Measuring the Cost of Open Source Software Innovation on GitHub

Authors José Bayoán Santiago Calderón, U.S. Bureau of Economic Analysis; Carol A. Robbins and Ledia Guci, National Center for Science and Engineering Statistics; Gizem Korkmaz, The Coleridge Initiative; and Brandon L. Kramer

Contact Jose.Santiago-Calderon@bea.gov

Date July 2022

Abstract Open source software (OSS) is software that anyone can study, inspect, modify, and distribute freely under very limited restrictions, generally attribution. While OSS is vital to virtually all aspects of modern society, there is no standard methodology to satisfactorily measure the scope and impact of these intangible assets. Today, GitHub is the world's largest forge with over 80 million users and 118 million public repositories. This study presents a framework based on GitHub's administrative data to discover, profile, and measure the development of OSS. The data include over 7.75 million original, nondeprecated repositories with a machine detectable OSI-approved license. For each repository, we collect metadata such as commits, license, and information about contributors. Adopting a cost estimation model from software engineering and national accounting methods for measurement of software, we develop a methodology to generate estimates of investment in OSS that are consistent with measures of software investment in the U.S. national accounts. Our current estimates show that the U.S. investment in OSS in 2019 was $36.2 billion.

Keywords Cost Estimation, GitHub, Innovation, Intangible Asset, Open Source Software

JEL Code C82, E22, H42, L17, O3, O51

Bureau of Economic Analysis
U.S. DEPARTMENT OF COMMERCE

1. Introduction

Open source software (OSS) is software that anyone can study, inspect, modify, and distribute freely under very limited restrictions such as attribution (St. Laurent 2004). In practice, the Open Source Initiative (OSI) certifies licenses that comply with the principles of open source; software that is licensed under any OSI-approved licenses is deemed open source. OSS is everywhere, both as specialized applications nurtured by devoted user communities and as digital infrastructure underlying platforms used by millions daily.

OSS is developed, maintained, and extended both within the private sector and outside of it through the contribution of independent developers and people from businesses, universities, government research institutions, and nonprofits. Examples include the Linux operating system, Apache server software, and R statistical programming software.

While the extent and impact of OSS is currently unknown, recent estimates suggest that its magnitude is significant. For example, Apache is estimated to hold the largest market share of domains (35%) and active websites (41%) as of November 2017 (Netcraft 2017). The Apache server, developed with federal and state funds at the National Center for Super-computing 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).

Firms use open source software in different ways, including directly and, through providing supplementary software or services. Indirectly, these interactions can be described as symbiotic, commensalistic, or parasitic (Dahlander and Magnusson 2005). The same dynamics occur outside the private sector with open source being a cornerstone for scientific inquiry (e.g., in academia and national laboratories) and mission critical for operations (e.g., NASA). For this reason, it is not surprising that a large source of contributions to OSS originate from the private sector as well as academic and government-affiliated contributors.

Many OSS projects create long-lived tools that are often outputs of public spending, a kind of freely shareable intangible asset that in many cases have been developed outside the business sector and subsequently used within the business sector. The scale and use of these modifiable software tools highlight an aspect of technology diffusion and flow that is not captured in market measures. Measures of creation and use of OSS would complement existing science and technology indicators on peer-reviewed publications and patents that are calculated from databases covering scientific articles and patent documents. Many well-developed methodologies and extensions exist, and a research community continues to grow, invigorated by improved computing power and algorithms.

We are motivated to better account for both the scale of OSS overall and the contribution of public spending to investments in open source software, a vital component of science activity. Measuring OSS fills an important gap in the measurement of U.S. investment in intangible assets and what implications that has for measurement of productivity and economic growth.

In this paper, we use non-survey data to measure the scope and value of OSS, focusing on the projects hosted and shared on the most popular source-code hosting facility, GitHub, with over 80 million users and 118 million public repositories available to query from their public API. Our strategy incorporates the lines of code for each project as a way to estimate the time/effort it took to develop the projects (B. W. Boehm 1984; Barry W. Boehm, Clark, et al. 2000). The estimated nominal development time computed according to our approach serves as an alternative estimate of OSS investment inspired by the national accounts methodology (U.S. Bureau of Economic Analysis 2018). Applying these alternative measure gives a comparable estimate for the share of investment in software that results in OSS.

2. Measurement of Software in the National Accounts

Software investment in the national accounts is categorized into six accounts depending on who produced it and for what purpose. The six accounts in the U.S. national accounts are: (1) software investment by the federal government for defense, (2) software investment by the federal government for non-defense purposes, (3) software investment by state and local governments, (4) private investment in prepackaged software (software that was purchased), (5) private investment in custom software (commissioned software to be used internally), and (6) private software investment on own-account (in-house development to support internal operations). Annual investment in software in the U.S. for 2019 is estimated at $490 billion, with $62.6 billion from the public sector and $427.7 billion from the private sector (Figure 1). Table 1 shows the economic sectors and software categories that are relevant for OSS.

The current methodology for the measurement of software in the national accounts presents challenges for measuring OSS. For example, OSS developed by the federal, state (including public higher education institutions), or local governments would be captured under the corresponding account (e.g., defense, non-defense, S&L) bundled with what would be prepackaged and custom software. The macroeconomic statistics do not offer a higher resolution that would allow us to answer questions such as how much OSS is being developed annually.

OSS developed in the private sector is categorized under custom software or own account, as it is not purchased, which excludes prepackaged software. However, custom software is developed for internal use, meaning it is designed to work with internal systems, and might not be easily adapted for re-use by others absent access to internal components. Software of this kind has limited usefulness to others and given the fewer incentives to open-source, it is more likely that OSS is developed as own account software compared to custom software.

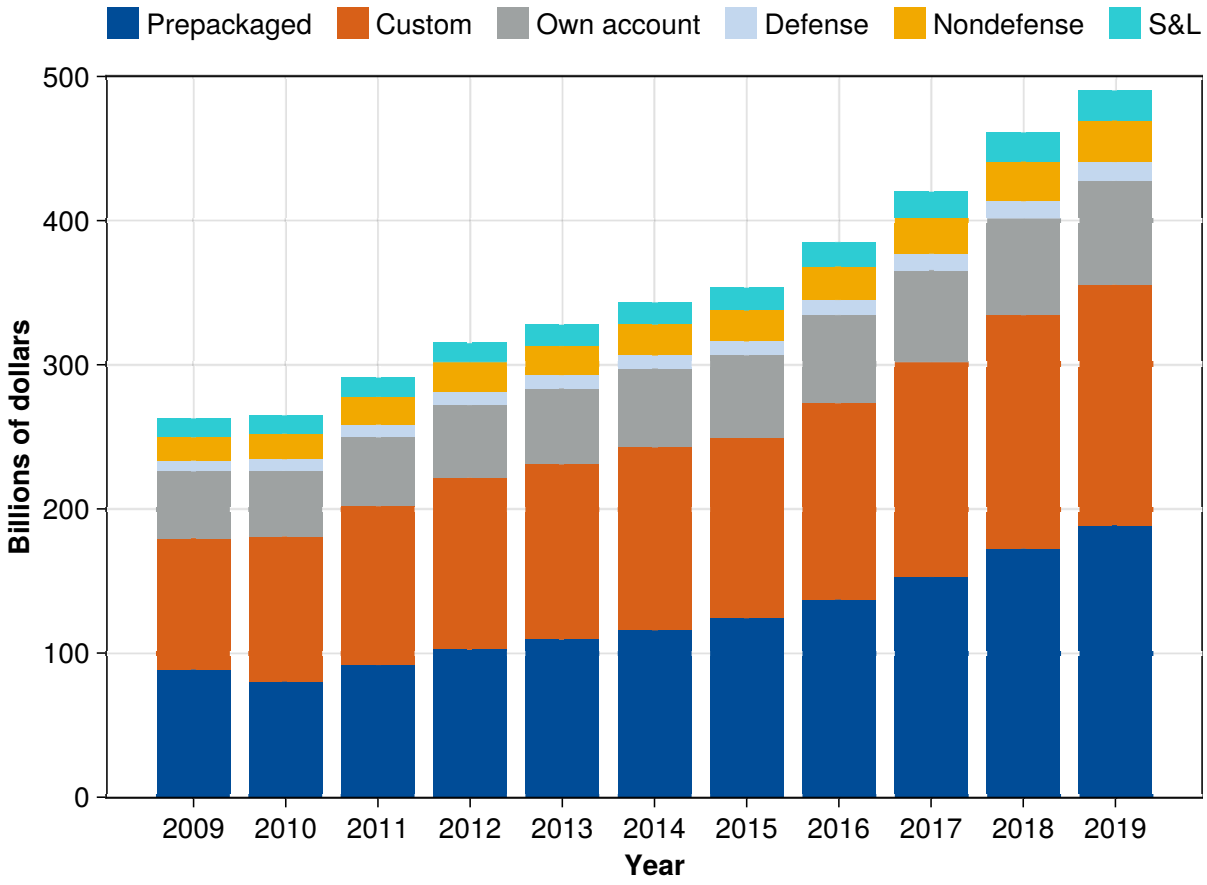
Table 1. Sector and Account for OSS Investment

Sector	Account
Business	Own Account
Federal Govt	Defense/Non-Defense
Households	-
NPISH	Own Account
Private Higher Ed	Own Account
Public Higher Ed	Defense, State & Local
S&L Govt	State & Local

Source: NIPA Handbook: Concepts and Methods of the U.S. National Income and Product Accounts (U.S. Bureau of Economic Analysis 2018).

The estimation of the own account software in the U.S. follows the recommendations of the System of National Accounts by estimating the costs of production (United Nations 2010). A description of the methodology is available through BEA publications (U.S. Bureau of Economic Analysis 2018; Chute, McCulla, and Smith 2018; Parker, Grimm, et al. 2000). For our purposes, the key elements for this study are that we develop estimates comparable to those in the accounts with the main difference that rather than using the estimates of the employment numbers across industries in the private sector, we "directly" observe a measure of the produced asset. Our series uses the same occupations for determining the wage series as well as the mark-up factor to obtain a total production cost from the wage bill. Our estimates do not include the 50% factor applied to the time-use factor in the published accounts given that our time-effort is based on a measure of the produced assets. Lastly, contrary to investment in software in the national accounts, our estimates of open-source software do not exclude software R&D.

Figure 1. U.S. Investment in Software 2009–2019



Source: Software investment in the public sector is reported in BEA National Income and Product Accounts NIPA 3.9.5: Government Consumption Expenditures and Gross Investment lines 23, 31, 39. Software investment in the private sector is reported in the NIPA 5.6.5: Private Fixed Investment in Intellectual Property Products by Type lines 3–5.

The current methodology makes certain assumptions such as which occupations engage in capital formation of own account software. These may have been well suited in traditional software developed by businesses, but it may introduce a downwards bias in estimating OSS development from non-industry sources such as private academic institutions leading to an underestimate of investment in OSS and in software overall. For instance, OSS contributions in the academic sector tend to include fewer programming-related occupations compared to a traditional business model (e.g., researchers based on domain knowledge such as bio-statistics or earth science rather than just computer science). Other issues with the existing methodology have also been identified, suggesting that there may be significant discrepancies with current figures based on the standard productivity analysis (Greenstein and Nagle 2014).

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 projects have enjoyed wide adoption becoming the leading solutions for various web solutions such as content management, JavaScript libraries, and Web Servers.[1] The presence of OSS is ubiquitous whether it is when interacting with the Internet or owning a mobile, most which are powered by Android—a Linux-based OS.[2] Apache is server software developed with federal and state funds at the National Center for Supercomputing Applications in Illinois. Greenstein and Nagle (ibid.) estimated the value of capital stock of Apache software in use in 2013 at between $2 and $12 billion.

To highlight some of the innovators and their motivations concerning OSS we present a few examples of highly successful projects. The first example is Linux—a kernel that powers a family of operating systems. While the original author Linus Torvald's initial motivation was just to create the project for fun (providing a great 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). Today, Linux is used across a number of sectors with a report suggesting that employees from more than 200 companies have contributed to the project over its past couple years of releases (Linux v4.8–v4.13).

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

[2] Android has an OS market share for mobile devices of around 70% as of November 2021 (https://gs.statcounter.com/os-market-share/mobile-tablet/worldwide).

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" (Rossum 1996). It is not uncommon for companies to rely heavily on open source components. Microsoft recently made waves when they hired Python's creator out of retirement for their Developer Division (Rossum 2020). Microsoft's acquisition of GitHub in 2018 is evidence of its stake in OSS. Python is an example of how innovation spurred by OSS and its inventors moves fluidly across different economic sectors.

Table 2. Influential Open Source 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

PostgreSQL, one of the most commonly used relational databases, is an example of an open source project that is a descendant of 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 originated within firms before the company open-sourced the projects while continuing to make large contributions. The goals of this paper are to document how to track open source projects and their development process, and to provide a way for that information to be incorporated in the federal statistics framework to improve current measurement of intangible assets.

3.1. Repositories and Source Code Hosting

Many OSS projects are developed and shared through *source code hosting platforms* or *forges* such as GitHub, GitLab, SourceForge, and Bitbucket. These platforms are used to develop, download, review, and publish projects and computer code. 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 for tracking changes and coordinating work on files among multiple developers. GitHub is by far the largest forge, with over 80 million users worldwide and 118 million public repositories (GitHub 2021).[3] Information embedded in the repositories and websites, including the code, contributors, and development activity, is publicly available through web-scraping, and 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.[4] 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. Various initiatives such as the Software Package Data Exchange (SPDX) provide standardization and tools such as standard license codes to make it easier to communicate specific license terms for software and compliance.

[3] Counts of users and public repositories were obtained by querying the GitHub API in December 2021.

[4] 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. The term "permissive" indicates that a license sets very few restrictions on the use/re-use and sharing such as attribution that is not required under public domain. Other types of licenses are copyleft licenses such as the GNU General Public (GPL) Licenses, which forbid proprietization (i.e., derivatives must remain under the same terms). While copyleft licenses used to be the norm in many early communities such as R, the trend for new development has been to adopt permissive licenses. For example, tidyverse, a popular collection of packages for the R language, has recently made efforts to re-license the components of the ecosystem to the permissive MIT license.[5]

4. Related Work

Our goal is to measure an intangible intellectual property product that is a public good. This work is related to the growing literature on how to value goods and services without explicit market transactions such as the case of many digital goods. Many digital products are used by consumers without a direct payment: similar to network television programming, in which in some cases their costs are supported by advertising. This kind of free content that is bundled with advertising can be understood as a barter transaction, content in exchange for being exposed to the advertising. In the absence of a direct price, this content created in the business sector can be valued based on its production cost (Nakamura and Soloveichik 2015; Nakamura, Samuels, and Soloveichik 2017). Other digital goods of the kind of online platforms tend to engage in mixed approaches such as transaction fees or engaging in data collection that provides high value to the businesses (Li, Makoto, and Kazufumi 2019). In those scenarios, an income approach can underestimate the true value of the economic activity given the different ways the assets may be monetized. For example, OSS can serve as the basis for a number of services around core components such as with Anaconda (Python), RStudio (R), and Julia Computing (Julia). These companies contribute to the open source ecosystem and in return add value to the companies' products and services.

Innovation is typically captured and measured using surveys, patent analysis, case studies, and peer reviews, and 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, Moylan, and Robbins 2009). Because of the link to economic growth, policymakers and researchers are interested in understanding and supporting activities that lead to innovation.

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

Traditional approaches to measuring innovation (using surveys, patent analysis, and peer reviews) leave many types of innovation not captured, 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, and occurs when the product is used, rather than sold in the market (Gault 2018), and is referred to as free innovation (Hippel 2017) or household production (Bockstael and McConnell 1983). Most intangible inputs are considered assets, because they are used repeatedly in production. Not valuing these intangibles misses changes in the economy, and it leads to underestimation of productivity and misallocation of resources.

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 U.S. have treated computer software as investment since 1999, extending this to R&D expenditures and entertainment and literary originals in 2013 (U.S. Bureau of Economic Analysis 2013). Beyond these three categories, Corrado, Hulten, and Sichel (2005) provides 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, Haskel, and Jona-Lasinio (2017) proposes a public investment category, information, scientific, and cultural assets, which 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 (ibid.).

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 2004). 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 (Hippel 2005).

Keller et al. (2018) presents a framework to observe and measure intangible inputs to innovation using non-survey data sources focusing on administrative data and opportunity data (e.g., repositories captured on web pages). OSS innovation is used as a case study, allowing us to describe the challenges and processes to both create and measure these intangibles using a data science framework that outlines processes to discover, acquire, profile, clean, link, explore the fitness-for-use, and statistically analyze the data. 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 scraped from online software repositories, and provide evidence and insights about how these data could be used to estimate value and impact.

Robbins et al. (2018a) and Robbins et al. (2018b) propose and prototype a framework to measure the cost of OSS as intangible capital created outside the business sector (such as in universities, federal government agencies, and households), thereby extending existing measures of publicly funded research output. The authors use data from GitHub repositories (where these projects are developed) to obtain information about the development activity of OSS projects, (e.g., lines of code). They adopt cost models developed in software engineering, and methods used by Bureau of Economic Analysis (BEA) to estimate the resource cost of developing packages for four open source programming languages: R, Python, Julia, and JavaScript. The preliminary estimates show that the resource cost for developing packages for these languages exceeds $3 billion dollars, based on 2017 costs. Applying this approach to OSS available on federal government's code.gov (Castle 2020) (part of an effort to make custom-developed code broadly available across federal government agencies) results in an estimated value of over $1 billion, based on 2017 costs, as a lower bound for the resource cost of this software (Robbins et al. 2018a).

Korkmaz et al. (2018) develops statistical models to identify factors that affect the impact of OSS, measured by number of downloads and citations, with a case study of R packages. The authors generate dependency and contributor networks of R packages using data collected from Depsy.org, and develop Quasi-Poisson models that use the network characteristics, as well as author and package attributes. They find that the more dependencies a package has, the less likely it is to have a high impact. The authors also show that package attributes, including the number of authors, and the centrality of a package in the dependency network measured by out-degree, closeness centrality, and pagerank have significant effects on both downloads and citations.

5. Data and Methods

A cost approach is an appropriate estimation framework for OSS given its nature as an intangible public good. For a cost approach, we need to opt for an estimation model that can attribute a dollar amount representation of the resources that it took to produce. The challenge of keeping large software projects on schedule and within budget motivates a literature in cost estimation within software engineering (T. N. Sharma, Bhardwaj, and A. Sharma 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 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, and models based on function points (Barry W. Boehm and Valerdi 2008). Our approach is a close adaptation of a COCOMO II model.

The intuition of the constructive cost model is that we can approximate the development time of software based on observable factors such as the lines of code and certain parameters that reflect assumptions about the effort and complexity of the project. A calibration factor represents the person-months needed for a set number of lines of code, unadjusted for effort factors. Effort multipliers account for complexity, reliability, and scale for these models; they lead to increased cost. In our use of this model, we multiply lines of code by a COCOMO II calibration factor to estimate person-months per project (Barry W. Boehm, Clark, et al. 2000). The effort multipliers from COCOMO II are parameters that we selected for the organic software class, which consists of software dealing with a well-known programming language and a small, but experienced team of contributors. While we held these consistent across all projects, the model allows for these parameters to be adjusted based on additional data.

$$\text{Effort} = 2.4(\text{KLOC})^{1.05}$$
$$\text{Nominal development time} = 2.5(\text{Effort})^{0.38}$$
$$\text{Development cost} = (\text{Monthly resource cost})\ (\text{Nominal development time})$$

KLOC stands for kilo (thousand) lines of code. We use the number of lines added to each project as the measure of effort. The resulting nominal development time in person months are then multiplied by our estimated person month production cost based on the wage bill and blow-up factor to account for total expenses (i.e., labor costs, intermediate consumption, and capital services). The wage series and blow-up factors are consistent with those of the own account software methodology described in section 2. We used the average wages for the three occupations considered for own-account software in the business sector using the data from the Occupational Employment and Wage Statistics (OEWS) (U.S. Bureau of Labor Statistics 2021).[6] We use a blow-up factor of 2.02 based on the multi-year total expenses to gross payroll ratio based on the Services Annual Survey (SAS) data for the representative industry Computer Systems Design and Related Services (NAICS: 5415). One important aspect to highlight is that in contrast to the own account software top-down approach, our bottom-up approach first identifies the assets rather than assuming the allocation of resources based on employment numbers. An implication of the differences in approaches is that we do not apply a capital investment factor to our estimated nominal development time, since our time allocation is based on "observing" the asset from software capital formation.

[6] Three occupation codes based on the 2019 taxonomy.

5.1. Data

Mining software repositories is an active area of research with the community developing archival projects like the Software Heritage (Di Cosmo and Zacchiroli 2017), databases of repository data like World of Code (Ma et al. 2019), and snapshots of the data accessible from GitHub like GHTorrent (Gousios 2013). However, using the data is not always straightforward without a deep understanding of the nature of version control systems and the administrative data of Git hosting platforms (Kalliamvakou et al. 2016). A further complication is the mixed nature of the data spanning administrative, social, and version control. For example, one key feature in GitHub is that contributions are linked to an account by matching the email associated with a commit and those in an account's verified emails list. In other words, if someone removes an email (e.g., lost access or email is no longer active and is unaware of the effect), all contributions attributed to that email would become orphan and no longer attributed to that user (Dohm 2017).

We follow Keller et al. (2018) overall approach to explore data sources beyond surveys to improve and extend indicators of science and engineering activity and of innovation. This approach includes structured processes to discover, acquire, profile, clean, link, explore the fitness-for-use, and statistically analyze the data. Here we gather and use publicly available metadata about individual OSS projects on GitHub and their contributors and organizations, as well as information within the code.

We first queried the GitHub database for all public repositories with a machine-detectable OSI-approved license that was original (i.e., non-fork, non-mirror) and not deprecated (i.e., non-archived).[7] GitHub uses the Ruby Gem Licensee in order to detect a license based on the text of the LICENSE file in the repository. The lion's share of the repositories are MIT licensed (57%), followed by Apache-2.0 (15%), and a combination of GPL-3.0 (14%) and GPL-2.0 (5%). The reported distribution of licenses was computed based on our set of repositories of interest but mirrors the distribution for all public repositories on GitHub as of 2020.

After establishing our universe of repositories with these OSS licenses, we collected all of the commit activity from January 2009 through December 2019 using the GHOST.jl software package (Santiago Calderón 2020), a wrapper of the GitHub API, including when the code was committed, lines added/deleted for the base branch of each repository, and all users that contributed to those repos. From this information, we were able to construct a table for all users' contributions to each project as well as when and how much they contributed in terms of lines added/deleted.

[7] Forks refer to copies of another repository that can be used to take the project in a different direction or maintain a modified version of it. Forks may also be used temporarily to make contributions and offer them as changes to the parent repository. Mirrors are copies of the development of a repository that is synchronized to a history being recorded elsewhere.

While these data provide a strong baseline for resource cost estimates, our goal is to repeat the process each year to provide a consistent picture of the latest snapshots (e.g., capturing collaborations within commits and recording the multiple authors rather than just the primary), and how the OSS ecosystem evolves over time.

In full, our final dataset includes 7.75 million GitHub repositories associated with 3.2 million distinct contributors. One refinement we apply to the data is to avoid shared commits across repositories. This may occur for multiple reasons such as various projects using a template, applying patches to sections of the code, or "forks" that have been generated through copy/paste and not an explicit fork procedure. In those cases, we identify the common commits and the repository which contains the "longest chain" in order to avoid multiple counting the same contribution. These shared commits are identified by fields such as the Git commit hash, author / author email, and timestamps. Other refinements we considered included a special treatment for automated contributions (authored by bots), but we decided to include such users in our final analyses, since the actions performed by bots would need to be replaced by human labor. Lastly, we have developed a flexible method to handle multi-authored commits that may be more common in the future and is now better supported by the GitHub platform.

5.2. Country and Sector Assignment

After obtaining the annual nominal development time for each repository, we then have to allocate the investment across countries and sectors. For attributing investment to a country we first consider the public information provided by the GitHub users in fields such as biography, location, company, and public email address. We developed and used various utilities for the autocoding assignments such as diverstidy and tidyorgs (Kramer 2021a; Kramer 2021b). These tools make use of other datasets such as concordance files for U.S. federal agencies, educational institutions (e.g., IPEDS), list of cities, and public domain list of legal entities. Once users have been allocated to a country or sector, the fractions weighted by contributions are used to distribute the annual investment estimates per repository. Total country/sector/year estimates are the aggregated sum of the investment across each contributor. Not all contributors or contributions are allocated to a country or sector, even among those that provide some information. Table 3 shows a breakdown of how users were assigned to country and sectors. Figure 2 shows a breakdown of contributions per country for the top 10 countries. Table 4 shows a summary of contributions to open source software on GitHub by users with institutional emails of the federal government. The table's addenda provide context on how the contribution by the federal government compares to the contributions by the largest contributors from other sectors.

Figure 2. Top Countries with Most Contributors

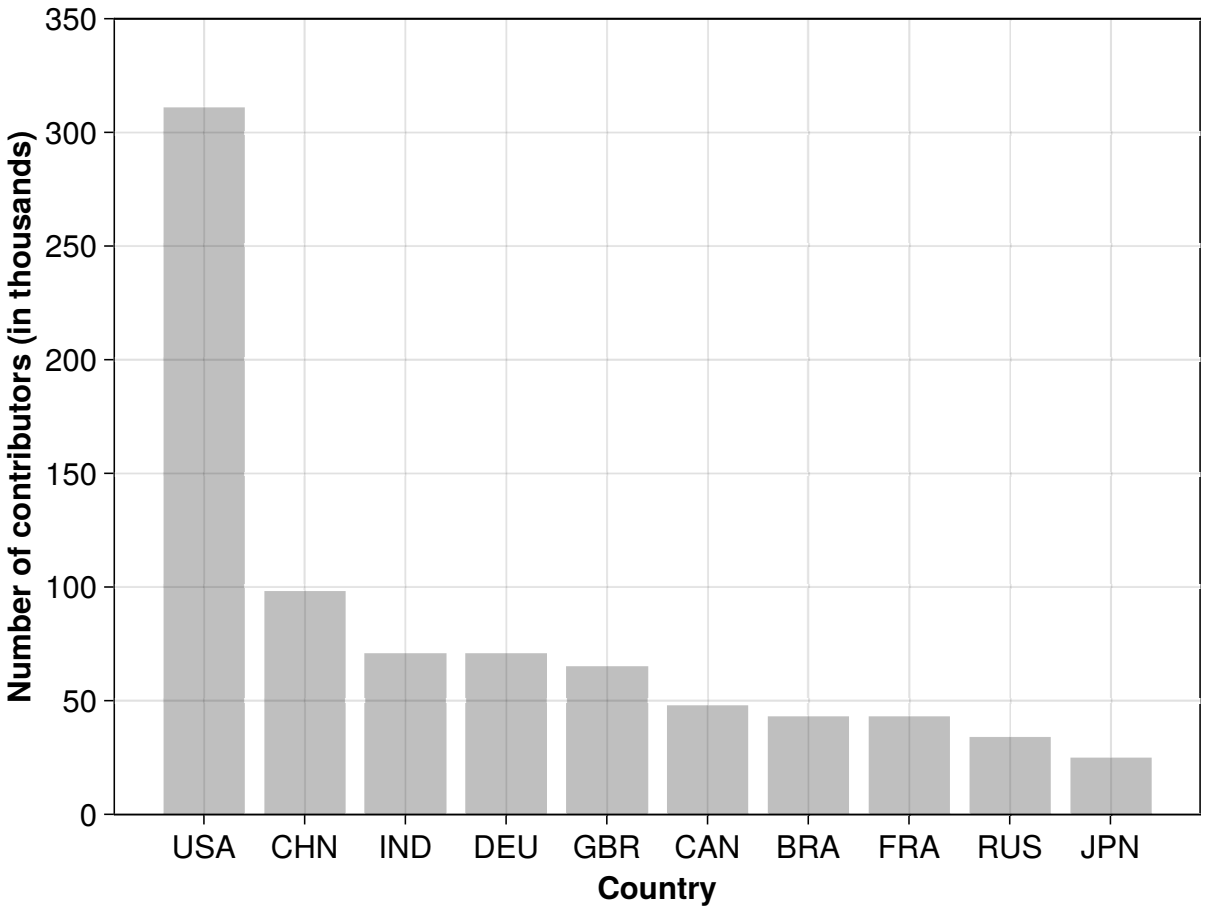


Table 3. Country and Sector Assignment to Contributors

	Total (k)	% All	% Valid
Total Users	3,212.5	100	-
Any Location Information	1,424	44.3	100
Valid Country	1,235	38.5	86.7
Any Sector Information	873.6	27.2	100
In Any Sector	385.5	12	44
Business	261	8.1	29.8
Academic	105	3.2	12
Household	12.2	0.4	1.4
Government	4.1	0.1	0.5
Non-Profit	2	0.1	0.2

6. Results

Table 5 shows our estimates of the U.S. nominal investment in OSS by producing sector for the 2009–2019 period. The annual investment in OSS for 2019 is about 36.2 billion dollars. This is an imprecise estimate. We suspect that the investment estimate for 2019 is imprecise with downward bias due to the timing of the data collection, which was early 2020. We suspect that we are not capturing, for example, development that has occurred but has not yet been published. Table 6 provides the estimated real investment in OSS for the U.S. in constant 2019 dollars using the own account software price index. However, the own account price index is an input-based index that is quite conservative. Figure 3 shows both the nominal and real investment series assuming the own account price index and the one for prepackaged software. 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 prepackage software price index would result in an average log growth increases of 3.2% through 2009–2019. We also report the net-stock current cost estimates (levels and growth rates) for OSS in Figure 5. Using the perpetual inventory method (PIM) and own account depreciation rates and price index we estimate a net-stock of OSS of $72.3 billion. Using either deflator, the growth rate of investment is considerable with an annualized percentage change between 2009 and 2018 of over 50% and over 45% for the net-stock.

Table 4. Repositories Contributed to by Institutional Members of the Federal Government

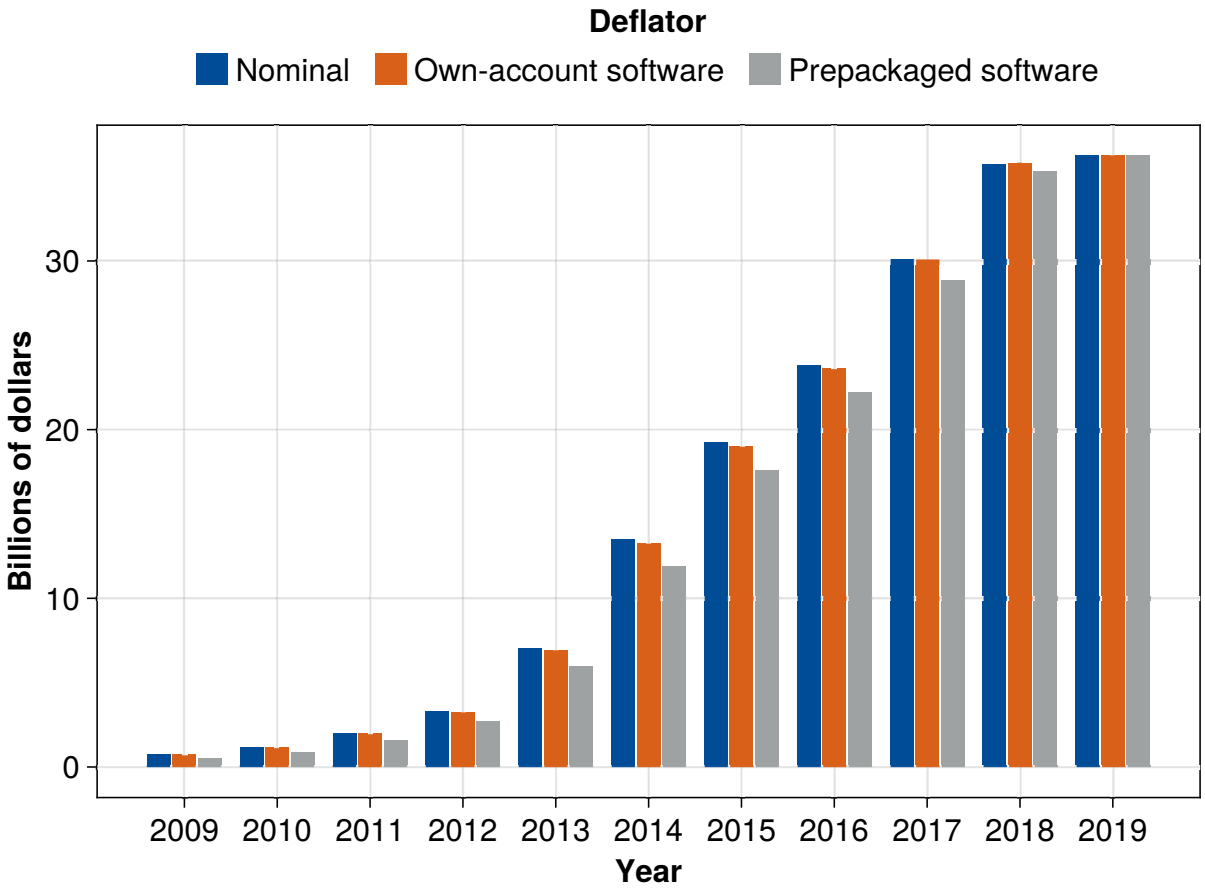
U.S. Institution	Count
Federal total	15,716
Department of Energy	11,156
NASA	1,102
HHS	863
DOC	819
Department of the Interior	537
DOD	321
GSA	319
Smithsonian Institution	107
Department of Agriculture	104
VA	76
All others	312
Addenda	
Microsoft	25,365
RedHat	24,767
UC Berkeley	7,152

Table 5. U.S. Nominal Investment in OSS, by Producing Sector (in Millions USD)

	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
All Activity	737	1,155	1,993	3,303	7,031	13,516	19,262	23,800	30,086	35,743	36,238
Sector ID'ed	283	431	752	1,264	2,382	4,266	5,860	6,902	8,311	9,159	8,719
Private	277	421	736	1,240	2,322	4,155	5,705	6,706	8,053	8,848	8,420
Business	269	407	713	1,203	2,244	4,014	5,504	6,425	7,724	8,472	8,022
NPISH	4	10	12	18	33	60	85	122	129	136	144
Household	4	4	11	19	45	81	116	159	200	240	254
Government	6	10	16	24	60	111	155	196	258	311	299
Addenda											
Academia	53	95	186	285	687	1,531	2,554	3,471	4,886	6,033	6,402

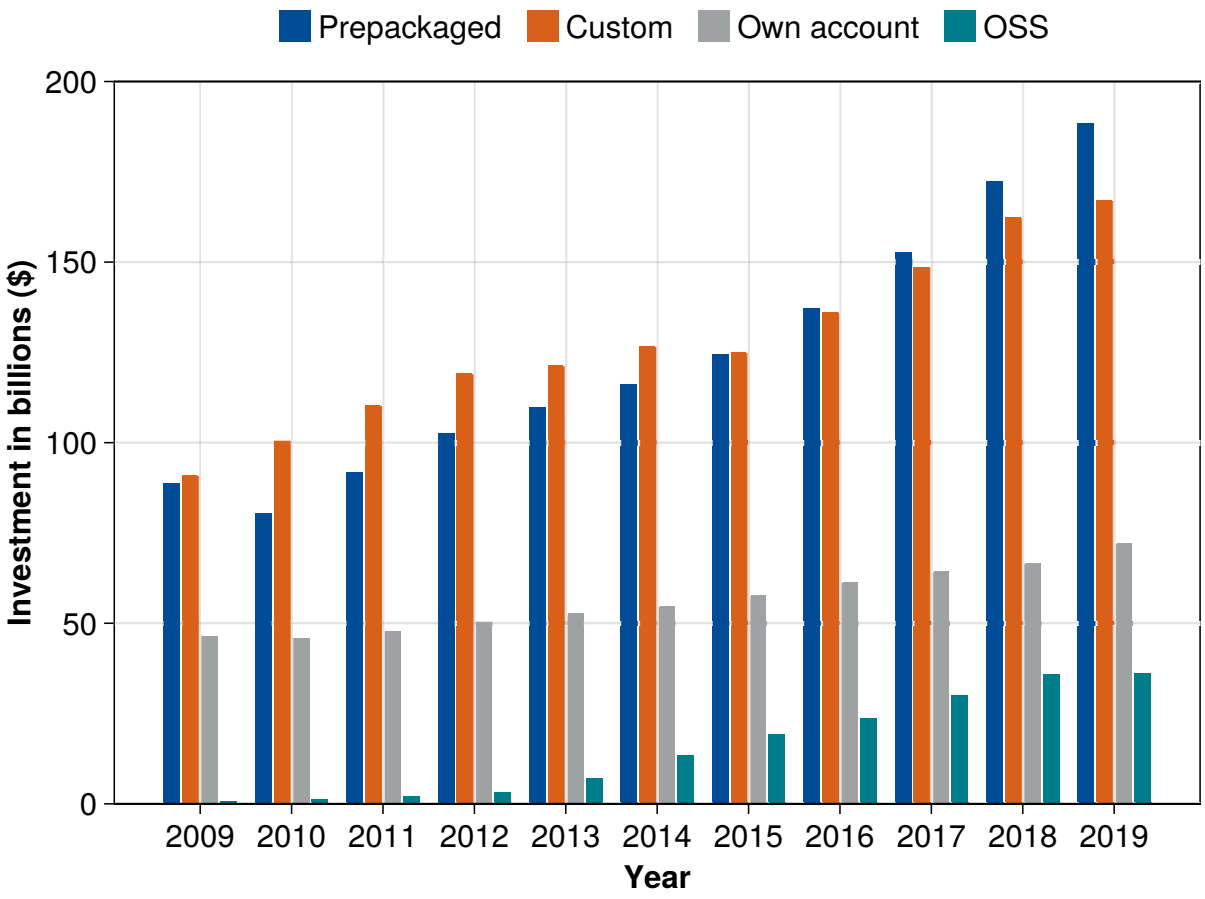
Note: Investment is estimated as person-months x monthly wage rate x 2.02.

Figure 3. Nominal and Real Investment in OSS



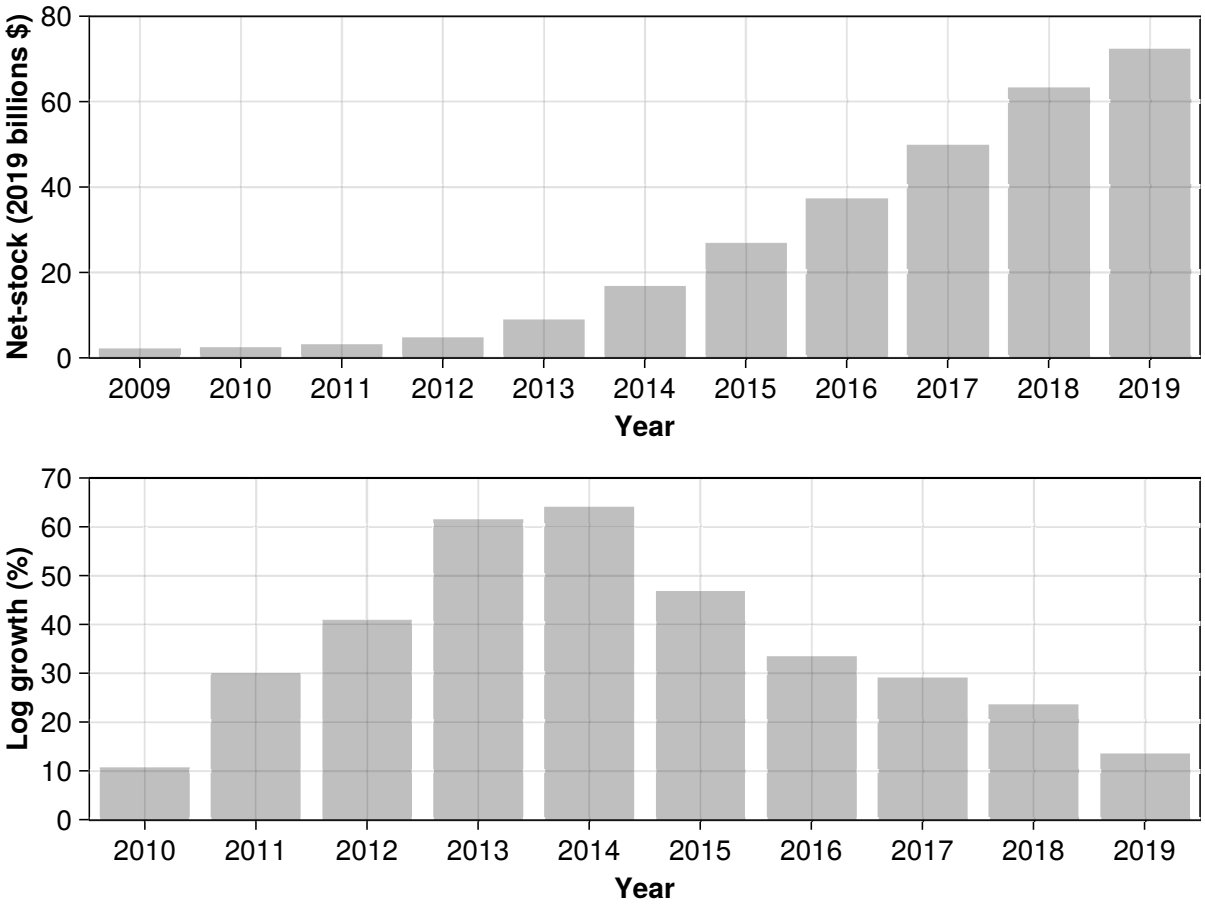
Source: NIPA 5.6.4: Price Indexes for Private Fixed Investment in Intellectual Property Products by Type line 3 prepackaged and line 5 own account.

Figure 4. Comparison in Software Investment Trends



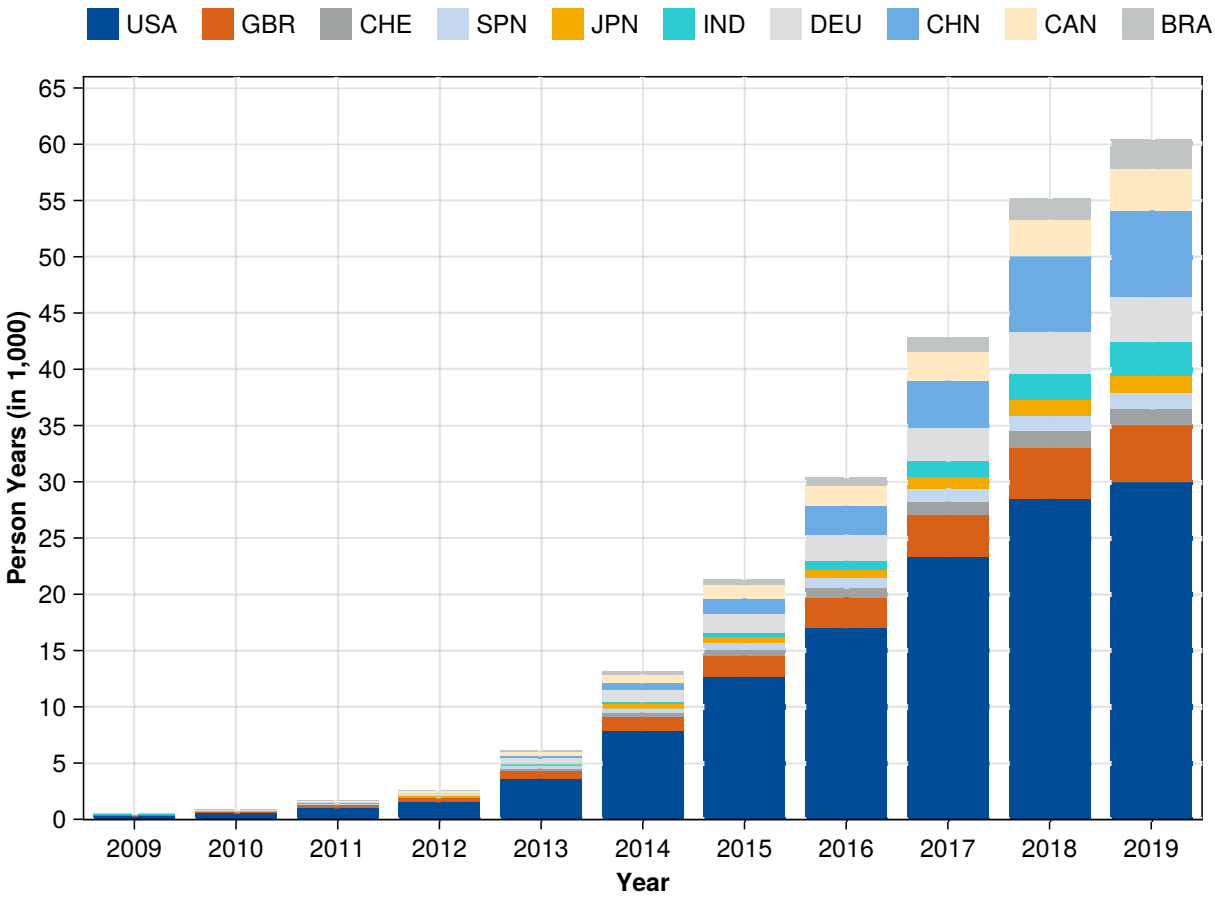
Note: Net-stock current cost estimates were computed using our annual investment estimates and a perpetual inventory method (PIM) assuming own-account depreciation rate (1/3) and price index for current cost basis.

Figure 5. Open Source Software Net-Stock Estimates



Note: Net-stock current cost estimates were computed using our annual investment estimates and a perpetual inventory method (PIM) assuming own-account depreciation rate (1/3) and price index for current cost basis.

Figure 6. Top Countries by Contributions (in Person Months)



Note: Estimates for 2019 are preliminary and are likely to underestimate investment given the timing of when the data were collected (early 2020).

Table 6. U.S. Real Investment in OSS, by Producing Sector (in 2019 Millions USD)

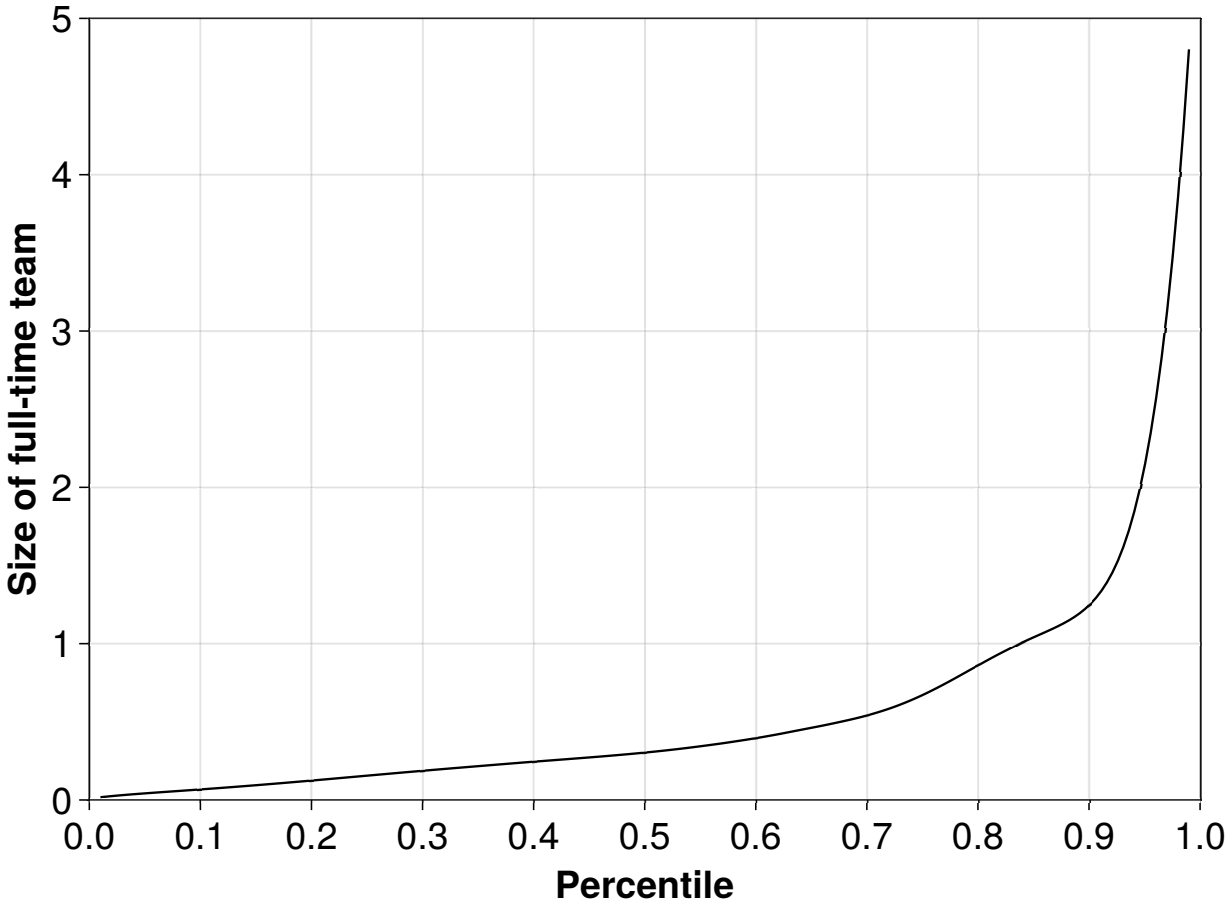
	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
All Activity	719	1,142	1,957	3,244	6,896	13,273	18,972	23,601	30,050	35,805	36,238
Sector ID'ed	270	416	723	1,218	2,277	4,080	5,619	6,650	8,043	8,863	8,420
Private	270	416	723	1,218	2,277	4,080	5,619	6,650	8,043	8,863	8,420
Business	262	403	700	1,181	2,201	3,942	5,421	6,371	7,715	8,487	8,022
NPISH	4	10	12	18	32	59	84	121	129	136	144
Households	4	4	11	19	44	80	114	158	200	240	254
Government	6	10	16	24	59	109	153	194	258	312	299
Addenda											
Higher Ed	52	94	183	280	674	1,503	2,516	3,442	4,880	6,044	6,402

Note: Price deflator is NIPA 5.6.4: Price Indexes for Private Fixed Investment in Intellectual Property Products by Type line 5 own account software.

The monetary value of a person month production cost will depend on the country, sector, and year. Two-thirds of the commit activity was authored by users to which we have not confidently assigned a country (e.g., using Gmail as their email). Currently, the best way to assess the distribution of time/effort across countries and sectors is to do so on the basis of person-months which are comparable regardless of the country or sector of the contributors. Figure 6 shows investment in person months for the top 10 countries.

We can also get a sense of how much investment is being allocated to different projects based on our data and methodology. The distribution shown on Figure 7 indicates that 80% of projects in 2019 had an investment equivalent lower than a full-time contributors and only the top 1% of projects had an effort equivalent above a 5-person full-time team.

Figure 7. Distribution of Annual Person-Month Development in 2019



Note: Data are trimmed at the upper 99th percentile.

7. Discussion

In this section we discuss assumptions and challenges within our approach. The first one 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 various orders of magnitude more projects and contributors than other services such as SourceForge with 500,000 public repositories or Bitbucket with 10 million users according to the documentation on their websites. Furthermore, not all open source 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 (e.g., notes about history of the project, license text of third-party components). However, all these introduced biases have a downwards direction suggesting our estimates to be a lower bound.

An alternative approach would be to rely on intellectual property records such as in the case of patents. However, contrary to proprietary software, which is usually registered with the U.S. Copyright Office, open source projects typically are not. One of the reasons is that there are few incentives for author, contributors, and maintainers to file applications considering U.S. law automatically grants the copyright holder all the protections it would generally need should any dispute end up in court.

Projects may have different conventions to which our approach is not entirely robust for analysis at higher resolution. However, these distortions should be relative negligible at a national accounts level. For example, we examine only the base branch of the repository which is a good heuristic but may undercount contributions for projects that use different branches for production. Likewise, 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 not observable until those are made public, which could happen in the following years, giving rise to a delay effect with downward bias for recent years. One source of bias in the opposite direction relates to the lines added not being part of the source code but rather artifacts such as data files (e.g., JSON, CSV), or auto-generated documentation (e.g., HTML), which might give the impression that more code was implemented than in reality. Moreover, the lines of code do not account for the quality of the code.

Household production is an ever-increasing sector that poses various challenges in order to properly be captured in official statistics. While our approach might be better suited than other common strategies, it would be well complemented with other approaches such as survey data. The National Center for Science and Engineering Statistics (NCSES) has recognized that a dual approach of survey and non-survey methods can provide a better picture and has ongoing efforts to establish better survey data on the role of households as producers including their role in open source software.

Many assumptions we make regarding the development process are very sensible from a national accounts framework, especially when taking into account the common OSS development process. Still, as the methodology matures, some refinements could be implemented. For example, increasing the sector classification of users into different sectors such as the federal/state & local would allow us to more closely adapt 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 matter for general estimates but would be useful for analysis at a higher resolution (e.g., per company or university).

8. Conclusion

In summary, inspired by a concern that OSS is not adequately measured in the current macroeconomic statistics, acknowledging the importance and value in obtaining good measurements, and recognizing a body of research that suggests the current distortions are significant, we developed a framework that can serve as a basis to address the problem. This approach adapts the current national accounts methodology while incorporating a cost estimation literature for the particular issue of estimating the cost of developing software. It overcomes challenges such as cataloguing the universe, taking into account the differences with other intellectual property that are relatively easier to capture through administrative records (e.g., copyrights, patents) by adapting 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: $36.2 billion (≈50% of own account software investment).

The importance of improving how we measure OSS is highlighted by the exponential rise in this asset in terms of development and importance to the economy and society. We find that contribution to OSS is growing rapidly; 2,350% increase in the number of repositories from 2008 to 2019. While our work does not propose or imply any change to the definition of investment in BEA's GDP accounts, from the perspective of identifying sources of innovation and technology diffusion, we see value in quantifying OSS created both in the market and outside of it, as well as the international contribution. The value of our work for GDP measurement is in the use of alternative methodologies and source data that focuses on a subset of an investment category in the national economics accounts. The bottom-up method we use may reveal some currently unaccounted for software investment, and that may be of broader interest because it would affect the level and composition of software investment.

While we believe the work presented here makes a significant contribution to the measurement of OSS and intangible assets more broadly, it does not fully meet our larger goal. Current and future work will keep refining our methodology and identifying areas where it may be improved or concerns to be addressed. For example, we are focusing on areas such as: sectoring contributions, developing tools to gather and analyze OSS data, and developing economic indicators based on the methodology that can reach a production ready standard in the future.

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.

Acknowledgements

This material is based on work supported by the U.S. Department of Agriculture (USDA) (FAIN: 583AEU70074) and National Science Foundation (NSF) (PIID: 49100420C0015). The authors acknowledge Research Computing at the University of Virginia for providing computational resources that contributed to the results reported within this publication. We also acknowledge the Data Science for the Public Good Program participants Cong Cong, Calvin Isch, Eliza Tobin, Daniel Bullock, Morgan Klutzke, and Crystal Zang. Lastly, we want to thank discussants and reviewers who get gave us feedback to improve the work to its current version. We thank Dylan Rassier, Shane Greenstein, and Juan Mateos Garcia.

References

[1] Ana M. Aizcorbe, Carol E Moylan, and Carol A. Robbins. *BEA Briefing: Toward Better Measurement of Innovation and Intangibles*. Survey of Current Business. 2009. URL: https://fraser.stlouisfed.org/title/46/item/10199/toc/359344.

[2] Nancy E. Bockstael and Kenneth E. McConnell. "Welfare Measurement in the Household Production Framework". In: *The American Economic Review* 73.4 (1983), pp. 806–814. ISSN: 00028282. URL: https://www.jstor.org/stable/1816580.

[3] B. W. Boehm. "Software Engineering Economics". In: *IEEE Transactions on Software Engineering* SE-10.1 (Jan. 1984), pp. 4–21. ISSN: 2326-3881. DOI: 10.1109/TSE.1984.5010193.

[4] Barry W. Boehm, Clark, Horowitz, Brown, Reifer, Chulani, Ray Madachy, and Bert Steece. *Software Cost Estimation with COCOMO II (with CD-ROM)*. 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2000. ISBN: 0130266922.

[5] Barry W. Boehm and Ricardo Valerdi. "Achievements and Challenges in Cocomo-Based Software Resource Estimation". In: *IEEE Softw* 25.5 (2008), pp. 74–83. ISSN: 0740-7459. DOI: 10.1109/MS.2008.133.

[6] Joseph Roland Castle. "An Organizational Analysis of Publishing the People's Code". PhD thesis. Virginia Polytechnic Institute and State University, 2020. URL: https://vtechworks.lib.vt.edu/handle/10919/97952.

[7] Jason W. Chute, Stephanie H. McCulla, and Shelly Smith. "Preview of the 2018 Comprehensive Update of the National Income and Product Accounts". In: *Survey of Current Business* 98 (4) (2018). URL: https://apps.bea.gov/scb/2018/04-april/0418-preview-2018-comprehensive-nipa-update.htm.

[8] Jonathan Corbet and Greg Kroah-Hartman. *2017 Linux Kernel Development Report*. Annual Report. The Linux Foundation, 2017.

[9] Carol Corrado, Jonathan Haskel, and Cecilia Jona-Lasinio. "Public Intangibles: The Public Sector and Economic Growth in the SNA". In: *Rev Income Wealth* 63 (2017), S355–S380. ISSN: 00346586. DOI: 10.1111/roiw.12325.

[10] Carol Corrado, Charles Hulten, and Daniel Sichel. "Measuring Capital and Technology: An Expanded Framework". In: *Measuring Capital in the New Economy*. Ed. by Carol Corrado, John Haltiwanger, and Dan Sichel. University of Chicago Press, 2005, pp. 11–46.

[11] Linus Dahlander and Mats G. Magnusson. "Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms". In: *Res Policy* 34.4 (2005), pp. 481–493. ISSN: 00487333. DOI: 10.1016/j.respol.2005.02.003.

[12] Fariborz Damanpour. "Organizational Innovation: A Meta-Analysis Of Effects Of Determinants and Moderators". In: *AMJ* 34.3 (1991), pp. 555–590. ISSN: 0001-4273, 1948-0989. DOI: 10.5465/256406.

[13] Roberto Di Cosmo and Stefano Zacchiroli. "Software Heritage: Why and How to Preserve Software Source Code". In: *iPRES 2017 - 14th International Conference on Digital Preservation*. Kyoto, Japan, Sept. 2017, pp. 1–10. URL: https://hal.archives-ouvertes.fr/hal-01590958.

[14] Lee Dohm. *GitHub Community Forums*. 2017. URL: https://github.community/t/how-to-change-author-name-and-email-of-commits/285/6.

[15] Alfonso Gambardella and Bronwyn H. Hall. "Proprietary Versus Public Domain Licensing of Software and Research Products". In: *Research Policy* 35.6 (2006), pp. 875–892. ISSN: 00487333. DOI: 10.1016/j.respol.2006.04.004.

[16] Fred Gault. "Defining and Measuring Innovation in All Sectors of the Economy". In: *Res Policy* 47.3 (2018), pp. 617–622. ISSN: 00487333. DOI: 10.1016/j.respol.2018.01.007.

[17] GitHub. *The State of the Octoverse*. 2021. URL: https://octoverse.github.com.

[18] Georgios Gousios. "The GHTorent Dataset and Tool Suite". In: *Proceedings of the 10th Working Conference on Mining Software Repositories*. MSR '13. San Francisco, CA, USA: IEEE Press, 2013, pp. 233–236. ISBN: 978-1-4673-2936-1. URL: http://dl.acm.org/citation.cfm?id=2487085.2487132.

[19] Shane Greenstein and Frank Nagle. "Digital Dark Matter and the Economic Contribution of Apache". In: *Res Policy* 43.4 (2014), pp. 623–631. ISSN: 00487333. DOI: 10.1016/j.respol.2014.01.003.

[20] Eric von Hippel. *Free Innovation*. ISBN: 978-0-262-03521-7. Cambridge, MA: The MIT Press, 2017. 228 pp. ISBN: 978-0-262-03521-7.

[21] Eric von Hippel. "Perspectives on Free and Open Source Software". In: ISBN: 978-0-262-06246-6. MIT Press, 2005. Chap. 14, pp. 267–278.

[22] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. "An In-Depth Study of the Promises and Perils of Mining GitHub". In: *Empir Software Eng* 21.5 (2016), pp. 2035–2071. ISSN: 1382-3256, 1573-7616. DOI: 10.1007/s10664-015-9393-5.

[23] Sallie A. Keller, Gizem Korkmaz, Carol A. Robbins, and Stephanie S. Shipp. "Opportunities to Observe and Measure Intangible Inputs to Innovation: Definitions, Operationalization, and Examples". In: *Proc. Natl. Acad. Sci. U.S.A.* 115.50 (2018), pp. 12638–12645. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1800467115.

[24] G. Korkmaz, C. Kelling, C. A. Robbins, and S. A. Keller. "Modeling the Impact of R Packages Using Dependency and Contributor Networks". In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Vol. 00. Aug. 2018, pp. 511–514. DOI: 10.1109/ASONAM.2018.8508255.

[25] Brandon Lee Kramer. *diverstidy: A tidy package for detection and standardization of geographic, population, and diversity-related terminology in unstructured text data*. 2021. URL: https://github.com/brandonleekramer/diverstidy.

[26] Brandon Lee Kramer. *tidyorgs: A tidy package that standardizes text data for organizational analysis*. 2021. URL: https://github.com/brandonleekramer/tidyorgs.

[27] Josh Lerner and Jean Tirole. "Economic Perspectives on Open Source". In: *Intellectual Property and Entrepreneurship*. Emerald Group Publishing Limited, 2004, pp. 33–69.

[28] Wendy C.Y. Li, Nirei Makoto, and Yamana Kazufumi. *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), Mar. 2019. URL: https://ideas.repec.org/p/eti/dpaper/19022.html.

[29] Yuxing Ma, Chris Bogart, Sadika Amreen, Russell Zaretzki, and Audris Mockus. "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. Montreal, Quebec, Canada: IEEE Press, 2019, pp. 143–154. DOI: 10.1109/MSR.2019.00031.

[30] Ben R. Martin. "Twenty Challenges for Innovation Studies". In: *Sci Public Policy* 43.3 (2016), pp. 432–450. ISSN: 0302-3427, 1471-5430. DOI: 10.1093/scipol/scv077.

[31] Leonard I. Nakamura, Jon Samuels, and Rachel H Soloveichik. *Measuring the 'Free' Digital Economy within the GDP and Productivity Accounts*. 2017. URL: https://www.bea.gov/research/papers/2017/measuring-free-digital-economy-within-gdp-and-productivity-accounts.

[32] Leonard I. Nakamura and Rachel H Soloveichik. "Valuing 'Free' Media Across Countries in GDP". In: *SSRN* (2015). ISSN: 1556-5068. DOI: 10.2139/ssrn.2631621.

[33] Netcraft. *Web Server Survey*. https://news.netcraft.com/archives/2017/11/21/november-2017-web-server-survey.html. 2017.

[34] Robert P Parker, Bruce T Grimm, et al. *Recognition of Business and Government Expenditures for Software as Investment: Methodology and Quantitative Impacts, 1959-98*. Tech. rep. Bureau of Economic Analysis, 2000. URL: https://www.bea.gov/research/papers/2000/recognition-business-and-government-expenditures-software-investment.

[35] Carol A. Robbins, Gizem Korkmaz, José Bayoán Santiago Calderón, Claire Kelling, Stephanie S. Shipp, and Sallie A. Keller. "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). 2018, p. III1. URL: https://www.imf.org/en/News/Seminars/Conferences/2018/04/06/6th-statistics-forum.

[36] Carol A. Robbins, Gizem Korkmaz, José Bayoán Santiago Calderón, Claire Kelling, Stephanie S. Shipp, and Sallie A. Keller. "The Scope and Impact of Open Source Software: A Framework for Analysis and Preliminary Cost Estimates". In: *35th International Association for Research on Income and Wealth (IARIW) General Conference*. IARIW. 2018, 2A5. URL: http://www.iariw.org/c2018copenhagen.php.

[37] Guido van Rossum. *Foreword for "Programming Python" (1st ed.)* 1996. URL: https://www.python.org/doc/essays/foreword/.

[38] Guido van Rossum. *Twitter Status*. 2020. URL: https://twitter.com/gvanrossum/status/1326932991566700549.

[39] José Bayoán Santiago Calderón. *GHOST.jl*. 2020. URL: https://github.com/team-oss/GHOST.jl.

[40] T. N. Sharma, Anil Bhardwaj, and Anita Sharma. "A Comparative Study of COCOMO II and Putnam Models of Software Cost Estimation". In: *IJSER* 2.11 (2011). ISSN: 2229-5518. URL: https://www.ijser.org/onlineResearchPaperViewer.aspx?A-Comparative-study-of-COCOMO-II-and-Putnam-models-of-Software-Cost-Estimation.pdf.

[41] Andrew M. St. Laurent. *Understanding Open Source and Free Software Licensing*. 1st ed. Sebastopol, CA, US: O'Reilly Media, Inc., 2004. 193 pp. ISBN: 978-0-596-00581-8.

[42] The PostgreSQL Global Development Group. *PostgreSQL 13.1 Documentation*. 2020. URL: https://www.postgresql.org/docs/13/history.html.

[43] Linus Torvalds and David Diamond. *Just for Fun: the Story of an Accidental Revolutionary*. 1st. New York, NY: HarperBusiness, 2001. 262 pp. ISBN: 978-0-06-662072-5.

[44] U.S. Bureau of Economic Analysis. *NIPA Handbook: Concepts and Methods of the U.S. National Income and Product Accounts*. 2018. URL: https://www.bea.gov/resources/methodologies/nipa-handbook.

[45] U.S. Bureau of Economic Analysis. *Preview of the 2013 Comprehensive Revision of the National Income and Product Accounts: Changes in Definitions and Presentations*. 2013. URL: https://www.bea.gov/information-previous-updates-nipa-accounts.

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

[47] United Nations. *System of National Accounts 2008*. United Nations, 2010. ISBN: 9789210544603. DOI: 10.18356/4fa11624-en.