues.[27] For each website and browser version, we can measure the loading time in milliseconds, which allows us to objectively assess the performance of historical versions of web browsers. To ensure consistent results, we iterated thirty times for each web page and every browser version of Microsoft Edge, Google Chrome, and Mozilla Firefox from 2017 onwards. Our final sample contains 3,323 observations.

3.2.3. Market Shares. To measure firm performance, we rely on market share data across devices. These data come from StatCounter Global Stats, a web traffic analysis service. In particular, it provides market share details for web browsers and operating systems by country, device, and version. StatCounter's tracking tool is installed on more than 2 million websites worldwide with billions of page views every month.[28] We collected market share data for 35 industrial countries: Australia, Canada, the member states of the European Union, Israel, Japan, the United Kingdom, the US, South Korea, and Switzerland. For a robustness check, we also gather web traffic data from US government websites. Compared with StatCounter Global Stats, web traffic data allows us to observe unique visits for each web browser and not only market shares.

3.3. External Validity

Our primary analysis examines Microsoft's entry into the Chromium project, an open source initiative already managed by a commercial entity, Google. While this case offers insights into how commercial participation influences development activity, it also raises questions about the generalizability of these findings to projects with different governance structures. Specifically, it is unclear whether similar effects would occur if Chromium had been initially governed by a nonprofit or community-led organization.

To address this limitation and enhance the external validity of our study, we extend our exploratory analysis to more cases from an important and rapidly evolving technology field. In particular, we study OSS repositories related to AI. In this setting, we observe both community-governed projects as well as projects with contributions from commercial actors, offering a diverse and relevant setting to assess the effects of commercial entry on development activity. By analyzing a large sample of AI repositories, we aim to determine whether the introduction of a commercial player leads to increased or decreased development activity—reflecting crowding in or crowding out—depending on the governance structure of an OSS project.

We followed the methodology developed in Gonzalez et al. (2020) to compile our list of AI repositories, making use of the fact that owners can tag their GitHub repositories according to the contents with so-called topics. We then filtered out repositories that contained only educational content (tutorials, courses, etc.) using a manually coded training data set and basic natural language processing methods. With this clean list of 13,740 AI repositories, we obtained all push events,[29] and of these repositories from GHArchive, we parsed the email address domain of all authors of commits within the event's payload to identify their affiliation (corporate/noncorporate). We implemented a multistep procedure combining data from multiple authoritative sources to systematically categorize email domains into four distinct groups of contributors: company, university, government/not-for-profit, and individual. First, we classified company domains using Crunchbase data, where we identified organizations as either commercial, academic, or government/nonprofit based on their descriptions, category listings, and domain patterns. Academic institutions were identified through keywords like "Research Institute" and specific domain patterns (e.g., .edu.cn), while government/nonprofit organizations were identified through keywords like "National Institute," "Nonprofit," government domains (.gov), and specific organizational domains (e.g., cern.ch). Second, we identified additional academic institutions using the university-domain-list database,[30] and we also curated a list of university subdomains if part of the email domain matches with the domains present in the list. For instance, 'fas.harvard.edu' is a subdomain associated with the faculty of arts and sciences at Harvard University, which we identified through our subdomain lookup. Third, we identified individual/personal email accounts using a comprehensive list of free and disposable email providers.[31] Finally, we conducted manual checks at all steps and resolved the remaining ambiguous cases by manually checking domains to ensure consistent classification. The final classification was implemented hierarchically—domains were first checked against university lists, then free email providers, government/not-for-profit, and finally against company domains, with manual verification as the final step.

In this setting, we can observe a large amount of variation. First, some repositories are initially entirely community-driven and managed by an individual or nonprofit organization (e.g., Scikit-learn managed by the INRIA Research Institute Foundation), while others resemble the characteristics of Chromium in that they are managed by a commercial firm (e.g., PyTorch and TensorFlow, which have been launched and sponsored by Meta and Google, respectively).[32] Second, some repositories attract a large number of contributors (like Chromium), while others are substantially smaller. Third, some repositories see the entry of a commercial player while others do not. Fourth, within those repositories where a commercial player enters, there is substantial variation in how well-embedded the entrant is in open source communities (measured by the number of projects they contributed to in the past).

Our definition of a commercial entrant varies based on the initial ownership structure of the repository. For repositories without corporate ownership (i.e., no commits from a company in the first month of existence), a commercial entrant is defined as the first company to make commits to the repository. Conversely, for repositories with existing corporate ownership/involvement, a commercial entrant is defined as the first company, different from the original corporate owners, to make commits to the repository. In Scikit-learn's case, which began as a community project without corporate ownership, the commercial entrant would be the first company to start contributing through commits. In TensorFlow's case, which had corporate involvement from the start, the commercial entrant would be the first company other than the original corporate owners to contribute through commits.

This multicase analysis enables us to qualify the boundary conditions of our Chromium case study while extrapolating higher-level insights about the commercialization of open source software more generally, irrespective of prior commercialization.

3.4. Estimation Strategy

We study the consequences of Microsoft's decision to abandon the proprietary technology used for their web browser Edge and to adopt Chromium technology. For competitors, this move came as a surprise. For Microsoft, this decision was strategic, a major paradigm shift with lots of uncertainty on how competitors, open source contributors, and consumers would react. To study the effects on the open source community, we compare Chromium to WebKit and Gecko. Being the two other major players in the browser engine space makes WebKit and Gecko the natural choice for a control group. Similarly, we compare Chromium-based browsers to other browsers to study security reporting and patching, as well as product quality and firm performance.

We compare various outcomes across browsers before and after Microsoft announced the switch to Chromium, and before and after Microsoft released the first version of their Chromium-based browser. In particular, we distinguish between two groups of projects (Chromium and non-Chromium) and three groups of browsers (Chromium, non-Chromium, and Microsoft). When comparing the two groups of projects, our baseline specification can be written as follows:

$$Y_{it} = \alpha + \beta_1 Announcement_t + \beta_2 PostRelease_t + Cr_i(\beta_3 + \beta_4 Announcement_t + \beta_5 PostRelease_t) + \varepsilon_{it}, \quad (1)$$

When comparing the three groups of browsers, our baseline specification can be written as follows:

$$Y_{it} = \alpha + \beta_1 Announcement_t + \beta_2 PostRelease_t + MS_i(\beta_3 + \beta_5 Announcement_t + \beta_7 PostRelease_t) + Cr_i(\beta_4 + \beta_6 Announcement_t + \beta_8 PostRelease_t) + \varepsilon_{it}, \quad (2)$$

where Y_{it} is the number of contributions to the underlying open source projects, number of vulnerabilities, number of functionalities, page load time, as well as market shares. MS_i and Cr_i are dummies respectively indicating if a web browser is developed by Microsoft and if a web browser is based on Chromium technology (but not developed by Microsoft). These dummies enable us to group observations and compare the evolution between the groups. For example, MS_i is equal to zero while Cr_i is equal to one for Opera 83 or Chrome 99. Symmetrically, MS_i is equal to one while Cr_i is equal to zero for Edge 98. Given our panel data structure, we cluster standard errors at the browser level to control for serial correlation between error terms across time. $Announcement_t$ indicates the time period between the Microsoft announcement and the release of the first version of Chromium Edge, whereas $PostRelease_t$ indicates the time period after the release of the first version of Chromium Edge. We use two reference points because we expect some of the consequences of Microsoft's decision to appear right after the announcement (e.g., on software development efforts), while other effects will only be measurable once the product is released (e.g., the number of functionalities supported by the newly Chromium-based Edge).

Our preferred specifications estimate the coefficients of interest using OLS, and we use a $\log + 1$ transformation of the dependent variable because it allows for a convenient interpretation. However, we are aware of the potential issues this can create (e.g., Wooldridge 2023, Chen and Roth 2024) and demonstrate in Section 4.4.1 that our results are robust to a Poisson model or unlogged OLS estimation.

Standard diagnostics such as leads-and-lags plots (see Figures A.3, A.4, A.5 and A.6)[33] suggest that the identifying assumption of the difference-in-differences method—parallel pretrends—is satisfied. Of course, there could have been spillover effects to related projects and products, for example when the attention of security specialists, whose supply is fixed in the short run, shifts away from other OSS projects toward Chromium. Further, a potential limitation of using WebKit and Gecko as a comparison group is that both are substantially smaller than Chromium in terms of commits. We, therefore, also use the Linux repository as a control group, which is similar in size but plausibly unaffected by Microsoft's decision to adopt Chromium. As discussed in detail in Section 4.4.2, this exercise yields very similar results. We are, therefore, reasonably certain that the changes we observe are indeed causally linked to Microsoft's involvement in the Chromium project. We have further carried out extensive research, gone through community forums and the trade press, conducted interviews with senior developers, and have not encountered other major events that coincide with the announcement or with the first release of Chromium Edge.

4. Results

The discussion of results is structured such that we first report results concerning the effects on the community and then results concerning product quality and firm performance. Throughout, we will first discuss model-free evidence, then quantify the patterns using the regression specifications outlined above.

4.1. Effects on Development Activity

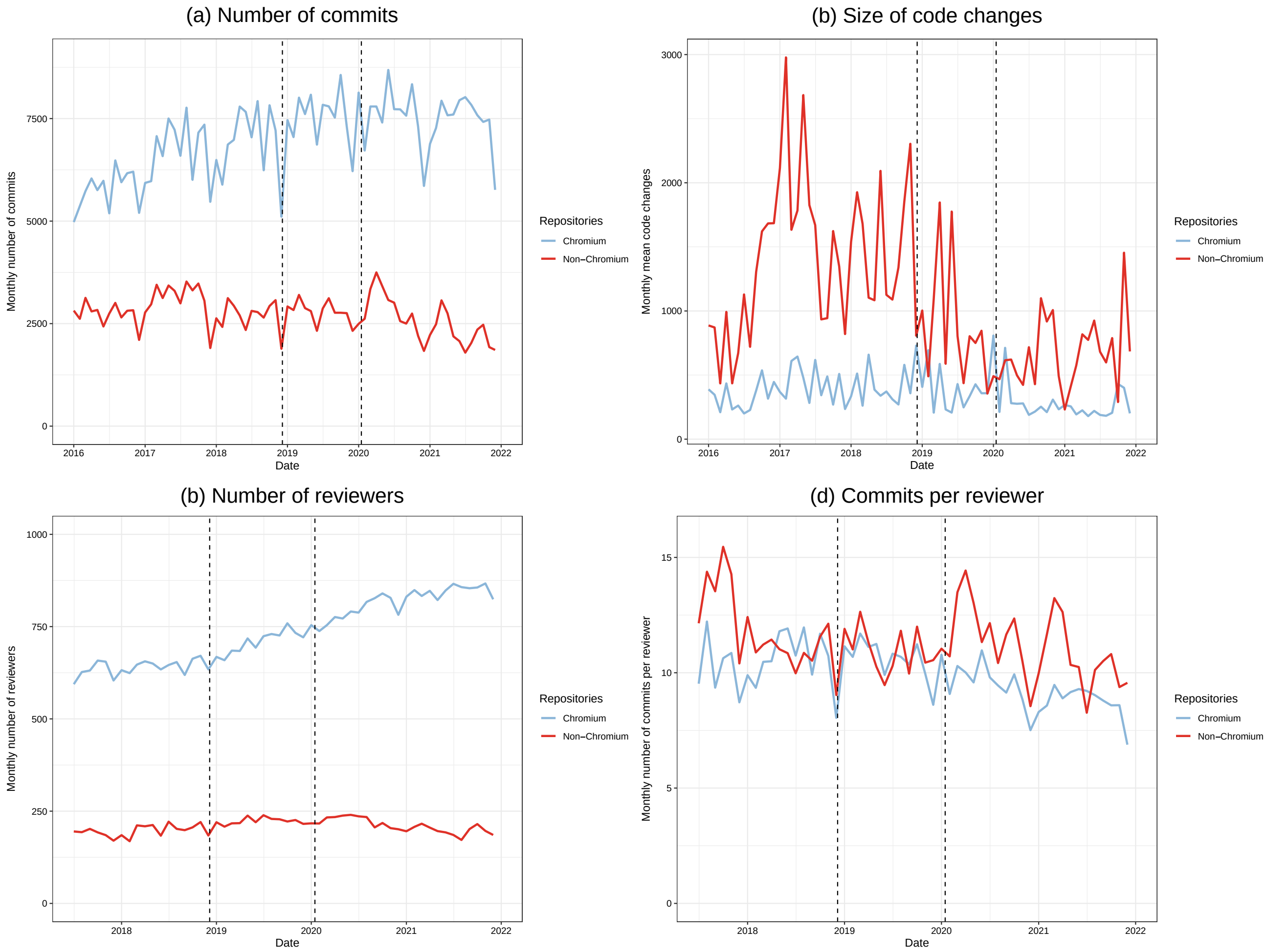
We first compare changes in development activity between Chromium and the other OSS web browser engines WebKit and Gecko. We then zoom in to Chromium and investigate how the activities of different actors have changed, namely Google, Microsoft, and all other ("external") contributors.

4.1.1. Comparing Chromium to Other Projects. We first focus on model-free evidence in Figure 1. Panel (a) shows that the total number of commits made to the Chromium repository experienced a slow increase beginning in 2016 with seasonal decreases in activity during year-end periods. However, this weak upward pattern tapers off and flattens after Microsoft's announcement to join Chromium (first vertical dashed line). The average total number of monthly commits in non-Chromium projects is stable over the entire period. In Panel (b), the average number of monthly code changes in the Chromium project is stable throughout the observed period. In non-Chromium projects, we observe strong variations in the size of code changes from month to month. Besides, we see a drop after Microsoft's announcement and a continued decrease after the first release of Chromium Edge. Panel (c) shows that the number of code reviewers in the Chromium project follows a similar trend as the number of code reviewers in the average non-Chromium project before the announcement. After the announcement, the number of Chromium code reviewers increases, and this trend continues after the first release. In Panel (d), we look at the workload of code reviewers, an important measure of software quality. We see that the number of commits per reviewer follows a similar trend in Chromium and the average non-Chromium project during the announcement period, but trends seem to diverge after the release of the first Chromium-based Edge.

We now turn to regression results, which let us quantify changes in development activity. Estimates in column (1) of Table 3 suggest that the difference between commits to the Chromium project and other projects is 19% ($e^{0.17} - 1$) higher during the announcement period compared with the average in the period before, and not statistically different from zero after the first release. This suggests there was substantial development activity leading up to the first release of Chromium Edge. The estimates in column (2) suggest no corresponding increase in the size of code changes.[34] Column (3) implies that the number of code reviewers in the Chromium project increased more than for the non-Chromium projects following the first release of Chromium Edge. However, as shown in column (4), there is no significant change in the workload of code reviewers during the announcement period, and after the first release of Chromium Edge.

4.1.2. Comparing Contributions Within Chromium. We now focus on the evolution of the number and size of contributions inside the Chromium project. In particular, we distinguish between contributors from Google, Microsoft, and other external contributors.

Figure 1. (Color online) Development Activity: Chromium vs. Non-Chromium



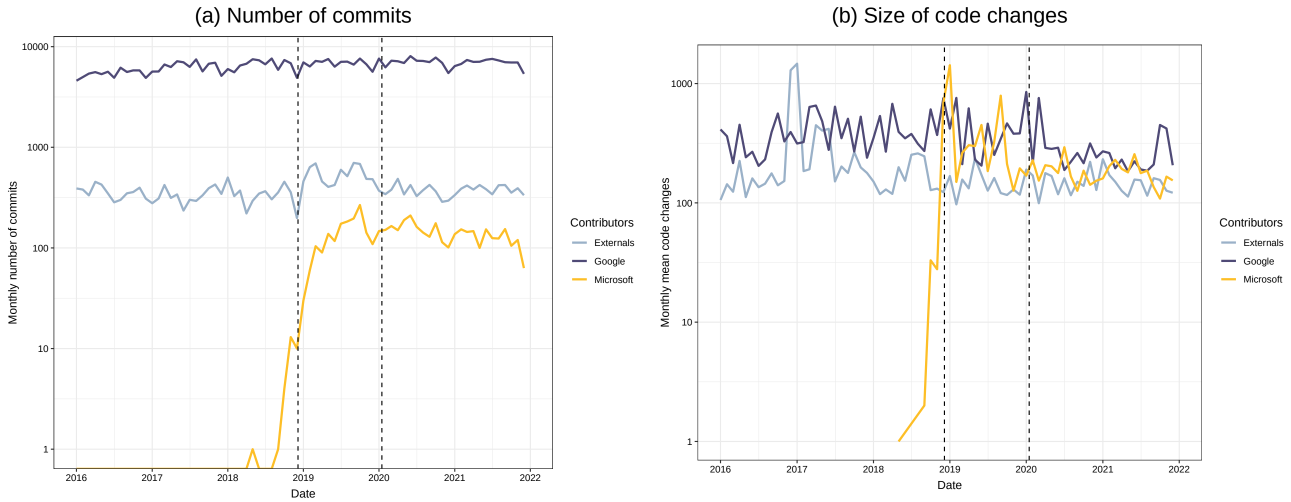
Notes. Data from Github.com. Panel (a) depicts the monthly mean number of commits for Chromium and non-Chromium repositories. Panel (b) depicts the monthly mean code changes, measured as the number of lines of code edited, for Chromium and non-Chromium repositories. Panel (c) depicts the monthly mean number of reviewers for Chromium and non-Chromium repositories. Panel (d) depicts the monthly mean ratio of commits per reviewer for Chromium and non-Chromium repositories. The first vertical line corresponds to Microsoft's announcement to adopt Chromium. The second vertical line marks the release of the first Chromium-based version of Edge.

Table 3. Changes in Development Activity

	(1) Commits	(2) Code changes	(3) Reviewers	(4) Commits per reviewer
Constant	4.07** (0.65)		4.92** (0.79)	11.78*** (0.74)
Chromium	0.88 (0.65)	−0.07 (0.03)	1.54 (0.79)	−1.32 (0.74)
Announcement	−0.14** (0.03)	−0.03 (0.01)	0.11** (0.02)	−0.81 (0.74)
PostRelease	−0.14 (0.19)	−0.06*** (0.01)	0.09 (0.03)	−0.68 (1.18)
Chromium × Announcement	0.17** (0.03)	−0.04* (0.01)	0.00 (0.02)	0.98 (0.74)
Chromium × PostRelease	0.19 (0.19)	−0.01 (0.00)	0.16** (0.03)	−0.68 (1.18)
Clustered SEs	Yes	Yes	Yes	Yes
Committer Organization FE	No	Yes	No	No
Observations	6,573	901,114	162	162
R^2	0.14	0.02	0.52	0.14

Notes. The omitted category is non-Chromium. In column (1), the dependent variable is the log(+1) number of daily commits. In column (2), the dependent variable is the log(+1) number of code changes. In column (3), the dependent variable is the log of the monthly number of reviewers. In column (4), the dependent variable is the monthly number of commits per reviewer. Standard errors in parentheses are clustered at the repository level in (1), (3), and (4), and at the committer organization level in (2).
$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Figure 2. (Color online) Development Activity Within Chromium



Notes. Data from Github.com. Panel (a) and (b) show respectively the monthly number of commits and the monthly mean code changes, measured as the number of lines of code edited, made to the Chromium repository by Google employees (users with @google.com and @chromium.org emails), Microsoft employees (users with @microsoft.com email), and external contributors. The y-axis is log-10 transformed. The first vertical line corresponds to Microsoft's announcement to adopt Chromium. The second vertical line marks the release of the first Chromium-based version of Edge.

Model-free evidence is depicted in Figure 2. In Panel (a), we see that Google's contributions remain at roughly the same levels over the entire period. Microsoft does not contribute at all to Chromium before 2018 and only starts to do so a few months before the official announcement. This timing coincides with an earlier partnership between Microsoft and Google.[35] The number of commits made by external contributors is stable before the announcement, increases during the announcement period, and returns to earlier levels after the first release.

In Panel (b), looking at the size of code changes, we see a stable trend for Google, followed by a slight drop during the announcement period, and a further drop after the first release of Chromium Edge. The size of code changes of external contributors does not appear to change systematically. Microsoft's code changes increase dramatically in size during the announcement period and decrease to about the same level as Google and external contributors after the first release.

Results of the corresponding regressions are reported in Table 4. Column (1) suggests that Microsoft increases the daily number of commits 2.5 times more than Google during the announcement period, and close to 3 times more than Google after the first release. External contributors also increase their commits more than Google, by 30% in the announcement period and by 1% in the period after the first release of Chromium Edge. These estimates suggest that most of the increase in activity we observe in the Chromium repository is driven by Microsoft's engagement and to a smaller extent by external contributors during the announcement period. The results in column (2) imply that Microsoft's code changes increase close to 3 times more than Google's code changes during the announcement period and after the first release. The dynamics in code changes by external developers are not statistically different from those of Google in either period. It is important to note that we cannot separately identify whether an increase in non-Microsoft contributions is caused only by Microsoft entering Chromium or by Microsoft's increased development activities after they have entered Chromium (i.e., spillovers from Microsoft's development activities).

Table 4. Changes in Development Activity Within Chromium

	(1) Daily number of commits	(2) Code changes
Constant	4.89*** (0.00)	3.79*** (0.01)
Microsoft	−4.88*** (0.00)	−1.00*** (0.01)
External	−2.66*** (0.00)	−0.07 (0.12)
Announcement	−0.02*** (0.00)	−0.06*** (0.01)
PostRelease	0.05*** (0.00)	−0.06** (0.03)
Microsoft × Announcement	1.26*** (0.00)	1.38*** (0.01)
External × Announcement	0.26*** (0.00)	0.06 (0.12)
Microsoft × PostRelease	1.38*** (0.00)	1.38*** (0.03)
External × PostRelease	0.01*** (0.00)	−0.07 (0.29)
Clustered SEs	Yes	Yes
Observations	6,558	503,556
R^2	0.76	0.001

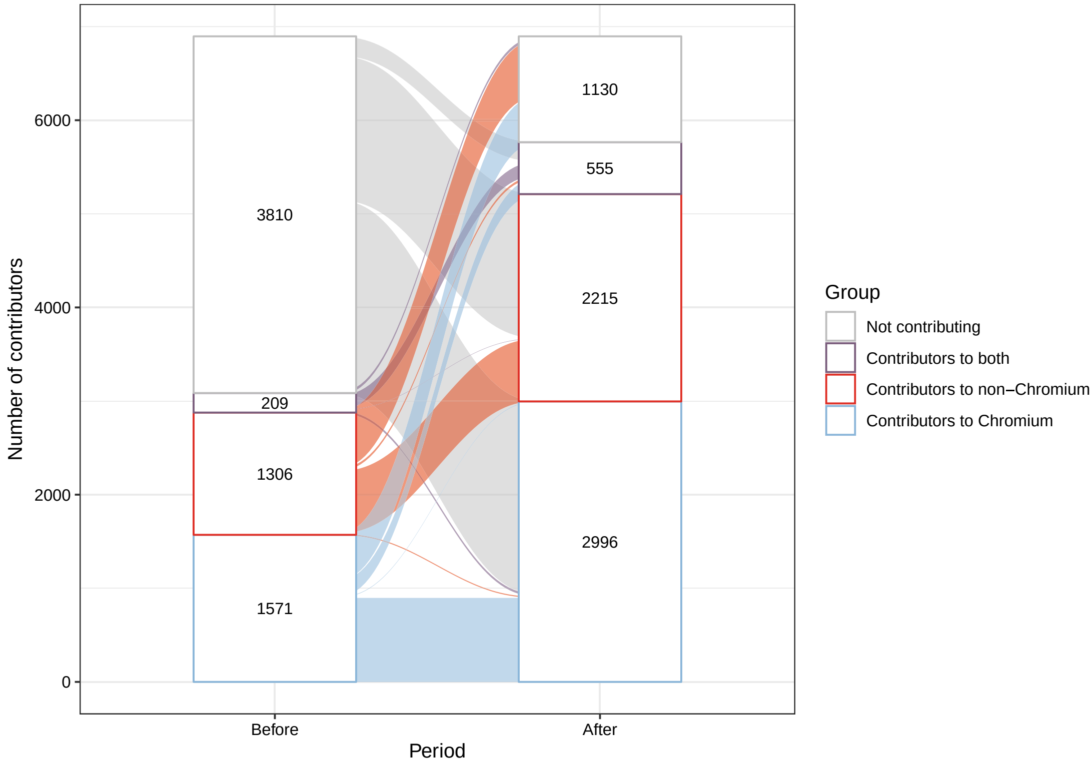
Notes. The omitted category is Google. In column (1), the dependent variable is the log(+1) number of daily commits, and standard errors are clustered at the organization group level. In column (2), the dependent variable is the log(+1) number of code changes, and standard errors in parentheses are clustered at the committer organization level.
$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

4.1.3. Changes in the Composition of Contributors. We now turn to comparing the pool of individual contributors to Chromium and non-Chromium repositories before and after Microsoft's announcement to join Chromium. This lets us quantify how many Chromium contributors continue to contribute, switch to competing projects, or stop contributing following Microsoft's announcement to join Chromium.

Figure 3 visualizes changes in the composition of contributors to Chromium and other projects. Out of the 1571 contributors exclusive to the Chromium repository initially, 897 continued to contribute to Chromium, 494 stopped contributing, 174 started contributing to both Chromium and non-Chromium repositories while only 6 stopped contributing to Chromium and started instead to contribute to non-Chromium projects after Microsoft's announcement. We see a large increase in contributors after Microsoft's announcement to join Chromium. The largest chunk of those additional contributors is new and has no history of contributing to either Chromium, WebKit, or Gecko. We observe a substantial refresh in the composition of the pool of contributors with 1,130 users ceasing their contributions replaced by 3,810 newcomers to the three projects. About 6% of these new contributors are employees of Microsoft. Within Chromium, Microsoft employees account for 10% of the new contributors.

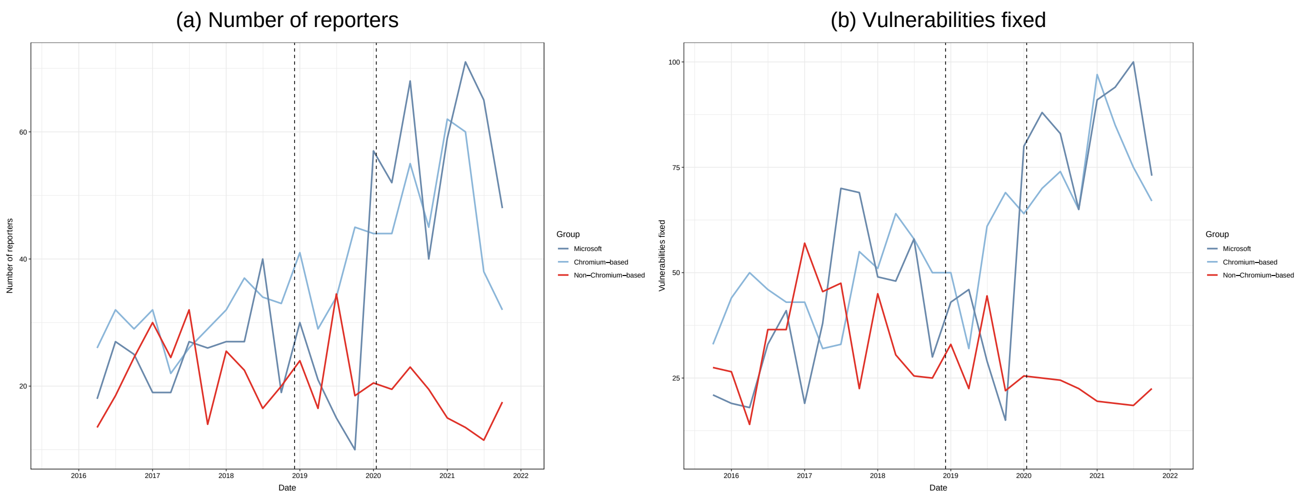
To better understand how the development activity within Chromium has changed, we further explore heterogeneity in Online Appendix A.2.1. First, we differentiate between individual and company contributors based on their email domains. We find that while both increased their activity postannouncement, individual contributors sustained higher levels postrelease, whereas contributors affiliated with companies decreased their activities after the release of the first version of Chromium-based Edge. Second, as Edge moved to Chromium, it also became available on Apple's MacOS, which could have motivated Mac-specific developers to contribute to Chromium. To test this, we examine contributions from individuals who previously contributed to WebKit (labeled as Apple Community contributors), the technology underlying Apple's Safari browser. We find that Apple Community contributors actually reduced their involvement in Chromium, while non-Apple Community contributors increased theirs. Finally, we show that the influx of external contributions to Chromium is driven by

Figure 3. (Color online) Changes in the Composition of Contributors



Notes. Data from Github.com. The figure depicts changes in the composition of contributors to Chromium and non-Chromium repositories. We differentiate four groups of contributors: the first group is composed of contributors who do not contribute during one period, the second group is composed of contributors to both Chromium and non-Chromium repositories, while the third and fourth groups contribute exclusively to Chromium or non-Chromium repositories. The Before period consists of the year before Microsoft's announcement to join Chromium and the After period starts after Microsoft's announcement until January 2022.

Figure 4. (Color online) Security Vulnerabilities and Patching



Notes. Data are collected from the browser security update release notes, on the corresponding web pages of browsers. Panel (a) shows the quarterly number of vulnerability reporters for Microsoft, Chromium, and non-Chromium browsers. Panel (b) shows the quarterly number of vulnerabilities fixed in Microsoft, Chromium, and non-Chromium browsers. Microsoft group is composed of Edge Legacy and Chromium-based Edge while Chromium covers Google Chrome, and the non-Chromium group consists of Mozilla Firefox and Apple's Safari. The first vertical line corresponds to Microsoft's announcement to adopt Chromium. The second vertical line marks the release of the first Chromium-based version of Edge.

completely external individuals as well as a community close to Microsoft. We show that individuals who are active in Microsoft's open source repositories contribute more to Chromium after Microsoft's decision to join Chromium and after the release of the first version of Chromium-based Edge. Overall, the findings suggest that Microsoft managed to pull its OSS community into the Chromium project—at least that part of the community that is not affiliated with commercial entities.

4.2. Effects on Security and Patching

Panel (a) in Figure 4 shows the quarterly number of vulnerability reporters over time for the three browser families Chromium (Chrome), non-Chromium (Safari and Firefox), and Microsoft (Edge Legacy and Edge). While the number of reporters stagnates for non-Chromium-based browsers, we see a large increase for Microsoft and Chromium browsers after the first release of Chromium Edge. Panel (b) shows similar trends. The number of vulnerabilities fixed in the preannouncement period appears similar among the three groups before a divergence occurs after Microsoft adopted Chromium.

Quantitative estimates of these changes are provided in Table 5. The results in column (1) imply that there was no significant movement in terms of the number of security reporters during the announcement period. After Microsoft releases the first version of their Chromium-based browser, the number of reporters related to Edge increases by 271%. This result is consistent with the idea that reporters respond to monetary incentives because Microsoft introduced a bounty program for Chromium-based Edge in August 2019.[36] The results in column (2) suggest that Microsoft and Google fix 27% and 32% more vulnerabilities in the announcement period and 421% and 239% more after the first release of Chromium Edge, respectively.

4.3. Effects on Product Quality and Firm Performance

In this section, we report the results of our exploratory analysis of how Microsoft's decision to join Chromium has affected the product quality of Edge, using

Table 5. Changes in Vulnerability Reporting and Fixing

	(1) Number of reporters	(2) Vulnerabilities fixed
Constant	3.00*** (0.16)	3.43*** (0.11)
Microsoft	0.19 (0.16)	0.03 (0.11)
Chromium	0.40* (0.16)	0.36** (0.11)
Announcement	0.09 (0.10)	−0.09 (0.05)
PostRelease	−0.46 (0.40)	−0.68 (0.51)
Microsoft × Announcement	−0.17 (0.10)	0.24** (0.05)
Chromium × Announcement	0.16 (0.10)	0.28** (0.05)
Microsoft × PostRelease	1.31** (0.40)	1.65** (0.51)
Chromium × PostRelease	0.91 (0.40)	1.22* (0.51)
Clustered SEs	Yes	Yes
Observations	92	104
R^2	0.46	0.42

Notes. The omitted category is non-Chromium. In column (1), the dependent variable is the log number of quarterly reporters. In column (2), the dependent variable is the log number of quarterly vulnerabilities fixed. Data are aggregated at the quarter level. Standard errors are clustered at the browser level in parentheses.
*$p < 0.1$; **$p < 0.05$; ***$p < 0.01$.

measures of product functionality and user experience. We then focus on the effects on firm performance.

4.3.1. Product Functionality. A first measure of product quality is the count of functionalities. We also look at the pace of implementation of new functionalities, and then the relative improvement across browsers. Finally, we compare overlap in functionalities across browsers and browser versions to understand spillovers across and within Chromium and non-Chromium-based browsers.

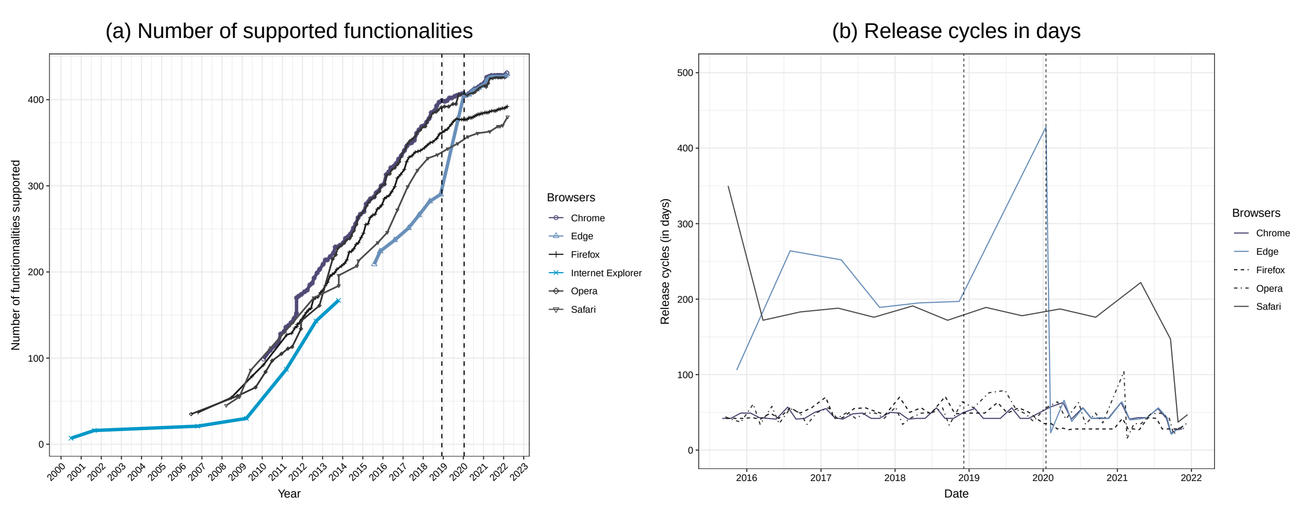
Panel (a) of Figure 5 shows that the number of functionalities offered by web browsers overall has increased extensively in the past two decades. The dashed vertical lines indicate the dates of Microsoft's announcement to join Chromium and the release of the first Chromium-based version of Edge. It is striking that Microsoft's browsers Internet Explorer and Edge have lagged behind the competition for a long time. Only with the switch to Chromium technology Microsoft was able to catch up. Note that this is not obvious ex-ante, because proprietary browsers may add or remove functionalities that are supported by the underlying open source core. Indeed, we also observe on Panel (a) that even if Chrome, Opera, and Chromium-based Edge are built on the same open source core they do not support the exact same set of functionalities.

Each data point plotted in Panel (a) represents a released version. With Microsoft's intervals between two points becoming shorter after the adoption of the Chromium technology, we see that Microsoft adopts a faster release cycle. This is confirmed in Panel (b), showing the evolution of release cycles (i.e., the number of days between two major releases). Microsoft used to ship new web browser versions around the releases of their operating system, which is also what Apple does with Safari. Looking at the trends over time indicates that there is continuous improvement in all technologies. Microsoft, however, updates Edge at a significantly slower pace than the average Chromium and non-Chromium browsers. After switching to Chromium, Microsoft also moves to fast release cycles like most other browsers.[37]

There are likely decreasing returns to the number of functionalities a browser offers and variations in the importance of functionalities. Qualitative evidence from a survey among web developers carried out in 2019 suggests that missing interoperability across browsers is a key barrier to adopting a new technology.[38] To capture that, we also study relative changes, that is, the distance of a browser toward the market leader in terms of functionalities as well as a browser's overlap in functionalities with the market leader.

Table 6 provides estimates of the magnitudes of the effects. In columns (1) and (2), we see that, compared with the rest of the market, Microsoft's change in the number of functionalities is 36% higher, and that Microsoft can reduce its distance to the market leader by 100 functionalities after the release of Chromium-based Edge. Further, we see that the distance toward the market leader only reduces for Microsoft and not for the competitors. In column (3), we estimate that Chromium Edge has 29 more functionalities that overlap with the market leader compared with the average non-Chromium browser. Finally, the estimates in column (4) suggest that Microsoft's release cycles become 72% shorter, reducing to an average of about 53 days, which is faster than the average release cycle in the market.

Figure 5. (Color online) Functionalities and Release Cycles



Notes. Data from caniuse.com. Panel (a) depicts the number of functionalities supported by each browser version. Each data point corresponds to the release dates of new versions for each browser. Panel (b) depicts the number of days between releases of browser versions. The first vertical line corresponds to Microsoft's announcement to adopt Chromium. The second vertical line marks the release of the first Chromium-based version of Edge.

Table 6. Changes in Functionality and Release Cycles

	Functionalities (1)	Distance (2)	Overlap (3)	Release cycles (4)
Constant	5.70*** (0.05)	42.18** (13.41)	78.93*** (3.32)	4.20*** (0.34)
Microsoft	−0.20** (0.05)	54.73** (13.41)	−13.35** (3.32)	1.07** (0.34)
Chromium	0.13** (0.05)	−40.83** (13.43)	20.36*** (3.33)	
Announcement	0.18*** (0.02)	3.17 (2.33)	3.72*** (0.78)	0.06 (0.07)
PostRelease	0.22*** (0.02)	4.69 (4.31)	4.80*** (0.94)	−0.24* (0.10)
Microsoft × Announcement	−0.01 (0.02)	12.39*** (2.33)	−1.41 (0.78)	−0.04 (0.07)
Chromium × Announcement	−0.02 (0.02)	−1.32 (2.99)	−4.07** (0.94)	
Microsoft × PostRelease	0.31*** (0.02)	−100.28*** (4.31)	29.19*** (0.94)	−1.28*** (0.10)
Chromium × PostRelease	−0.02 (0.02)	−4.63 (4.32)	−4.59*** (0.94)	
Clustered SEs	Yes	Yes	Yes	Yes
Observations	405	405	498	11,385
Adjusted R^2	0.79	0.90	0.94	0.33

Notes. The omitted category is non-Chromium in columns (1), (2), (3), and non-Microsoft in column (4). The dependent variable in column (1) is the log number of functionalities. In column (2), the dependent variable is the distance in the number of functionalities with the web browser supporting the most functionalities. In column (3), the dependent variable is the overlap in the number of functionalities with the web browser supporting the most functionalities. The dependent variable in column (4) is the log duration of days between two major version releases. Standard errors are clustered at the browser level in parentheses.

*$p < 0.1$; **$p < 0.05$; ***$p < 0.01$.

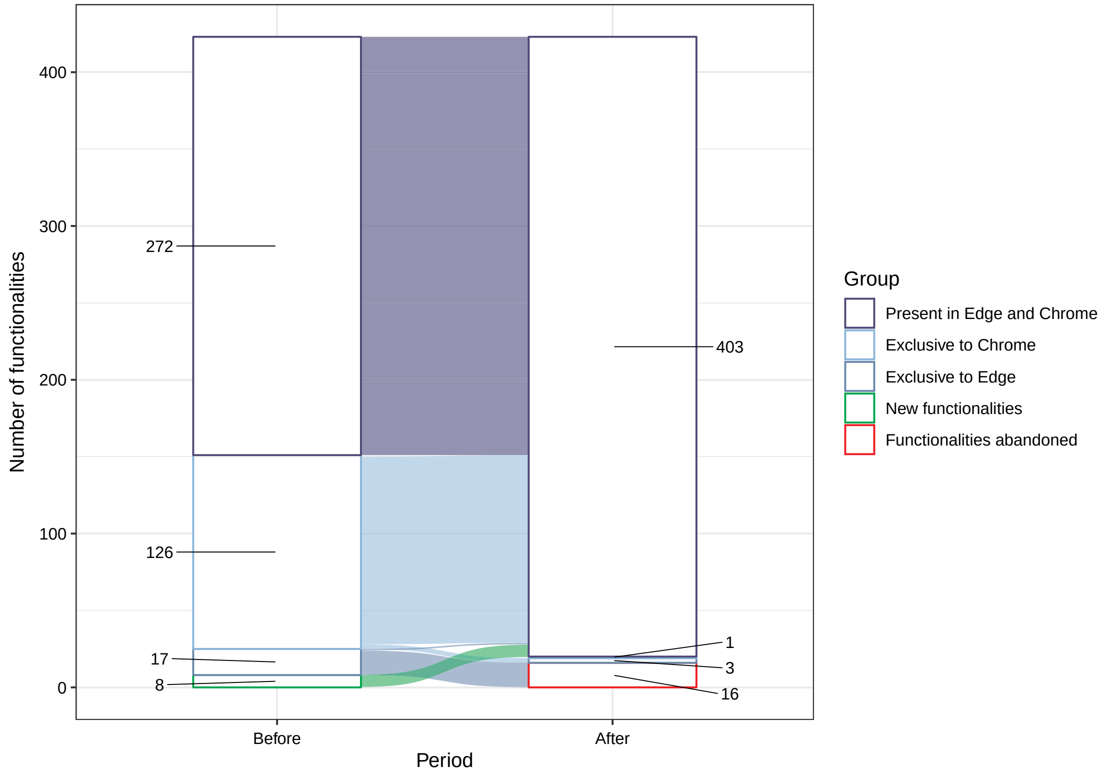
We now look at the overlap between Chrome and Edge in more detail. Figure 6 shows the flows and the overlap of features supported by Microsoft Edge and Chrome comparing functionalities supported by versions released just before and after Microsoft's adoption of Chromium. Edge 18 is the last version of Microsoft Edge based on proprietary technology, released in November 2018. Edge 79 is the first version of Microsoft Edge based on Chromium technology, released in January 2020. Between these two releases, Google had multiple updates to their Chrome browser. We therefore compare Edge 18 to Chrome 71, released in December 2018, and Edge 79 to Chrome 80, released in February 2020. Only one functionality from Legacy Edge is present in both Edge 79 and Chrome 80 while Edge adopted 125 functionalities that were present in Chrome 71. Finally, five functionalities were jointly adopted (125 versus 130), three were only adopted by Edge and one was only added to Chrome. Out of the 17 functionalities exclusive to Edge in the preperiod, 16 were abandoned.

The finding that Chromium-based Edge has substantially more functionalities than Legacy Edge may not be entirely surprising. However, it is important to note that it is not impossible for Microsoft to decide not to support specific functionalities. Hence, it is a choice to support as many functionalities as they do, most likely as a result of a cost-benefit consideration. Further, it is not obvious that Microsoft did not add exclusive functionalities for web development. Given that Microsoft is the market leader for productivity applications (i.e., Microsoft Office and Outlook), they could have added functionalities that make it more convenient for users to use the online version of Office or Outlook in the Edge browser. However, there might be functionalities that are outside the scope of the data we have from caniuse.com. For example, Edge now has a toolbar that integrates an AI assistant, which Chrome and the other Chromium browsers do not offer.

4.3.2. Page Load Time. We now move to page load time as a measure of product quality from a user experience perspective. Model-free evidence in Figure 7 depicts the evolution of the average loading time for a simulated web browsing session for different browser versions. The dates on the horizontal axis correspond to the release dates of each browser version used in our tests. First, we notice that both Chrome (Chromium) and Firefox (non-Chromium) tend to get faster with consecutive updates. Second, we see that Edge was actually faster before Microsoft adopted the Chromium technology.

Estimates are provided in Table 7. In column (1), we estimate that the loading time of Microsoft Edge increased by more than 1,000 milliseconds after adopting Chromium. This suggests that there might be a trade-off between loading time and functionalities. A striking element of Figure 7 is that all browsers seem to converge to a common loading time in the period after Microsoft adopts Chromium. Taking into consideration that there might be a natural limit to how fast a web page can be rendered, we scale our measure of loading time by relating it to the minimum observed value in column (2). The results show the same patterns as in column (1). To further investigate the potential trade-off between loading time and functionalities, we look at the average loading time per functionality in column (3). Also here, we find the same patterns. While Edge outperformed the other browsers before Microsoft adopted Chromium, Edge is the

Figure 6. (Color online) Overlap in Functionalities



Notes. Data from caniuse.com. In the Before period, we compare Edge 18 (the last version of Microsoft Edge based on their proprietary technology, released in November 2018) with Chrome 71 (released in December 2018). We look at the flows of functionalities after Microsoft adopts Chromium. We, therefore, compare Edge 79 (the first version of Microsoft Edge based on Chromium technology, released in January 2020) with Chrome 80 (released in February 2020). The plot reads as follows: 272 functionalities that were present in Edge 18 and Chrome 71 are still present in Edge 79 and Chrome 80. Out of the 126 functionalities exclusive to Chrome in the Before period, 122 are present in both Edge and Chrome in the After period. Out of the 17 functionalities exclusive to Edge in the Before period, one is present in both Edge and Chrome in the After period while 16 were abandoned. 8 new functionalities are present in both Edge and Chrome in the After period.

only browser that got slower in the period after the first version of Chromium-based Edge was released.[39]

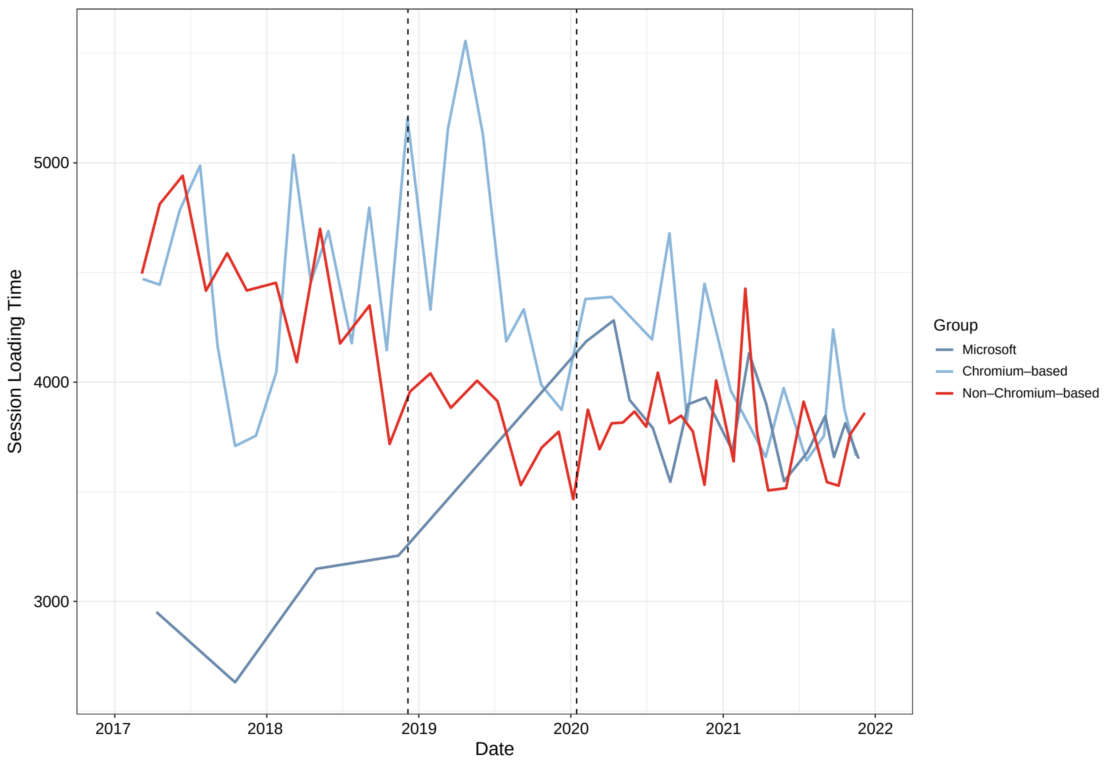
4.3.3. Firm Performance. Having established that Microsoft's decision to join Chromium has had mostly positive effects on product functionality, we now turn to firm performance. Data from StatCounter provides us with market shares of different browser versions and device types.

In this section, we take the perspective of Microsoft and divide the analysis into two layers. First, we look at the firm level. Microsoft offered two web browsers during our sample period, Edge and Internet Explorer, hence we will first combine the two market shares to compute aggregate figures for Microsoft. In a second level of analysis, we treat Edge as the focal product for Microsoft as it was the one actively developed and managed during that period.

In columns (1) and (3) of Table 8, we can see that Microsoft's aggregate market share was not significantly different from the non-Chromium browsers in the postrelease period, both overall and when only looking at desktop devices. At the firm level, the decision to adopt Chromium did not significantly change Microsoft's aggregate market share, not even on desktop devices. In columns (2) and (4), we define the *Microsoft* dummy to only include Edge, leaving Internet Explorer in the comparison group. Estimates suggest that, relative to all non-Chromium browsers, the market share of Edge increased by 4.6 percentage points overall, and by 8 percentage points on desktop devices after the first release of the Chromium-based version.

We report additional evidence in tables in the Online Appendix. In Table A.2 we directly compare the market share of Legacy Edge to the market share of Chromium-based Edge. Our estimates suggest a gain of 1.8 percentage points overall and 4 percentage points increase on desktop devices. It also enables us to dig more precisely into how web browser market shares' dynamics shifted after Microsoft joined Chromium. We see that Edge's increased market share on all devices and desktop markets did not come at the expense of Google Chrome but rather from all the other, less popular, Chromium participants and non-Chromium browsers. Microsoft's increase in market share on all devices was primarily

Figure 7. (Color online) Page Load Time



Notes. Data comes from simulations run on browserstack.com. The first vertical line corresponds to the announcement made by Microsoft to adopt Chromium. The second vertical line marks the release of the first Chromium-based Edge. Microsoft group is composed of pre-Chromium and Chromium-based Edge, Chromium covers Google Chrome while the non-Chromium group consists of Mozilla Firefox. Dates correspond to the release dates of each version for each web browser present in our sample. The web browsing session consists in loading selected web pages from craiglist.org and wikipedia.org on the same Windows 10 environment. Each data point corresponds to the release dates of new versions for each web browser.

driven by desktop usage, which represented about 40% of total web traffic in 2020.[40] We see no changes in usage on mobile devices after Microsoft releases Chromium-based Edge.

To separate a demand-side effect (i.e., users choosing Chromium-based Edge) from a supply-side effect (i.e., Microsoft effectively forcing users to adopt Chromium-based Edge), we ran another analysis. The supply-side effect can only operate on Windows because Microsoft has no power to force non-Windows users to use Edge. Exploiting variation across operating systems can therefore help to remove the supply-side effect from

Table 7. Changes in Page Load Time

	(1) Loading time	(2) Relative distance	(3) Time/functionality ratio
Constant	4,163.34*** (0.00)	1,543.34*** (0.00)	11.77*** (0.00)
Microsoft	−1,178.34*** (0.00)	−736.34*** (0.00)	−0.80*** (0.00)
Chromium	327.90*** (0.00)	794.90*** (0.00)	−0.02*** (0.00)
PostRelease	−340.94*** (0.00)	−340.94*** (0.00)	−1.84*** (0.00)
Microsoft × PostRelease	1,189.19*** (0.00)	1,189.19*** (0.00)	0.02*** (0.00)
Chromium × PostRelease	−155.81*** (0.00)	−155.81*** (0.00)	−0.42*** (0.00)
Clustered SEs	Yes	Yes	Yes
Observations	3,323	3,323	3,323
R^2	0.06	0.10	0.10

Notes. The omitted category is non-Chromium. Column (1) reports the combined loading time (in milliseconds) of craiglist.org and wikipedia.org web pages. In column (2), the dependent variable is the relative distance to the minimum observed value. In column (3), the dependent variable is a ratio of the loading time divided by the number of functionalities supported by the web browser version considered. Standard errors are clustered at the browser level in parentheses.

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

Table 8. Changes in Market Shares

	All devices		Desktop	
	Firm-level (1)	Product-level (2)	Firm-level (3)	Product-level (4)
Microsoft	−4.45* (2.13)	−8.83*** (2.53)	3.48 (4.23)	−7.01** (2.81)
Chromium	−4.27 (8.80)	−1.84 (8.86)	3.23 (13.90)	3.56 (13.40)
Announcement	−1.21 (1.98)	−1.97 (1.47)	−1.61 (2.34)	−2.70 (1.81)
PostRelease	−1.37 (2.77)	−2.79 (2.21)	−1.67 (3.59)	−3.99 (3.11)
Microsoft × Announcement	−2.00 (1.98)	2.25 (1.47)	−2.19 (2.34)	3.77* (1.81)
Chromium × Announcement	2.41 (2.24)	3.17 (1.80)	3.43 (2.91)	4.51 (2.49)
Microsoft × PostRelease	−2.49 (2.77)	4.56* (2.21)	−2.93 (3.59)	8.00** (3.11)
Chromium × PostRelease	2.78 (3.03)	4.20 (2.52)	3.70 (3.98)	6.01 (3.55)
Clustered SEs	Yes	Yes	Yes	Yes
Country FE	Yes	Yes	Yes	Yes
Observations	22,995	25,550	17,885	20,440
R^2	0.01	0.02	0.02	0.04

Notes. The omitted category is non-Chromium. In columns (1) and (3), Microsoft's market share is measured as the sum of Internet Explorer and Edge market shares. In contrast, in columns (2) and (4), we measure Microsoft's market share only for Edge web browser, and Internet Explorer is treated as a non-Chromium competitor browser. In all four columns, market shares cover 35 industrial countries. Standard errors are clustered at the browser level in parentheses.
$^*p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

our estimates, by specifying fixed effects for the operating system. For this, we need to access data at the browser-operating system level, which is not available from StatCounter. We therefore use data on visits to U.S. federal government agencies' websites.[41] These data, albeit with the caveat of a restricted set of visited websites, provide granularity daily usage data at the browser-operating system level. Table A.3 presents estimates consistent with the baseline effects reported in Table 8. We find a positive effect at the product level (treating Internet Explorer as a competitor) on all devices. We even estimate a slight positive increase in usage at the firm level (i.e., when summing Internet Explorer and Edge usages) when controlling for time-invariant effects at the operating system level.

4.4. Robustness

4.4.1. Alternative Econometric Models. In several specifications of the main paper, we use a log transformation of the dependent variable ($log(y+1)$). Given that our outcome variable is a count in some of these specifications, that is, a nonnegative integer, one might be worried that OLS is not the optimal choice of estimator (Wooldridge 2023, Chen and Roth 2024). Therefore, we report another set of estimates to show that our results are robust to alternative specifications. Tables A.4–A.10 report the results of OLS (with and without log transformation) and Poisson specifications without log transformation for all the count variables included in our study (number of commits, number of code changes, number of reviewers, number of vulnerability reporters, number of vulnerabilities fixed, number of functionalities, and duration of release cycles).

Tables A.4–A.6 reproduce the results related to changes in development activity. The results remain quantitatively similar among the three specifications for the number of commits, the size of code changes, and the number of reviewers. In Table A.4, the general interpretation remains that the number of commits in the Chromium repository increases after Microsoft announced joining the Chromium project relative to the non-Chromium repositories. Columns (2) and (3) suggest that the increase persists after Microsoft releases the first version of Chromium-based Edge. In Table A.5, Columns (2) and (3) show that the size of code changes in the Chromium repository increases comparatively to the non-Chromium in the announcement and after the first release of Chromium Edge, suggesting a stronger impact of Microsoft's entry into Chromium compared with our baseline results. Table A.6 confirms that the number of reviewers increases for the Chromium repository after Microsoft releases the first version of Chromium-based Edge.

Tables A.7 and A.8 reproduce the results related to vulnerability reporting and fixing. The results remain quantitatively similar among the three specifications for the number of reporters and the number of vulnerabilities fixed. We note an increase in the number of reporters and vulnerabilities fixed for both Microsoft and Chromium web browsers compared with non-Chromium web browsers after Microsoft adopted the Chromium technology.

Finally, Tables A.9 and A.10 reproduce the results related to the number of functionalities supported and release cycles. The results are consistent among the three specifications for the two dependent variables. We find an increase in functionalities supported by

Microsoft Edge and a reduction of the duration between two major releases.

4.4.2. Alternative Control Group. To address potential concerns regarding the use of Gecko and WebKit repositories as the comparison group, we conducted a robustness check using the Linux repository as an alternative comparison. This choice was motivated by several factors: Linux is similar in size to Chromium, and both Microsoft and Google are substantial contributors to Linux, suggesting similar general dynamics might be at play.[42] Moreover, using Linux as a control group mitigates potential substitution effects, as operating system developers and web browser developers are likely to be specialized in their respective technologies.

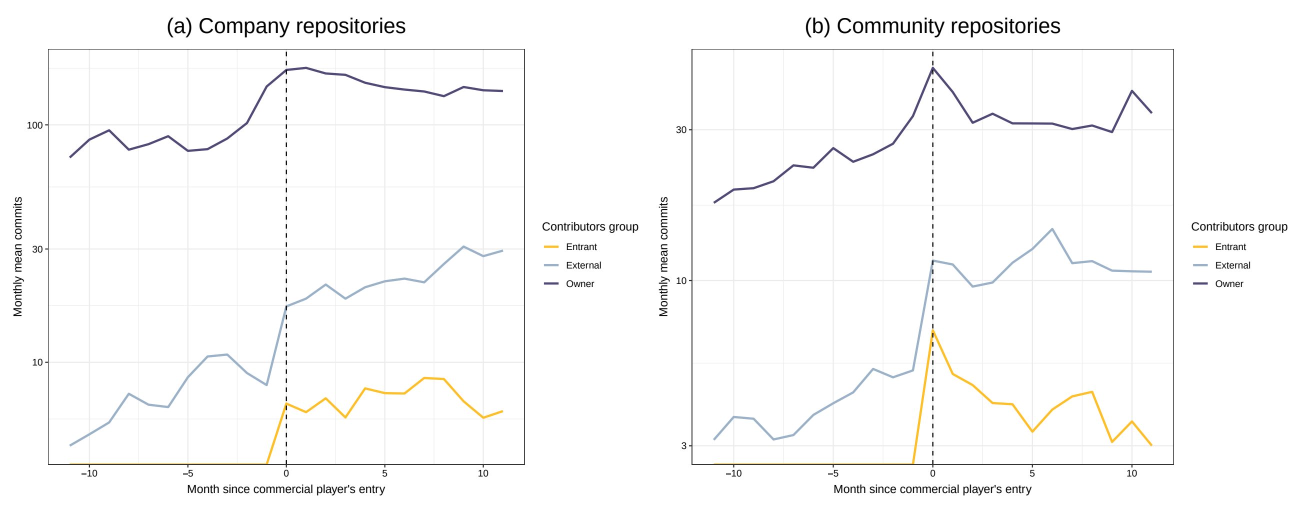
We collected data on commits to the Linux repository from GitHub and replicated our main analysis. Figure A.7 suggests that after the announcement of Microsoft's decision to join Chromium, contributions to Chromium increased at a higher rate than those to Linux. Table A.11 presents results from three econometric specifications. Column (1) shows an OLS model with $log(1 + commits)$ as the dependent variable. Column (2) presents an OLS model with the number of commits (without log transformation) as the dependent variable. Column (3) displays results from a Poisson model on the number of commits. Columns (2) and (3) corroborate the visual inspection of Figure A.7, demonstrating a significant effect in the postannouncement period, and an additional significant, albeit smaller, effect in the postrelease period. These results align with our main findings using Gecko and WebKit as control groups. However, it is important to note a caveat. The $log + 1$ transformation appears to perform poorly in this case (see Chen and Roth 2024). The OLS regression without log transformation and the Poisson model in columns (2) and (3) contradict the results we get in column (1). Despite this limitation, the overall consistency of results using Linux as an alternative control group strengthens the robustness of our main findings.

4.5. External Validity: Commercial Entry into Open Source AI

Finally, we explore the boundary conditions of the findings of the particular case of Microsoft joining an open source project that was already under the stewardship of a commercial entity. We aim to get answers to the hypothetical question of how the developer community would have responded had Microsoft entered a community-led/noncommercial open source project. To explore this question, we analyzed 13,740 AI repositories on GitHub, which vary in governance (community-driven versus commercial), contributor numbers, and the presence of commercial entities. By comparing these cases with Microsoft's involvement in Chromium, we aim to understand the broader effects of commercial entry on open source development across different project types.

Figure 8 shows model-free evidence. In both company repositories (panel (a)) and community repositories (panel (b)), we observe a sudden increase in the

Figure 8. (Color online) Development Activity in AI Open Source Repositories Before and After Commercial Entry



Notes. Data from Github.com. Panel (a) and (b) show respectively the monthly number of commits made to company repositories and community repositories by repository owner contributors, commercial entrant contributors, and external contributors. Company repositories are defined as repositories that do experience at least one company commit during the first month of existence. Community repositories are defined as repositories that do not experience a company commit during the first month of existence. The vertical line corresponds to the first entry of a commercial player. We define the entry of a commercial player as either the first commit made to a community repository or the earliest commit made by a company that was not contributing in the first month of existence in company repositories. We use the same definition to define entrant commits. Repository owners are defined as contributors from organizations who contribute in the first month of the repository's existence. Externals are organizations that are neither owners nor entrants. The y-axis is log-10 transformed.

number of commits of external contributors and the repository owner in the month following the entry of a commercial player. The number of commits sustains in the twelve months after the entry and remains higher than pre-entry levels. These patterns are in line with what we observe in the case of Microsoft's entry into Chromium panel (a) of Figure 2.

To study external validity in the context of AI repositories, we estimate the following specification:

$$Commits_{it} = \beta_1 Post_{it} + External_{it}(\beta_2 + \beta_4 Post_{it}) + Entrant_{it}(\beta_3 + \beta_4 Post_{it}) + \alpha_i + \gamma_t + \varepsilon_{it}, \quad (3)$$

where $Commits_{it}$ is the number of commits in repository i during month t, α_i and γ_t are repository and month fixed effects. Equation 3 enables us to estimate the average treatment effect. $Post_{it}$ is a dummy taking a value of 1 if the observation occurs after the entry of a commercial firm in repository i. $External_{it}$ is a dummy taking a value of 1 if the commits are made by external contributors in repository i during month t. $Entrant_{it}$ is a dummy taking a value of 1 if the commits are made by commercial entrant contributors in repository i during month t. We split our sample in two to distinguish between community and company repositories.

Table 9 reports regression results. Looking at both company and community projects, we find an increase in commits made by external contributors as well as repository owner contributors after the entry of a commercial player. The fact that we do not find any negative effects in either setting corroborates our core findings in the case of Microsoft entering Chromium, where we document overall positive effects, both in terms of commits from Microsoft as well as from other external developers. It also suggests that the surge in activity following the entrance of a commercial player is not conditional on the initial nature of the project (company versus community). Hence, what we find in the case of Microsoft and Chromium seems to bear external validity and the insights translate to other domains of substantial commercial appeal. However, the generalizability to projects in less commercially driven settings may be limited. Additionally, while we observe short to medium-term effects, long-term impacts and potential shifts in project governance or community dynamics remain areas for further investigation.

Table 9. Changes in Development Activity in AI Open Source Repositories After Commercial Entry

	Commits	
	Company (1)	Community (2)
Entrant	−2.70*** (0.09)	−1.53*** (0.04)
External	−1.89*** (0.08)	−1.06*** (0.04)
Post	0.33*** (0.06)	0.29*** (0.03)
Entrant × Post	0.25*** (0.08)	0.35*** (0.04)
External × Post	0.34*** (0.07)	0.28*** (0.03)
Clustered SEs	Yes	Yes
Repository FE	Yes	Yes
Month FE	Yes	Yes
Observations	30,411	76,596
Adjusted R^2	0.55	0.42

Notes. The omitted category is the owner of the repository. In columns (1) and (2), the dependent variable is the log(+1) number of monthly commits. In column (1), we focus on company repositories, defined as repositories that do experience at least one company commit during the first month of existence. In column (2), we focus on community repositories, defined as repositories that do not experience a company commit during the first month of existence. Standard errors are clustered at the repository level in parentheses.

$^*p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

5. Discussion and Conclusion

Our goal in this study was to understand how the OSS community reacts when a private company switches from a proprietary to an open source core for a commercial product, as well as how that affects cybersecurity, product quality and firm performance.

Contrary to concerns raised by industry observers and prior theoretical work (Johnson 2002, Lifshitz-Assaf and Nagle 2021), we do not find evidence of crowding out. We document an increase in the number of individuals that report security vulnerabilities, potentially because the entry of a large commercial firm makes it (financially) more worthwhile to do so. The daily number of contributions to the Chromium repository made by Microsoft increased, as did the number of contributions from external developers. Moreover, we show substantial churn in the pool of contributors, suggesting drastic compositional changes over a short period.

The results that we present in this paper might be specific to Microsoft, its market position in the web browser market, its market position in the operating systems market, and its reputation among consumers and external open source contributors. However, Microsoft joining Chromium is not the only instance where a private company decided to use an open source project as the technological shell of their product. We find very similar patterns by looking at thousands of comparable cases in the setting of open source AI. Similar to Chen et al. (2021), who study community reactions to acquisitions, we observe an increase in activity from external contributors but our results differ as internal contributors do not reduce contributions to the focal project following the involvement of a new actor in the project. We also add to previous work on the impact of change in the competitive landscape on contributions. Our results are complementary to Nagaraj and Piezunka (2024) as we do not observe a decrease in the number of contributors joining the projects affected by competition. We also contribute to a better empirical understanding of the relationship between open source and firm performance (Nagle 2019, Lin and Maruping 2022, Conti et al. 2025). In our empirical context, we find that open source adoption is associated with an increase in market share for the focal firm.

One of the main negative implications of our findings is the potential damage to internal motivation that can occur when an organization adopts open source technology. Our study draws attention to the experience of Microsoft engineers who, after joining the Chromium project, gave up the majority of the functionalities they had previously developed. This phenomenon is often referred to as the "not invented here syndrome" and can reduce the benefits of adoption if it triggers internal resistance during the implementation process (Katz and Allen 1982). Future research could explore the potential spillovers on internal development within the organization. A deeper understanding of the potential challenges and drawbacks associated with piggybacking on an open source project will enable organizations to make more informed decisions about their technology adoption strategies.

For policymakers, it is important to recognize that commercial entry into existing open source projects results in higher concentration, which might lead to market inertia and concerns related to innovation over the long run (Aghion et al. 2005, Goettler and Gordon 2014). Evidence from a software development platform indicates that increasing competition tends to have a stronger negative effect on innovation rates among paid developers compared with unpaid ones (Miric and Jeppesen 2023). Translated into the context of web browsers, where many developers contribute to open source projects, a reduction in competition due to concentration could disproportionately impact paid contributors (i.e., employees of commercial organizations), leading to a slowdown in innovation. Innovation in web browsers is key to enabling subsequent innovations from web developers or content creators. With a single project exercising significant influence over the technological direction, there are legitimate concerns that areas like user privacy, ad-blocking, and platform openness may become deprioritized if they conflict with the commercial interests of the market leader. Competition authorities might also be concerned with the fact that web browser vendors benefiting from market power in operating systems may use it to favor their products over their competitors.[43] On the other hand, the reduction in the number of technologies can provide more standardization and compatibility between products mitigating the importance of network effects for market dominance (Chen et al. 2009).[44] Hence, the aggregate effect on social welfare is difficult to predict. It is worth stressing that the reduction in the number of web browsing technologies does not imply an automatic diminution of differentiation between web browsers. Indeed, web browser vendors still remain in control of adapting the technology to their specific offerings and to provide specific features to differentiate their web browsers from competitors. Finally, we found evidence of crowding-in effects following Microsoft's entry. Our finding that some of the new external contributors are developers that have been involved in projects maintained by Microsoft before suggests that commercial actors can have the power to influence the composition of open source communities beyond paying their own employees to contribute to open source projects.

Our exploration of open source adoption in the commercial sphere points to several important areas for future research. First, our study highlights the need to understand cross-company security vulnerability reporting practices and their impact on product reliability, especially as multiple firms collaborate on OSS projects. As major companies transition to open source cores, such as Microsoft's adoption of Chromium, there is a need to quantify efficiency gains in corporate R&D and explore whether these savings are sufficient to drive sustained innovation over time. The potential influence of proprietary features layered on top of OSS, a trend observed with Microsoft's Chromium-based Edge browser, warrants further study to assess how these add-ons impact user adoption and competitive positioning within the market. Finally, the choice of licensing models in OSS, which determines how companies can commercially leverage and contribute to projects, poses questions about the long-term legal and operational impact of such models, particularly in projects heavily influenced by commercial players. Addressing these gaps could deepen our understanding of the trade-offs and strategic choices shaping the evolution of OSS ecosystems in commercial contexts.

Acknowledgments

The authors are grateful to Lucas Franco for his excellent research assistance as well as Vincent Vandersluis for copyediting. The authors thank the editors and three anonymous referees for their helpful comments and guidance. Daniel Blaseg, Jean-Philippe Bonardi, Tim Bresnahan, Hannes Datta, Luise Eisfeld, Tobias Kretschmer, Patrick Haack, Sarit Markovich, Poonacha Medappa, Frank Nagle, Rob Seamans, Ananya Sen, Sonali Shah, Shiva Shekhar, Joy Wu, Dogukan Yilmaz and Giorgio Zanarone provided advice on prior drafts and helpful suggestions. The authors also thank conference and seminar participants at the NYU Open Source Workshop, LMU Munich, HEC Lausanne, the OFA Symposium Berlin, University of Bremen, Esade Business School, Tilburg University, the AFREN 2023 Summer School, and the Digital Economy Workshop Norwich.

Endnotes

[1] High-profile examples include Walmart building their own e-commerce architecture based on the open source technologies Node.js and React and releasing it as an open source project named Electrode and Goldman Sachs releasing software toolkits for quantitative finance on the open source development platform GitHub.

[2] For an in-depth discussion of the history of the web browser market and technologies, see Online Appendix A.1.

[3] On the rising importance of Chromium in web standardization, see https://dev.to/jehrhardt/let-s-learn-about-standardization-from-microsoft-s-move-to-chromium-5970.

[4] For more details on qualifications to become a reviewer, see https://www.chromium.org/getting-involved/become-a-committer/ and for code owners, see https://chromium.googlesource.com/chromium/src/+/HEAD/docs/code_reviews.md\#expectations-of-owners.

[5] See https://github.com/chromium/chromium/blob/main/LICENSE.

[6] See https://venturebeat.com/business/microsoft-edge-on-windows-10-the-browser-that-will-finally-kill-ie/.

[7] See https://www.theverge.com/2018/12/4/18125238/microsoft-chrome-browser-windows-10-edge-chromium and https://www.theverge.com/2020/1/16/21068387/browser-wars-microsoft-edge-google-chrome-apple-safari-chromium-webkit-goat-rodeo.

[8] See https://blogs.windows.com/windowsexperience/2018/12/06/microsoft-edge-making-the-web-better-through-more-open-source-collaboration/.

[9] See https://www.theverge.com/2019/5/6/18527550/microsoft-chromium-edge-google-history-collaboration.

[10] See https://blog.mozilla.org/en/mozilla/goodbye-edge/.

[11] See https://www.microsoft.com/en-us/msrc/bounty-new-edge.

[12] Various press reports indicate that Microsoft Edge was missing a significant number of features compared to competitors, see for instance https://www.computerworld.com/article/3513973/microsofts-new-edge-browser-third-times-the-charm.html and https://www.bleepingcomputer.com/news/microsoft/a-year-after-it-added-support-for-extension-edge-has-only-70-add-ons/.

[13] Microsoft CEO Satya Nadella was concerned about Edge's lack of capabilities to run Office collaboratively, see https://www.theverge.com/2019/5/6/18527550/microsoft-chromium-edge-google-history-collaboration.

[14] See https://www.ft.com/content/2ba17784-6862-11e8-b6eb-4acfcfb08c11.

[15] See https://www.theregister.com/2000/07/31/ms_ballmer_linux_is_communism/ and https://www.theregister.com/2001/06/02/ballmer_linux_is_a_cancer/.

[16] See for example, https://www.reuters.com/article/us-github-microsoft-gitlab-idUSKCN1J12BR.

[17] See for example, https://techcrunch.com/2022/07/01/open-source-developers-urged-to-ditch-github-following-copilot-launch.

[18] See https://www.forbes.com/sites/zakdoffman/2021/03/20/stop-using-google-chrome-on-apple-iphone-12-pro-max-ipad-and-macbook-pro/.

[19] Anecdotal evidence suggests that Microsoft has used such a strategy in the past. See, for example, how they forced webmail users to open links with Edge https://www.theguardian.com/technology/2018/mar/19/windows-10-microsoft-force-people-edge-browser-windows-mail-chrome-firefox or for the lack of compliance with an EU antitrust commission decision, https://ec.europa.eu/commission/presscorner/detail/en/IP_13_196.

[20] See for instance https://www.theverge.com/2018/12/6/18128648/microsoft-edge-chrome-chromium-browser-changes.

[21] According to interviews with senior developers, a general policy at Google and Microsoft is to use corporate accounts for contributions to open source projects. Our data include full names and email addresses, allowing us to identify cases where individuals use multiple email addresses. This occurs in a minority of cases (86% of the commits come from users that use one unique email address). While individuals may use pseudonyms for private contributions, this aligns with our definition of non-corporate committers. Measurement error would reduce the precision of our estimates if the choice of using a corporate over a private account is time-invariant. Our estimates would be biased if the choice of using a corporate account over a private account is correlated with the timing of Microsoft's decision to enter Chromium. It is possible, yet unlikely, that engineers at Microsoft—in an official capacity—contributed to Chromium or non-Chromium projects with their private accounts before and then changed to their corporate accounts.

[22] In principle, any contributor to Chromium can ask for a chromium.org address. However, statistics we received from senior developers at Chromium indicate that only less than 0.02% of the commits made by individuals using chromium.org e-mail addresses are not Google employees. We also received information on a small number of individuals for which we then amended affiliations. While there is some measurement error, we are confident that it is negligibly small.

[23] See https://www.mozilla.org/en-US/security/advisories/, https://support.apple.com/en-us/HT201222 and https://chromereleases.googleblog.com/. For Microsoft security release notes, we directly used their API to collect information on all vulnerabilities related to Internet Explorer and Edge web browsers: https://api.msrc.microsoft.com/sug/v2.0/us-EN/vulnerability.

[24] For users, the decision to adopt a specific web browser might also be driven by the number of available extensions (i.e., plug-ins) a browser offers. Anecdotal evidence suggests that Legacy Edge only had about 70 available extensions compared to hundreds of thousands that were available to Chromium-based browsers (see: https://blogs.windows.com/msedgedev/2017/09/29/microsoft-edge-extensions-one-year-later/ and https://blog.symalite.com/chrome-extensions-statistics/). However, we cannot perform a more detailed analysis because we were not able to locate comprehensive panel data on browser extensions at the browser-month level.

[25] See https://www.forbes.com/sites/theyec/2022/09/12/the-slow-death-of-third-party-cookies/.

[26] Used by more than 1.6 million developers according to GitHub repository statistics (June 2022): https://github.com/Fyrd/caniuse.

[27] The web pages selected are the following: https://en.wikipedia.org/w/index.php?title=Web_browser&oldid=1083238014 and https://newyork.craigslist.org/.

[28] A breakdown of monthly sample sizes by country is available on Statcounter website, see https://gs.statcounter.com/faq\#sample-size.

[29] Push events capture all contributions that are merged into the main codebase, including both direct commits from maintainers and accepted pull requests from external contributors. The GitHub Events API records a push event whenever new commits are added to a repository, regardless of whether they originated from a direct commit or from an accepted pull request. Importantly, our data allows us to track the original author of each commit (through the author field), not just the maintainer who approved it. This means we capture contributions from all developers whose code is ultimately incorporated into the projects, not only those with maintainer rights. This approach is consistent with the Chromium analysis, as it similarly tracks all accepted contributions to the codebase. We focus on push events because they represent actual contributions to the codebase, rather than just proposed changes that may or may not be accepted.

[30] See https://github.com/Hipo/university-domains-list.

[31] See https://gist.github.com/drakodev/e85c1fd6d9ac8634786d6139e0066fa0.

[32] Meta gave away the governance of PyTorch to the Linux Foundation in 2022 to make it vendor-free: https://www.linuxfoundation.org/blog/blog/welcoming-pytorch-to-the-linux-foundation/.

[33] Figure A.3 displays differences between Chromium and non-Chromium repositories while Figures A.4 and A.5 and show differences between Microsoft and non-Microsoft browsers. Figure A.6 shows differences between Microsoft and respectively Chromium and non-Chromium browsers. Comparison groups differ from one category to another to reflect the unit of analysis of our estimations.

[34] We add committer organization fixed effects in column (2) to control for time-invariant organizational characteristics. It is a fixed effect for the organization that is associated with the email address of the user, for example, Microsoft, Google, or other organizations. In the other specifications, we cannot specify these fixed effects because we use different levels of aggregation (daily versus monthly totals) as the unit of observation, which mechanically removes the information of the committers' organization.

[35] Microsoft announced in November 2018 that they would invest to improve support of Chrome for Windows on ARM processors. See https://9to5google.com/2018/11/19/microsoft-google-chrome-windows-10-arm/.

[36] The Edge program is comparable to the existing bug bounty program of Chrome in terms of scope and rewards. See https://www.microsoft.com/en-us/msrc/bounty-new-edge and https://bughunters.google.com/about/rules/chrome-friends/5745167867576320/chrome-vulnerability-reward-program-rules. This can explain that both the number of reporters and the number of fixed vulnerabilities follow very similar trends for Chromium-based Edge and Chrome in Figure 4.

[37] It is important to acknowledge that it is not clear whether the faster release cycles of Chromium-Edge are caused by a general open source effect (perhaps because of different modes of development in the open source community), a Chromium-specific effect (Microsoft needs to follow the community), or Microsoft's contemporaneous switch from an update-with-OS strategy to shipping Edge as a standalone application.

[38] See https://mdn.dev/archives/insights/reports/mdn-web-developer-needs-assessment-2019.html.

[39] We are not aware of any study that empirically estimates the determinants of browser choice for users, in particular not regarding a potential trade-off between the number of functionalities and speed. However, there is some indirect evidence suggesting that both matter for the web browsing experience. First, based on a 2020 survey carried out by the Mozilla Developer Network, we have evidence that web developers ($N = 3{,}237$) care about the number of functionalities browsers support (see https://mdn.dev/archives/insights/reports/mdn-browser-compatibility-report-2020.html). When asked "What are the biggest pain points for you when it comes to browser compatibility?", 47% named "Lack of browser support for a given feature" as one of their three biggest concerns. Second, a few facts underline that load speed is an important general factor for the web browsing experience. A study by Deloitte looking at 30 million user sessions across 37 brands, shows a sizeable correlation between page load speed and economic outcomes (See https://www.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds_Make_Millions_report.pdf). A page load speed improvement of 100 milliseconds is associated with a 5% increase in page views per session, an 8% increase in conversion, and a 9% increase in order value. It is difficult to directly translate these website-level results to the browser level, but it gives some indication that page load speed in general can have important economic effects.

[40] See https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/2020.

[41] See https://www.usa.gov/website-analytics/.

[42] Linux and Chromium are written in similar languages (C and C++, respectively), have a similar number of commits (1.3 million and 1.5 million, respectively), a similar total number of lines of code (35.8 million, and Chromium has 33.6 million, respectively), which translates into a similar estimated cost (using the basic COCOMO model; $668 million and $605 million, respectively); see https://openhub.net/p/_compare?project_0=Linux+Kernel\&project_1=Chromium+%28Google+Chrome%29.

[43] See https://www.theverge.com/22630319/microsoft-windows-11-default-browser-changes and https://www.macrumors.com/2022/02/25/should-apple-ban-rival-browser-engines/.

[44] Recent efforts of by web browsers vendors support this notion, at least in the short run. See https://www.zdnet.com/article/web-devs-need-to-keep-an-eye-on-interop-2022-benchmark-to-make-life-easier/.

References

Aghion P, Bloom N, Blundell R, Griffith R, Howitt P (2005) Competition and innovation: An inverted-u relationship. *Quart. J. Econom.* 120(2):701–728.

Altman EJ, Nagle F, Tushman ML (2022) The translucent hand of managed ecosystems: Engaging communities for value creation and capture. *Acad. Management Ann.* 16(1):70–101.

August T, Chen W, Zhu K (2020) Competition among proprietary and open-source software firms: The role of licensing in strategic contribution. *Management Sci.* 67(5):3041–3066.

August T, Shin H, Tunca TI (2013) Licensing and competition for services in open source software. *Inform. Systems Res.* 24(4):1068–1086.

August T, Shin H, Tunca TI (2017) Generating value through open source: Software service market regulation and licensing policy. *Inform. Systems Res.* 29(1):186–205.

Baldwin CY, Clark KB (2006) The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Sci.* 52(7):1116–1127.

Baron J, Kanevskaia O (2023) Wearing multiple hats – The role of working group chairs' affiliation in standards development. *Res. Policy* 52(9):104822.

Belenzon S, Schankerman M (2015) Motivation and sorting of human capital in open innovation. *Strategic Management J.* 36(6):795–820.

Bitzer J, Geishecker I (2010) Who contributes voluntarily to OSS? An investigation among German IT employees. *Res. Policy* 39(1):165–172.

Boudreau KJ (2021) Promoting platform takeoff and self-fulfilling expectations: Field experimental evidence. *Management Sci.* 67(9):5953–5967.

Boudreau KJ, Jeppesen LB (2015) Unpaid crowd complementors: The platform network effect mirage. *Strategic Management J.* 36(12):1761–1777.

Bresnahan TF, Yin PL (2006) *Standard Setting in Markets: The Browser War* (Cambridge University Press, Cambridge, UK), 18–59.

Burtch G, Ghose A, Wattal S (2013) An empirical examination of the antecedents and consequences of contribution patterns in crowd-funded markets. *Inform. Systems Res.* 24(3):499–519.

Caloghirou Y, Ioannides S, Vonortas NS (2003) Research joint ventures. *J. Econom. Surveys* 17(4):541–570.

Casadesus-Masanell R, Ghemawat P (2006) Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Sci.* 52(7):1072–1084.

Chen J, Roth J (2024) Logs with zeros? Some problems and solutions. *Quart. J. Econom.* 139(2):891–936.

Chen J, Doraszelski U, Harrington JE (2009) Avoiding market dominance: Product compatibility in markets with network effects. *RAND J. Econom.* 40(3):455–485.

Chen W, Jin F, Xue L (2021) Flourish or perish? The impact of technological acquisitions on contributions to open-source software. *Inform. Systems Res.* 33(3):867–886.

Chesbrough H (2023) *Measuring the Economic Value of Open Source: A Survey and a Preliminary Analysis* (The Linux Foundation, San Francisco).

Conti A, Peukert C, Roche MP (2025) Beefing it up for your investor? Engagement with open source communities, innovation, and startup funding: Evidence from github. *Organ. Sci.* Forthcoming.

Dubé J-PH, Hitsch GJ, Chintagunta PK (2010) Tipping and concentration in markets with indirect network effects. *Marketing Sci.* 29(2):216–249.

Economides N, Katsamakas E (2006) Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Sci.* 52(7): 1057–1071.

Foerderer J, Kude T, Mithas S, Heinzl A (2018) Does platform owner's entry crowd out innovation? Evidence from google photos. *Inform. Systems Res.* 29(2):444–460.

Folta TB (1998) Governance and uncertainty: The trade-off between administrative control and commitment. *Strategic Management J.* 19(11):1007–1028.

Fosfuri A, Giarratana MS, Luzzi A (2008) The penguin has entered the building: The commercialization of open source software products. *Organ. Sci.* 19(2):292–305.

Gambardella A, von Hippel E (2019) Open sourcing as a profit-maximizing strategy for downstream firms. *Strategy Sci.* 4(1):41–57.

Goettler RL, Gordon BR (2014) Competition and product innovation in dynamic oligopoly. *Quant. Marketing Econom.* 12:1–42.

Gonzalez D, Zimmermann T, Nagappan N (2020) The state of the ml-universe. *Proc. 17th Internat. Conf. Mining Software Repositories*, 431–442.

Greenstein S, Nagle F (2014) Digital dark matter and the economic contribution of apache. *Res. Policy* 43:623–631.

Grimpe C, Kaiser U (2010) Balancing internal and external knowledge acquisition: The gains and pains from R&D outsourcing. *J. Management Stud.* 47(8):1483–1509.

Gross DP (2020) Collusive investments in technological compatibility: Lessons from us railroads in the late 19th century. *Management Sci.* 66(12):5683–5700.

Haefliger S, von Krogh G, Spaeth S (2007) Code reuse in open source software. *Management Sci.* 54(1):180–193.

Hanaki N, Nakajima R, Ogura Y (2010) The dynamics of R&D network in the it industry. *Res. Policy* 39(3):386–399.

Hann IH, Roberts JA, Slaughter SA (2013) All are not equal: An examination of the economic returns to different forms of participation in open source software communities. *Inform. Systems Res.* 24(3):520–538.

Horton JJ, Tambe P (2025) The death of a technical skill. *Inform. Systems Res.* Forthcoming.

Iansiti M, Richard GL (2006) The business of free software: Enterprise incentives, investment, and motivation in the open source community. Harvard Business School Working Paper, No. 07-028.

Jeppesen LB, Frederiksen L (2006) Why do users contribute to firm-hosted user communities? The case of computer-controlled music instruments. *Organ. Sci.* 17(1):45–63.

Johnson JP (2002) Open source software: Private provision of a public good. *J. Econom. Management Strategy* 11(4):637–662.

Kamien MI, Muller E, Zang I (1992) Research joint ventures and R&D cartels. *Amer. Econom. Rev.* 82(5):1293–1306.

Karhu K, Ritala P (2021) Slicing the cake without baking it: Opportunistic platform entry strategies in digital markets. *Long Range Planning* 54(5):101988.

Katz R, Allen TJ (1982) Investigating the not invented here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R&D project groups. *R&D Management* 12(1):7–20.

Kochhar PS, Kalliamvakou E, Nagappan N, Zimmermann T, Bird C (2019) Moving from closed to open source: Observations from six transitioned projects to GitHub. *IEEE Trans. Software Engrg.* 47(9):1838–1856.

Laursen K, Salter AJ (2014) The paradox of openness: Appropriability, external search and collaboration. *Res. Policy* 43(5):867–878.

Leone MI, Reichstein T (2012) Licensing-in fosters rapid invention! The effect of the grant-back clause and technological unfamiliarity. *Strategic Management J.* 33(8):965–985.

Lerner J, Tirole J (2002) Some simple economics of open source. *J. Indust. Econom.* 50(2):197–234.

Lifshitz-Assaf H, Nagle F (2021) The digital economy runs on open source. Here's how to protect it. *Harvard Business Review* (September 2), https://hbr.org/2021/09/the-digital-economy-runs-on-open-source-heres-how-to-protect-it.

Lin Y-K, Maruping LM (2022) Open source collaboration in digital entrepreneurship. *Organ. Sci.* 33(1):212–230.

Loh J, Kretschmer T (2023) Online communities on competing platforms: Evidence from game wikis. *Strategic Management J.* 44(2): 441–476.

MacCormack A, Rusnak J, Baldwin CY (2006) Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Sci.* 52(7):1015–1030.

Maillart T, Zhao M, Grossklags J, Chuang J (2017) Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *J. Cybersecurity* 3(2):81–90.

Malladi SS, Subramanian HC (2020) Bug bounty programs for cybersecurity: Practices, issues, and recommendations. *IEEE Software* 37(1):31–39.

Miric M, Jeppesen LB (2023) How does competition influence innovative effort within a platform-based ecosystem? Contrasting paid and unpaid contributors. *Res. Policy* 52(7):104790.

Mollick E (2016) Filthy lucre? Innovative communities, identity, and commercialization. *Organ. Sci.* 27(6):1472–1487.

Nagaraj A, Piezunka H (2024) The divergent effect of competition on platforms: Deterring recruits, motivating converts. *Strategy Sci.* 9(3):277–296.

Nagle F (2018) Learning by contributing: Gaining competitive advantage through contribution to crowdsourced public goods. *Organ. Sci.* 29(4):569–587.

Nagle F (2019) Open source software and firm productivity. *Management Sci.* 65(3):1191–1215.

Nagle F, Seamans R, Tadelis S (2024) Transaction cost economics in the digital economy: A research agenda. *Strategic Organ.* 23(2): 351–365.

Raymond E (1999) The cathedral and the bazaar. *Knowledge Tech. Policy* 12(3):23–49.

Reisinger M, Ressner L, Schmidtke R, Thomes TP (2014) Crowding-in of complementary contributions to public goods: Firm investment into open source software. *J. Econom. Behav. Organ.* 106:78–94.

Rothaermel FT, Hitt MA, Jobe LA (2006) Balancing vertical integration and strategic outsourcing: Effects on product portfolio, product success, and firm performance. *Strategic Management J.* 27(11):1033–1056.

Simcoe T, Watson J (2019) Forking, fragmentation, and splintering. *Strategy Sci.* 4(4):283–297.

Song P, Xue L, Rai A, Cheng Z (2018) The ecosystem of software platform: A study of asymmetric cross-side network effects. *MIS Quart.* 42(1):121–142.

Spaeth S, von Krogh G, He F (2014) Research note—Perceived firm attributes and intrinsic motivation in sponsored open source software projects. *Inform. Systems Res.* 26(1):224–237.
Spinellis D (2008) A tale of four kernels. *Proc. Internat. Conf. Software Engrg.* 381–390.
Spinellis D (2011) Quality wars: Open source versus proprietary software. Oram A, Wilson G, eds. *Making Software: What Really Works, and Why We Believe It* (O'Reilly Media, Sebastopol, CA), 259–293.
Stam W (2009) When does community participation enhance the performance of open source software companies? *Res. Policy* 38(8):1288–1299.
Stewart KJ, Ammeter AP, Maruping LM (2006) Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Inform. Systems Res.* 17(2):126–144.
Tellis GJ, Yin E, Niraj R (2009) Does quality win? Network effects versus quality in high-tech markets. *J. Marketing Res.* 46(2):135–149.
Tiwana A (2015) Evolutionary competition in platform ecosystems. *Inform. Systems Res.* 26(2):266–281.
von Hippel E, von Krogh G (2003) Open source software and the "private-collective" innovation model: Issues for organization science. *Organ. Sci.* 14(2):209–223.
von Krogh G, Haefliger S, Spaeth S, Wallin MW (2012) Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quart.* 36(2):649–676.
Wang Y, Chen Y, Koo B (2020) Open to your rival: Competition between open source and proprietary software under indirect network effects. *J. Management Inform. Systems* 37(2):1128–1154.
Wen W, Zhu F (2019) Threat of platform-owner entry and complementor responses: Evidence from the mobile app market. *Strategic Management J.* 40(9):1336–1367.
Wen W, Ceccagnoli M, Forman C (2016) Opening up intellectual property strategy: Implications for open source software entry by start-up firms. *Management Sci.* 62(9):2668–2691.
Wen W, Forman C, Graham SJ (2013) Research note—The impact of intellectual property rights enforcement on open source software project success. *Inform. Systems Res.* 24(4):1131–1146.
Wooldridge JM (2023) Simple approaches to nonlinear difference-in-differences with panel data. *Econom. J.* 26(3):C31–C66.
Wright NL, Nagle F, Greenstein S (2023) Open source software and global entrepreneurship. *Res. Policy* 52(9):104846.
Xu L, Nian T, Cabral L (2019) What makes geeks tick? A study of stack overflow careers. *Management Sci.* 66(2):587–604.
Yilmaz ED, Meyer T, Miric M (2023) Preventing others from commercializing your innovation: Evidence from creative commons licenses. Preprint, submitted September 1, https://arxiv.org/abs/2309.00536.
Yin, Bresnahan, Yin (2005) Economic and Technical Drivers of Technology Choice: Browsers. *Annales D'économie Et De Statistique*. (79/80):629
Yue D, Nagle F (2024) Igniting innovation: Evidence from PyTorch on technology control in open collaboration. Harvard Business School Working Paper No. 25-013, Harvard Business School, Boston.
Zhang XM, Zhu F (2011) Group size and incentives to contribute: A natural experiment at Chinese Wikipedia. *Amer. Econom. Rev.* 101(4):1601–1615.
Zhu F, Iansiti M (2012) Entry into platform-based markets. *Strategic Management J.* 33(1):88–106.
Zhu F, Liu Q (2018) Competing with complementors: An empirical look at amazon.com. *Strategic Management J.* 39(10): 2618–2642.
Zrahia A, Gandal N, Markovich S, Riordan M (2024) The simple economics of an external shock to a bug bounty platform. *J. Cybersecurity* 10(1):6.