Research Policy 53 (2024) 104954

Contents lists available at ScienceDirect

Research Policy

journal homepage: www.elsevier.com/locate/respol

From GitHub to GDP: A framework for measuring open source software innovation

Gizem Korkmaz [a,*], J. Bayoán Santiago Calderón [b], Brandon L. Kramer [c], Ledia Guci [b], Carol A. Robbins [d]

[a] Westat, 1600 Research Blvd, Rockville, 20850, MD, USA
[b] U.S. Bureau of Economic Analysis, 4600 Silver Hill Rd, Suitland, 20746, MD, USA
[c] Edge & Node, New York, NY, USA
[d] U.S. National Science Foundation, National Center for Science and Engineering Statistics, 2415 Eisenhower Ave, Alexandria, 22314, VA, USA

ARTICLE INFO

JEL classification:
C82
E22
H42
L17
O3
O51

Keywords:
Open source software
Cost measurement
GitHub
Innovation
Gross domestic product
Software investment
National accounts

ABSTRACT

Open source software (OSS) is software that anyone can review, modify, and distribute freely, usually with only minor restrictions such as giving credit to the creator of the work. The use of OSS is growing rapidly, due to its value in increasing firm and economy-wide productivity. Despite its widespread use, there is no standardized methodology for measuring the scope and impact of this fundamental intangible asset. This study presents a framework to measure the value of OSS using data collected from GitHub, the largest platform in the world with over 100 million developers. The data include over 7.6 million repositories where software is developed, stored, and managed. We collect information about contributors and development activity such as code changes and license detail. By adopting a cost estimation model from software engineering, we develop a methodology to generate estimates of investment in OSS that are consistent with the U.S. national accounting methods used for measuring software investment. We generate annual estimates of current and inflation-adjusted investment as well as the net stock of OSS for the 2009–2019 period. Our estimates show that the U.S. investment in 2019 was $37.8 billion with a current-cost net stock of $74.3 billion.

1. Introduction

Open source software (OSS) is developed, maintained, and used both within the business sector and outside of it through the contribution of independent developers and people from universities, government research institutions, and nonprofits. Because OSS can be studied, modified, and distributed freely, typically with only minor restrictions (St. Laurent, 2004), this software can undergo fairly rapid innovation and be repurposed across various industries (Raymond, 1999). The Open Source Initiative (OSI) certifies licenses that comply with the principles of open source; any software with an OSI-approved license is considered open source. Notable examples of OSS include the Linux operating system, Apache server software, and programming languages R and Python.

When renowned entrepreneur and Netscape co-founder Marc Andreessen declared "software is eating the world" in 2011 (Andreessen, 2011), private investment in computer software was about $250 billion (U.S. Bureau of Economic Analysis, 2023j). At that time, computer software made up 13% of U.S. private nonresidential (e.g., commercial, industrial, institutional) fixed investment, which accounts for the nonresidential structures, equipment, and intellectual property products created, purchased, or improved (see Section 2 for definitions). By 2022, private fixed investment in computer software had grown to 17% of nonresidential private fixed investment (U.S. Bureau of Economic Analysis, 2023k).[1] Over this time, the open source share of private software investment in the United States has grown dramatically. With the emergence of generative artificial intelligence, large language models, and computer vision models, it is likely that the importance of software overall and the open source component will only increase. While the potential for software innovation to support economic and productivity growth and transform various sectors of the economy is indisputable,

* Corresponding author.
E-mail addresses: gizemkorkmaz@westat.com (G. Korkmaz), jose.santiago-calderon@bea.gov (J.B. Santiago Calderón), brandonleekramer@gmail.com (B.L. Kramer), ledia.guci@bea.gov (L. Guci), crobbins@nsf.gov (C.A. Robbins).

[1] Calculation available at: https://alfred.stlouisfed.org/graph/?g=1cK5U.

https://doi.org/10.1016/j.respol.2024.104954
Received 27 September 2022; Received in revised form 19 December 2023; Accepted 3 January 2024
Available online 26 January 2024

measurement of innovation in software has been limited, particularly for OSS. In this paper, we provide an approach to measure it.

Many OSS projects create long-term tools that are products of public spending. Often, these tools have been developed outside of the business sector and subsequently used within it. While limited, existing estimates of publicly funded OSS suggest its magnitude is significant. For example, by 2017, Apache was estimated to hold the largest market share of active websites (44.5%) (Netcraft, 2017).[2] The Apache server, developed with federal and state funds at the National Center for Supercomputing Applications at the University of Illinois, is estimated to be equivalent to between 1.3% and 8.7% of the stock of prepackaged software currently accounted for in U.S. private fixed investment (Greenstein and Nagle, 2014). The scale and use of these modifiable software tools highlight an aspect of technology flow that needs to be captured in market measures.

Better measurement supports better policy, and as an intangible asset created and used across the economy, OSS developed by applicable sectors should be fully accounted for in gross domestic product (GDP). Economic measurement that is integrated with the overall accounting of goods and services produced in the economy provides the basis for understanding the impact of OSS on sector-level productivity as well as overall productivity. Understanding the role that each sector plays in developing, funding, and promoting OSS can help inform public policy that supports innovation and economic growth.

This paper makes several important contributions. First, it fills a measurement gap by providing a novel approach to measuring investment in OSS. It thus contributes to the economic measurement literature by developing a methodology consistent with those of national accounts that measure software investment in the United States. We adopt a cost model from software engineering and apply it to GitHub data. More specifically, our methodology draws from the current economic measurement of own-account software — software created using internal resources as opposed to purchased or outsourced. Based on the development cost, we extend this cost measure to the measurement of OSS as a useful asset used in production.

While we follow the general framework for measuring own-account software in the national accounts, we deviate when we can use unique aspects of our approach to overcome current measurement challenges. For instance, the national accounts methodology for estimation of own-account software considers only certain occupation types (such as computer programmers, software developers). Moreover, it assumes shares for each occupation to determine the allocation of time spent on software development. These assumptions may have been well suited in traditional software developed by businesses but may introduce a downward bias in estimating OSS development from non-industry sources such as private academic institutions (that involve contributions from researchers such as bio-statisticians or earth scientists).[3] This may lead to an underestimation of investment in OSS and in software overall. In a notable deviation, using data on open source projects hosted and shared on the most widely-used source-code hosting facility, GitHub (GitHub, 2023), we use the lines of code for each project to estimate the time and effort required to develop the projects (Boehm, 1984; Boehm et al., 2000). To ensure the reliability of our results, we conduct robustness tests that consider team sizes and types of software. Because our framework is consistent with measuring other types of software, intangible assets, and other intellectual property products in the national accounts, it allows for comparing OSS to other kinds of investments.

Second, this paper provides estimates of U.S. annual OSS investment based on the prices prevailing during the period the investment took place (nominal) and adjusted for inflation and quality changes (real) which allows for comparisons across time. We also provide the corresponding net stock estimates (i.e., the cumulative value of the asset) for the 2009–2019 period. Although the totality of OSS is unknown, we believe our estimates account for a significant portion of OSS development and make a significant contribution to a growing literature on measuring OSS innovation poorly captured by traditional approaches. These OSS measures complement existing science and technology indicators on peer-reviewed publications and patents that are calculated from databases covering scientific articles and patent documents. In addition, with these estimates, we aim to contribute to the understanding of productivity and economic growth, both within and outside the business sector, and encourage further research into the importance and contribution of OSS in the digital economy.

Third, with this paper we have developed methods, tools, and administrative datasets that we used to collect the GitHub data and to classify the data into countries and economic sectors. We have made these tools and datasets publicly available to encourage further research and analyses into the measurement of OSS and its contribution to productivity. Overall, the paper makes an important contribution to a better understanding of the scope of OSS investment and its growth over time and provides a consistent and robust framework for measuring it.

The remainder of the paper is structured as follows. Section 2 describes the treatment of software in the national accounts framework, its limitations, and how our approach overcomes these challenges. Section 3 provides a broad description of OSS, highlights successful projects, and summarizes the role of repositories and licenses. The related literature on OSS and its relationship to our work is described in Section 4. It includes the motivations for creation and use of OSS, economic measurement of software and other intangibles, and the implications of OSS for productivity. Data and methods for this paper are described in Section 5, with results shown and described in Section 6. Section 7 provides a discussion of limitations and extensions to this work, and Section 8 presents our conclusions. Technical details on the methodology are provided in Appendix A.

2. Software investment in the national accounts

In the U.S. national accounts, fixed or produced assets are products that are intended, have the potential, or are used repeatedly or continuously in the production process of goods and services for at least one year. Fixed assets consist of intellectual property products (IPPs), equipment, and structures. IPPs are research and development (R&D); software; and entertainment, literary, and artistic originals (ELAO). Examples of ELAO include theatrical movies, long-lived television programs, books, and music. Equipment includes products such as new machinery, furniture, and vehicles. Lastly, structures are products that are usually constructed at the location where they will be used for a long time, such as buildings and oil wells, and embedded equipment, such as electrical systems, plumbing, and indoor temperature control (U.S. Bureau of Economic Analysis, 2022).

The System of National Accounts (SNA) provides two main ways to account for assets (United Nations, 2010). The first is through *investment*, which refers to the procurement of assets in a given period. Procurement can be done through purchases or through internal production, also known as *own-account*. Investment includes the value of the procurement of new assets, the improvement of existing assets, and the replacement of assets that have become obsolete. The second is a measure of the value of all accumulated/available assets at some point in time, referred to as the *stock*. Assets, however, suffer from physical deterioration, normal obsolescence, or typical accidental damage, which decreases their usefulness in the production process. The term *net stock* refers to the value of accumulated assets after adjustment for depreciation (United Nations, 2010). Current-cost net stock uses the cost of replacing the assets at the current prices to value the investment.

[2] In 2023, Apache held 20.79% market share of active websites followed by another open source web server, Nginx (Netcraft, 2023).

[3] Survey data on the background and occupations of OSS developers provides an opportunity for future refinement (Nagle et al., 2020).

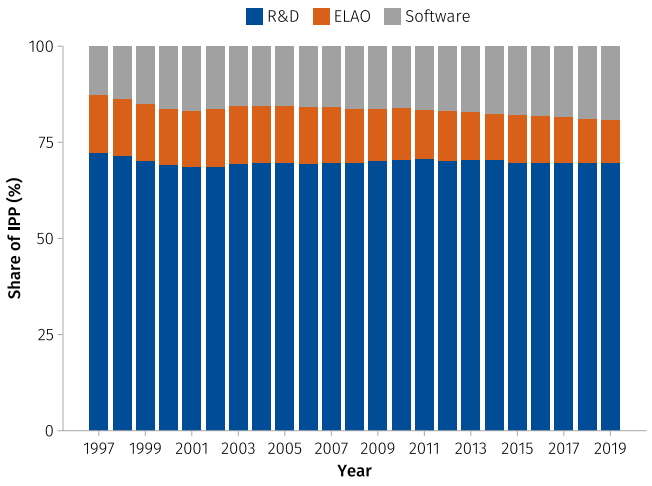


Fig. 1. Current-cost net stock of intellectual property products (IPPs) that consist of research and development (R&D); entertainment, literary, and artistic originals (ELAO); and software.
Source: Current-cost net stock estimates for private and public IPPs (U.S. Bureau of Economic Analysis, 2023m,p).

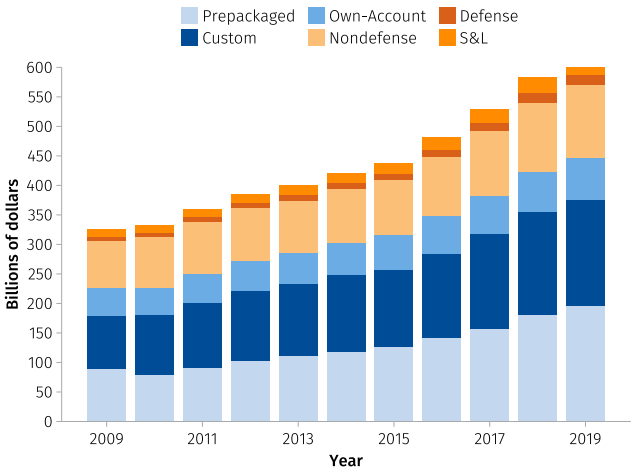


Fig. 2. U.S. investment in software 2009–2019.
Source: Published series (U.S. Bureau of Economic Analysis, 2023a,b,c,g,h,i).

The annual investment in software as a share of the U.S. total investment in fixed assets has doubled from 6.2% in 1997 to 12.8% in 2022.[4] In 2022, annual investment in software is estimated at $702 billion ($598 billion from the private sector and $104 billion from the public sector). This accounts for the private investment in software accounting for about 42.5% of total private investment in IPPs. The software share of private nonresidential investment – the investment in equipment, structures, and IPPs by the private sector that excludes residential investment – is 17.4%.[5]

Fig. 1 shows the asset composition of the net stock of IPPs, with R&D making up the largest share of IPPs. As the figure shows, investment in software has been growing faster than the other components of IPPs. This increases the share of the net stock of IPPs that are attributed to software. Software's share of the net stock of IPPs increased from 12.6% in 1997 to 17% in 2022.

The current classification for the measurement of software in the national account is shown in Table 1. In the national accounts, private software investment includes the investment by businesses and nonprofit institutions serving households (NPISHs) and is classified into three categories based on how it was procured. The three categories are (1) *prepackaged software* that is bought in the market, (2) *custom software* which includes commissioned and integration services, and (3) *own-account* that covers in-house development. Software originals, or the initial investment in producing the asset of which its copies are sold in the market as prepackaged or potentially custom software solutions, are considered R&D investment but are included in the stock attributed to software.

Public investment in software is classified based on who is procuring it and for what purpose. Defense-related investment in software (from the Department of Defense (DOD)) is counted and published in the *defense* account. The investment of software from the federal government, excluding DOD, is allocated to *nondefense*. Lastly, investment in software from state and local governments, including public state and local institutions of higher education, are published separately from the federal government.

[4] See https://alfred.stlouisfed.org/graph/?g=1bFiA for the calculation using data from the U.S. Bureau of Economic Analysis (BEA) (U.S. Bureau of Economic Analysis, 2023a,b,c,d,j,k,l).

[5] See https://alfred.stlouisfed.org/graph/?g=1bFjH for the calculation.

Of note is the measurement of the investment in software by institutions of higher education. This investment is not separately measured. The investment by private for-profit higher education institutions is included in the investment by the business sector. The investment by nonprofit higher education institutions is attributed to the NPISHs sector. The contribution from public institutions is captured under the relevant accounts of federal (defense/nondefense) or state and local government.

A breakdown of the private and public software investment components is provided in Fig. 2 for our period of study (2009–2019). In 2019, the latest year for which we estimate investment in OSS, annual investment in software in the United States is estimated at $527 billion. Of this total, $79 billion (15%) was from the public sector and $448 billion (85%) was from the private sector.

The current classification system has limitations for measuring OSS. First, it fails to distinguish between open source and proprietary software. Given the different implications for productivity between these types of software, it is vital to provide separate measures. Conceptually, OSS developed by the federal, state and local governments is accounted for under the corresponding sectoral account (e.g., defense, nondefense, S&L). As noted, OSS developed in academic institutions is divided into public and private investment based on the institution. Further, own-account software is counted together with prepackaged and custom software. This approach obscures the contributions of government and academic institutions to the creation of OSS. Our approach explicitly measures investment in OSS within the current national account framework. Our estimates explicitly break out these categories of contribution.

Considering the private software investment accounts, OSS is more likely to be developed as own-account. However, some scenarios make the analysis more nuanced. For example, if a firm develops a software original and later releases a copy with economic rights similar to an original as open source, it may not be considered an investment, but would be included in the stock of software assets. If the development of an open source project is funded by various parties which are not the employers of the developers, it may be considered a procurement that falls within custom software.

Lastly, while the contribution to the creation of OSS by the various sectors is not explicitly measured in the national accounts, the investment in software by the household sector is not measured at all. Our approach allows for the estimates of investment in OSS to be provided at the following levels: (1) business sector, (2) NPISH, (3) government (federal, state and local), (4) households, and (5) higher education

Table 1
Sectors and respective accounts for OSS investment.

	Prepackaged	Custom	Own-account	Software total
Private sector				
Business NPISH	Published (Proprietary)	Published (Proprietary + OSS)	Published (Proprietary + OSS)	Published (Proprietary + OSS)
Household	Not measured	Not measured	Not measured	Not measured
Public sector				
Federal Gov't (excl. defense)	Detail not published	Detail not published	Detail not published	Published (proprietary + OSS)
Dept. of Defense	Detail not published	Detail not published	Detail not published	Published (proprietary + OSS)
State and Local	Detail not published	Detail not published	Detail not published	Published (proprietary + OSS)

Notes: The accounts are described based on the National Income and Product Accounts (NIPA) Handbook (U.S. Bureau of Economic Analysis, 2022). *NPISH* refers to nonprofit institutions serving households, and includes private higher education. Public higher education is included in the federal and state and local government accounts.

Table 2
Widely adopted open source software projects.

Project	Category/Type	Initial release
LaTeX	Typesetting	1983
Linux	Operating system	1991
Apache	Web server	1995
GIMP	Graphics editor	1996
PostgreSQL	RDBMS	1996
VLC	Media player	2001
Firefox	Web browser	2002
QGIS	GIS	2002
LLVM	Compiler	2003
Thunderbird	Email client	2003
WordPress	CMS	2005
jQuery	Front-end	2006
LibreOffice	Productivity suite	2011
OpenBLAS	BLAS/LAPACK	2011
React	JavaScript library	2013
Project Jupyter	Shell	2015
TensorFlow	ML framework	2015
Visual Studio Code	IDE	2015
Hugo	SSG	2017

(private and public). This level of detail makes our estimates more internationally comparable with other measures of intangible assets such as R&D.

3. Landscape of open source software

Beginning in the early 1980s, OSS projects have provided users with zero-dollar cost and freely modifiable software tools. Table 2 lists some of the widely used OSS projects and the year of their initial release. These widely adopted projects have become the leading solutions for various web applications such as content management, JavaScript libraries, and web servers.[6] Android, another prominent open source project, is a Linux-based operating system for mobile devices that holds 70% of market share as of November 2021 (GlobalStats, 2021).

To highlight some of the innovators and their motivations concerning OSS, we present a few examples of highly successful projects. The first example is the aforementioned Linux — a core program that powers a family of computer operating systems. While the original author Linus Torvald's initial motivation was to create the project for fun (an example of software developed as a form of household innovation) (Torvalds and Diamond, 2001), Linux has swiftly evolved into one of the most important software tools in the world. As of 2017, Linux "runs 90 percent of the public cloud workload, has 62 percent of the embedded market share, and 99 percent of the supercomputer market share [as well as] 82 percent of the world's smartphones and nine of the top ten public clouds" (Corbet and Kroah-Hartman, 2017). In 2019, Linux was used across a number of sectors, with a report suggesting that employees from more than 400 companies have contributed to the project since 2009 (The Linux Foundation, 2020).

Guido van Rossum was an employee at a national research institute when he started Python—a "hobby programming project that would keep [him] occupied during the week around Christmas" (van Rossum, 1996). It is not uncommon for companies to rely heavily on open source components. Microsoft made waves when they hired Python's creator out of retirement for their Developer Division (van Rossum, 2020). Python is an example of how innovation spurred by OSS and its inventors moves fluidly across different economic sectors.

PostgreSQL, one of the most commonly used relational databases, is an example of an open source project that originated from a federally sponsored academic project at a public university. The POSTGRES project at the University of California at Berkeley was sponsored by a number of federal entities including the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF) (The PostgreSQL Global Development Group, 2020). In contrast, many popular web frameworks such as Twitter Bootstrap and Facebook React were created within firms before the company open-sourced the projects while continuing to make large contributions.

3.1. Repositories and source code hosting

Many OSS projects are developed and shared through *source code hosting platforms* such as GitHub, GitLab, SourceForge, and Bitbucket. These platforms are used to develop, download, review, and publish code for projects that are stored and managed in central locations referred to as repositories. These platforms host public and private repositories, while providing a suite of features such as access and permissions by teams, issue tracking, wikis, web-hosting, and continuous integration. Version control systems, such as Git, serve to track changes and coordinate work on files among multiple developers. Information embedded in the repositories and websites, including the code, contributors, and development activity, is publicly available through techniques used to collect web-based data. This creates a very rich source of data to study the scope and impact of these projects (Keller et al., 2018).

3.2. Licenses

When someone develops code, the developer is automatically granted copyright over the work just as with other artistic literary work regardless of whether the work has been registered with a copyright office.[7] The copyright holder becomes the author or potentially the employer, depending on applicable laws and contracts. It is highly encouraged for developers to use standard licenses that allow users to quickly understand the terms under which the software is governed.

[6] The market shares of various technologies are estimated by W3Techs available at: https://w3techs.com/technologies.

[7] See the Berne Convention for the Protection of Literary and Artistic Works for additional information on the current international framework.

One of the most popular licenses is the MIT license, an open source permissive license developed at the Massachusetts Institute of Technology. The term permissive means that a license sets very few restrictions on the use/reuse and sharing such as requiring attribution that is not required under public domain. Other types of licenses are "copyleft" licenses such as the GNU General Public (GPL) Licenses, which allow users to modify and reuse intellectual property without restriction, but requires that derivatives must remain under the same terms. While copyleft licenses used to be the norm in the early years of OSS development, the recent trend has been to adopt permissive licenses. For example, tidyverse, a popular collection of packages for the R programming language, has made efforts to relicense a bulk of their packages by applying the permissive MIT license.[8]

4. Related work

This work aligns with the growing body of literature on valuing goods and services that do not involve explicit market transactions, many of which are digital goods. Our work adds to this literature by providing a new method to estimate the inputs to the production of OSS. Section 4.1 addresses economic measurement of intangibles, in particular that of software. In Section 4.2, we summarize relevant literature on different motivations for firms to develop and use OSS. We look at the potential implications of a shift from proprietary software to OSS. Section 4.3 concludes with the relationship between OSS and productivity.

4.1. Economic measurement of innovation, intangibles, and software

Innovation is typically captured and measured using surveys, patent analysis, case studies, and peer reviews. Most available statistics are focused on the business sector. Innovation is measured through its incidence (survey measurement), activities (primarily science, technology, engineering, and mathematics education and workforce), outputs (products and processes), and outcomes (economic growth and societal benefits) (Aizcorbe et al., 2009). Because of the link to economic growth, policymakers and researchers are interested in understanding and supporting activities that lead to innovation.

Traditional approaches to measuring innovation leave many types of innovation uncaptured, because they focus on business sector activity and often represent intangible assets that are hard to put a price on, such as knowledge and OSS (Damanpour, 1991). This "dark innovation" (Martin, 2016) takes place in households, universities, and governments. It occurs when the product is used, rather than sold in the market (Gault, 2018), and is referred to as "free innovation" (von Hippel, 2017) or household production (Bockstael and McConnell, 1983).

Interest in better measurement of the economic impact of computer software and the increased digitization of knowledge led to parallel development in national economic accounting. For example, GDP statistics for the United States have treated computer software as investment since 1999, extended this treatment to R&D expenditures and entertainment and literary originals in 2013 (U.S. Bureau of Economic Analysis, 2013). Beyond these three categories, Corrado et al. (2005) provide a framework for consistent accounting for expenditures on intangibles that generate future benefits. Arguing that public expenditures on intangibles yielding future benefits should be understood as investment, Corrado et al. (2017) propose a public investment category — information, scientific, and cultural assets. This category includes software and databases along with R&D, mineral exploration and cultural products. They argue that better accounting of public investment in intangibles would provide a more complete picture of economic growth.

In the absence of a direct price, in-house intangible assets (like other forms of internally created investment in the business sector) can be valued based on its production cost (Nakamura and Soloveichik, 2015; Nakamura et al., 2017). Other digital goods, such as online platforms, often use mixed approaches such as collecting transaction fees or engaging in data collection that provides high value to the businesses (Li et al., 2019). Since assets may be shared or monetized in different ways, a production cost approach can underestimate the true economic value of this investment. For example, OSS can serve as the basis for a number of services around core components such as with Anaconda (Python), Posit (formerly known as RStudio) (R), and JuliaHub (formerly known as Julia Computing) (Julia). These companies contribute to the open source ecosystem and in return add value to the companies' products and services.

Our approach is to observe and measure intangible inputs to innovation using non-survey data sources (e.g., software repositories). Keller et al. (2018) uses OSS innovation as a case study, and describe the challenges and processes to measure these intangibles using a data science framework. Through a process of data discovery, acquisition, statistical data integration, and visualization, the authors show the feasibility of measuring innovation related to OSS through data collected from online software repositories. They provide evidence and insights about how these data could be used to estimate value and impact.

A production cost-based approach to measure OSS as intangible assets created both within and outside of the business sector (such as in universities, federal government agencies, and households) using GitHub data was proposed and prototyped in Robbins et al. (2018a,b). The work adopts cost models developed in software engineering and methods used by the Bureau of Economic Analysis (BEA) to estimate the resource cost of OSS. They focus on packages developed for four open source programming languages (R, Python, Julia and JavaScript) and OSS available on federal government's Code.gov (Code.gov, 2016) (part of an effort to make custom-developed code broadly available across federal government agencies). The authors estimate that the resource cost for developing all of the associated packages and projects shared on Code.gov exceed $3 billion and $1 billion, respectively, based on 2017 costs.

Korkmaz et al. (2018, 2020) present an impact-focused approach to measure the value of OSS and use the number of downloads and citations of OSS as measures of their impact. Using data collected from Depsy.org (NSF-funded initiative to track the impact of research code compiling R and Python packages) (Piwowar and Priem, 2016), the authors generate dependency and contributor networks of R and Python packages and develop statistical models to identify factors that affect the impact of OSS.

4.2. Technological and economic motivations for open source software

The motivations for investing in the development of freely available products can be hard to understand. In 2018, Microsoft acquired GitHub for $7.5 billion and IBM Red Hat for $34 billion in 2019 (Microsoft, 2018; Red Hat, 2019-07-09). Despite the high purchase prices, Microsoft and IBM determined that positioning themselves closer to these epicenters of OSS innovation was a worthwhile endeavor.

Dahlander and Magnusson (2005) describes three types of motivation for participating in OSS projects: economic, social, and technological. Economic motivation for firms includes competitiveness and a rapid pace of development with a dedicated community. For communities, the economic rewards include low-cost access to software and the monetary rewards that may be derived from reputation effects. Firms and development communities share the social motivation to participate in a code-sharing community. Technological motivations for both firms and communities include access to technical feedback. Firms can see their new technologies diffuse rapidly, influencing adoption and standard-setting.

[8] https://www.tidyverse.org/blog/2021/12/relicensing-packages.

The ability of firms to combine OSS with complementary resources, such as custom software or other proprietary inputs, provides additional motivation for firms (Fosfuri et al., 2008). As part of an open source business model, motivations can also include the utilization of all the aspects of a value network, including customer interface, communication, and governance (Duparc et al., 2022).

Surveys can give us a better understanding of the composition of OSS contributors and the dynamics of OSS development (e.g., Nagle et al. (2020)). David et al. (2003) describes the composition of U.S. OSS contributors, featuring a majority of contributors being employees, followed by students, and lastly, self-employed. The differences within firms, seems also to have important ramifications, such as in the decision to adopt OSS at the firm level (Alexy et al., 2011).

For many academics and researchers, software tools and databases are by-products of their own work that can also be used by other academics as well (Gambardella and Hall, 2006). Advantages of OSS include the ability to scale customization projects and to resolve program bugs quickly through many users (Lerner and Tirole, 2005). OSS communities can also be viewed as user innovation networks, where contributors more successfully develop solutions to their own software needs through the OSS community (von Hippel, 2005).

The technological and economic motivations for using OSS extend to government applications as well. The U.S. Department of Defense (DOD) uses OSS for tasks including geospatial imaging, remote sensing, training simulations, and software-defined radio frameworks (DOD CIO, 2021). While risk is an often-cited concern (Hauge et al., 2010) with OSS adoption, the DOD's Chief Information Officer notes that any commercially available software has the risk of malicious code. When a program is released as open source, these vulnerabilities can be found and fixed (DOD CIO, 2021).

4.3. Productivity and open source software

Improved measurement of OSS has implications both for productivity measurement and to inform policy decisions that advance innovation and drive economic growth. Productivity analysis (relationship between inputs and the associated outputs) is one of the most significant motivations for improved measurement of OSS. The literature has identified concerns about applying conventional productivity analysis to measure the impact of OSS (Greenstein and Nagle, 2014). At the macroeconomic level, improved accounting for OSS is important for two reasons. First, OSS is unpriced, and as such, can distort current input cost-based measures of business productivity. Second, OSS packages are part of the capital stock used in further production. At the firm level, OSS has changed or has the potential to change the way companies are run by impacting technological capabilities, skill sharing, shared resources, and skill sets. Accurate measurement of the adoption of OSS can shed light on strategic and competitive factors within the business sector.

Productivity analysis measures how efficiently inputs are being converted into outputs. These inputs and outputs are essential components for understanding and predicting economic growth. When productive assets like OSS are not fully counted in annual investment, both inputs and outputs are underestimated. While this study does not provide productivity analysis related to OSS, it provides a first step toward better measurement of OSS as an input, that is to say a source of capital services, and as an output, an investment good, that contributes to GDP. This analysis can provide insights into the nature and development of software, the use of software development resources, and the relative efficiency gained by using these resources.

Firms engage with software, both proprietary and OSS, in multiple ways. These interactions between firms and software systems may be mutually beneficial, beneficial to one but not the other, or beneficial to one at the expense of the other (Dahlander and Magnusson, 2005). Further, while the licensing of software enables its value to be shared, proprietary software developed and deployed within large firms can enable those firms to lock out their competitors and allow them to increase their profits. OSS and other open platforms allow for a standardization of software that can be accessed by any user, increasing diffusion (Bessen, 2022).

Thus, use of OSS has the potential to reduce barriers to entry and weaken the market of large incumbent firms. This usage of OSS to lower costs of entry to new firms has been described as part of free innovation (von Hippel, 2017). Based on more than 10,000 U.S. firms surveyed for the years 2000 to 2009, firm-level productivity increased for firms that either produce information technology or are heavy users of information technology (Nagle, 2019). The study finds that the use of OSS increases firm-level productivity, but the effect is more significant for small firms than for larger ones. Another way of looking at lower entry costs through OSS is to compare Microsoft's proprietary server software with Linux. Based on what equivalent software would have cost, Greenstein and Nagle (2014) estimated the unmeasured value of the capital stock of Apache software in use in 2013 at between $2 and $12 billion.

An implication of this is that reducing spillovers benefits large firms while increasing them benefits to small firms (Bessen, 2022). As a counterweight, scale effects can lower the cost of storing and maintaining OSS, thus supporting large firm growth (Crouzet et al., 2022). Productivity trends can be largely influenced by events that make valuable assets accessible to all. For example, large language models such as ChatGPT are extremely expensive to develop, train, and operate. While firms invest many resources in developing their own models and solutions, Meta takes a different approach than other firms (e.g., Google, Open AI), by making these models more accessible and releasing the code and weights from the trained models with permissive terms. The ways in which these software assets are distributed could have a significant impact on productivity in the future.

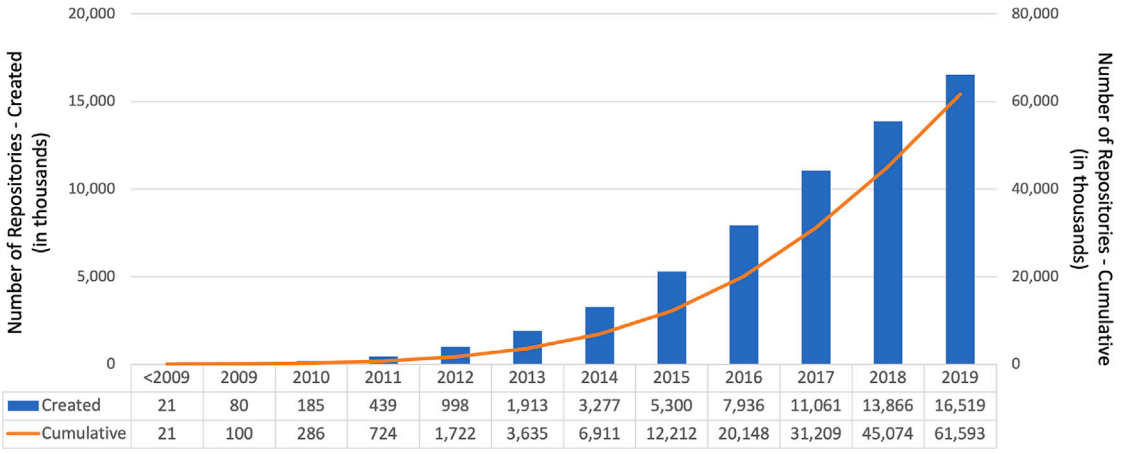
5. Data and methods

While economic measurement of goods and services is best accomplished using a market price for a well-defined quantity of output, OSS cannot be distinctly measured based on observable prices and quantities or based on total revenue. Many OSS projects are developed in free repositories and information embedded in these repositories such as the code, contributors, and development activity is publicly available. Current and comprehensive survey data do not exist for the contributions of OSS. However, as OSS is disseminated online, a wealth of information is available in the code and headers of the software programs themselves. This paper shows how these data can be used to develop a measurement framework for OSS, inspired by the measurement of own-account software investment in the national accounts. In this section, we describe the data and methods used to estimate the total costs involved in the production of software. Details are provided in Appendix A.

5.1. Data

Mining software repositories is an active area of research. Researchers have developed archival projects like the Software Heritage (Di Cosmo and Zacchiroli, 2017), which aims to preserve software, databases of repository data, such as World of Code (Ma et al., 2019), and snapshots of GitHub data like GHTorrent (Gousios, 2013). Additionally, GitHub has developed products such as the GitHub Innovation Graph to publish data and metrics about activity on GitHub, highlighting trends in different global economies and international collaboration patterns.

For this analysis, we collect and use publicly available metadata about OSS projects developed on GitHub, the largest platform in the world with over 100 million developers and 284 million public repositories (GitHub, 2023). The data include information about their contributors and organizations, as well as information within the code.



	<2009	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Created	21	80	185	439	998	1,913	3,277	5,300	7,936	11,061	13,866	16,519
Cumulative	21	100	286	724	1,722	3,635	6,911	12,212	20,148	31,209	45,074	61,593

Fig. 3. Number of public GitHub repositories (created and cumulative), 2009–2019.
Source: GitHub API v4, data collected on May 2023.

Fig. 3 provides the trend in the creation of public repositories as well as the cumulative number of public repositories on GitHub. In 2019, 16 million repositories were created, bringing the total number of active repositories to over 60 million.

We first queried the GitHub database for all public repositories with a machine-detectable[9] OSI-approved license that was original (i.e., non-fork, non-mirror) and non-archived.[10] The majority (57%) of the repositories in our dataset are licensed under the MIT license, followed by the Apache-2.0 license (15%), and a mixture of GPL-3.0 (14%) and GPL-2.0 (5%) licenses. It is worth noting that the reported license distribution closely aligns with the distribution of licenses for all public repositories available on GitHub as of 2020 (GitHub Innovation Graph).

After establishing our universe of repositories with OSI-approved licenses, we collected all of the development activity (approved code edits referred to as commits) from January 2009 through December 2019 using the GHOST.jl software package (Santiago Calderón, 2020). The information includes the time of the code edit and the lines of code added and deleted by each developer. We constructed a table for all developers' contributions to each project as well as when and how much they contributed (measured by lines added). Our final dataset includes 7.6 million repositories and 3.2 million unique contributors associated with these repositories.

5.2. Cost estimation method

This paper presents a methodology that is inspired by the measurement of own-account software investment in the national accounts. This involves the estimation of the total production cost of software. The methodology to estimate own-account software investment relies on assumptions regarding the number of employees in certain occupations and industries (see Appendix A.2 for details). In contrast, our approach is to use the observable characteristics of the output (i.e., lines of code added each year during development) to estimate the time and effort required to develop the software. To calculate the estimated person-months from the lines of code, we use an adaptation of the basic Constructive Cost Model (COCOMO) from software engineering, which is described below.

Using lines of code as a measure of effort stems from the need to ensure large software projects are on schedule and completed within budget. It has motivated development of a body of literature on cost estimation within software engineering (Sharma et al., 2011). While costs can be estimated as a function of the number of instructions, as software projects grow, effort increases nonlinearly. Different cost models account for complexity, reliability, and scale in a variety of ways. These are based on characteristics of the product, the platform, the contributors, and the project. Examples of these estimation models include Constructive Cost Model (COCOMO II), the Putnam Software Life Cycle Management model (Sharma et al., 2011), and models based on function points (Boehm and Valerdi, 2008).

Our approach is a close adaptation of the basic COCOMO (Boehm, 1984), which was originally developed for estimating the cost and effort of developing proprietary software projects. It was designed to be flexible and adaptable to various software development projects. The intuition behind this model is that the required effort for software can be estimated by observable characteristics such as the project size (which can be measured in total lines of code). The model provides a calibration factor – the effort required in person-months to produce a set number of lines of code – and a range of parameters that can be used to adjust for the type of the software project. COCOMO provides different parameter values for different software types (organic, semi-detached, and embedded). For our main results, we selected the parameters for the organic software, which consists of software dealing with a well-known programming language and a small, but experienced team of contributors. We used the parameters for the other types for sensitivity analysis. A more detailed description of our methodology is provided in Appendix A.2.

The effort in person-months and, consequently, the nominal development time is computed by applying the calibrated model parameters for the organic software to the total lines of code added. The resulting development cost in person-months is multiplied by a monthly resource cost to estimate the total production cost. The monthly resource cost includes the wages and salaries of the occupations included for own-account software and the respective adjustment factor to incorporate non-wage components (e.g., additional labor costs, capital services, intermediate inputs). The wage series and adjustment factors are consistent with those of the own-account software methodology described in Appendix A.1. This results in the current (nominal) investment series. Lastly, we apply a price deflator to the nominal series to adjust for inflation and quality changes over time. This yields the real investment series which are used to compute the net stock estimates.

[9] GitHub uses the Ruby Gem Licensee to detect license detail using the LICENSE file in the repository.

[10] Forks refer to copies of a repository created to make modifications without altering the original project. They may be used temporarily to test features to be incorporated to the parent repository. Mirror is an automatically synchronized copy of a repository that is commonly used to keep a backup copy of the repository.

Table 3
Cumulative number of repositories contributed to by the institutional members of the U.S. federal government (2010–2019).

U.S. institution	Count
Federal total	15,716
Department of Energy (DOE)	11,156
National Aeronautics and Space Administration (NASA)	1,102
Department of Health and Human Services (HHS)	863
Department of Commerce (DOC)	819
Department of the Interior (DOI)	537
Department of Defense (DOD)	321
General Services Administration (GSA)	319
Smithsonian Institution	107
Department of Agriculture (USDA)	104
Department of Veterans Affairs (VA)	76
All others	312
Addenda	
Microsoft (business)	25,365
Red Hat (business)	24,767
University of California, Berkeley (academic)	7,152

Note: The OSS projects of the agencies and the respective GitHub repositories were obtained from Code.gov (2016). Details are provided in National Science Board (2022).

5.3. Country and sector assignment

Our goal is to estimate total annual U.S. investment in OSS and investment by the different sectors of the U.S. economy (i.e., business, academic, government, nonprofit, household). This requires assigning developers to countries and sectors using publicly available information. We used the information provided by the GitHub users in their profile such as short biography (bio), location, company, and public email address. We also used text analytics methods utilizing software packages (Kramer, 2021a,b) that make use of other datasets for respective sectors. The datasets include Hipo Labs' university domain list[11] for the academic sector, U.S. Government Domain list,[12] A-Z Index of Government Departments and Agencies,[13] and a list of nonprofits that administer government laboratories extracted from the U.S. Government Federally Funded Research and Development Centers dataset.[14] Robbins et al. (2021) use a similar approach to obtain the OSS contributions from the federal government agencies. Table 3, which was previously published in National Science Board (2022), shows the number of repositories contributed to by GitHub users that provide institutional emails associated with the U.S. federal government. The addenda of Table 3 provide context on how the contributions by the federal government compare to those by the largest contributors in the business and academic sectors.

Table 4 illustrates a breakdown of users' assignment to country and sectors based on our method. Note that not all developers provide complete information in their profiles, hence we were not able to map those contributors to countries or sectors.

For the academic sector, we found that academic users based in the United States are the most prominent OSS contributors, with approximately one-third of the institutions being from the United States. Users from those institutions make up around 55% of all users in the academic sector. When we examined the top universities, we found that 15 of the top-20 universities were based in the United States. Fig. 4 plots the top-20 U.S.-based universities based on the number of OSS developers. We observed that half of these are private institutions.

After assigning users to their respective countries or sectors, the annual investment estimates per repository were allocated to the United States and to respective public and private sectors using fractions that were weighted by the contributions of the users (measured by lines of code added). If a contributor is identified as being affiliated with multiple countries the country weight is divided equally among each. The total country/sector/year estimates presented in Section 6 are the aggregated sum of the investment across all contributors.

[11] https://raw.githubusercontent.com/Hipo/university-domains-list/master/world_universities_and_domains.json.
[12] https://home.dotgov.gov/data/.
[13] https://www.usa.gov/federal-agencies.
[14] https://www.nsf.gov/statistics/ffrdclist/.

Table 4
Country and sector assignment to contributors (2009–2019).

	Total (k)	% All	% Valid
Total users	3188.5	100	–
Any location information	1247	39.1	100
valid country	1202	37.7	96.5
Any sector information	1204.9	37.8	100
In any sector	381.7	11.9	31.7
Business	254.4	7.9	21.1
Academic	99.6	3.1	8.3
Household	11.1	0.3	0.9
Government	3.7	0.1	0.3
Nonprofit	1.7	0.1	0.1

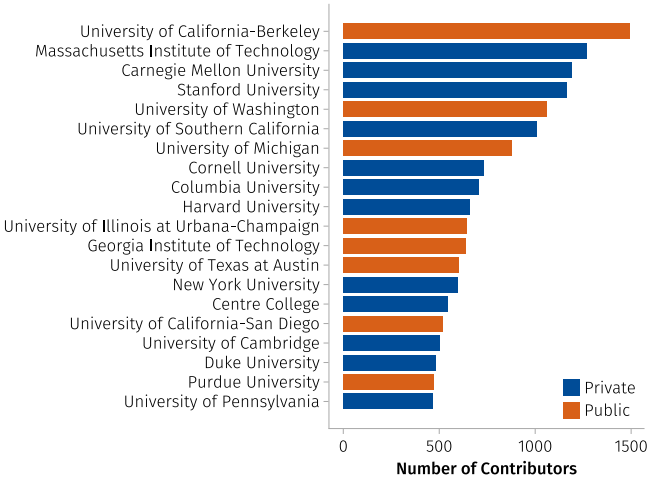


Fig. 4. Top U.S. universities based on the number of developers.

6. Results

In this section, we summarize our findings. In Section 6.1, we illustrate contributions to OSS by the top contributing countries to demonstrate the relative size of U.S. investment in OSS. In Section 6.2, we provide estimates of U.S. investment in OSS measured by nominal and real investment, and net stock estimates, respectively.

6.1. International investment in OSS

Here we show countries with the highest level of OSS contributions on GitHub for the 2009–2019 period. Fig. 5 presents top 10 countries based on the number of contributors. We observe that there are over 300,000 U.S.-based developers, making the United States the top contributor, followed by China (CHN), India (IND), Germany (DEU), and the United Kingdom (GBR).

As described in Section 5.2, our method estimates the effort (in person-months) required to produce the software asset. Fig. 6 shows effort in person-years for the top 10 countries. We observe that the effort contributed by each country increases over time. In 2019, about 41% of the contributions came from U.S-based developers, corresponding to the effort of approximately 185 thousand full-time contributors. As noted above, two-thirds of the commit activity is associated with users that have not been assigned to a country, leading to an underestimate (see Section 5.3).

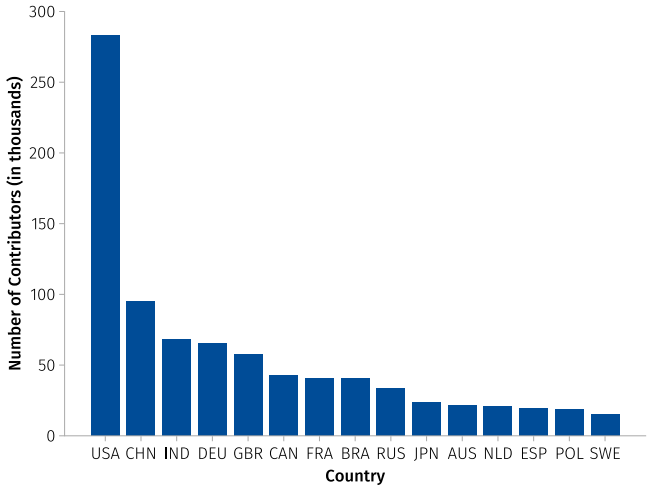


Fig. 5. Top countries with most contributors (2009–2019).

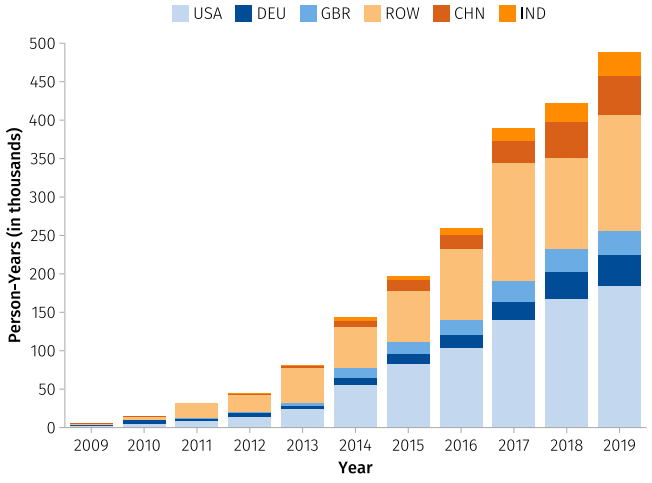


Fig. 6. Top countries by contributions (in person-years). Note: The top 5 countries in 2019 were the United States (USA), Germany (DEU), the United Kingdom (GBR), China (CHN), and India (IND). ROW stands for the *rest of the world*.

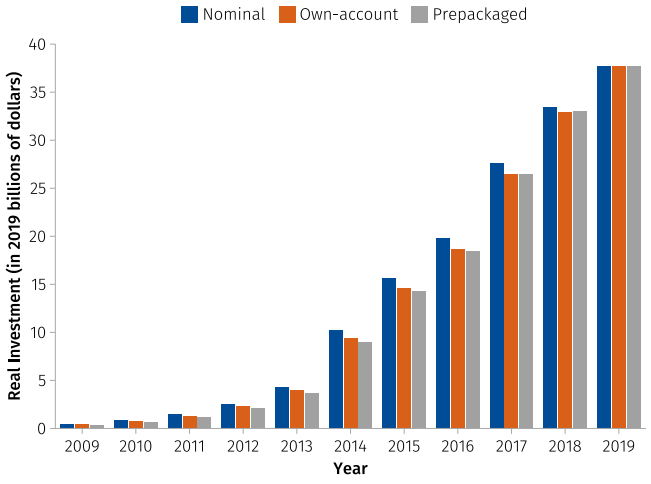


Fig. 7. U.S. real investment in OSS using alternative deflators (2009–2019). The nominal deflator uses a value of one for all periods resulting in the real series being the same as the nominal series. The series is included for reference.
Source: Published series (U.S. Bureau of Economic Analysis, 2023e,f).

The COCOMO assumes 132 productive hours per month for the United States (Jones, 2007). Based on the total number of hours worked by employees and the total number of employees and self-employed workers engaged in production published by BEA (U.S. Bureau of Economic Analysis, 2023n,o), we calculate that in 2019 about 0.115% of U.S. workers' time was employed contributing to the development of OSS.

6.2. U.S. investment in OSS

In this section, we focus on U.S. investment in OSS. Table 5 presents the estimates of the U.S. nominal investment in OSS by sector for the 2009–2019 period. The annual investment in OSS for 2019 is estimated at approximately $37.8 billion.[15] This compares to BEA's estimate for own-account software of $71.2 billion in 2019 and $448 billion for private investment in software of all types (prepackaged, custom, and own-account). For context, BEA reports private investment in R&D in 2019 at $533.2 billion.

We also provide real investment estimates, since they are important for productivity analysis. These are the nominal estimates adjusted for changes in quality and changes in price over time. Table 6 provides the estimated real investment in OSS for the United States in 2019 dollars using the own-account software price index. However, the own-account price index is an input-based index that is quite conservative. Fig. 7 shows the real investment series assuming a nominal price index (i.e., price = 1 at every period), the own-account price index, and prepackaged software price index. While development of OSS likely mirrors that of own-account, OSS usage is more likely to follow that of prepackaged software in which many non-developer actors use the software. Using the prepackaged software price index would result in an average log growth increase of 2.38% through 2009–2019.

In Fig. 8, we report the net stock estimates including growth rates for the 2009–2019 period. Current-cost net stock estimates were computed using the annual investment estimates, own-account software prices, and a perpetual inventory method (PIM). Using PIM, we calculate the net stock in each year as the cumulative value of gross investment less the cumulative value of depreciation. Using PIM and own-account depreciation rates and price index we estimate a net stock of OSS of $74.3 billion.[16] The annualized percentage change between 2009 and 2018 for annual investment and net stock are both over 60%. In Fig. 9, we provide the net stock estimates (the value of fixed assets adjusted for depreciation) for OSS compared to the series for software in private fixed assets.

As both Figs. 8 and 9 illustrate, the net stock of OSS is growing fast (average log growth rate of 42%) and even faster than other types of software such as total software which grew at an average of 6.3%. This emphasizes the importance of improved measurement and the potential impact of this fast-growing asset.

6.3. Robustness and sensitivity analysis

The framework used to obtain the estimated effort and nominal development time uses various assumptions. The appendix explains these more thoroughly (see Appendix A.2). Many sources of variability impact estimates of studies such as this one, including team-effects (Yu and Kumbier, 2020). For example, when employing COCOMO, a number of factors change depending on the type of software being developed. These different software project types are meant to reflect

[15] We suspect that the investment for 2019 is underestimated due to the timing of the data collection, which was early 2020. The estimate does not capture the development activity of projects that were not yet published.

[16] https://www.bea.gov/help/glossary/perpetual-inventory-method.

Table 5
U.S. nominal investment in OSS by producing sector (in millions of dollars).

	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
All activity	473	862	1480	2529	4361	10,210	15,644	19,826	27,590	33,482	37,761
In a sector	223	212	371	584	1601	4,104	6,647	8,325	13,092	14,754	17,432
Private	181	145	272	387	907	2,062	3,074	3,866	5,466	6,715	6,729
Business	181	135	269	381	889	1,911	2,978	3,610	5,127	6,287	6,406
Nonprofit	–	10	2	4	9	89	27	149	107	166	134
Household	–	–	1	2	9	62	69	107	232	262	189
Government	1	3	6	8	29	68	64	199	263	229	445
Academic	41	64	93	189	665	1,974	3,509	4,260	7,363	7,810	10,258

Note: See Appendix A.1 for details.

Table 6
U.S. real investment in OSS by producing sector (in 2019 millions of dollars).

	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
All activity	424	784	1336	2283	3983	9409	14,600	18,698	26,496	32,903	37,761
In a sector	200	193	335	528	1461	3782	6,203	7,853	12,574	14,498	17,432
Private	162	132	246	350	828	1900	2,868	3,647	5,250	6,598	6,729
Business	162	123	243	344	812	1761	2,779	3,405	4,924	6,178	6,406
Nonprofit	–	9	2	4	8	82	25	141	103	163	134
Household	–	–	1	2	8	57	64	101	223	257	189
Government	1	3	5	7	26	63	60	188	253	225	445
Academic	37	58	84	171	607	1819	3,275	4,018	7,071	7,675	10,258

Note: Real investment adjusts for inflation and quality difference across time. The series uses the own-account software prices (U.S. Bureau of Economic Analysis, 2023e).

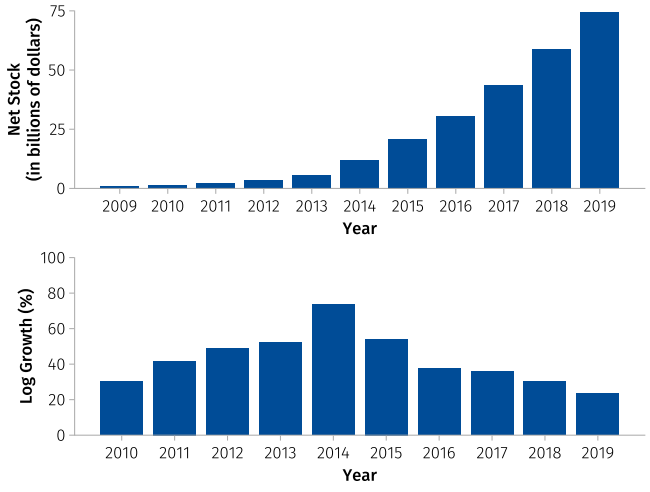


Fig. 8. U.S. net stock estimates of OSS (2009–2019). Note: Current-cost (2017) net stock estimates were computed using the annual investment estimates and the perpetual inventory method (PIM). We used the own-account software (OAS) depreciation rate and prices.

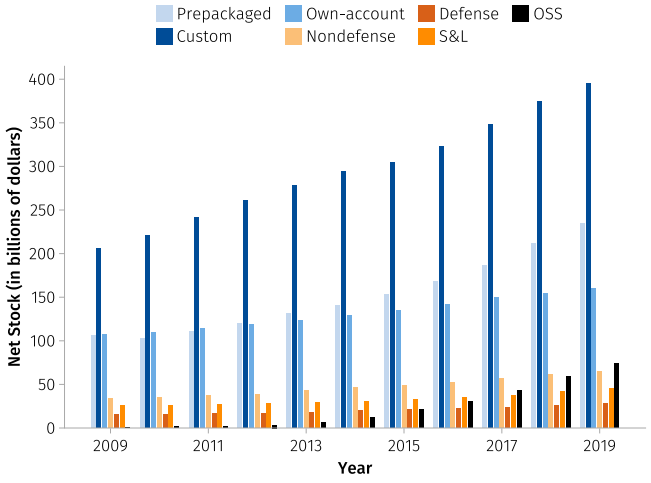


Fig. 9. Trends of net stock estimates of software by type (2009–2019). Note: Current-cost (2017) net stock estimates were computed using the annual investment estimates and the perpetual inventory method (PIM). The OSS series is compared to the software assets published series (U.S. Bureau of Economic Analysis, 2023m,p).

distinguishing aspects such as the complexity of the implementation and team sizes. The results presented in this study assume the projects included in our data are best described as "organic" (see COCOMO's definition of software types in Appendix A.2). This is the project type with the smallest parameter values, leading to shorter relative development time based on the same lines of code observed.

To assess the stability of the estimates and how much variability is introduced due to different team sizes, we conducted two exercises. The first one was to relax the assumption that all projects in our data were best described as the organic type. We allowed for them to be heterogeneous based on the team sizes. Then we categorized the repositories into three groups – small, medium, and large – based on the number of distinct contributors. We assigned them different parameter values based on the COCOMO's definitions of software types (organic, semidetached, and embedded). Repositories with small teams (1-5 contributors), which accounted for 83.5% of all repositories, were classified as organic. Medium-sized repositories, with 6–10 contributors, were assigned parameter values for semidetached software, and the repositories with larger teams which involved 11 or more contributors (comprising 7% of all repositories), were classified as embedded.

This first exercise yielded estimates indistinguishable from those we obtained assuming all projects are of the same type. The second exercise explored the upper bounds by assigning all repositories in the data the highest parameter values (i.e., assuming embedded). On average, the percentage difference between the lowest and highest estimates for each year was 2.8%. The variation was relatively small, with the highest difference being less than 4% and the lowest at 2.8%. This slight difference can be attributed to two main factors. Firstly, most repositories have small team sizes, which means that the organic software type carries significant weight. Secondly, the size of contributions follows a power law distribution, as depicted in Fig. 10. This distribution indicates that the majority of repositories have relatively small values, while a few repositories account for the majority of contributions. The COCOMO parameters, functioning as an exponent, have a lesser impact

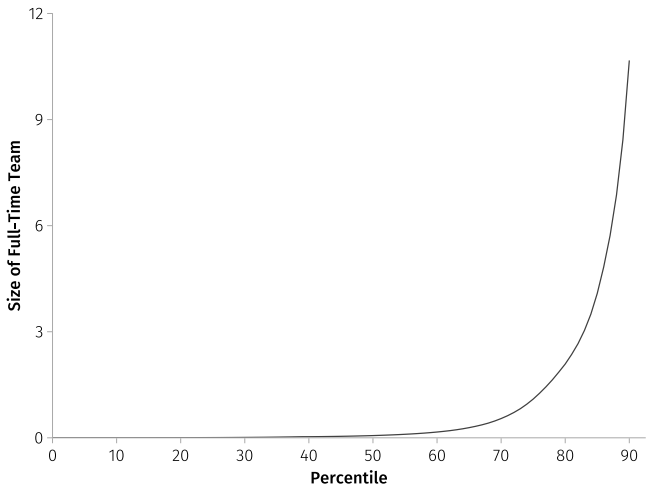


Fig. 10. Distribution of annual person-month development in 2019. Note: Data are trimmed at the upper 90th percentile.

in cases like these. Additionally, the distribution shown in Fig. 10 reveals that in 2019, only 25% of projects had an investment equivalent higher than that of a full-time contributor.

Other sources of variability, such as the annual average wages and production-to-payroll ratios, also failed to alter the study results significantly. It is our conclusion, having considered multiple potential sources of variability, that the results are relatively stable, at least at the relevant precision commonly accepted in national accounting.

7. Discussion

In this section, we discuss assumptions and limitations of our approach. The first relates to finding the universe of OSS or a representative sample through a census of GitHub. While there are certainly many projects that are not on GitHub (e.g., large projects use other version control systems like Apache Subversion SVN), GitHub has significantly more projects and contributors than other services such as SourceForge with 500,000 public repositories or Bitbucket with 10 million users based on the information provided on their websites. Furthermore, not all OSS projects have a machine-detectable license. Large projects with OSI-approved licenses on GitHub such as Python and Julia are not detected as such due to the "license" text not being standard. However, all these introduced biases have a downward direction, suggesting our estimates to be a lower bound.

The second challenge relates to the data and methods used to assign developers to countries and sectors. As described in Section 5.3, we use the self-reported information in GitHub users' profiles such as short biography (bio), location, company, and public email address. First of all, not all developers provide complete or accurate information in their profiles, or the information about contributors and repositories (e.g., emails, licenses) may change over time. This leads to inaccurate measurement of the contributions. Moreover, GitHub tracks a user's contribution (i.e., commits) to a repository by linking the email associated with the commit and those provided by the user in their account. If an email account associated to a user is removed (e.g., user loses access or email is no longer active), the contributions can no longer be attributed to that user (Dohm, 2017), affecting the data quality. Using this type of data requires a deep understanding of the nature of version control systems and Git hosting platforms (Kalliamvakou et al., 2016).

The use of lines of code as a measure of effort could also be problematic. First of all, lines of code do not account for the quality of the code. This may be addressed by capturing the type of code added (e.g., complex functions vs. standard commands) and the programming language using more advance methods. Moreover, some repositories include items other than code such as data files (e.g., JSON, CSV), or auto-generated documentation (e.g., HTML). This results in overestimation of the time and effort.

Project teams may have different conventions which may also limit the scope of our data. For example, we examine only the main branch of the repository, which may lead to under counting contributions for projects that use different branches for production. Similarly, work that was not committed (e.g., open/not-merged pull requests) would not be captured in our estimates, discounting time/effort from users. Moreover, current contributions to projects in development are only observable once those projects are made public. This could happen in the following years, giving rise to a delay effect with a downward bias for recent years. However, these distortions should be relatively negligible at a national accounts level.

The production cost approach implies that the monetary value assigned to an intangible good that is not traded in the marketplace, also known as the shadow price, is accounted for with changes in the prices of inputs. Changes in labor costs for software developers can partially account for some of this, but the absence of an output price for OSS leaves the actual value of the software unmeasured. As a result, this approach misses potential changes in the productivity of the production process (Fleming, 2022). This challenge is not exclusive to our measurement framework. As OSS and other new products emerge, existing methodologies may not account for price declines, leading to an underestimation of productivity (Nakamura, 2022).

Many assumptions we make regarding the development process are very sensible from a national accounts framework. This is especially the case when we take into account the common OSS development process. Still, as the methodology matures, some refinements could be implemented. For example, increasing the number of users who are assigned to sectors, such as the federal or state and local, would allow us to more closely align with the national accounts methodology (which uses different inputs depending on whether the contributions came from the public or private sector). These refinements will likely not affect for overall estimates but would be helpful for analysis at a higher resolution (e.g., per firm or institutions of higher education). In terms of better assignment of contributors, there are multiple promising approaches. GitHub, for example, recently enabled accounts to link up to four social media profiles. This allows for better access to understanding contributors' backgrounds (e.g., education, experience, affiliations), and social networks, which may provide insights into their work.

There are other areas where improvements could have a significant impact. For example, the COCOMO81, the first version of the framework, was released in the late 1970s and aimed to be reflective of the software development practices of the 1980s. The COCOMO II model was developed using regression models on a sample of 163 projects and aimed to be reflective of software development practices in the 1990s. It is not hard to imagine that these have changed considerably in the last several decades, especially, with the emergence of the internet, newer programming languages, and more recently, the potential disruption of large language models (LLMs) such as ChatGPT and AI tools such as GitHub's Copilot. Nowadays, administrative records would allow for access to larger sample sizes of projects, practices, and features to refine and obtain higher confidence in cost models.

8. Conclusion

In summary, inspired by a concern that OSS is not adequately measured in the current macroeconomic statistics, acknowledging the importance and value of obtaining good measurements, and recognizing a body of research that suggests the present distortions are significant, we developed a framework that can serve as a basis to address the measurement problem. Our approach adapts the current national accounts methodology while incorporating a cost estimation

method developed in software engineering to estimate the cost of developing software. It overcomes challenges such as defining and cataloging the OSS universe, it takes into account the differences between OSS and other IPPs that are relatively easier to capture through administrative records (e.g., copyrights, patents), and it adapts strategies from bibliometrics (i.e., repositories in hosting platforms serve the function of academic articles on journals). Finally, we obtain a sensible estimate of the share of software investment that fuels OSS and report the value for 2019: $37.8 billion (≈53% of own-account software investment). These data provide a strong baseline for resource cost estimates and our goal is to repeat the process each year to provide a consistent picture of how the OSS ecosystem evolves over time.

The practical implications of our measurement framework are as follows. OSS has the potential to create spillover effects in many areas that impact productivity in various sectors. This makes it an important subject for policy consideration. Measuring OSS will fill an important gap in the measurement of public investment in intangible assets and innovation. It will also highlight what implications there will be for measuring productivity and economic growth. In the business sector, our measurement framework will allow for disentangling proprietary software and OSS and measuring their impact on productivity. Outside the business sector, reliable measures of the scope and impact of OSS are scarce. For example, our framework can allow for measurement of university contributions in a broader way supplementing measures of publications and patents.

To build on this framework and method for the measurement of OSS and intangible assets more broadly, more work is needed to improve measurement by type of OSS (Fleming, 2022) and by sector. Further, the application of artificial intelligence tools may greatly increase developer productivity. Future extensions of this work will focus on contributions by economic sector, development of tools to gather and analyze OSS data at product level granularity, and development of scientific and economic indicators based on the methodology that can reach a production-ready standard in the future.

CRediT authorship contribution statement

Gizem Korkmaz: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **J. Bayoán Santiago Calderón:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing, Conceptualization. **Brandon L. Kramer:** Data curation, Formal analysis, Software, Validation, Visualization, Writing – review & editing. **Ledia Guci:** Data curation, Formal analysis, Methodology, Validation, Writing – review & editing. **Carol A. Robbins:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is made available on a repository.

Acknowledgments

This work was supported by the National Science Foundation under Grant Numbers 2306160 and 2224441, and PIID: 49100420C0015; and the U.S. Department of Agriculture (USDA) [FAIN: 583AEU70074].

The authors would like to thank anonymous referees for their valuable comments and feedback, and would also like to express their thanks to Martin Fleming, Frank Nagle, Shane Greenstein, Wesley Cohen, Dylan Rassier, Juan Mateos Garcia, Manuel Hoffman, and Eric von Hippel whose comments and suggestions have significantly improved the paper.

The authors acknowledge Aaron Schroeder and the Data Science for the Public Good Young Scholars program participants Daniel Bullock, Daniel Chen, Cong Cong, Calvin Isch, Morgan Klutzke, Eliza Tobin, and Crystal Zang. The authors also acknowledge the University of Virginia Research Computing for providing computational resources that contributed to the results reported within this publication. We are also grateful to GitHub for providing programmatic access to the data through their Application Programming Interface (API). The earlier versions of the paper were presented at the New York University's Economics of Open Source Virtual Workshop (2022); at the International Association for Research in Income and Wealth (IARIW) and the UK Economics Statistics Centre of Excellence (ESCoE) 2021 Conference on Measuring Intangible Assets and their Contributions to Growth; and at the Society of Government Economists Session on Innovation, Growth, and Trade at the American Economic Association/Allied Social Science Associations (ASSA) Annual Meeting (2022).

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, the National Center for Science and Engineering Statistics, the U.S. Department of Agriculture, the U.S. Bureau of Economic Analysis, or the U.S. Department of Commerce.

Appendix A. Technical appendix

A.1. Measurement of own-account software in the national accounts

The development of OSS aligns more closely with own-account software development compared to other software categories. The estimation of own-account software in the national accounts follows the recommendations of the System of National Accounts by estimating the costs of production (United Nations, 2010). These costs include labor costs (wages and salaries and other components of total employee compensation), intermediate inputs, capital services (including the consumption of fixed capital, and normal profits), and other costs incurred in production, such as business taxes.

The first step in estimating the production of own-account software is to identify the work activities related to its production and the occupations and industries that perform these activities. Until 2013, the occupations considered in scope for own-account software were computer systems analysts, computer programmers, software developers, and software quality assurance analysts and testers. Since 2013, the occupations have been expanded to include web developers and digital interface designers, database administrators and architects, network and computer systems administrators, computer network architects, and computer and information research scientists. For each occupation, BEA assumes only a certain percentage of their work is allocated to developing software.

The number of employees and their annual average wages and salaries associated with own-account software investment (i.e., procurement of a software asset using internal resources) are multiplied to

obtain a payroll estimate. The payroll estimate is adjusted for non-wage employee compensation (e.g., employer contributions for employee pension and insurance funds), intermediate inputs (e.g., materials, energy, purchased services), capital services, and other production costs. This adjustment is estimated using the ratio of total revenue to wages and salaries of the representative industry Computer Systems Design and Related Services (NAICS5415). The value of the adjustment factor has historically ranged from 2.02 (in 2007) to 2.07 (since 2013). The source for occupational employment and wage data is the Occupational Employment and Wages Survey (OEWS) from the Bureau of Labor Statistics (U.S. Bureau of Labor Statistics, 2021). The source for the computation of the markup factor is the Service Annual Survey (SAS) from the Census Bureau (U.S. Bureau of Economic Analysis, 2022).[17]

BEA makes several adjustments to the number of employees to account for R&D in software (e.g., software originals), custom software, capital formation rates, and time spent on activities outside of the definition of the production of the asset (e.g., maintenance). Adjustments for software originals and custom software are made based on the concentration of occupations within industries. R&D software adjustments are made based on R&D surveys from the National Center for Science and Engineering Statistics (NCSES). Lastly, the adjustment for the time use has been updated from a 50% factor applied to all occupations based on a study (Boehm, 1984) using software developer surveys (Parker et al., 2000) to estimating these based on online job postings data. A more detailed description of the methodology is available through BEA publications (U.S. Bureau of Economic Analysis, 2022; Chute et al., 2018; McCulla et al., 2023).

A.2. Cost estimation methodology

Investment in OSS cannot be estimated based on observable prices and quantities or based on total revenue. Instead, we use a production-cost approach inspired by the national accounts methodology for own-account software investment. To translate lines of code into estimates of effort in person-months, we use an adaptation of the basic Constructive Cost Model (COCOMO) from software engineering. COCOMO makes its estimates of required effort (measured in person-months) based on the estimate of the project's size (as measured in thousands/kilo of lines of code (KLoC)), and the equation is given as follows:

$$\text{Effort} = a_i(\text{KLoC})^{b_i} \tag{A.1}$$

COCOMO defines three types of software: organic, semidetached, and embedded; and provides different parameter values for each type (Boehm, 1984). A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem. The projects classified as semi-detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. An embedded software project requires the highest level of complexity, creativity, and experience. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

We selected the coefficients a_i, b_i to match those of organic software type. While we held these consistent across all projects, the model allows for these parameters to be adjusted based on additional data. We selected the organic software type parameters for main results, and used the other parameters for sensitivity analysis.

$$\text{Effort} = (2.4)(\text{KLoC})^{1.05} \tag{A.2}$$

[17] The estimates using the published BEA tables yield very similar estimates.

Given the required amount of effort to develop the lines of code, the schedule or nominal development time can be calculated as

$$\text{Nominal development time} = (2.5)\text{Effort}^{c_i} \tag{A.3}$$

where $c_i = 0.38$ for organic software type.

Finally, we estimate the average staffing for a project based on the observed activity in our data. We calculated the percentile of average monthly contributions across all projects by each contributor. We selected the value at the 88th percentile (i.e., 65,000 lines of code) to represent the typical maximum lines of code a contributor could write in a month. Note that this estimate may not necessarily be comparable beyond our data.

The annual development cost estimate for the repository is then calculated as

$$\text{Cost} = 2.5\left(2.4\,(\text{KLoC})^{1.05}\right)^{0.38}\left(\frac{\text{KLoC}}{65}\right) \tag{A.4}$$

The calculated annual development cost (per repository) is allocated to countries based on the fraction of the lines added by the contributors of that country. If a contributor is identified as being affiliated with multiple countries the country weight is divided equally among each. The process is similar for the sectors (i.e., business, nonprofit, household, government, and academia) in the United States.

Finally, we compute the total production cost in dollars by multiplying the development cost in person-months with a monthly resource cost which includes the wages and salaries of the occupations included for own-account software and the appropriate adjustment factor to incorporate non-payroll costs. The wage series and adjustment factors are consistent with those of the own-account software methodology described in Appendix A.1.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.respol.2024.104954.

References

Aizcorbe, A.M., Moylan, C.E., Robbins, C.A., 2009. BEA briefing: Toward better measurement of innovation and intangibles. URL: https://fraser.stlouisfed.org/title/46/item/10199/toc/359344, Survey of Current Business.

Alexy, O., Henkel, J., Wallin, M.W., 2011. From closed to open: Job role changes, individual predispositions, and the adoption of commercial open source software development. Res. Policy 42.

Andreessen, M., 2011. Why software is eating the world. URL: https://www.wsj.com/articles/SB10001424053111903480904576512250915629460.

Bessen, J., 2022. The New Goliaths: How Corporations Use Software To Dominate Industries, Kill Innovation, and Undermine Regulation. Yale University Press.

Bockstael, N.E., McConnell, K.E., 1983. Welfare measurement in the household production framework. Am. Econ. Rev. 73 (4), 806–814, URL: https://www.jstor.org/stable/1816580.

Boehm, B.W., 1984. Software engineering economics. IEEE Trans. Softw. Eng. SE-10 (1), 4–21. http://dx.doi.org/10.1109/TSE.1984.5010193.

Boehm, B.W., Clark, Horowitz, Brown, Reifer, Chulani, Madachy, R., Steece, B., 2000. Software Cost Estimation with COCOMO II (with CD-ROM), first ed. Prentice Hall, Upper Saddle River, NJ, USA.

Boehm, B.W., Valerdi, R., 2008. Achievements and challenges in COCOMO-based software resource estimation. IEEE Softw. 25 (5), 74–83. http://dx.doi.org/10.1109/MS.2008.133.

Chute, J.W., McCulla, S.H., Smith, S., 2018. Preview of the 2018 comprehensive update of the National income and product accounts. Surv. Curr. Bus. 98 (4), URL: https://apps.bea.gov/scb/2018/04-april/0418-preview-2018-comprehensive-nipa-update.htm.

Code.gov, 2016. Sharing America's Code: Unlock the tremendous potential of the Federal Government's software. https://code.gov/.

Corbet, J., Kroah-Hartman, G., 2017. 2017 Linux Kernel Development Report. Annual Report, The Linux Foundation, URL: https://web.archive.org/web/20221205085006/https://storage.pardot.com/6342/188781/Publication_LinuxKernelReport_2017.pdf.

Corrado, C., Haskel, J., Jona-Lasinio, C., 2017. Public intangibles: The public sector and economic growth in the SNA. Rev. Income Wealth 63, S355–S380. http://dx.doi.org/10.1111/roiw.12325.

Corrado, C., Hulten, C., Sichel, D., 2005. Measuring capital and technology: An expanded framework. In: Corrado, C., Haltiwanger, J., Sichel, D. (Eds.), Measuring Capital in the New Economy. University of Chicago Press, pp. 11–46.

Crouzet, N., Eberly, J.C., Eisfeldt, A.L., Papanikolaou, D., 2022. The economics of intangible capital. J. Econ. Perspect. 36 (3), 29–52. http://dx.doi.org/10.1257/jep.36.3.29.

Dahlander, L., Magnusson, M.G., 2005. Relationships between open source software companies and communities: Observations from Nordic firms. Res. Policy 34 (4), 481–493. http://dx.doi.org/10.1016/j.respol.2005.02.003.

Damanpour, F., 1991. Organizational innovation: A meta-analysis of effects of determinants and moderators. Acad. Manag. J. 34 (3), 555–590. http://dx.doi.org/10.5465/256406.

David, P., Waterman, A., Arora, S., 2003. The Free/Libre Open Source Software Survey for 2003. Technical Report, Stanford Institute for Economic Policy Research, URL: https://web.archive.org/web/20170829111521/http://web.stanford.edu/group/floss-us/report/FLOSS-US-Report.pdf.

Di Cosmo, R., Zacchiroli, S., 2017. Software Heritage: Why and how to preserve software source code. In: IPRES 2017 - 14th International Conference on Digital Preservation. Kyoto, Japan, pp. 1–10, URL: https://hal.archives-ouvertes.fr/hal-01590958.

DOD CIO, 2021. DOD open source software FAQ. URL: https://dodcio.defense.gov/Open-Source-Software-FAQ.

Dohm, L., 2017. GitHub community forums. URL: https://github.community/t/how-to-change-author-name-and-email-of-commits/285/6.

Duparc, E., Möller, F., Jussen, I., Stachon, M., Algac, S., Otto, B., 2022. Archtypes of open-source business models. Electron. Mark. 32.

Fleming, M., 2022. Enterprise Information and Communications Technology Software Pricing and Developer Productivity Measurement. Working Paper No. 037, The Productivity Institute, http://dx.doi.org/10.2139/ssrn.4454144.

Fosfuri, A., Giarrantana, M.S., Luzzi, A., 2008. The penguin has entered the building: The commercialization of open source software products. Organ. Sci. 19.

Gambardella, A., Hall, B.H., 2006. Proprietary versus public domain licensing of software and research products. Res. Policy 35 (6), 875–892. http://dx.doi.org/10.1016/j.respol.2006.04.004.

Gault, F., 2018. Defining and measuring innovation in all sectors of the economy. Res. Policy 47 (3), 617–622. http://dx.doi.org/10.1016/j.respol.2018.01.007.

GitHub, 2023. The state of the octoverse. URL: https://octoverse.github.com.

GlobalStats, 2021. Mobile & tablet operating system market share worldwide. https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide.

Gousios, G., 2013. The GHTorent dataset and tool suite. In: Proceedings of the 10th Working Conference on Mining Software Repositories. MSR '13, IEEE Press, pp. 233–236, URL: http://dl.acm.org/citation.cfm?id=2487085.2487132.

Greenstein, S., Nagle, F., 2014. Digital dark matter and the economic contribution of apache. Res. Policy 43 (4), 623–631. http://dx.doi.org/10.1016/j.respol.2014.01.003.

Hauge, Ó., Soares Cruzes, D., Conradi, R., Sandanger Velle, K., Skarpenes, T.A., 2010. Risks and Risk Mitigation in Open Source Software Adoption: Bridging the Gap between Literature and Practice, Vol. 319. Springer, Berlin, Heidelberg, pp. 105–118. http://dx.doi.org/10.1007/978-3-642-13244-5_9.

Jones, C., 2007. Estimating Software Costs: Bringing Realism To Estimating, second ed. McGraw Hill.

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M., Damian, D., 2016. An in-depth study of the promises and perils of mining GitHub. Empir. Softw. Eng. 21 (5), 2035–2071. http://dx.doi.org/10.1007/s10664-015-9393-5.

Keller, S.A., Korkmaz, G., Robbins, C.A., Shipp, S.S., 2018. Opportunities to observe and measure intangible inputs to innovation: Definitions, operationalization, and examples. Proc. Natl. Acad. Sci. (PNAS) 115 (50), 12638–12645. http://dx.doi.org/10.1073/pnas.1800467115.

Korkmaz, G., Kelling, C., Robbins, C.A., Keller, S.A., 2018. Modeling the impact of R packages using dependency and contributor networks. In: In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 511–514. http://dx.doi.org/10.1109/ASONAM.2018.8508255.

Korkmaz, G., Kelling, C., Robbins, C.A., Keller, S.A., 2020. Modeling the impact of Python and R packages using dependency and contributor networks. Soc. Netw. Anal. Min. 10, 1–12.

Kramer, B.L., 2021a. Diverstidy: A tidy package for detection and standardization of geographic, population, and diversity-related terminology in unstructured text data. URL: https://github.com/brandonleekramer/diverstidy.

Kramer, B.L., 2021b. Tidyorgs: A tidy package that standardizes text data for organizational analysis. URL: https://github.com/brandonleekramer/tidyorgs.

Lerner, J., Tirole, J., 2005. The economics of technology sharing: Open source and beyond. J. Econ. Perspect. 19 (2), 99–120. http://dx.doi.org/10.1257/0895330054048678.

Li, W.C., Makoto, N., Kazufumi, Y., 2019. Value of Data: There's No Such Thing as a Free Lunch in the Digital Economy. Discussion Papers 19022, Research Institute of Economy, Trade and Industry (RIETI), URL: https://ideas.repec.org/p/eti/dpaper/19022.html.

Ma, Y., Bogart, C., Amreen, S., Zaretzki, R., Mockus, A., 2019. World of code: An infrastructure for mining the universe of open source VCS data. In: Proceedings of the 16th International Conference on Mining Software Repositories. MSR '19, IEEE Press, pp. 143–154. http://dx.doi.org/10.1109/MSR.2019.00031.

Martin, B.R., 2016. Twenty challenges for innovation studies. Sci. Public Policy 43 (3), 432–450. http://dx.doi.org/10.1093/scipol/scv077.

McCulla, S.H., Turner, D.L., Mataloni, L., 2023. Preview of the 2023 comprehensive update of the national economic accounts. Surv. Curr. Bus. 103 (6), URL: https://apps.bea.gov/scb/issues/2023/06-june/0623-nea-preview.htm.

Microsoft, 2018. Microsoft acquires GitHub. URL: https://news.microsoft.com/announcement/microsoft-acquires-github/.

Nagle, F., 2019. Open source software and firm productivity. Manage. Sci. 65 (3), 1191–1215. http://dx.doi.org/10.1287/mnsc.2017.2977.

Nagle, F., Wheeler, D.A., Lifshitz-Assaf, H., Ham, H., Hoffman, J.L., 2020. Report on the 2020 FOSS Contributor Survey. Technical Report, The Linux Foundation, URL: https://web.archive.org/web/20230527013309/https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/2020FOSSContributorSurveyReport_121020.pdf.

Nakamura, L.I., 2022. Deflating research and development investment: Some new ideas and estimates. URL: https://iariw.org/wp-content/uploads/2022/08/Nakamura-IARIW-2022.pdf.

Nakamura, L.I., Samuels, J., Soloveichik, R.H., 2017. Measuring the 'free' digital economy within the GDP and productivity accounts. URL: https://www.bea.gov/research/papers/2017/measuring-free-digital-economy-within-gdp-and-productivity-accounts.

Nakamura, L.I., Soloveichik, R.H., 2015. Valuing 'Free' Media Across Countries in GDP. FRB of Philadelphia Working Paper, http://dx.doi.org/10.2139/ssrn.2631621.

National Science Board, 2022. Invention, Knowledge Transfer, and Innovation. Science and Engineering Indicators 2022: Table INV-4. Cumulative Contribution of Selected Entities to Open-Source Software on GitHub: 2010–19. Technical Report, URL: https://ncses.nsf.gov/pubs/nsb20224/table/INV-4.

Netcraft, 2017. Web server survey. https://news.netcraft.com/archives/2017/11/21/november-2017-web-server-survey.html.

Netcraft, 2023. Web server survey. https://www.netcraft.com/blog/may-2023-web-.

Parker, R.P., Grimm, B.T., et al., 2000. Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98. Technical Report, Bureau of Economic Analysis, URL: https://www.bea.gov/research/papers/2000/recognition-business-and-government-expenditures-software-investment.

Piwowar, H., Priem, J., 2016. Depsy: Valuing the software that powers science. https://github.com/Impactstory/depsy-research/blob/master/introducing_depsy.md.

Raymond, E., 1999. The cathedral and the bazaar. Knowl., Technol. Policy 12 (3), 23–49.

Red Hat, 2019-07-09. IBM closes landmark acquisition of Red Hat for $34 billion; defines open, hybrid cloud future. https://www.redhat.com/en/about/press-releases/.

Robbins, C.A., Korkmaz, G., Guci, L., Santiago Calderón, J.B., Kramer, B., 2021. A first look at open source software investment in the United States and in other countries, 2009–2019. In: International Association for Research on Income and Wealth (IARIW) ESCoE Conference. IARIW, URL: https://iariw.org/wp-content/uploads/2021/11/robbins-paper.pdf.

Robbins, C.A., Korkmaz, G., Santiago Calderón, J.B., Kelling, C., Shipp, S.S., Keller, S.A., 2018a. Open source software as intangible capital: Measuring the cost and impact of free digital tools. In: The Sixth IMF Statistical Forum: Measuring Economic Welfare in the Digital Age: What and how?. International Monetary Fund (IMF), p. III1, URL: https://www.imf.org/en/News/Seminars/Conferences/2018/04/06/6th-statistics-forum.

Robbins, C.A., Korkmaz, G., Santiago Calderón, J.B., Kelling, C., Shipp, S.S., Keller, S.A., 2018b. The scope and impact of open source software: A framework for analysis and preliminary cost estimates. In: Presented At the 35th International Association for Research on Income and Wealth (IARIW) General Conference. IARIW, p. 2A5, URL: http://old.iariw.org/copenhagen/robbins.pdf.

Santiago Calderón, J.B., 2020. GHOST.jl. URL: https://github.com/team-oss/GHOST.jl.

Sharma, T.N., Bhardwaj, A., Sharma, A., 2011. A comparative study of COCOMO II and Putnam models of software cost estimation. Int. J. Sci. Eng. Res. 2 (11).

St. Laurent, A.M., 2004. Understanding Open Source and Free Software Licensing, first ed. O'Reilly Media, Inc..

The Linux Foundation, 2020. 2020 Linux Kernel Development Report. Linux Kernel History Report, The Linux Foundation, URL: https://web.archive.org/web/20220919201352/https://project.linuxfoundation.org/hubfs/Reports/2020_kernel_history_report_082720.pdf.

The PostgreSQL Global Development Group, 2020. PostgreSQL 13.1 documentation. URL: https://www.postgresql.org/docs/13/history.html.

Torvalds, L., Diamond, D., 2001. Just for Fun: The Story of an Accidental Revolutionary, first ed. HarperBusiness.

United Nations, 2010. System of National Accounts 2008. United Nations, http://dx.doi.org/10.18356/4fa11624-en.

U.S. Bureau of Economic Analysis, 2013. Preview of the 2013 comprehensive revision of the National income and product accounts: Changes in definitions and presentations. URL: https://www.bea.gov/information-previous-updates-nipa-accounts.

U.S. Bureau of Economic Analysis, 2022. NIPA handbook: Concepts and methods of the U.S. National income and product accounts. URL: https://www.bea.gov/resources/methodologies/nipa-handbook.

U.S. Bureau of Economic Analysis, 2023a. Government gross investment: Federal: National defense: Gross investment: Intellectual property products: Software. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y053RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023b. Government gross investment: Federal: Nondefense: Gross investment: Intellectual property products: Software. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y068RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023c. Government gross investment: State and local: Gross investment: Intellectual property products: Software. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y072RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023d. Gross government investment. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, A782RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023e. Price index for private fixed investment in intellectual property products: Software: Own account. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y005RG3A086NBEA.

U.S. Bureau of Economic Analysis, 2023f. Price index for private fixed investment in intellectual property products: Software: Prepackaged. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y003RG3A086NBEA.

U.S. Bureau of Economic Analysis, 2023g. Private fixed investment in intellectual property products: Software: Custom. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y004RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023h. Private fixed investment in intellectual property products: Software: Own account. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y005RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023i. Private fixed investment in intellectual property products: Software: Prepackaged. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, Y003RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023j. Private fixed investment: Nonresidential: Intellectual property products: Software. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, B985RC1A027NBEA.

U.S. Bureau of Economic Analysis, 2023k. Private nonresidential fixed investment [PNFIA]. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, URL: https://alfred.stlouisfed.org/graph/?g=15t9g.

U.S. Bureau of Economic Analysis, 2023l. Private residential fixed investment [PRFI]. Retrieved from ALFRED, Federal Reserve Bank of St. Louis, URL: https://alfred.stlouisfed.org/series?seid=PRFI.

U.S. Bureau of Economic Analysis, 2023m. Table 2.1. Current-cost net stock of private fixed assets, equipment, structures, and intellectual property products by type. URL: https://apps.bea.gov/iTable/?ReqID=10&step=2, Series: k1ntotl1ens0, k1ntotl1rd00, k1ntotl1ae00.

U.S. Bureau of Economic Analysis, 2023n. Table 6.8D. Persons engaged in production by industry. URL: https://apps.bea.gov/iTable/?reqid=19&step=2&isuri=1&categories=survey, Series: A4602C.

U.S. Bureau of Economic Analysis, 2023o. Table 6.9D. Hours worked by full-time and part-time employees by industry. URL: https://apps.bea.gov/iTable/?reqid=19&step=2&isuri=1&categories=survey, Series: B4702C.

U.S. Bureau of Economic Analysis, 2023p. Table 7.1. Current-cost net stock of government fixed assets. URL: https://apps.bea.gov/iTable/?ReqID=10&step=2, Series: k1gtotl1ens0, k1gtotl1rd00.

U.S. Bureau of Labor Statistics, 2021. Occupational Employment Statistics: National industry-specific and by ownership. URL: https://www.bls.gov/oes/tables.htm.

van Rossum, G., 1996. Foreword for "programming python" (1st ed.). URL: https://www.python.org/doc/essays/foreword/.

van Rossum, G., 2020. Twitter status. URL: https://twitter.com/gvanrossum/status/1326932991566700549.

von Hippel, E., 2005. Perspectives on Free and Open Source Software. MIT Press, ISBN: 978-0-262-06246-6, pp. 267–278, chapter 14.

von Hippel, E., 2017. Free Innovation. The MIT Press, ISBN: 978-0-262-03521-7.

Yu, B., Kumbier, K., 2020. Veridical data science. Proc. Natl. Acad. Sci. 117 (8), 3920–3929. http://dx.doi.org/10.1073/pnas.1901326117.