The Rise of FOSS in India

EMPIRICAL EVIDENCE AND INSIGHTS FROM CROSS-SECTORAL CASE STUDIES

NATIONAL LAW SCHOOL OF INDIA UNIVERSITY

CONTRIBUTORS

Research and Writing:

Dr. Arul George Scaria
Dr. Suryaprakash Mishra
Mr. Shubham Shinde
Ms. Rashi Singhal

Design:

Mr. Gavin Morris
Mr. Joy Rana

Editing:

Ms. Ishani K B

Published by

National Law School of India University, Bengaluru (Centre for Intellectual Property Research and Advocacy [CIPRA]; DPIIT IPR Chair; and Reliance Chair on Corporate Law and Governance)

Funding Disclosure:

This project was funded by the Samagata Foundation. However, the funders were not involved in the framing of research questions, methodology, selection of respondents for the study, or any other research-related matters.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This report would not have been possible without the unwavering support of many people and many institutions. We are deeply grateful to all of them and these short acknowledgements is only an attempt to indicate this. It can in no manner cover all persons or the extent of our gratitude to all of them.

We would like to first thank Samagata Foundation for funding this research. Without their generous support, an empirical study like this would not have been possible. The colleagues at Samagata need to be appreciated not just for facilitating research on non-mainstream topics like this, but also for their complete non-interference in all aspects of the research, including the framing of research questions and the methodological choices we made. They are a model for other funding organisations. We are also equally grateful to the National Law School of India University (NLSIU) and the entire administration team at NLSIU, for facilitating a research project like this and for providing all the necessary support at all stages of research and publication of the report. We are also thankful to all our other colleagues at NLSIU, particularly faculty members and researchers, for their support in this regard.

Mr. Venkatesh Hariharan needs to be specifically mentioned and thanked here as he was the first to persuade us of the need for undertaking an empirical study on FOSS, specifically in the Indian context. To the best of his ability, he helped us in connecting with diverse stakeholders in the area. It needs to be added here that he gave his support and time only out of his passion for FOSS, and not for any material benefit.

We also thank all the respondents who participated in our interviews for taking time out of their extremely busy schedules to speak with us. They gave us invaluable and diverse perspectives about FOSS. The excitement about FOSS, and the desire to see changes in India towards FOSS adoption, were very much evident in almost all those intellectually enriching conversations.

We would also like to specifically thank the following scholars for providing their inputs and feedback on various aspects of our research: Prof. Ajay Shah (Co-founder, XKDR Forum), Prof. Frank Nagle (Assistant Professor, Harvard Business School), Mr. Mirko Böhm (Senior Director, Linux Foundation Europe), Mr. Pramod Varma (CTO, EkStep Foundation), Prof. Rahul De' (Professor, IIM Bangalore) and Mr. Vikram Aggarwal (Engineering Lead, Google). We would also like to thank Karan M.V. (Director, International Relations, GitHub) for the thoughtful conversations on the subject and for connecting us with different stakeholders.

We should also thank Ms. Janani Ganeshan (Consulting Editor – Research, NLSIU) for her excellent support in the publication of this report. Ms. Ishani K B needs to be thanked for carefully reading the draft and making language related corrections. We must also thank Mr Gavin Morris for his design inputs in making this report aesthetically appealing.

This acknowledgements section would be incomplete if we did not specifically mention the two people who played the most important role in this project: Mr. Shubham Shinde and Ms. Rashi Singhal. Shubham, with his technological background and attention to detail, provided solid support for all the conversations we had as part of this project. Rashi, with her expertise in law and interdisciplinary approaches to law, complemented this with all her energy. The result is this report. We thank both of them for their dedication, enthusiasm, and hard work. But we would also like to add here that mistakes, if any (including missing out important names in this acknowledgements section), are our own!

DR. ARUL GEORGE SCARIA

DR. SURYAPRAKASH MISHRA

GLOSSARY

ACS: Access Control Server

API: Application Programming Interface

BFSI: Banking, Financial Services and Insurance

CSP: Cloud Service Provider

CSR: Corporate Social Responsibility

CXO: Chief Experience Officer

DMS: Donor Management System

DPG: Digital Public Goods

DPI: Digital Public Infrastructure

ECI: Election Commission of India

EMR: Electronic Medical Record

EOL: End of Life

EVA: Economic Value Addition

FSF: Free Software Foundation

GDP: Gross Domestic Product

HCX: Health Claims Exchange

ICT: Information and Communications Technology

ICU: Intensive Care Unit

IP: Intellectual Property

IS: Information Systems

KYC: Know Your Customer

LMS: Learning Management System

MeitY: Ministry of Electronics and Information Technology, Government of India

MSME: Micro, Small and Medium Enterprises

NSE: National Stock Exchange

OCR: Optical Character Recognition

OEM: Original Equipment Manufacturer

OS: Operating System

OSI: Open Source Initiative

PoS: Point of Sale

PSU: Public Sector Undertaking

ROI: Return on Investment

SaaS: Software as a Service

SBOM: Software Bill of Materials.

SDK: Software Development Kit

SDLC: Software Development Life Cycle

SLA: Service Level Agreement

SPV: Special Purpose Vehicle

TCO: Total Cost of Ownership

TSP: Technology Service Provider

WCAG: Web Content Accessibility Guidelines

EXECUTIVE SUMMARY

Free and Open Source Software (FOSS) is defined as software that is free to study, modify, run and re-distribute. Its usage has grown exponentially in the past decade, with FOSS comprising 70-90% of software in all modern-day software solutions. It offers a multitude of benefits such as lower costs, improved reliability, security, ability to customise the software, etc. While FOSS poses some challenges as well, such as a lack of technical support, the need to perform maintenance of the solution, and that of in-house technical capacity, they are outweighed by its benefits.

The characteristic of FOSS as a global commons, or a public good, makes quantification of its value in economic terms difficult. Hence, with the help of the case study method, this research aims to highlight the salience of FOSS by evaluating the extent of, and reasons for FOSS adoption by diverse types of organisations in India, the tangible and non-tangible benefits they experience as a result, and the various kinds of challenges they face in the process of FOSS adoption. These case studies were prepared with the help of semi-structured interviews with key personnel in these organisations. For the case studies, we have chosen organisations operating across four sectors (healthcare, education, finance and software and IT services) and four categories (start-ups, non-profits, medium and large organisations), wherever possible, in addition to two public sector organisations.

Our study indicates that for the organisations we studied, there is a clear preference for FOSS and it dominates their technology stack, though there may be other types of software that are employed for specific use cases. Email services and internally used applications are two categories wherein we noticed the use of proprietary software in an otherwise FOSS-oriented organisation.

Among the organisations analysed as case studies for this study, cost savings and a lower Total Cost of Ownership (TCO) are most often cited as the benefits of FOSS. The flexibility offered by FOSS to customise the software as per their requirement was also cited by many organisations. A third of the organisations also saw the need for technical capability to work with FOSS as a blessing in disguise, as it aided in developing their internal technical and engineering capacity.

Our study indicates that organisations are also facing some challenges on account of adopting FOSS, which includes tracking and managing updates and patches, unavailability of personnel skilled in FOSS technologies, and the lack of community support.

The study finds considerable awareness among organisations regarding the impact of licensing on their usage of FOSS. Most of them indicated a preference for using FOSS components with permissive licenses, while avoiding restrictive or viral licenses that require them to release any derivative works.

Lastly, the report also lists some recommendations for policy-makers which are primarily based on the conversations with diverse stakeholders. For example, during our interviews many organisations advocated for policies that favour procurement of FOSS by the government. Some were also of the view that all software developed through taxpayer funds should be released as FOSS. We support such policy reforms, and the report specifically recommends changes to procurement guidelines, making procuring FOSS solutions the norm. The report also recommends imparting Information & Communications Technology (ICT) education in schools using FOSS. The mandatory adoption of FOSS by state governments and their departments is also an important recommendation, and experiences from Kerala illustrate the potential of this approach.

INTRODUCTION

Free and Open Source Software (FOSS) is an umbrella term used to refer to 'free software' and 'open source software'.[1] 'Free software' is a term coined by the Free Software Foundation (FSF) where 'free' refers to freedom of usage rather than absence of cost. It defines a free software as one that gives the user the freedom to share, study and modify it.[2] Open source software is defined by the Open Source Initiative (OSI) as software that provides users the freedom to run, make copies, study, change and improve, and distribute original or modified versions of it.[3] We use the term FOSS throughout this report to refer to open source software.

This report provides insights from our empirical study on FOSS in India. The previous studies in this direction were conducted in 2009[4] and 2015[5] and so we hope that this study will be able to account for the transformation in prevalence and capabilities of FOSS from 2015 onwards. It aims to showcase the salience of FOSS ecosystem in India by highlighting the extent of FOSS usage, as well as the benefits and challenges involved in FOSS usage, as experienced by adopting organisations. We hope the report will serve as evidence for suitable policy measures that seek to effectively leverage FOSS for the benefit of the country.

FOSS from its humble beginnings as a movement among hobbyist programmers based in universities like Berkeley, Stanford and MIT sharing code with one another has grown exponentially to be the dominant type of software in use today.[6] In fact, FOSS comprises 70-90% of software in all modern-day software solutions.[7]

FOSS offers a multitude of benefits to its adopters. The source code being open enables one to read and understand the working of the software.[8] The ability to modify the code enables customisation and incremental development, allowing one to

1. CivicDataLab, 'The State of Free and Open Source Software in India' <https://state-of-foss.in/the-state-of-foss-report.pdf> accessed 8 January 2025.

2. 'What Is Free Software and Why Is It So Important for Society?' (*Free Software Foundation - Working together for free software*) <https://www.fsf.org/about/what-is-free-software> accessed 3 February 2025.

3. 'The Open Source Definition' (*Open Source Initiative*) <https://opensource.org/osd> accessed 3 February 2025.

4. Rahul De', 'Economic Impact of Free and Open Source Software – A Study in India' (Indian Institute of Management Bangalore 2009) <https://www.iimb.ac.in/sites/default/files/inline-files/Rahul-De-Economic-Impact-on-Free-and-Open-Source-Software-A-Study-in-India.pdf> accessed 8 January 2025.

5. Rahul De', Lewin Sivamalai and Ravi A Rao, 'Economic Impact of Free and Open Source Software Usage in Government' (Indian Institute of Management Bangalore 2015) <https://icfoss.in/doc/ICFOSS_economic-impact-free(v3).pdf> accessed 3 March 2025.

6. Josh Lerner and Jean Tirole, 'Some Simple Economics of Open Source' (2002) 50 The Journal of Industrial Economics 197.

7. Jason Perlow, 'A Summary of Census II: Open Source Software Application Libraries the World Depends On' (*The Linux Foundation*, 2022) <https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on> accessed 2 March 2025.

8. Maha Shaikh and Emmanuelle Vaast, 'Folding and Unfolding: Balancing Openness and Transparency in Open Source Communities' (2016) 27 Information Systems Research 813.

build on existing software.[9] FOSS is generally more secure and stable than proprietary software[10] given its distributed model of development and debugging.[11] It also acts as a viable alternative to commercially licensed proprietary software.

FOSS adoption or the presence of FOSS ecosystem also has a positive effect on a country's economy. A study analysing macroeconomic parameters of 32 large economies (primarily European Union [EU] countries) alongside their contributions to GitHub found that in the absence of all OSS contributions, their cumulative average Gross Domestic Product (GDP) would be lowered by 2.2%.[12] Another study estimated that a 10% increase in OSS contributions annually would generate an additional 0.4% to 0.6% GDP in the EU.[13] Both studies also found the GDP to be positively correlated with global code contributions to FOSS, rather than with domestic contributions i.e., those from within the country's borders. This exemplifies the characteristic of FOSS as a public good and global commons.

Despite its criticality and significance to the economy, FOSS being a public good makes it challenging to develop an estimate of its true value in economic terms.

One study estimated the value of contributions and investments into FOSS to be in the range of €60 to €95 billion in the EU for 2018.[14] Another estimated the demand and supply-side value of FOSS to be $4.15 billion and $8.8 trillion respectively.[15]

While quantification of the economic value addition of FOSS in the Indian context was one of our intended goals, the same was difficult to deduce and isolate for a variety of reasons. This included the multitude of costs involved in procuring and running a software, non-economic motivations for FOSS adoption such as the time value of development velocity and the lack of closed-source alternatives for certain categories of software like data analytics, containers etc.

---

9. Bruce Kogut and Anca Metiu, 'Open-Source Software Development and Distributed Innovation' (2001) 17 Oxford Review of Economic Policy 248.

10. Henry Chesbrough, 'Measuring the Economic Value of Open Source: A Survey and a Preliminary Analysis' (*The Linux Foundation*, 2023) <https://www.linuxfoundation.org/research/measuring-economic-value-of-os> accessed 17 February 2025.

11. Kogut and Metiu (n 9).

12. Knut Blind and Torben Schubert, 'Estimating the GDP Effect of Open Source Software and Its Complementarities with R&D and Patents: Evidence and Policy Implications' (2024) 49 The Journal of Technology Transfer 466.

13. Directorate-General for Communications Networks, Content and Technology (European Commission) and others, *The Impact of Open Source Software and Hardware on Technological Independence, Competitiveness and Innovation in the EU Economy: Final Study Report* (Publications Office of the European Union 2021) <https://data.europa.eu/doi/10.2759/430161> accessed 9 August 2024.

14. Ibid.

15. Hoffmann M, Nagle F and Zhou Y, 'The Value of Open Source Software' [2024] SSRN Electronic Journal <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4693148> accessed 3 March 2025.

METHODOLOGY

The study uses a mixed methods framework, wherein insights from doctrinal research are complemented with insights from diverse case studies.

We began with a comprehensive literature review, covering research papers and survey reports on FOSS across a broad timescale. This helped in understanding the historical evolution of FOSS, alongside its benefits and challenges. Based on insights from the literature review, the core research questions were identified. These are:

1. How and to what extent are organisations using FOSS?
2. What are the benefits (tangible and non-tangible) they experience by virtue of adopting FOSS?
3. What are the challenges of working with FOSS?
4. What are the factors behind the organisation's choice of software?
5. What potential legal and policy measures can support and promote FOSS in India?

As indicated earlier, within our mixed methods research framework, we adopted the case study approach to comprehensively address these research questions. To build methodologically rigorous case studies, we prepared a detailed, semi-structured questionnaire.

While it would have been preferable to build case studies from all sectors, we had to limit our case studies to four sectors (healthcare, education, finance and software and IT services) due to time and resource constraints. However, efforts were made to ensure greater diversity by trying to have four categories in each of the sectors — start-ups, non-profits, and both medium and large organisations (as defined by Ministry of MSME)— wherever possible. Additionally, we studied two government organisations.

The organisations for the case studies were identified through purposive sampling, keeping in mind the sectors as well as types of organisations required. Semi-structured interviews were conducted through video conferencing with the relevant individuals in the organisations. The interviews were mostly with founders, Chief Experience Officers (CXOs) or technical architects of organisations, to better capture their usage of, and views on FOSS.

In total, 16 organisations were studied, and they are listed in Table 1. We have redacted the name of one of the organisations at their request.

TABLE 1. CASE STUDIES

CATEGORY SECTOR	STARTUP	NON-PROFIT	MEDIUM	LARGE	UNDECIDED CATEGORY
Finance	PocketATM			• [One of the largest private sector banks in India] • Zerodha • Razorpay	
Software and IT services	Dhiway	Tech4Good Community	Remiges	Thoughtworks	NxtGen Datacenter & Cloud Technologies
Healthcare		• Swasth Alliance • KG Hospital			Open Healthcare Network
Education	• PupilFirst • Kalvium				
Government	• NPCI • KITE				

LIMITATIONS

Like many other studies, this study also faces some limitations. First, and the most obvious one, is the sample size. We acknowledge that there are only 16 case studies in this study, and this is primarily due to the time and resource constraints we faced. The adoption of purposive sampling, and the lack of random sampling may have also affected the diversity of organisations chosen for case studies. For this study, we have only approached organisations known to have adopted FOSS and hence it is possible that some of the responses might have an inherent bias towards FOSS adoption. We must also re-emphasise that the interviews were conducted with selected representatives from the organisations (and not with all employees), whose expertise or knowledge could have been limited by their specific roles in the organisations. Finally, most of the surveyed organisations could not provide statistical data on cost comparisons of FOSS, and this made quantification (in percentage terms) of the extent of FOSS usage in their technology stack extremely difficult, if not impossible. Future studies may address these challenges through more innovative methodological approaches.

SECTOR-WISE CASE STUDIES

As indicated earlier, the primary contribution of this report is the case studies from diverse sectors, and different types of organisations, that are using FOSS. In each of these case studies we provide an overview of the extent of adoption of FOSS in the organisation, the factors that prompted the adoption, the benefits received, and the challenges faced in FOSS adoption. The systematic analysis of these case studies, along with extended literature reviews and interviews with other stakeholders, contribute to the framing of recommendations to fuel the adoption of FOSS in India.

Government Sector

KITE

Kerala Infrastructure and Technology for Education (KITE) is a public sector organisation set up by the Government of Kerala. It began its journey as the IT@School project in 2001, to impart ICT-enabled education at the state's public schools. In 2017, KITE was established as a Section 8 Company and a Special Purpose Vehicle (SPV), to serve as the nodal e-governance agency for the state's Department for General Education.

KITE is primarily tasked with initiating ICT-enabled activities in over 16,000 schools in Kerala. This includes upgrading infrastructural facilities, capacity-building of teachers, e-governance and content development. They have developed different kinds of software and mobile applications for the Department. VICTERS, an educational satellite TV channel, is also operated by the organisation.

Public school students in Kerala are provided laptop computers loaded with a customised, open source OS, developed by KITE and based on Edubuntu, called KITE-GNU-Linux that is bundled with a suite of more than 50 FOSS applications.

One of their flagship programs in the state is the Little KITES IT Clubs, as part of which every year over 1.8 lakh students are provided specialised training in five different areas, such as animation, cyber safety, hardware, electronics, Malayalam computing, Artificial Intelligence (AI), robotics etc.[16] KITE also procures and supplies necessary computing hardware and robotic kits (Arduino boards) for this program.

KITE is allocated a budget of ₹40 crore. It has about 100 employees, in addition to 220 master trainers who impart training to the teachers to conduct this program.

FOSS at KITE

Kerala has mandated ICT education for all school students since 2003. Initially, both proprietary software and FOSS were used to impart education largely related to IT skills, which included knowledge on the use of tools such as Microsoft Excel and PowerPoint.

16. 'KITE' <https://kite.kerala.gov.in/KITE/index.php/welcome/about_us> accessed 3 February 2025.

Students were provided the option to undertake their Secondary School Leaving Certificate (SSLC) exam for Information Technology on both Windows and Linux systems, and most opted for the former. A shift to exclusively using FOSS for education was undertaken in 2007, in compliance with the new ICT policy of the state. The same year, 5 lakh students took this exam using FOSS systems without any glitches. This boosted the organisation's confidence in using FOSS, and KITE has never looked back since.

'The scale of what we are doing with ICT education would not have been possible without FOSS.'
Anvar Sadath, CEO, KITE

A shift to IT-enabled education and cross-subject integration in 2009 saw software being leveraged to teach various subjects, using specialised open source applications. The examples in this regard include Geogebra for mathematics and Phet for physics. Today, AI is also included in textbooks and taught using Scratch, a programming language developed by the Massachusetts Institute of Technology (MIT).

KITE is also collaborating and sharing its expertise by expanding their programs to other geographies. Discussions are ongoing with Finland's Ministry of Education and Culture to adapt the Little KITES program for their schools. Additionally, 10 states in India have expressed interest in the program and are being assisted by KITE.

KITE's in-house team works to release required security updates for their FOSS applications, while also supporting other organisations.

Benefits and Challenges

The following are the key benefits of FOSS adoption at KITE:

1. Allows unrestricted usage with the ability to customise and localise the software as per their needs.
2. Provides substantial cost savings. The Kerala government estimates they have saved around ₹3,000 crore annually by migrating to FOSS.[17]
3. Allowed them to operate completely independent of vendors, which has provided the organisation the confidence to implement any software solution.
4. Enhanced security has obviated the need to use antivirus software and firewalls.
5. By providing students with access to innovative learning tools and resources, the program has fostered a more engaging and effective learning environment, which may potentially lead to improved academic outcomes.

'More than financial savings, it is the academic freedom that FOSS provides which is more valuable.'
Anvar Sadath, CEO, KITE

The larger benefit of adopting FOSS has been that the teachers, who comprise 40% of state government employees, are empowered with improved digital skills. Overall, it has helped in making Kerala a better knowledge society.

17. Email communication with Mr. Anvar Sadath, dated 3 July 2025.

However, there were also challenges in the adoption of FOSS at KITE. The following are two that were identified:

1. FOSS is not very user-friendly, and they have encountered challenges when connecting to and using external devices like printers.
2. With the advent of AI models, it is challenging to identify which ones are truly 'open source'.

Despite such challenges, we observe that KITE, alongside the school teachers, has been successfully and consistently using FOSS.

National Payments Corporation of India

The National Payments Corporation of India (NPCI) was established in 2008 as a Section 8 company, by the Reserve Bank of India (RBI) and the Indian Banks Association (IBA), to create a robust infrastructure for payments and settlements in India. Initially, 10 core banks served as promoters, with the shareholding being expanded to 56 member banks in 2016. Payment service operators, payment banks and small finance banks were also inducted in 2020.

NPCI has been at the forefront of innovation in building India's digital financial infrastructure, with UPI (Unified Payment Interface) being their most widely used service. It is a real-time, mobile-based, interoperable, peer-to-peer digital payments system that allows users to transact using any bank account and identify payees by scanning QR codes. Allowing non-banking entities to develop mobile applications while the NPCI acts as the network and settlement service provider has greatly aided the adoption of UPI.

In the month of November 2024, the service processed an average of 516 million transactions daily, with an average daily transaction value of ₹71,839 crore. This service is now available in 7 other countries, apart from India. The NPCI today processes 150 billion transactions annually, close behind Mastercard's 170 billion and Visa's 212 billion.[18]

UPI is just one among the many innovative products developed by the organisation. Others include:

1. **RuPay:** An electronic retail payment system that supports the issuance of debit, credit and prepaid cards, which support contactless and offline payments, by banks. It can even be issued as a National Common Mobility Card.
2. **IMPS (Immediate Payment Service):** An electronic, inter-bank, real-time fund transfer service that is accessible on multiple channels like mobile applications, the internet, bank branches, ATMs and SMS.
3. **AePS (Aadhaar Enabled Payment System):** A payment service that allows users to transact electronically from any Aadhaar-linked bank account through a PoS device or micro-ATMs carried by a bank's business correspondents.
4. **National Electronic Toll Collection:** An interoperable, electronic toll payment solution

18. Anand Adhikari, 'National Payments Corporation of India's Growing Clout: How NPCI Is Setting Itself Up to Conquer the Globe' [2024] *Business Today* <https://www.businesstoday.in/magazine/deep-dive/story/national-payments-corporation-of-indias-growing-clout-how-npci-is-setting-itself-up-to-conquer-the-globe-418938-2024-02-26> accessed 3 February 2025.

that employs an RFID-based device called a 'FastTag' to make payments from a linked bank account.

5. **Bharat Bill Payment System:** An integrated bill payment service for billers, payment service providers and consumers that supports multiple payment modes.

The organisation has about 2000 employees and earned a revenue of ₹1,986 crore in 2023 financial year. After accounting for expenses, the organisation generated a surplus of ₹809 crore, having created cumulative assets of ₹5,571 crore.[19]

FOSS at NPCI

'Open source in India has gained prevalence only over the last eight or ten years.'
Dilip Asbe, MD and CEO, NPCI

NPCI has shifted to using FOSS extensively for its products and business applications since 2015, driven by the following factors:

1. The back-end systems of Aadhaar being built using open source gave the organisation confidence and served as proof of concept for the development and operation of a largescale solution built using FOSS.
2. The unviability of using licensed software at such a massive scale given the price sensitive nature of the Indian market.
3. The desire to eliminate dependency on vendors and shift away from a black box model, where the organisation is unaware of the internal working of its systems.

NPCI is highly keen on open sourcing technical solutions they have developed, but is constrained by limited human resources and largely focuses on solving business problems. While the organisation has open-sourced a solution to deploy and manage a Hyperledger fabric-based blockchain network on Kubernetes, named Falcon, its own contributions to it have been limited.

'In fact, NPCI suggests setting up an experience sharing platform that is open to all, where beneficiaries of open source share their experience for collective benefit.'
Dilip Asbe, MD and CEO, NPCI

FOSS is extensively deployed at NPCI to solve business and scalability problems while ancillary systems necessary for operation such as email, ERP, payroll etc. are consumed as SaaS offerings. In addition, NPCI deploys 20-25 information security tools, some of which are proprietary, to ensure foolproof and multi-layered data security.

Benefits and Challenges

Our conversations suggest that NPCI has experienced the following benefits by means of adopting FOSS:

'We are an engineering shop rather than a procurement organisation.'
Dilip Asbe, MD and CEO, NPCI

19. National Payments Corporation of India, 'Annual Report 2022-23' (*NPCI*, 2023) <https://www.npci.org.in/PDF/npci/corporate-governance/financials/NPCI-financials-2022-23.pdf>. accessed 3 March 2025.

1. **Technical capacity:** FOSS enabled the organisation to build a technology ecosystem in-house. They process 10 billion API requests a day with a latency of less than 100 milliseconds, which is a testament to their technical capability.
2. **Flexibility:** FOSS makes it possible to adopt a 'plug and play' model, wherein individual components of a product or solution (like DBMS for instance) can be replaced with minimal effort and impact on the overall system.
3. **Scalability:** FOSS is very well suited and tested for applications requiring horizontal scaling, which matches the organisation's use case.

Some challenges faced by NPCI with FOSS include:

1. Troubleshooting technical issues.
2. Need for maintaining backups.
3. Mitigating risks and vulnerabilities associated with FOSS.
4. Evaluating multiple options as part of the software selection and adoption process.

The latter is particularly necessary to mitigate the risk posed by mergers and acquisitions of FOSS organisations. Such developments could potentially lead to changes to the licensing model of their products, particularly shifting to a subscription-based model.

For instance, the merger of Cloudera and Hortonworks, two companies that provide enterprise support for Hadoop, an open source Big Data framework, resulted in increased licensing costs imposed by the combined entity.[20]

Finance Sector Case Studies

Zerodha

Zerodha Broking Ltd was founded in 2010 as a fully online brokerage platform, with the objective of removing barriers to broking for traders. In fact, the name Zerodha is a combination of the English word 'zero' and the Sanskrit word '*rodha*', meaning barrier.

Zerodha was the first discount broking organisation in India and lowered the cost of trading in two ways.[21] Firstly, the cost savings resulting from a minimal physical presence (branches and associated staffing) were passed on to the trader. Secondly, they charged a low brokerage fee, commensurate with the actual costs incurred by the organisation to execute trade orders.

The organisation started out with a proprietary trading platform provided by NSE (National Stock Exchange) and later moved to their own platform that was developed and maintained by the same vendor. For the first three years, they focused solely on operations, and gradually built a tech vertical where all applications and associated functionality were developed using FOSS.

20. Joni Hoppen and Sigmundo Preissler Jr., 'Migration of Cloudera Hadoop Data Lake to Open Source Hadoop: Challenges and Benefits' (*Aquarela*, 20 September 2023) <https://aquare.la/en/migration-of-cloudera-hadoop-data-lake-to-open-source-hadoop-challenges-and-benefits/> accessed 3 February 2025.

21. Nithin Kamath, 'Our Story till Now...' (*Z-Connect by Zerodha*, 28 September 2012) <https://zerodha.com/z-connect/general/our-story-till-now> accessed 3 February 2025.

The founders made a conscious choice in opting for bootstrapping and relying on their own revenue to finance operations. This helped overcome the pitfalls of VC funding, which prioritises growth above all. Spending on advertising, marketing and hiring were also strategically limited. Despite this, the organisation attracted 30,000 clients in its first three years.[22]

Zerodha has effectively leveraged technology, using Aadhaar-based consumer verification since 2016 to seamlessly onboard customers. They have also maintained the size of their tech team at just 30 personnel despite a five-fold growth in consumers from 2 to 10 million.[23] As of today, the organisation has approximately 1200 employees and handles about 15% of the retail trade volume in India.[24]

FOSS at Zerodha

The technology vertical of Zerodha employs a structured decision-making framework when deciding on the adoption of a software component or the development of a solution, which reflects a strong preference for FOSS. Off-the-shelf FOSS components are given the highest preference, followed by those in which minimal customisation would be needed. In case neither is available, a solution is developed in-house.

Proprietary solutions are procured in scenarios where the product is cumbersome to build in-house and does not perform a function critical to the core business. Employees are encouraged to contribute to FOSS projects on GitHub using the 'Zerodha tech' tag. Additionally, the organisation provides financial grants to open source projects (even those not directly related to or used for the organisation's operations) as well as support in the form of hardware resources. Recently, the organisation committed to provide $1 million annually as financial assistance to support FOSS development.[25]

Additionally, around 60% of the code developed within Zerodha is reused internally and their GitHub page lists 27 FOSS projects.

Despite being an avid adopter of FOSS across all software categories, the organisation has identified use cases where FOSS alternatives are either unavailable or inadequate, and hence proprietary or in-house developed solutions are being used.

1. Secure email service: The organisation uses Google Workspace despite its high cost since an equivalent FOSS alternative has not emerged, possibly, in the organisation's view, due to the complex nature of the application.

2. Lending component in Enterprise Resource Planning (ERP): The unavailability of this component in ERPNext, their ERP system, required the organisation to sponsor the development of this feature.

22. ibid.

23. V Keshavdev, 'How Zerodha Went from Zero to Hero' [2023] *Fortune India* (2023) <https://www.fortuneindia.com/long-reads/how-zerodha-went-from-zero-tohero/111627> accessed 3 February 2025.

24. Zerodha, 'Our Company, History, and the People behind It' (*Zerodha*) <https://zerodha.com/about/> accessed 3 February 2025.

25. Kailash Nadh, 'Announcing FLOSS/Fund: $1M per Year for Free and Open Source Projects' (*FLOSS/fund*, 15 October 2024) <https://floss.fund/blog/announcing-floss-fund/> accessed 3 February 2025.

Additionally, about half of their Customer Relationship Management (CRM) system is developed in-house.

Benefits and Challenges

Zerodha, as a case study, highlights the following benefits from adopting FOSS solutions:

1. Cost effectiveness.
2. Improved risk management by means of enhanced control and visibility.
3. Freedom to use software in the manner desired and the flexibility to make customisations.
4. Aiding of learning and development of internal technical capacity.
5. Freeing up of human capacity by reducing the burden of vendor management.
6. Freedom from vendor lock-in.
7. Faster rollout of changes, which aids in quicker compliance with new regulatory requirements.
8. Facilitation of scaling as proprietary vendor-based products place limits on usage through their licensing terms.

While FOSS adoption has yielded benefits, Zerodha's experiences also highlight some challenges.

1. **Evaluation errors:** Since the assessment of tools and solutions is not foolproof, some FOSS implementations have resulted in costly changes, and at times forced the organisation to abandon projects midway or drop existing features.

   However, it was also pointed out that this is a standard process in the adoption and usage lifecycle of technology. Technologies (both proprietary and FOSS) can become unviable for various reasons (technical or otherwise) and organisations drop and shift to alternatives. FOSS has in fact enabled Zerodha to switch away in scenarios of unviability easily, saving costs and avoiding any pitfalls of vendor locking.

2. **Support discontinuation:** In the event where support for a project is discontinued, it causes problems with downstream applications.

   Again, it was highlighted that support can disappear with any service provider or technology, FOSS or proprietary. With FOSS, because the organisation has full liberty and access to the entire code, the necessity of any external support is very low as they can build the necessary technical understanding internally.

Meanwhile, Zerodha mitigates the competitive risks arising from free riding of their internally developed open source tools such as ListMonk, a mailing list manager, by releasing them under the AGPL license. This prohibits their packaging and resale as commercial offerings.

Economic Value Addition

The organisation uses more than 50 FOSS libraries and tools for various functions.[26] This has certainly led to massive cost savings. Below are examples of some products, compared with the costs of proprietary alternatives, to indicate the extent to which savings might have happened within an organisation like Zerodha due to FOSS.

26. 'Zerodha's FOSS Stack - Zerodha Tech Blog' <https://zerodha.tech/stack/> accessed 3 February 2025.

1. **Email marketing:** Zerodha has internally developed ListMonk, an open source, self-hosted newsletter and mailing list manager to manage email communications with customers. The organisation estimates that using an equivalent SaaS offering like MailChimp would cost them approximately $10,000 per month in subscription fees, whereas ListMonk costs just $210 to operate.
2. **Laptop operating systems:** All the laptops issued by the organisation use the Ubuntu operating system, resulting in cumulative savings of $170,000 they would pay in licensing costs if they were to use Microsoft Windows Professional instead.
3. **Support ticketing:** Similarly, using an open source support ticketing system, OS Ticket, saves the organisation approximately $10,000 per month, based on their usage and pricing of an equivalent proprietary offering like FreshDesk.

These examples clearly indicate a considerable economic advantage gained by an organisation like Zerodha due to the adoption of FOSS, though we must add that accurate economic quantification of benefits is difficult even for an organisation like Zerodha, which keeps track of the benefits of FOSS to some extent.

Razorpay

Razorpay Software Pvt. Ltd was founded in 2014 to 'make movement of money simpler and easier'.[27] The organisation's online payment solution or payment gateway supports a variety of digital payment options including net banking, debit and credit cards, and multiple digital wallets. This provides businesses, merchants and e-commerce organisations, among others, a quick, secure and affordable solution to send, receive and manage money. It also offers APIs, enabling seamless integration with customer-facing applications.[28]

The organisation is an RBI-authorised online payment aggregator and its services are availed by 76 of India's 100 start-up unicorns.[29] In 2024, it processed payments of $150 billion and held 3% of the global market share for digital payments.[30] It has approximately 3,000 employees and an annual turnover of ₹2,068 crore in F.Y. 2023-24.

Other services provided by Razorpay include a neo-banking platform, PoS devices and e-KYC for businesses that wish to incorporate payment gateways. The validation process for the latter being completely automated has made the process highly scalable, fast and efficient compared to manual verification.

---

27. 'Razorpay's Simple Fundamental: Here's What Makes the Fintech Major a Force to Reckon With' [2024] *Business Today* (2024) <https://www.businesstoday.in/magazine/deep-dive/story/razorpays-simple-fundamental-heres-what-makes-the-fintech-major-a-force-to-reckon-with-424147-2024-04-03> accessed 3 February 2025.

28. 'About Razorpay and The Team Behind It' (*Razorpay*) <https://razorpay.com/about/> accessed 3 February 2025.

29. 'Razorpay's Simple Fundamental: Here's What Makes the Fintech Major a Force to Reckon With' (n 27).

30. 'We Plan to Reach $750 Billion Payments Volume Goal by 2030: Razorpay's Shashank Kumar' *Business Today* (2024) <https://www.businesstoday.in/technology/news/story/we-plan-to-reach-250-million-payments-volume-goal-by-2030-razorpays-shashank-kumar-444944-2024-09-08> accessed 3 February 2025.

FOSS at Razorpay

Razorpay's commitment towards open source is reflected in their FOSS usage across their technology stack, products, services and throughout their Software Development Life Cycle (SDLC), in the form of developer productivity tools.

The organisation's mechanism to evaluate software components for use in their solutions has the following decision-making workflow. Firstly, FOSS components that fulfil their requirements and that can readily solve the problem are sought. Then, based on an assessment of their TCO, the solution is either self-hosted, or a managed FOSS solution is opted for, if the organisation believes that managing the solution would distract them from their strategic objectives. Commercial proprietary solutions are also explored in scenarios where there are time constraints, for compliance purposes, and depending on the maturity of the solution.

The organisation also strategically contributes to FOSS, and has developed and open-sourced solutions used across the industry. Some of which are listed below:

1. Upon discovering that their open source asset management tool was not enterprise grade, the organisation contributed to upgrading the same to meet enterprise standards.
2. The organisation open-sourced their Access Control Server (ACS) software, which performs multi-factor authentication and authorisation to verify the identity of users transacting with debit and credit cards. A hosted version of this is currently used by around 25 banks.
3. Another key piece of software is the IFSC toolkit that validates IFSC codes and performs the conversion from code to branch name. This is open-sourced under the MIT license and is accessible through API as a hosted service.

The organisation also participates in Hacktoberfest[31], an event where volunteers contribute to FOSS projects. It hosts an internal hackathon annually, alongside organising weekly sessions called 'Open source Fridays' where developers fix bugs in FOSS projects.

The organisation could not provide an estimate on code reuse based on LOC (lines of code), since they extensively employ a micro-services architecture wherein teams reuse services through APIs. There are 53 projects currently listed on GitHub that have been developed by Razorpay.

The discussions indicated that there are some use cases for which the organisation uses commercial proprietary software such as Active Directory and SAP Financial Reporting.

Benefits and Challenges

Discussions indicate that the following are the major benefits that Razorpay derives from FOSS:

1. Lower TCO.
2. Ability to build differentiated products and services.
3. Eliminates the need to re-invent the wheel, i.e., allows incremental development on existing solutions.
4. Aids development of modular solutions where individual components can be swapped.

31. 'Hacktoberfest' <https://hacktoberfest.com> accessed 15 February 2025; 'Razorpay' (*GitHub*) <https://github.com/razorpay> accessed 16 February 2025.

5. Developing open source solutions internally helps the organisation to build credibility, especially if their product becomes the standard.

Banks and FOSS: Insights from Razorpay

Modularity in IT system architecture enabled by FOSS has greatly benefited banks, which have traditionally relied on a single technology service provider (TSP). With a modular architecture, banks can choose different TSPs for different components and each TSP can provide services in their area of specialisation.

The discussions also highlighted the challenges the organisation faced in working with FOSS.

1. Discovering, post adoption of a component, that a community to maintain and support the project is non-existent.
2. FOSS projects becoming EOL (End of Life) i.e., no longer being updated or supported.
3. Managing updates.

PocketATM

PocketATM is a fintech start-up founded in 2021 that is working towards expanding financial inclusion and has around 15 employees as of date. Its offering aims to digitise the process of cash withdrawal by enabling users to transact at neighbourhood stores through UPI. This offers consumers enhanced convenience while also providing merchants or shop owners an avenue to deposit surplus cash and receive instant credit to their account, obviating the need to visit a bank for this purpose. Lastly, banks also benefit from increased touch points to serve consumers, and this particularly benefits those residing in areas underserved by ATMs.

FOSS at PocketATM

Our discussions indicate that this start-up has been an adopter of FOSS since its inception, having instituted a clear mechanism to evaluate and decide on the adoption of a software component. For any software requirement that is related to their core product, FOSS components are evaluated exclusively. As for any software needs apart from their core product, a subscription to a managed open source solution is purchased, which is then integrated with their IT systems.

It is also observed that the developers at the organisation are part of multiple FOSS communities on Telegram, where they help with resolving issues faced by other community members. They also regularly engage with their counterparts at other organisations to mutually find solutions to common problems.

Only about a third (30%) of the code developed is reused internally, since most projects are new implementations that require new logic. However, the organisation has set a target to achieve code reuse in the range of 40-50%.

The organisation exclusively uses FOSS solutions across five software categories, while proprietary software dominates across the rest. They are forced to use proprietary licensed software in scenarios where compatibility and integration with a bank's IT infrastructure or that of their vendors is necessary. This is driven by the largely proprietary technology stack of banks, and the unavailability of APIs for FOSS solutions. Despite this limitation, licensing costs constitute only 3-4% of their overall IT budget.

Banks and FOSS: Insights from PocketATM

Many banks continue to use a lot of proprietary software since making the shift to FOSS is technically complex and could be expensive at times. It also requires personnel proficient in both proprietary and FOSS. Specifically, there is a demand for organisations that specialise in migrating an organisation's technology stack from proprietary software to FOSS.

Benefits and Challenges

Discussions with PocketATM suggest that the organisation has experienced the following benefits from adopting FOSS solutions:

1. Lower costs through savings on licensing fees.
2. Faster implementation of solutions as compared to proprietary alternatives.
3. Shorter training time, as new employees can be sufficiently trained in approximately 15 days to produce output and contribute to the organisation's work. However, merely understanding the documentation of licensed software is challenging.
4. Given a software requirement and use case, the choice of open source tools or solutions to be used is very clear and segmented.

However, experiences within PocketATM suggest that FOSS adoption brings with it some challenges:

1. Integration with other systems (particularly proprietary ones) is difficult.
2. At times, the community takes significant amount of time to provide resolutions for technical issues.
3. FOSS offers limited features, which works well for start-ups. However, the organisation is uncertain whether FOSS would continue to fulfil future requirements as they expand.

That said, there is a general sense within the organisation that FOSS has witnessed improvement in some aspects over the past few years. These include the general availability of support and considerably better user interfaces aided by advancements in AI.

One of the largest private sector banks in India

One of the largest private sector banks in India (name withheld on request) has also adopted FOSS for diverse applications. This bank has a network of more than 9,000 branches and 20,000 ATMs. Their annual revenue for F.Y. 2024 was more than ₹1 trillion.

This bank has been a pioneer in leveraging technology to provide enhanced services to its customers. We are avoiding specific mentions of examples to protect their confidentiality.

Banks traditionally have been laggards regarding, and largely consumers of technology. For the most part, they have been buying platforms and stitching them together. The bank which we have taken as a case study realised that it was critical to own their platforms in order to differentiate themselves, respond to changes faster, maintain an edge in the market and also compete with fintech organisations.

They balance their regular day-to-day operations while developing new strategic initiatives in parallel. To address the latter, the bank has established software development centres with engineering expertise that follow a software development

model called the 'factory method'. Factories are organisations with 100-150 technical personnel, alongside personnel ranging from sales to compliance to operations teams. They are operated like a lean, self-contained fintech organisation and hence are unburdened by bureaucracy.

FOSS adoption in the bank

The bank began its software development journey in 2020, marking a strategic shift towards building its engineering capabilities. Over time, the organisation has significantly evolved, with 20-25% of its software now developed from its own engineering efforts and using FOSS. The bank has moved from merely accessing open source solutions to building open source tools at scale for its consumers.

Open source is extensively integrated into their production environment. It is primarily employed within software base products and services for clients, and to explore new features, technologies and opportunities for innovation. While the bank recognises the value of open source, it considered it non-strategic for internally used applications, and opts not to allocate engineering resources to these low-risk areas.

The bank has adopted an unconventional approach to arriving at its technology stack/ architecture. Banks generally prescribe a set of approved software components and/ or IT architectures to be employed for developing software solutions. These prescriptions are often based on partnerships of the bank with software support service providers.

This bank believes that such centralised architectures limit creative freedom. Instead, it has a governance body called the Enterprise Architecture Council (EAC) that comprises of technology architects and domain experts. This body evaluates proposed software solutions alongside the FOSS components they are based on, and then takes a call on their support strategy primarily by considering factors like suitability and performance. The bank also has an internal OSPO (Open Source Program Office) that oversees overall FOSS adoption.

Currently, the bank interacts with open source as a consumer, with no contributions. According to our interactions, while future plans of the bank include contributing to the open source ecosystem, their regulatory, licensing and enterprise policies need to be addressed first. With this in mind, the bank follows meticulous coding practices to ensure compliance with licensing requirements.

Open source is seen as an essential part of the bank's development ecosystem, supported by licensing and governance frameworks. They frequently acknowledge the open source components they use in conferences and presentations. To foster innovation and adoption, the bank actively involves employees and partners in open source initiatives.

They conduct five or six hackathons and coding challenges annually, engaging internal teams, vendors, and fintech and start-up communities. The bank has also involved non-technical employees in low or no-code challenges, which have been widely praised for promoting inclusivity and encouraging creative problem solving.

Benefits and Challenges

This bank has experienced the following benefits from adopting open source solutions:

1. Faster adoption and software development.

2. Agility to adapt and respond to evolving business needs effectively.
3. Reduction in total cost of ownership, while maintaining efficiency.
4. High security.

Our interactions also suggested that the bank has faced some challenges.

1. Subscribing to support services for an open source solution from the principal OEM or project creator can negate the cost advantage of open source. Thus, organisations need to maintain internal technical capacity to support their FOSS solutions effectively.
2. Identifying vulnerabilities and implementing timely patches.

To address these challenges, the bank is currently relying on their in-house back-end team to support their platforms. This effort is made easier by generative AI tools that aid in understanding and optimising the codebase.

Education Sector Case Studies

PupilFirst

PupilFirst is an education technology, or ed-tech start-up founded in 2011 that aims to usher in a digital transformation of classroom education. The organisation has developed an open source LMS (Learning Management System) that hosts online courses alongside having mentors for personalised guidance. This system also allows education content creators to host their courses, create assessments and check for plagiarism in student/participant submissions. In a nutshell, they are aiming to provide a complete suite of digital tools to run an online school.

The organisation has around 20 employees and generated a revenue of ₹1.3 crore in F.Y. 2022-23. Being a completely remote team, they extensively leverage IT for communication, and project and document management.

FOSS at PupilFirst

PupilFirst's core product, its LMS, is open source and released under the MIT license. When evaluating software components, either FOSS or proprietary, the costs of both acquiring and running them (i.e. TCO) are compared. In case of FOSS solutions, those that in their assessment are easy to run, maintain and upgrade are adopted. Else, the organisation opts for a managed FOSS solution. For requirements met only by proprietary solutions, those known to be reliable, originating from known brands and having SLAs in place are considered, while those with user based licenses are avoided. Lastly, a software solution is considered for in-house development only if it forms an integral part of the organisation's IT systems.

In addition to consuming FOSS, the PupilFirst team is extensively involved in fixing bugs in FOSS projects. Upon encountering an issue, they work towards developing a fix, undertake testing and send a pull request to the project maintainers or its community. Furthermore, they follow the process put in place by the project's core group to have the request containing the fix merged.

The organisation participates in Hacktoberfest, an annual event organised by DigitalOcean to encourage contribution to FOSS. PupilFirst largely uses FOSS solutions, exclusively in seven software categories and partially in four others. Proprietary software is their next preferred choice with four

categories comprising only these solutions alongside a similar number having proprietary solutions in combination with other types of software. In-house software is the sole choice for three categories.

PupilFirst also uses multiple proprietary software solutions such as Google Workspace, Discord (an instant messaging platform), and Typeform (SaaS solution for creating online forms and surveys) alongside AI models from OpenAI, which are generally closed models.

'The internet would not have been possible without open source.'
Bodhish Thomas, Head of DPG, PupilFirst

Benefits and Challenges

The discussions suggest that PupilFirst has derived the following benefits using FOSS:

1. Lower total cost of ownership.
2. Open and accessible source code enables examination and thereby aids in understanding working of the software.
3. Increased trust by means of visibility of the working of the solution.
4. The FOSS community comprises ethical hackers i.e. members who check for security vulnerabilities and report issues with the code. This form of community audit enhances security.
5. Negligible maintenance is required for operation.
6. Open standards and protocols aid in development of software.

'It is a gift to be able to be able to examine and understand the working of software.'
Bodhish Thomas, Head of DPG, PupilFirst

However, discussions also highlighted some challenges they faced.

1. FOSS projects are at times not maintained by the community or become inactive due to the lack of a revenue model.
2. Even if a FOSS project is regularly updated, some of its dependencies that constitute the core functionality might not receive updates, thereby creating a 'dependency mismatch'.
3. FOSS projects that undergo changes to licensing terms get forked into multiple versions.

Kalvium

Kalvium is an ed-tech start-up founded in 2021. The organisation has developed curriculum for a B.Tech. computer science programme that is delivered in collaboration with 15-20 universities. The organisation was founded to address the gap in the skill levels of engineering graduates, which exist despite their spending four years or about 5000 hours in their undergraduate programs, and bring them on par with industry needs.

Kalvium's program is fully digital and focuses on hands-on practical experience in an office-like environment, while being low on lecture time. It is developed by working backwards from industry inputs and the founder's own understanding of industry requirements.

The organisation has leveraged technology to capture hourly learnings of students and is able to provide deep data-backed insights on individual student progress. It has grown rapidly in a short time span, currently employing 250-300 personnel, and has an annual revenue in the range of ₹15-50 crore.

FOSS at Kalvium

Kalvium's choice of FOSS is strongly driven by their desire to be in control of their core technology in the long run. The organisation feels that since the impact of their work would be evident on a timescale spanning decades, they cannot rely on proprietary software vendors, as their existence cannot be assured over such a long timeframe.

When evaluating software components for use, the organisation considers FOSS alongside the best proprietary solutions. The expected Return on Investment (ROI) is a crucial factor in deciding on the adoption of a solution.

The organisation releases all its students' projects as FOSS. It also leverages its technical expertise and experience of working with FOSS by employing it in the teaching of computer science. This knowledge acts as an added benefit since real world SDLCs are comprised between 70-80% by FOSS. However, FOSS being used for education could be a correlation uniquely suited to the organisation's business and specific use case. In other words, FOSS might not have been leveraged as much if the organisation were to deliver a program not related to software programming.

As part of Kalvium's curriculum, coding challenges are organised once or twice a week and around 200 of them have been organised till date.

The extent of code reuse is not a metric that is tracked by the organisation. However, its software development process closely matches that followed in the development of FOSS.

Kalvium uses a mix of software types equally distributed across all categories, with FOSS being used the most, followed by proprietary and software developed in-house. Additionally, across all except nine categories, at least two or all three types of software were used.

Proprietary solutions such as cloud services and those customised for the organisation's use case, with usage-based rather than user-based licensing are used.

Benefits and Challenges

Our discussions suggest that Kalvium has experienced the following benefits of FOSS:

1. Control over the firm's core technology.
2. Ability to customise software as per requirement.
3. Freedom from vendor lock-in and pricing mechanisms of commercial proprietary software providers.

The need to carry out support and maintenance activities and the requirement of internal technical capacity to work with FOSS were highlighted, during discussions, as the major challenges involved in their adoption of FOSS.

VGLUG: An Inspiring Community-Driven FOSS Education Story

Though not strictly falling within the definition of an organisation for the purpose of our case studies, in the context of education, we also consider it important to highlight an initiative called Villupuram GNU Linux User Group (VGLUG). It was founded in 2013 as a non-profit based in Villupuram, Tamil Nadu. It engages volunteers to educate students from marginalised, and economically and socially backward communities on FOSS. The organisation currently hosts around 240 students across five batches.

Villupuram, a Tier-III town and a district in Tamil Nadu, is characterised by a lack of industries and plagued by low literacy and poor class 10 and class 12 pass percentage rates (the district is featured sixth from last in class 12 pass percentage in the state in 2023).[32] The founder of this group, Karkee Udhayan, who hails from the town and is employed in Bengaluru, decided to establish the organisation to give back to society by utilising weekend visits to Villupuram thusly.

FOSS was specifically chosen as a subject on which to educate students, as it is very much in alignment with the principles of free knowledge-sharing. According to the group, this is particularly significant as FOSS knowledge is akin to scientific facts produced cumulatively over generations and not by a single entity or group of people.

> *'We are using FOSS as a tool for poverty eradication.'*
> **Karkee Udhayan, Founder, VGLUG**

VGLUG, through assistance from volunteers, conducts two-hour sessions every Sunday on technologies like Python, Node.js, Flutter, etc. The format of these sessions is unlike a traditional classroom, and they are instead referred to as meet-ups where facilitators share their knowledge. VGLUG's volunteers largely comprise of students from previous batches who are keen to contribute, on account of being beneficiaries of the same program and hailing from the town.

---

32. The Hindu Bureau, 'Tamil Nadu State Board results key updates | T.N. records 94.56% pass rate in class 12 State board exams' *The Hindu* (6 May 2024) <https://www.thehindu.com/news/national/tamil-nadu/tamil-nadu-state-board-results-live-updates-may-6-2024/article68144709.ece> accessed 30 January 2025.

VGLUG's Operating Model

VGLUG specifically aims to serve needy students, i.e., those who are marginalised in many ways. It has accordingly set an eligibility criteria and devised a process to select students who can attend their meet-ups.

Students satisfying any one of the following conditions are eligible to apply:

1. Those residing within a 60 km radius of Villupuram.
2. Whose parents do not own land.
3. Whose parents are working as agricultural or daily wage labourers.
4. Who come from single-parent household.

The shortlisted students are interviewed, and volunteers conduct visits to their homes. The objective of these visits is twofold: firstly, to verify the candidates' background and personal information and secondly, to educate their parents about VGLUG's activities and build trust to ensure that they allow their wards (particularly girls and women) to attend meet-ups on a regular basis.

VGLUG plans to establish GNU Linux User Groups (GLUGs) in each village, in true grassroots fashion, where residents of that village would serve as volunteers.

VGLUG operates using minimal IT infrastructure, limited to laptop computers and projectors which are secured through donations. They utilise the premises of schools, which are available on weekends, to conduct their meet-ups.

Despite working with limited resources, VGLUG's work has put Villupuram on the world stage. This was due to organising of two global events in the town, the 2022 Wikipedia meet-up and the International Debian Conference in 2023.

Students actively contribute to FOSS as part of their engagement with VGLUG. They have developed applications for Linux distributions in addition to four tools for Wikipedia, including those for translation, OCR (Optical Character Recognition) and spell-checking. There are currently 10 projects from VGLUG listed on GitHub.

At the culmination of the program, in lieu of an examination, a hackathon is conducted to assess students' learning.

VGLUG's Impact on Policy

Students from Villupuram are often forced to migrate to other cities in search of work. To address this challenge, VGLUG put forth a demand to establish an IT park at Villupuram. While the Union Government initially rejected the proposal put forward by the Member of Parliament, the organisation was successful in including it in the manifesto of a party contesting the state assembly elections. Today, not only has a 'TIDEL Neo' IT park been established in the town, but this has also served as a model to establish similar IT parks in two other districts, Salem and Thanjavur.

Some of the other social impacts of VGLUG include:

1. Meet-ups being utilised for infusing socially progressive and rational thoughts among students.
2. Knowledge gained by students helping to identify and combat cybercrime and fake news at the grassroot level, since people with low digital awareness are often targeted by online scammers and cyber criminals.
3. Increase in female participation rates in FOSS, as special attention was paid to the needs of female students upon noticing an equal or greater interest among them to learn FOSS.

Software/ IT Sector Case Studies

Dhiway

Dhiway is a blockchain and Web 3.0-focused technology start-up based in Bengaluru and founded in 2019. The name Dhiway is derived from the Sanskrit term '*dhi*'.[33]

The organisation attempts to address the lack of trustworthy and verifiable information on the internet. The company's chief product, CORD, is an enterprise-grade blockchain that generates tamper-proof and verified credentials, useful for applications such as registry management and decentralised digital commerce. It obviates the need for centralised systems, paving the way for peer-to-peer value transfer with the added benefit of security and data privacy.[34]

CORD is approved by the OSI as open source, and is designed to support population scale initiatives like the UN's Sustainable Development Goals.[35] It is accompanied by APIs, SDKs and developer-friendly documentation to facilitate easy integration with user or third-party applications.

The organisation has around 25 employees, and the company generates revenue from the enterprise version of its open source offering.

FOSS at Dhiway

The founders of Dhiway, having formerly worked with Red Hat, are highly familiar with FOSS and hence the organisation's products, engineering and revenue model are shaped largely by Red Hat's operating model.

Alongside FOSS, the organisation also extensively uses proprietary software and software developed in-house. When evaluating software components for use, the organisation has an inclination towards efficient solutions that also provide them a better experience. Costs are another major factor and those of software subscription are compared with that of in-house development, where staffing considerations also play a role.

Dhiway's engagement with FOSS extends beyond consumption, encompassing code contribution, bug-fixing and the creation of documentation. As per the organisation, their key contribution through CORD is making FOSS more usable while also providing support and after-sales services.

They also engage with groups that write protocols and specifications like the WCAG (Web Content Accessibility Guidelines). This involves designing for the future and conceptualising 'what is' (present state) and 'what could be' (imagined future), while incorporating aspects of privacy and security.

33. Dhiway Networks, 'Dhiway: A Journey Rooted in Knowledge and Trust' (*Dhiway*, 3 October 2024) <https://dhiway.com/dhiway-a-journey-rooted-in-knowledge-and-trust/> accessed 3 February 2025.

34. 'From Centralized Control to Decentralised Trust: All You Need to Know' *India Today* (16 February 2024) <https://www.indiatoday.in/information/story/from-centralized-control-to-decentralised-trust-all-you-need-to-know-2503060-2024-02-16> accessed 3 February 2025.

35. 'Dhiway Announces the General Availability of the CORD Blockchain for Building Digital Public Utilities' *Financial Express* (10 November 2022) <https://www.financialexpress.com/business/blockchain-dhiway-announces-the-general-availability-of-the-cord-blockchain-for-building-digital-public-utilities-2803263/> accessed 3 February 2025.

The organisation has also served in the Samagra Code4Gov Tech challenge, where participants develop software to be released as DPI (Digital Public Infrastructure) or DPG (Digital Public Goods), as a mentoring organisation. About 10% of their developed code is reused internally.

FOSS dominates the technology stack of Dhiway with most categories of their software employing only FOSS-based solutions. Proprietary software was the next popular choice, used across five categories, while in-house software was used heavily in two and partially across five categories.

Proprietary and SaaS solutions are specifically used to meet operational and administrative requirements like HR and finance. Since the users of these applications are non-technical staff, the organisation feels that it is imperative for these users to be familiar with the software being used. If they are not, it hinders their functioning and amounts to devaluation of their work.

The organisation employs GitHub for their software development pipeline, which is not open source. Also, applications are hosted in virtual environments and on public cloud infrastructure where they may or may not be run on FOSS.

Benefits and Challenges

The organisation feels that FOSS has undergone significant improvement in stability, and now facilitates easier, and faster, setup and installation. The number of entities selling managed FOSS services in the market has also increased.

'A firm like Microsoft is now sponsoring FOSS projects. This is a testament to FOSS as a preferred model to deliver software and content.'

Sankarshan Mukhopadhyay,
VP, Customer Experience, Dhiway

Our conversations suggest that following are some of the challenges Dhiway faces with FOSS implementation:

1. The long-term survivability of FOSS projects is contingent upon them being associated with a foundation that provides support and oversees its governance. Otherwise, they risk becoming EOL if the project's originators do not see value in continuing to maintain them. This has an adverse impact on downstream projects.
2. FOSS projects that drive impact both for businesses and society are not necessarily the ones that are attractive for developers to work on. This hinders the uptake of their development.
3. Universities have a preference towards having students working on end-to-end software projects. In contrast, contributing to FOSS projects involves numerous minor bug fixes and code improvements to a large project. Hence, universities are unwilling to recognise the contributions to FOSS projects for fulfilment of criteria for student internships, as the work done cannot be easily evaluated. In this context, it needs to be added that, in reality, end-to-end software projects as a category are seldom implemented, while work done towards FOSS contribution is often valuable.
4. When new workers enter the job market without adequate exposure to FOSS, they find it difficult to understand and work with FOSS, given its distributed architecture.

Economic Value Addition

Dhiway spends $10,000 annually on proprietary software for their internal needs, which includes Zoho Suite, Google Workspace and GreytHR. They estimate that using managed FOSS equivalents would cost them only around $8000. They also point out that developing equivalent solutions in-house would be expensive, and they estimate the cost to be around $15,000.

Thoughtworks

Thoughtworks is a leading global technology consultation organisation founded in Chicago in 1993. The organisation pioneered the application of agile software development to globally distributed teams known as distributed agile to develop custom software solutions for its clients.

The organisation is an extensive user of, and contributor to, open source since its inception. It views the philosophical concept of open source as a driver of software quality that provides the ability to build superior solutions.

In the span of just 30 years, the organisation has grown exponentially to more than 10,000 employees, and it has 48 offices across 19 countries.[36] The company generated a revenue of $1,126 million for the F.Y. 2023 with the figure for the Asia-Pacific region being $387 million.[37]

Note: The information garnered for this case study, though relating to Thoughtworks, is specific to the Digital Public Goods (DPG) vertical of the organisation. This vertical typically caters to clients like governments, foundations and NGOs, and multilaterals like the World Bank. It also operates with lower revenues and generates lower profits, with its operations slightly overlapping with the CSR function.

FOSS at Thoughtworks

Thoughtworks was ranked among the top 25 contributors to GitHub by the Open Source Contributor Index.[38] They have also contributed to the development of 14 DPGs, key among them being the development of Bahmni,[39] an open source hospital management system used in more than 50 countries. The organisation has also utilised CSR funds to finance FOSS projects.

The development of the following open source tools, popularly used across the industry today, was also undertaken by the organisation.

1. **Selenium:** A suite of tools for web browser automation, used for testing web applications.
2. **CruiseControl:** A Java-based framework for continuous integration.
3. **Mingle:** A project management and collaboration solution.

Since the organisation primarily builds software for clients, the choice of the type of software component largely depends on client-induced constraints like delivery timeline, budget, as well as operating and maintenance costs. While most clients are agnostic

36. 'Our History' (*Thoughtworks*) <https://www.thoughtworks.com/en-in/about-us/history> accessed 3 February 2025.

37.'Thoughtworks Reports Fourth Quarter and Full Year 2023 Financial Results' (*Thoughtworks*) <https://investors.thoughtworks.com/news-releases/news-release-details/thoughtworks-reports-fourth-quarter-and-full-year-2023-financial/> accessed 3 February 2025.

38. 'OSCI – Open Source Contributor Index' <https://opensourceindex.io/> accessed 3 February 2025.

39. 'Bahmni™ Open Source EMR & Hospital Information System (HMIS)' (*Bahmni*) <https://www.bahmni.org> accessed 3 February 2025.

to the underlying technology stack of their software solutions, those keen on providing a better user experience to their customers prefer a custom-built solution that differentiates them from their competition.

The DPG vertical's clients, on the other hand, generally prefer FOSS-based software solutions that are then hosted in their own data centres on basic commodity hardware. These include UIDAI, NPCI, ONDC etc., while some UN agencies use proprietary software.

The organisation's preference for FOSS is a consequence of multiple factors including the accessibility of available codebases and FOSS' fitments for custom software development, and is aided by the geek culture of the organisation where engineers enjoy working with FOSS. They regularly organise 'Geek nights', a form of internal hackathon where employees are provided a problem statement and tasked with developing a solution. Employees are also provided with a budget for R&D and self-learning, depending on the financial situation of the organisation and market conditions.

The organisation estimates that about 20% of their developed code is reused internally. However, this is only an estimate and not a calculated metric.

The organisation uses proprietary vendor-based solutions for internal software requirements despite sufficient interest amongst developers to build these applications.

Benefits and Challenges

Our conversations indicate that the following are the major benefits seen by Thoughtworks in using FOSS:

1. Lower TCO.
2. Highly secure, reliable and scalable.
3. Allows development of superior quality software.
4. Offers better privacy.
5. Provides freedom from vendor lock-in.

However, the conversations also indicated that there are concerns regarding the operational challenges of FOSS projects, particularly regarding sustainability. It was pointed out that when contributors lack a revenue stream and means to profit from the activity, FOSS development risks being dependent on the goodwill of these contributors.

Remiges

Remiges Technologies Pvt. Ltd, formerly known as Merce Technologies Pvt. Ltd and founded in 2000, is a medium-sized custom software development and IT services enterprise with approximately 350 employees and an annual turnover of ₹35.51 crore, as of March 2024. Remiges aligns its business practices with the robust adoption of FOSS, integrating it into software services and infrastructure management for its clients.

The organisation undertakes the development, deployment, maintenance and support of business applications. They specialise in app modernisation which encompasses database migration, DevOps, automation, UI/ UX development, and developing cloud native applications. Their skills in this area also include full lifecycle software development,

project management, on-site maintenance services, server infrastructure management, and the integration of heterogeneous applications.

Remiges possesses experience in building business applications in the verticals of trading, manufacturing, depository and financial services, e-governance applications, etc., catering primarily to large organisations in the Banking, Financial Services, and Insurance (BFSI) sector.

FOSS at Remiges

Our conversations indicate that Remiges has adopted FOSS since its inception as a cornerstone of its operations. Remiges extensively leverages FOSS in three primary areas: developing solutions for its clients, for internal software requirements, and for innovation of new projects. They find FOSS to be a fundamentally powerful and very valid idea, both from a pragmatic and philosophical point of view.

When selecting software components, Remiges assesses the technology stack based on client requirements. If the client specifies a particular technology, it is adopted. However, when the choice is left to them, Remiges evaluates functional and non-functional requirements to determine the most suitable components, favouring FOSS over proprietary components wherever possible. If the solution demands high performance, unusual feature sets, or a sophisticated component layer not adequately available in FOSS, proprietary components are chosen. In such cases, clients are informed about the licensing terms and associated considerations.

Approximately 80–90% of Remiges' employees use Linux-based systems, and the company relies on open source tools and platforms both for internal needs and to develop client deliverables. In fact, almost all software development tools and platforms used are open source.

In addition to consumption, their engagement with FOSS also includes:

1. Cultivating strong familiarity and acceptance of FOSS tools among their developers: As a service-oriented organisation, their advocacy for FOSS creates a continuous cycle of adoption, strengthening internal expertise and actively contributing to the broader open source ecosystem. This helps in creating a talent pool of developers skilled in FOSS and it becomes difficult for them to switch to working on proprietary software.
2. Remiges also encourages clients in the BFSI sector to explore FOSS solutions. It was learnt from our conversations that through Remiges' assurance and detailed demonstrations of FOSS' security and performance benefits, some clients have implemented their first mission-critical FOSS applications under Remiges' guidance.
3. The organisation has open sourced six products under the Apache 2.0 license that are accessible on GitHub.

While their focus on custom software products leads to low internal code reuse due to closed source delivery requirements for clients, Remiges identifies and integrates common FOSS subsystems wherever feasible.

The organisation largely uses FOSS solutions, and exclusively so in five software categories. Proprietary software is their next preferred choice with four categories using only these solutions. For the rest of the twelve categories, a mix of FOSS and proprietary solutions, with FOSS dominating the tech stack, is used across all categories except one, where largely proprietary solutions are used.

While FOSS is prevalent in their technology stack, Remiges also integrates proprietary solutions in specific contexts. Proprietary software is utilised when client requirements dictate its use, or when it demonstrates clear superiority over open source alternatives, such as in the case of Microsoft Office and Google Workspace.

Benefits and Challenges

During our conversation, it has been pointed out that Remiges experienced the following benefits from adopting FOSS solutions:

1. Better stability and performance consistently across most software categories, often surpassing proprietary alternatives.
2. Freedom from invasive licensing audits. Licenses of proprietary software contain clauses that authorise the conduct of audits to verify compliance with licensing terms. The Business Software Alliance (BSA), an organisation which represents major proprietary software companies, conducts these audits through on-site inspections to verify software installations against purchased licenses. For Remiges, the absence of these intrusive audits removes a significant operational burden, eliminating what can feel like 'legalised extortion'.
3. Increased security, scalability and flexibility, enabling unrestricted use without concerns over license violations. Furthermore, the collaborative ecosystem ensures that widely used FOSS solutions benefit from continuous scrutiny thereby enhancing security.

However, Remiges has also experienced some challenges.

1. The democratic nature of FOSS development, while empowering, can lead to project fragmentation. Internal disagreements within the project's core team can cause a project to split into multiple versions or forks. This affects the stability of the variants and forces organisations to carefully evaluate and choose among competing versions.
2. Financial and other highly regulated institutions must adhere to stringent Information Systems (IS) audit requirements, often mandated by regulatory authorities. They must obtain an audit from ISACA, an international association focused on IT governance. These audits often mandate the presence of a legal entity to provide support and accept liability in case of system failure. Unsupported FOSS solutions face challenges in meeting these requirements, making enterprises in regulated industries cautious about their adoption. While corporate-backed FOSS offerings (e.g., Red Hat) address this concern, the lack of formal contracts or guarantees of liability remains a barrier for unsupported solutions.
3. Although FOSS eliminates licensing fees, its total cost of ownership may not always be lower and is complex to calculate. Personnel costs, particularly for the skilled developers and administrators needed to maintain and support open source solutions, can offset the savings from free software.

4. Proprietary software is necessary only in fringe areas where FOSS options are inadequate. For some widely used software like Microsoft Office, open source alternatives are still not sufficiently competitive. For instance, while open source tools provide basic functionality, they lack the advanced features and sophistication of Microsoft PowerPoint for instance, which remains the superior product in its category.
5. Performing maintenance and support of solutions.

NxtGen Datacenter and Cloud Technologies

NxtGen Datacenter and Cloud Technologies (NxtGen) was founded in 2012 as a provider of high density data centre facilities. The organisation naturally progressed and moved up the value chain by becoming a provider of public cloud infrastructure services. They currently have around 400 employees and their services are utilised by major clients like BookMyShow and ECI (Election Commission of India).

NxtGen provides two types of cloud services: an enterprise cloud service called SpeedCloud, built using OpenStack (an open source cloud computing infrastructure platform); and a public and private cloud service built on VMware's hardware virtualisation, a proprietary technology. In fact, NxtGen is VMware's largest Cloud Service Provider (CSP) in the Asia-Pacific region.

FOSS at NxtGen

NxtGen adopted FOSS two or three years into their journey, coinciding with their foray into the cloud infrastructure services business. The organisation credits FOSS for enabling them to provide highly scalable cloud services at prices substantially (around 80%) lower than that of other large service providers.

'We would choose to use FOSS 10 out of 10 times.'
Abhisyant Anasapurapu,
Head of Product Development, NxtGen

Our conversations suggest that the choice of the type of software that serves as the underlying stack of the cloud service, is entirely customer-dependent. While the organisation has a strong preference for FOSS, sometimes customers opt for the proprietary offering.

NxtGen has developed, and made freely available for public use, four business productivity tools based on open source AI models by Meta and Hugging Face. The objective is to combat the low perceived value of AI as a result of the freemium model of most AI tools, and signalling the organisation's AI capabilities that could aid future business expansion.

Code reuse is low in the organisation, owing to their highly diverse technology stack which is guided by the compatibility of constituent components.

The organisation does not use cloud services from any other service provider in any form, including SaaS applications. The only exception in this regard is Microsoft Outlook for email services since it is essential for business continuity.

Overall, the organisation's usage of proprietary software and FOSS is reported to be in the ratio of 1:50.

Benefits and Challenges

Conversations with NxtGen suggested the following to be the key benefits of their FOSS adoption:

1. Its ability to customise software, which enables differentiation and value addition in their products and services.
2. Provides a readily available framework that serves as a good starting point for developing solutions.
3. Eliminates licensing costs.
4. Offers greater visibility, control, stability, and robustness.

The organisation uses Mattermost, an open source collaboration platform, with some customisations, as its internal messaging application. This ensures the security and confidentiality of company data since it can be self-hosted, as opposed to other proprietary solutions that are only available as SaaS offerings. Mattermost has also proved to be highly robust and functions without issues despite being hosted on a single docker container and being used by all the employees of the organisation.

The key challenge with FOSS for NxtGen is that they are solely responsible for ensuring performance and up-time of their services, alongside resolving any bugs and issues. This not only increases the operational workload but demands an in-depth understanding of the solutions and the possession of the requisite technical skillset. However, the organisation views this as a risk worth taking, because of the value they derive through FOSS adoption.

Tech4Good Community

Tech4Good Community (T4GC) is a non-profit organisation operating in the software and IT services sector. Founded in 2018, the organisation has grown from around five employees in 2022 to a team of approximately fourteen people in 2024. They focus on enhancing the technological capabilities of non-profit organisations and social enterprises. Their mission is to bridge the technology gap in the social sector, helping organisations leverage technology to amplify their impact across various domains, including climate change, disability, health, education and livelihoods, with particular attention to gender and community issues. It has formed partnerships with several organisations, such as the Rainmatter Foundation and Rohini Nilekani Philanthropies, who have supported its initiatives financially.[40] T4GC's operations are primarily financed through philanthropic grants.

T4GC raises awareness about the potential of FOSS for non-profits, helping organisations understand the benefits of adopting open source technologies within the constraints of their limited resources. Through a range of initiatives, T4GC not only raises awareness but also builds capacities around open source technologies, demonstrating their impactful applications within the non-profit sector. Some of these initiatives are:

40."Tech4Good Community' (*Tech4Good Community*) <https://www.tech4goodcommunity.com> accessed 3 February 2025.

1. **FOSSFwd program:** This is a comprehensive program through which an end-to-end solution for non-profits is developed using FOSS. The program involves either building custom solutions from scratch or configuring existing open source technologies, primarily using Frappe's offerings, to improve operational efficiency. Participating organisations pay a nominal fee for software development, making it an accessible option for non-profits with constrained budgets.
2. **FOSSFwd Grid:** T4GC's FOSSFwd Grid events go beyond traditional workshops by serving as dynamic platforms for sharing best practices, success stories and insights into implementing FOSS-based solutions.
3. **Impact stories:** Through post-engagement interactions, T4GC collects feedback to assess the impact of their developed solutions. This highlights real-world success stories from non-profits they have worked with, helping others understand the open source adoption journey and the resulting benefits.
4. **Climate coalition:** For this initiative, T4GC has partnered with climate organisations to develop programmatic interventions, turning data into actionable insights. Additionally, T4GC has facilitated roundtable discussions involving 5-8 organisations, and capacity-building sessions on data standardisation practices.

FOSS at T4GC

T4GC began primarily as a capacity-building organisation for non-profits through IT enablement. However, since 2023, T4GC has actively integrated FOSS into its operations. It now encompasses production environments, internal workflows, client interactions and back-end operations. This includes adopting FOSS-based solutions for project management, HR management, and custom-built tools like the Donor Management System (DMS), tailored specifically for Indian non-profits. The DMS is designed on the 80-20 principle, where 80% of workflows meet standard non-profit needs, and the remaining 20% are customisable by them or other developers and vendors.

T4GC's FOSS adoption has enabled them to develop custom apps using tools like ERPNext by Frappe, design standard modular and adaptable systems that can be customised based on client specific workflows, and enabled them to fill a gap for a DMS addressing the specific needs of Indian non-profits.

T4GC follows a structured, three-step evaluation process to assess potential solutions, focusing on workload capacity, workflow compatibility, long-term cost effectiveness (over 3-5 years) and a non-profit's specific use cases. This process ensures that the final software aligns with the client's requirements, operational realities and long-term sustainability. Post-deployment, T4GC also offers two or three rounds of staff training to ensure effective use and maintenance of the system.

While T4GC relies on FOSS, they also use proprietary solutions, like Google's Workspace, for certain internal operations.

Beyond adoption, T4GC engages with FOSS by:

1. Creating and sharing documentation for under-documented FOSS projects.
2. Presenting their FOSS based solutions at conferences, approximately ten in the last two years, such as the FOSS United's IndiaFOSS Summit, Tech4Good Conference, and others.

While they are relatively new to the FOSS journey, it is reported that the code developed within the organisation is reused internally, wherever it makes sense to. Moreover, T4GC is working on standardising their new DMS and contribute back to the community by providing it as an open source offering.

T4GC largely uses FOSS solutions, exclusively in seven and partially in two software categories. Proprietary software is their next preferred choice with two categories exclusively comprising these solutions alongside a same number having proprietary solutions in combination with other software types. In-house software is partially used for two categories of software.

Benefits and Challenges

FOSS has enabled T4GC to develop solutions tailored to the specific needs of non-profits. This flexibility allows organisations to define workflows that better suit their requirements rather than adapting to those of proprietary solutions, usually developed for corporates or large enterprises. Such solutions also tend to have many features that are irrelevant for non-profits. T4GC empathises with non-profits' need for flexibility and their unstructured nature of operations. Another advantage is that a solution developed once for a non-profit with relevant workflows can be deployed for multiple such organisations, with slight variations.

Open source tools also offer significant cost savings, as both the one-time and recurring costs associated with dependence on costly proprietary tools reduce, making them far more affordable for non-profits. For instance, it was reported that an activity tracking system that a proprietary vendor quoted at ₹18,00,000 was implemented by T4GC at a fraction of the cost (₹3,50,000).

Additionally, while some proprietary vendors impose rigid timelines, FOSS-based solutions offer flexibility in timelines, which is particularly valuable for non-profits operating on limited budgets and resources.

Despite its benefits, T4GC's experiences also suggest some challenges with FOSS:

1. Frameworks often lack active community support, making it difficult to resolve issues promptly. For instance, a bug fix request raised by T4GC is reported to have taken six months to receive a response.
2. Limited or outdated documentation complicates problem-solving and slows development. T4GC addresses this by reverse-engineering systems and creating the required documentation.
3. FOSS components face compatibility issues arising due to the restrictive nature of certain business models, such as tools tied exclusively to proprietary cloud hosting.
4. Progress in FOSS development projects for T4GC's area of work depends heavily on user feedback. Unlike a proprietary system with predefined functionality and a 'pay the subscription, get the system' model, custom-built FOSS solutions require feedback as input to evolve. Without this, changes cannot be implemented, and development stalls. In some cases, T4GC hosts systems on staging servers for extended periods, such as four months, waiting for non-profits to review and provide feedback.

5. Non-profits often lack the technical expertise or capacity to maintain open source solutions, which increases reliance on external support. T4GC mitigated this by offering low-cost services to assist with maintenance and knowledge transfer.
6. Despite T4GC offering training on use of the systems, organisations typically do not send their whole staff for training. This, combined with the high attrition rate in non-profits, exacerbates the issue of knowledge retention. For instance, when staff trained on HR systems leave, new employees often need training from scratch, especially if no internal knowledge transfer is done.
7. When developing solutions, deciding between 'must have' and 'good to have' features can be challenging.

'Freedom to make changes comes at a price.'
Akhila Somanath, Co-founder and Chief Growth Officer, T4GC

Despite these challenges, discussions with T4GC indicate that the benefits of adopting FOSS outweigh the drawbacks.

Economic Value Addition

T4GC highlights that calculating the cost of proprietary software is challenging due to its closed nature and uncertain pricing, which depends on multiple components. While a subscription fee of ₹3,000, for example, might seem affordable, vendor quotations can escalate significantly, often reaching around ₹8,00,000, as additional components are factored into the overall cost. There is often no frame of reference as to the rationale behind high quotations.

Healthcare Sector Case Studies

KG Hospital

KG Hospital (KGH) is a multi-speciality hospital located in Coimbatore established by the K. Govindaswamy Naidu Medical Trust in 1974. Committed to delivering advanced yet affordable healthcare, the hospital operates as a single-unit, super-specialty facility employing approximately 2,500 staff members.

The hospital is self-financed and reinvests its earnings into its operations, ensuring sustainability and continuous improvement. They also host a postgraduate medical institute and have trained over 300 doctors so far.[41]

The hospital's internal IT team, split into software and infrastructure teams, is critical to its business operations. About 80% of their applications are developed in-house.

FOSS at KGH

KGH employs a structured approach to assess and adopt software solutions, prioritising FOSS wherever feasible. However, for any requirement, they first try to identify a vendor software which is affordable and customisable. The vendor solution is adopted if it fits the requirement, as they prefer to use a readily available solution and not re-invent the wheel.

FOSS adoption at KGH has led to reduced licensing costs and increased software customisation, enabling the hospital to provide cost-effective healthcare services with software aligned to their needs.

41. 'About KG Hospital - Leading Multispecialty Hospital in Coimbatore' (*KG Hospital*) <https://www.kghospital.com/about-us.html> accessed 3 February 2025.

Examples of FOSS adoption include Zimbra for email management, open source alternatives for Microsoft Office suite, and customised vendor-developed applications.

The hospital's engagement with FOSS is primarily consumptive in nature. However, it was observed that it actively shares its technological advancements and best practices at various forums, including the following:

1. Presented at around four conferences over the last two years, including one organised by the Association of Health Care Providers (AHPI), and other health-focused events. They also present best practices in IT, alongside improvements they have made over existing solutions, and were felicitated with the Platinum Award by Quality Council of India and featured in an ISB Case Study.[42]
2. Organise at least one health hackathon annually and engage with students year-round through coding challenges and initiatives like VR-based stroke rehabilitation programs.

While FOSS dominates their technology stack, proprietary software is also adopted in specific cases where it offers the best fit, is affordable, and provides good security. These include:

1. **HR applications and pharmacy management software:** selected for features and good security.
2. **SIMS (Security Information Management System) integration for computerised prescription:** employs an external vendor solution to check for drug-drug interactions in real time.
3. **PACS (Picture Archiving and Management System:** chosen for its fit with their system, cost effectiveness, and security compared to other available options.

Benefits and Challenges

Conversations with KGH indicate that the hospital has experienced the below benefits from adopting FOSS solutions:

1. Significant cost savings on licensing, which is crucial for a hospital operating on a limited IT budget.
2. Enhanced flexibility and customisation to meet specific healthcare needs.

However, our interaction suggests they have also faced challenges with FOSS adoption. These include a requirement for regular upgrades, presence of bugs in some versions, and occasional system crashes. Despite the challenges, FOSS adoption remains a conscious and deliberate decision aligned with KGH's goals.

Swasth Alliance

Swasth Alliance (Swasth) is a non-profit operating in the healthcare sector. Established in response to the COVID-19 pandemic in 2020, Swasth represents a coalition of over 150 healthcare organisations united by a shared vision of improving health

42. The case study has two parts to it. Vijaya Sunder M and Meghna Raman, 'Health-Tech Strategy at KG Hospital Part A: Identification and Prioritization of Key Focus Areas' (Harvard Business Publishing) <https://cases.isb.edu/healthtech-strategy-at-kg-hospital-a.html> accessed 31 January 2025; Vijaya Sunder M and Meghna Raman, 'Health-Tech Strategy at KG Hospital Part B: Tech Strategy Design and Implementation' (Harvard Business Publishing) <https://store.hbr.org/product/health-tech-strategy-at-kg-hospital-part-b-tech-strategy-design-and-implementation/ISB293> accessed 31 January 2025.

outcomes across India.[43] Their team size varies between 10-16 and includes product and technical experts, specialists in healthcare and system operations, and personnel in clinical roles.

Swasth's mission is to leverage digital technologies to drive healthcare inclusion and improve health outcomes, with a special emphasis on enabling comprehensive, integrated care models. It aims to achieve Universal Health Coverage (UHC) and 'Health for All' in India through technology-enabled, value-based, integrated care.[44]

One of Swasth's flagship initiatives is the Health Claims Exchange (HCX), which represents a pioneering step towards building a digitised health benefits ecosystem. HCX is an open protocol for health insurance e-claims exchange specification, and a digital platform designed to simplify the flow of information between insurance holders and providers, facilitating faster and seamless claims processing. Developed as a DPG, HCX embodies a transparent and collaborative process involving over 180 volunteers across its evolution timeline from the healthcare, health tech and health insurance sectors.[45]

Although Swasth does not operate as a traditional healthcare service provider, its impact is rooted in leveraging FOSS to build ecosystems and community-driven solutions. IT and software development are crucial to its operations, with platform development outsourced by means of providing SDKs while Swasth focuses on conceptualisation, design and strategy.

FOSS at Swasth

Swasth's process for deciding on adoption of a software component prioritises FOSS as the default choice, unless a significant business or technical justification favours a proprietary alternative. Decisions often involve evaluating existing FOSS solutions or developing custom FOSS-based solutions.

> *'FOSS components are like LEGO blocks to be plugged into your technical architecture.'*
> **Abhishek Jain, CPTO, Swasth**

The organisation employs FOSS across its technology stack for production applications. For instance, Swasth's flagship platform, HCX, is built and operated using multiple FOSS components, including PostgreSQL for database management systems, Keycloak for authentication systems, HAPI FHIR programming framework, Redis for in-memory storage, and React for the front end.

While the organisation relies heavily on FOSS for product development, it notes that FOSS is not always ideal for internally used applications, especially for smaller organisations.

Swasth engages with FOSS in three key ways:

1. **Consumption:** using FOSS to build and operate platforms like HCX.
2. **Paying it forward:** leveraging FOSS to create additional open source contributions (e.g., HCX is released under the MIT License).

43 . 'History' (*Swasth*) <https://swasthalliance.org/history.html> accessed 3 February 2025.

44. 'Vision & Mission' (*Swasth*) <https://swasthalliance.org/vision-and-mission.html> accessed 3 February 2025.

45. ibid.

3. **Contributing in return:** addressing issues or improving components (e.g., resolving bugs in HAPI) and contributing those fixes to the community.

In addition to these technical contributions, Swasth engages in FOSS evangelisation to promote HCX adoption. Over the past two years, the organisation has presented its solutions at 20–25 conferences and forums for insurers in India, in addition to bi-weekly meetings within the HCX community. It has also participated in three or four international engagements. While the organisation has not conducted hackathons, it holds workshops where 20–25 community members gather to address specific issues collaboratively. They are also planning to advocate for recognition of their offering as a DPI.

FOSS dominates Swasth's technology stack with 13 software categories exclusively using FOSS solutions. Proprietary software is their next preferred choice with 6 categories comprised of only these solutions.

The organisation occasionally chooses proprietary tools for internal needs. For instance, they use Slack as their instant messaging tool since employees are comfortable with it and also since it is available to Swasth at zero cost on account of their being a non-profit. They also use ZohoBooks, and other HR and finance software.

'Organisations take comfort with proprietary software since they believe that in case of issues, there's always someone's neck to catch.'
Abhishek Jain, CPTO, Swasth

Benefits and Challenges

Conversations with Swasth suggested the following as the benefits of FOSS adoption:

1. Significant reductions in timelines and costs, According to them, without FOSS, the cost of developing HCX would have either increased four or five-fold, or the timeline would have tripled, i.e., the organisation achieved in three years what could have taken at least ten years with proprietary solutions.
2. Improved quality and aided development of innovative solutions.

However, Swasth also reported certain challenges in FOSS adoption. These include:

1. The need for sustained funding and contributions as critical elements for long-term viability of FOSS projects.
2. Many users fail to return the value they are deriving from FOSS in the form of contributing or extending code, or funding. This could potentially slow down the pace of technological evolution over time.

Open Healthcare Network

Note: OHC is an independent, community-maintained project, with eGov Foundation and PupilFirst funding the engineers that work on OHC projects.

Open Healthcare Network (OHC), formerly known as Coronasafe Network, is an initiative that took root during the early days of the COVID-19 pandemic in 2020. It was a collaborative effort to

address the emerging healthcare challenges through FOSS solutions. It has been financially supported by the eGov Foundation and ACT for Health, an organisation formed during COVID-19 crisis to support innovative solutions that addressed some of the primary challenges faced during the pandemic.

OHC is a volunteer-driven initiative, comprising of a team distributed across eGov Foundation, PupilFirst, and volunteers,[46] and supported by multiple entities including Kerala Startup Mission and the Michael and Susan Dell Foundation.[47] OHC's founders, who had prior experience in open source projects such as Kerala Rescue, took note of the threat posed by the pandemic and identified critical issues, such as hospital capacity management and patient data transfer, that could be addressed or mitigated by robust, scalable digital tools.

CARE was initially developed as an open source hospital capacity management tool to address the critical challenge of hospital resource-tracking during the COVID-19 pandemic. Given the high volume of cases, hospitals were overwhelmed, and patients were struggling to find facilities with available beds. CARE allowed hospitals to log in and report real-time numbers on available capacity, such as that of ICU beds, and provided live data to patients. This enabled better planning and resource allocation.

As the pandemic progressed, CARE expanded to include tools to seamlessly manage the shifting of patients within the different levels of the hospital based on symptom severity, and between primary, secondary and tertiary healthcare facilities. It helped optimise healthcare capacity by prioritising critical cases for higher-level care. The platform also addressed challenges of the pandemic, such as the inability to transfer paper-based medical records due to contamination risks and the unavailability of patient's kin to provide information about patient condition and diagnosis. CARE introduced digital solutions for securely transferring medical data and managing patient records. CARE was recognised as the 50th Digital Public Good by the United Nations, a testament to its robustness. Over time, the system has evolved to include an Electronic Medical Record (EMR) system.

As COVID-19 cases reduced, the project was upgraded to address other critical healthcare challenges, such as the 10BEDICU initiative led by Srikanth Nadhamuni. This initiative aimed to address the shortage of doctors in remote areas by setting up small ICUs connected to medical colleges, allowing specialist doctors to monitor and manage patients remotely. CARE was adapted to manage these networks, enabling patient monitoring and coordination between facilities. Currently, CARE is being expanded to include palliative care capabilities, focusing on simpler, location-based planning and coordination, in contrast with the detailed metrics required for ICUs.

FOSS at OHC

OHC has a small technology stack. The thought process behind it was to use the most common technologies that developers are familiar with to attract a wide community. The objective is to build solutions quickly, rather than spend time learning new technologies.

46. 'OHC Network Contributors' (*Open Healthcare Network*) <https://contributors.ohc.network/people> accessed 3 February 2025.

47. 'Open Healthcare Network: Transforming Healthcare with AI' (*Open Healthcare Network*) <https://ohc.network/> accessed 3 February 2025.

Beyond development, OHC actively contributes to the open source community by sharing its tools and fostering collaboration. For instance, the adoption of open source disaster management principles from previous projects influenced the architecture of CARE. By showcasing the advantages of open source collaboration, OHC exemplifies the potential of open source to address large-scale societal challenges. Additionally, OHC developed courses to teach students specific tech stacks (for example, Python, React and Django), so that they could contribute to the project in return. These courses were open source and free. They also conduct talks with various FOSS communities, such as GitHub Constellation, FOSS United, communities in Kerala, and Tinker Hub, a non-profit engaged in skilling and empowering students with FOSS education.

While there is no overlap or reuse of code among the solutions, the libraries are largely common.

Reasons for High FOSS Contributions

Apart from employing commonly used and familiar programming languages and encouraging contributions from students, the team also organised sessions with the contributors where doctors were invited to share how the solutions aided their work. The team also put out metrics such as the number of patients benefited, which enabled contributors to know the impact of their work. This impact being easily measurable was an added advantage which further motivated them to contribute.

Benefits and Challenges

OHC has experienced the following benefits from adopting open source:

> *'Had it not been for FOSS, our costs would have been higher by 10 or even 100-fold.'*
> **Vignesh Hari, Founding Member, OHC**

1. **Flexibility:** Open source approaches made it possible for the team to easily contribute while managing their full-time jobs. Contributors could chip in whenever they had time without the rigidity of a fixed schedule. The founders were of the view that in the absence of such an arrangement, their efforts would not have extended beyond two months.
2. **Support from FOSS enthusiasts:** The project being open source attracted contributions from people who genuinely cared about the problem being solved. Many contributors were students or professionals eager to pitch in because they were personally impacted or interested.
3. **Improved problem-solving:** Contributors brought fresh ideas and pointed out better ways to approach problems. They helped the team figure out easier and more efficient solutions.
4. **Impactful and fast development:** Projects like quarantine slot-booking systems and oxygen cylinder tracking were possible because of contributions from people who built and maintained parts of the system.
5. **Extensive collaborations:** By virtue of being open source, contributors pitched in from all parts of the world and added value. In a time of crisis when solutions had to be built quickly, this provided the added advantage of improved development velocity with contributions occurring at night as well.

'To this day we get requests from volunteers across the globe wanting to contribute to the project. At times, people are up at 5 a.m. to pick up issues.'

Vignesh Hari, Founding Member, OHC

6. **Government adoption:** The Kerala Rescue solution was adopted by the government in a matter of hours because of it being open source. CARE has also been adopted by around 11 government agencies in India over the last 4 years, and is actively used by around 7 Indian states.
7. **Shared responsibility:** The open nature fostered a sense of shared responsibility.

However, open source adoption has not been without its challenges. OHC's reported challenges include:

1. **Being overwhelmed by contributions**: In the early days, the influx of contributions was hard to manage, and the core team struggled to keep track of who was building what.
2. **Unclear tasks for contributors:** Contributors often didn't know where to start or what to work on, leading to confusion and delays.
3. **Misalignment of goals:** Without clear communication about the scope and needs of the project, contributors sometimes built features that weren't needed, creating more work to fix or having to scrap them.

'Clarity about requirements is utmost necessary.'

Vignesh Hari, Founding Member, OHC

4. **Short-lived projects:** Some projects were only relevant for a short period, like the quarantine slot-booking system, which became obsolete after the first few months. Maintaining enthusiasm for such projects was challenging. For instance, the development of Kerala Rescue for the Kerala flood lasted for around seven days, after which the contributions dropped.
5. **Time-intensive management:** Writing crystal-clear feature requests and issues, reviewing contributions, and guiding contributors took a lot of effort from the core team. In the initial period, this was not a challenge since contributions were largely from known persons whom the project originators could connect with.
6. **Coordination challenges with global contributors:** Managing contributions from people across different time zones, languages, and motivations added to the complexity.
7. **Government adoption:** Making governments adopt open source tooling is a challenge as it opens up questions on who will provide support, customisations, etc.

SUMMARY OF FINDINGS

Decision Factors

As is evident from the diverse case studies discussed, organisations follow different assessment approaches to decide on the type of software component to use for their requirements, and different factors guide their decisions. While in some organisations, strict assessment approaches are observed, in many other instances organisations are seen taking a flexible approach towards assessment, guided more by pragmatism than idealism.

Organisations generally use one or more of the following component types:

1. FOSS.
2. A managed FOSS solution.
3. In-house developed software.
4. Proprietary commercial software.

The existing literature suggests that in a scenario wherein organisations decide on adopting an FOSS component, they generally evaluate available solutions based on a set of criteria. These criteria can be broadly divided into two types: technical factors and business or organisational factors.

Technical factors relate to the technical aspects of the working of the software component. Scholars working in the area, particularly two studies in 2021[48] and 2022,[49] have identified the following technical factors:

1. Fulfilment of functional requirements.
2. Compatibility with existing systems.
3. Integration feasibility and adaptation required for integration.
4. Usability/ ease of usage.
5. Maintainability of the component and the system it is integrated into.
6. Compatibility of licensing terms (if used in a product) of the component with that of the product.
7. Extent of possible customisation.
8. Adherence to standards.
9. Security.

Business and organisational factors relate to how the software satisfies or caters to needs of the business and constraints of the organisation. Below is a list of factors identified from the existing literature:[50]

48. Rohan Patnaik and Rahul Kumar, 'Factors Affecting Open Source Software Adoption' (Centre for Digital Transformation, Indian Institute of Management Ahmedabad 2021) <https://www.iima.ac.in/sites/default/files/2023-06/Adopting%20Open%20Source%20Software%20Report%202021_7Feb22.pdf> accessed 2 March 2025.

49. Simon Butler and others, 'Considerations and Challenges for the Adoption of Open Source Components in Software-Intensive Businesses' (2022) 186 Journal of Systems and Software 111152 <https://linkinghub.elsevier.com/retrieve/pii/S0164121221002442> accessed 23 January 2025.

50. Patnaik and Kumar (n 48); Butler and others (n 49).

1. Economic or Financial cost effectiveness/ TCO.
2. Time required for adoption.
3. Extent of business process re-engineering required.
4. Availability of support (community support or as a managed service).
5. Availability of documentation.
6. Availability of training (if deemed necessary).
7. Ownership and governance of the component's parent project.
8. Extent of use in the industry.
9. Presence under active development.

Among all the organisations discussed as case studies in this report, not too surprisingly, there is a clear preference for FOSS, with most of them either favouring or evaluating FOSS components as one of their choices. As indicated earlier, our case studies suggest that organisations in India also engage in an evaluation process during decision-making for software components, though a strict step-by-step assessment based on all the factors was not found in most instances.

It is interesting to observe that managed FOSS solutions are explored for use cases wherein the organisations do not wish to spend resources towards operating and maintaining the solution, or when the organisations are of the opinion that doing so would distract them from their core business. In most instances, these are typically software needs apart from the core product, or applications for internal use.

The reasons mentioned by organisations for using commercial proprietary software components are also varied. They are either used as a last resort or when the organisations face time constraints. Other specific edge cases include requirements for high performance, unusual feature sets, regulatory compliance purposes, and the maturity and sophistication of the solution offered by the concerned proprietary software.

It is also noticed that organisations opt for in-house development of software primarily in scenarios wherein the solution is a core or integral part of their IT system/ product, or when the requirements are simple or straightforward, requiring minimal resources.

Not too surprisingly, the case studies of organisations delivering software and IT services indicate that their technology stack is largely determined by client preferences. Despite this, there is a clear preference for FOSS.

Cost-related considerations, particularly TCO, were also observed as playing a key role in the evaluation processes of different organisations, particularly start-ups, along with other considerations.

Large and medium-sized organisations, alongside start-ups, reported analysing the highest number of technical factors. The fulfilment of functional requirements, and the ease with which the component and system it is integrated into can be maintained, emerged as the two most important factors mentioned by the organisations. This was followed by the following technical factors:

1. Feasibility of integration of the component with existing/ external systems.
2. Ease of usage of the component.
3. Extent of customisation possible.
4. Favourable licensing terms (this refers to a preference for permissively licensed components).

Start-ups, followed by non-profits, reported considering the largest number of business or organisational factors. In this aspect, cost effectiveness of the component and the presence of an active community surrounding the project were the most important factors for organisations when evaluating FOSS components. The availability of support, either from the community or as a managed service, featured as the next most important factor followed by the extent of use of the component in the industry.

The presence of a community being evaluated as a factor (which is different from support from the community) emerged as a new finding, indicating that the community around an FOSS project fulfils a purpose beyond providing technical support. It could be an indicator for popularity of the project or could reflect the value it provides, which in turn could attract a larger community.

FOSS Alternatives

All of the organisations that responded to the question about the presence of proprietary or software developed in-house in their stack, reported using either proprietary, licensed or SaaS applications for various purposes.

We observe that proprietary software is largely employed for two main purposes, namely, email services, and/ or workspace collaboration tools and internally used applications.

The reasoning provided for this included the lack of equivalent open source alternatives due to the complexity of the application, and attempts by large email providers to maintain a leverage in the market by disincentivising emails not originating from their servers.

As indicated earlier, many organisations also use proprietary software or SaaS applications for internal needs, i.e., solutions used within the organisation by employees or for its operations. This includes instant messaging, HR, finance, payroll, ERP, etc. The reasoning adopted in this regard included:

1. Organisations do not have the capacity, or do not want to allocate technical resources for development of non-strategic and ancillary software.
2. Difficulties in supporting and maintaining these applications.
3. Preference and comfort of employees, particularly non-technical staff.

Interestingly, in-house development was opted by one organisation in our study, owing to ambiguity in the licensing of the FOSS component that was planned to be deployed. Some other reasons listed during our conversations for choice of proprietary software over FOSS include:

1. Better performance, security and advanced functionality (in case of specific applications).
2. Availability of cost-effective and customised solutions.
3. Compliance requirements.
4. Integration with vendor software.

We may also add here that while most organisations rely on proprietary email and instant messaging applications, our study also indicates the presence and use of open source alternatives, namely Zimbra and Mattermost respectively, by some organisations.

FOSS Consumers or FOSS Contributors?

Almost all organisations studied as part of this report indicated that they are consuming FOSS for their diverse requirements. However, the same cannot be said about their extent of FOSS contributions. Only two third of them reported releasing software developed by them as FOSS. But on the positive side, it has to be specifically mentioned here that 3 of them, mostly start-ups, released even their core offering as open source.

There could be multiple reasons that prevent active code contributions in this regard, and future studies may try to focus on this dimension. Our conversations indicate that even among those who expressed interest to contribute, some were unable to do so owing to limited resources and lack of regulatory clarity on the same.

It is observed that though some organisations did not directly contribute their code to FOSS, they engaged in fixing bugs in FOSS projects, which is certainly an important form of contribution to FOSS development. Only two large organisations, both avid adopters of FOSS, contributed financially to FOSS in some form. Some also contributed to the development of documentation.

Benefits

FOSS offers several advantages over proprietary software by virtue of its source code being accessible to view, modify, replicate and use. The ability to modify the code enables customisation and incremental development, allowing one to build on existing software. This reduces duplication of effort, promotes code reuse[51] and lowers development costs. The openness of code facilitates learning, enabling one to examine and understand the working of the software.[52]

It also aids in the development of better-quality, reliable and secure software. Since the code can be examined by a large number of developers, issues and vulnerabilities are likely to be discovered sooner and fixed ('given enough eyeballs, all bugs are shallow'[53]). Its modular architecture allows a large number of contributors to concurrently develop and debug the software.[54] This distributes and thereby lowers the cost of maintenance as compared to when the same activity is performed by a single organisation (as in the case of proprietary software).[55]

Proprietary software users are impacted by 'vendor lock-in', a scenario in which the consumer is unable to switch to a different software offered by another vendor owing to the prohibitive costs involved. FOSS, on the other hand, is built on open standards that facilitate interoperability among software components.

51. Kenneth Wong and Phet Sayo, 'Free/Open Source Software - A General Introduction' (*UNDP-APDIP*, 2004) <https://www.unapcict.org/sites/default/files/2019-01/FOSS%20-%20General%20Introduction.pdf> accessed 2 March 2025.

52. Shaikh and Vaast (n 8).

53. Eric S Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary* (O'Reilly Media 1999) 30.

54. Kogut and Metiu (n 9).

55. Lerner and Tirole (n 6).

Other key benefits experienced by organisations adopting FOSS, as highlighted by two surveys in 2023[56] and 2024[57] include:

1. Improved cost effectiveness through savings on licensing fees. Organisations also experienced lower TCO, which includes all costs incurred in operating the software such as hardware, personnel, training etc.
2. Improved software development velocity resulting from readily available functionality.
3. Faster rollout of software enhancements, releases and patches.
4. Stability derived from community support.
5. Reliability from reduced errors and bugs.
6. Enhanced security.
7. Impetus to innovation.

These benefits, highlighted in existing literature, were largely echoed by the organisations in our study.

Across all the categories surveyed, most of the organisations listed lower TCO or cost savings as a benefit of FOSS. This validates the finding that lower costs are a key factor driving the adoption of FOSS. This was followed by the ability to customise and localise the software being reported as a benefit by many organisations.

Some organisations reported improved development velocity resulting from incremental development on existing software, freedom from vendor lock-in associated with proprietary software vendors, and enhanced security of FOSS as experienced benefits. Some organisations also reported that FOSS provided them an impetus to innovation, either by allowing them to build differentiated products and services or through improved problem-solving as a result of new ideas and better solutions arising from contributors.

Interestingly, while working with FOSS requires an organisation to possess requisite technical capacity, six organisations reported this as a benefit rather than a limitation since it allowed them to develop their own internal technical and engineering capabilities.

FOSS systems being highly scalable and providing organisations enhanced visibility and control over their systems were reported as its benefits by some organisations, though most of the literature we had reviewed had not highlighted these benefits. An overview of the benefits reported is seen in Table 2.

56. Chesbrough (n 10).

57. OpenLogic, '2024 State of Open Source Report - Open Source Usage, Market Trends, & Analysis' (2024) <https://www.openlogic.com/sites/default/files/pdfs/report-ol-state-of-oss-2024.pdf> accessed 28 January 2025.

Table 2. Top Benefits of FOSS

Sl.No.	BENEFIT	ORGANISATIONS THAT REPORTED THE BENEFIT
1	Lower TCO/ improved cost effectiveness through savings on licensing fees	Zerodha, Razorpay, [A private bank], Pocket ATM, ThoughtWorks, Tech4Good, NxtGen, Swasth, KG Hospital, OHC, PupilFirst, KITE, Kalvium
2	Allows customisation as per requirement	Zerodha, Kalvium, KG Hospital, Tech4Good, KITE, NxtGen, VGLUG
3	Development of internal technical capacity	Zerodha, PupilFirst, VGLUG, KITE, NPCI
4	Improved development velocity resulting from readily available functionality	Razorpay, [A private bank], Pocket ATM, NxtGen, Swasth
5	Freedom from vendor lock-in and associated costs	Zerodha, Kalvium, KITE, ThoughtWorks
6	Enhanced security	[A private bank], PupilFirst, KITE, Thought-Works

Challenges

While there are many benefits derived from FOSS, it is undeniable that organisations working with FOSS also experience some challenges. For example, most FOSS lacks user-friendly interfaces and comprehensive documentation.[58] The unavailability of associated technical support also forces users to rely on the FOSS community for solutions in case of issues with the software. Users also have to keep track of, and apply security patches and updates. Their timely releases are also an important factor.

Performing these tasks requires good internal technical capacity, i.e., personnel with skills, experience, and proficiency in FOSS technologies. The cumulative costs incurred in the above activities can at times result in a situation where the FOSS solution is costlier than the corresponding proprietary software.[59]

FOSS projects also risk becoming end of life (EOL).[60] This happens when community members or the project originators do not see value in continuing to contribute to the project. In such a scenario, the user has to either maintain them on their own or switch to a different software.

58. Wong and Sayo (n 51).

59. Shahron Williams Van Rooij, 'Adopting Open-Source Software Applications in U.S. Higher Education: A Cross-Disciplinary Review of the Literature' (2009) 79 Review of Educational Research 682 <https://journals.sagepub.com/doi/10.3102/0034654308325691> accessed 2 March 2025.

60. OpenLogic (n 57).

Also, disagreements among the community members can cause a project to split into competing versions, otherwise known as forking,[61] which again requires the user to make a choice regarding the specific version to adopt.

Some of the other challenges of FOSS adoption which could be observed from the surveys mentioned in the benefits section include:

1. Unavailability of training in FOSS technologies and associated costs.
2. Cost of switching from proprietary solutions.
3. Legal uncertainty regarding FOSS licensing.
4. Need to undertake testing of FOSS solutions since they lack certification.
5. Revenue loss due to free-riding competitors as certain licenses require many modifications or enhancements to FOSS, to be released as open source.[62]

While organisations analysed in this study echoed many of the challenges identified in the existing literature, some also reported unique and specific challenges of working with FOSS in their own use contexts.

While FOSS often benefits from community scrutiny and rapid patching, managing dependencies and vulnerabilities in its components, which are potentially exploitable by malicious actors, is a serious challenge.[63] One effective approach to address this is through vulnerability tracking, facilitated by the generation of SBOMs (Software Bill of Materials) and the use of SCA tools.[64] In October, CERT-In released technical guidelines on SBOM[65] for its adoption in the Indian public sector, government, essential services, and organisations involved in the software export and software services industry.

As highlighted by Abhishek Jain of Swasth Alliance, dependency and vulnerability checks, while ideally performed for 100% of the code, are practically conducted more frequently on a need-basis, particularly when open source components are used for data privacy and security aspects, and are performed at broad intervals. This selective approach can be compared to a security guard who thoroughly checks visitors the first few times but reduces subsequent inspections unless specific issues arise. This is the typical approach in the industry as illustrated by the March 2024 SSH vulnerability incident.

Some of the surveyed organisations shared their measures to manage dependencies and vulnerabilities:

1. Automated dependency scanning tools.
2. DevSecOps cycles with vulnerability scans.

61. Kogut and Metiu (n 9).

62. Robert M Sauer, 'Why Develop Open-Source Software? The Role of Non-Pecuniary Benefits, Monetary Rewards, and Open-Source Licence Type' (2007) 23 Oxford Review of Economic Policy 605 <https://www.jstor.org/stable/23606749> accessed 23 January 2025.

63. OpenLogic (n 57).

64. '13 Open Source Software Security Risks' (*SentinelOne*, 5 November 2024) <https://www.sentinelone.com/cybersecurity-101/cybersecurity/open-source-software-security-risks/> accessed 27 January 2025; CERT-In, 'Technical Guidelines on Software Bill of Materials (SBOM)' <https://www.cert-in.org.in/PDF/SBOM_Guidelines.pdf> accessed 3 January 2025.

65. CERT-In (n 64).

3. SBOM to analyse licenses of all software.
4. Keeping track of the changelog of the project.
5. Information security practices based on the NIST Cybersecurity framework with multi-layer architecture.
6. Monthly authenticated scans.

Additionally, some organisations also reported using GitHub and community-driven auditing to address these challenges. The top challenges are listed in Table 3, alongside the organisations that reported each one. Interestingly, the following specific challenges identified from the literature were not highlighted by any organisation in our sample:

1. Revenue loss due to free-riding competitors.
2. Testing required due to lack of certification for FOSS.
3. Switching costs from proprietary solutions.

Table 3. Top Challenges of FOSS

SL.NO.	CHALLENGE	ORGANISATIONS THAT REPORTED THE CHALLENGE
1	Tracking and applying patches and updates	Zerodha, Razorpay, [A private bank], PupilFirst, KG Hospital
2	Availability of personnel with skills in FOSS technologies	[A private bank], Tech4Good, Dhiway, NxtGen
3	Lack of community support	Zerodha, Razorpay, Pocket ATM, Tech4Good, PupilFirst
4	Performing maintenance and support	Remiges, Kalvium
5	Integration/ compatibility with existing IT systems	KITE, Tech4Good, Pocket ATM
6	Lack of contributions to FOSS	Swasth, Dhiway, OHC
7	No revenue model to sustain FOSS projects	ThoughtWorks, Swasth, PupilFirst
8	Troubleshooting issues with FOSS	NPCI, NxtGen

IP and Licensing

Five types of IP are relevant in the context of software in many jurisdictions: trade secret, copyright, patent, designs, and trademark.[66] However, not all these forms of IP protection are utilised by individuals and organisations uniformly across software, as their usage depends on the developers' preferences as well as on applicable national laws.

The key critique of the traditional IP protection system in the context of FOSS is that it inherently conflicts with basic FOSS principles. For instance, trade secrets protect confidential information,and this is against the basic ethos of open source. In fact, the FSF explicitly regards trade secrets as a GPL (General Public License) violation.[67]

However, it needs to be noted that the open source community has also tried to leverage some forms of IP protection in a unique manner to protect themselves at times. For instance, trademark is a sign used for protecting source identification, building consumer trust, and thereby also building a brand, in the long term. So, while giving up other forms of IP protection such as copyright, patents and trade secrets, an organisation may rely on their trademark-related rights to maintain the competitive edge in the market. While the term 'open source' cannot be trademarked because of its descriptive character,[68] many members of the open source community can be now seen leveraging certification marks such as 'OSI Certified',[69] administered by the OSI, to signal compliance with open source principles.

The organisations surveyed for the study showed an awareness of issues related to licensing and patents but reported minimal direct challenges with IP protection. This could be partly due to the absence of any major litigations in India in the area of FOSS. However, it also needs to be added here that many respondents in our study noted the need for careful consideration of licensing terms and conditions, thereby indirectly indicating consciousness about the underlying IP issues.

Challenges with regard to patent protection

Patents can grant exclusive control over inventions, and their application in the realm of software has always been a contentious issue, particularly in the context of FOSS principles.[70] Richard Stallman equates software patents to land mines, where each design decision risks legal repercussions.[71] Open source proponents seek to revisit patent jurisprudence in the context of software programs altogether, citing it as an 'undeserved reward'.[72] They point to issues such as the highly collaborative

66. Vikrant Narayan Vasudeva, *Open Source Software and Intellectual Property Rights* (Kluwer Law International 2014).

67. 'Frequently Asked Questions about the GNU Licenses - GNU Project - Free Software Foundation' (*GNU Operating System*) <https://www.gnu.org/licenses/gpl-faq.en.html#TradeSecretRelease> accessed 22 January 2025.

68. Eric S. Raymond, 'Announcement of "OSI Certified" Open Source Mark' (*Open Source Initiative*, 16 June 1999) <https://opensource.org/pressreleases/certified-open-source-php> accessed 21 January 2025.

69. ibid.

70. Andres Guadamuz, 'Legal Challenges to Open Source Licenses' (2005) Script-ed <https://era.ed.ac.uk/handle/1842/2272> accessed 17 December 2024.

71. Richard Stallman, 'Fighting Software Patents - Singly and Together' (*GNU Operating System*) <https://www.gnu.org/philosophy/fighting-software-patents.en.html> accessed 26 January 2025.

72. Kirk Rowe, 'Why Pay for What's Free?: Minimizing the Patent Threat to Free and Open Source Software' (2008) 7 The John Marshall Review of Intellectual Property Law <https://repository.law.uic.edu/ripl/vol7/iss3/9> accessed 3 March 2025. As cited in Vasudeva (n 66).

and incremental character of innovations in the area of software, the challenges associated with prolonged patent terms, and the high potential for abuse in the patent system.[73]

The open source community has developed some strategies to mitigate some of these challenges.[74] They include integrating patent clauses in licenses, open patent movement, patent promises,[75] creation of patent pools,[76] creating prior art databases,[77] promotion of rigorous prior art examination and defensive publication strategies.[78]

However, patents still pose considerable challenges for FOSS by virtue of factors such as the high cost of patent litigation (making it difficult for most FOSS projects to defend themselves against infringement claims) and the existence of patent trolls who seek patents merely for the purpose of extorting money from others. The large number of patent applications filed in many of the jurisdictions and the manner in which patent applications are often drafted (concealing the fact that the invention in question is a software) poses additional challenges for FOSS community.

As highlighted by Shuvam Misra of Remiges, patents remain a persistent concern for creators of both open and closed-source software, necessitating proactive risk mitigation strategies. The increasing awareness of this threat within the FOSS community has led organisations to offer protection mechanisms. For instance, Red Hat assumes liability for lawsuits related to their offerings, shielding customers from legal consequences by taking responsibility for contesting such cases in court.

The open source community has also raised concerns about the long-term impact on innovation, particularly for smaller developers and companies.[79] In some instances, entire projects would have to be halted due to a minor infringement claim from the patent-holders of a proprietary program. While cross-licensing (mutual exchange of patent licenses rights between two or more parties) is a mitigation tactic, it is understood to mostly benefit large corporations as very few FOSS projects have patents to trade.[80]

Moreover, empirical research suggests that IPR enforcement actions can negatively impact FOSS projects by decreasing user interest and developer activity.[81] For example, in the context of the *SCO v. IBM* law suit, one study points out that the user interest (measured by project downloads) showed substantial decline following the initiation of the suit. More specifically, after the lawsuit was filed, FOSS projects having a high technology overlap

73. Malcolm Bain and P McCoy Smith, 'Patents and the Defensive Response' in Amanda Brock (ed), *Open Source Law, Policy and Practice* (Oxford University Press 2022) <https://doi.org/10.1093/oso/9780198862345.003.0010> accessed 23 January 2025; Vasudeva (n 66).

74. See Bain and McCoy Smith (n 73); Vasudeva (n 66).

75. Red Hat <https://www.redhat.com/en/about/patent-promise> accessed 3 March 2025. See also: Google; IBM, Microsoft.

76. For instance, Open Invention Network and License on Transfer Network.

77. Projects like Open Source as Prior Art and Peer to Patent.

78. Platforms like Technical Disclosure Commons.

79. Vasudeva (n 66).

80. Bain and McCoy Smith (n 73). Vasudeva (n 66).

81. Wen Wen, Chris Forman and Stuart JH Graham, 'Research Note: The Impact of Intellectual Property Rights Enforcement on Open Source Software Project Success' (2013) 24 Information Systems Research 1131.

with the concerned software witnessed around 15-16% decline in monthly downloads, as compared to a control group. Data also indicates a substantial decline in developer activity and illustrates that in FOSS projects with high technology overlap with the concerned software, FOSS projects experienced a 45% decrease in developer activity in comparison to a control group.[82]

However, as indicated earlier, during our interactions with different organisations as part of this study, most organisations did not report facing any direct software patent-related challenges. Only one organisation reported a challenge they faced relating to a cryptographic library patent. It needs to be added that some organisations expressed concerns about patents being a persistent issue, emphasising the need for vigilance. In other words, even if direct negative experiences in this area are limited in the case of Indian organisations (probably due to the restrictions on software patents under the Patents Act, 1970 of India), they may still be causing a chilling effect in the area.

Copyright Law

Copyright law plays the most important role in the context of software. It treats software as a 'literary work', protected under Indian copyright law. Some of the open movements like Creative Commons have used the copyright framework in creative ways to ensure wider dissemination of software and other subject matters covered under copyright law. As copyright protection is automatic (no registration is required for getting protection) in most jurisdictions, they achieve the objective of broader dissemination of such works by providing easy-to-use and easy-to-understand licenses that allow a broad range of activities. In some instances, this could mean relinquishing the entire copyright, and in most instances, the developers would only retain the specific right they wish to retain. Attribution is one of such rights.

However, due to the fact that there are different licensing options currently available, there is also considerable divergence on licensing terms, despite agreement on the OSS philosophy. To reduce conflict and promote the growth of OSS, groups like the OSI[83] and the FSF[84] have set definitional standards for these licenses. The different licenses can be classified in various ways, such as in terms of control (academic vs permissive vs partially closable vs reciprocal licenses)[85] and in terms of their functional differences (permissive vs restrictive vs highly restrictive licenses).[86]

Among the different open source license options, it is observed that the two licenses[87] most often used by developers (based on the number of unique pushes to GitHub) in India for 2024 (Q3) are MIT[88] and Apache-2.0.[89]

---

82. ibid.

83. 'Licenses' (*Open Source Initiative*) <https://opensource.org/licenses> accessed 28 December 2024.

84. 'FSF Licensing & Compliance Team' (*Free Software Foundation*) <https://www.fsf.org/licensing/> accessed 26 January 2025.

85. In terms of control over the software. See Van Lindberg, *Intellectual Property and Open Source* (O'Reilly 2008).

86. Maryna Manteghi, 'Understanding Open Source and Free Software Licensing Mechanism: A Close Review of the Alternative Approach to Traditional Notions of Software Licensing' (2017) SSRN Electronic Journal <https://www.ssrn.com/abstract=3082313> accessed 12 December 2024.

87. 'IN | GitHub Innovation Graph' <https://innovationgraph.github.com/economies/in#git-pushes> accessed 3 March 2025; 'Innovationgraph/Data/Licenses. Csv at Main · Github/Innovationgraph' (*GitHub*) <https://github.com/github/innovationgraph/blob/main/data/licenses.csv> accessed 3 March 2025.

88. 'MIT License' <https://mit-license.org/> accessed 31 January 2025.

89. 'Apache License, Version 2.0' <https://www.apache.org/licenses/LICENSE-2.0> accessed 31 January 2025.

Table 4 compares some of the popular open source licenses, in terms of the nature of rights granted to the users. It uses the information provided in the Choose A License Appendix[90] and the Open Comparison Grid, released by the Center for Technology Transfer and Enterprise Creation, Carnegie Mellon University.[91]

Table 4. Popular Open Source Licenses by Nature of Rights Granted

LICENSE	MIT	APACHE LICENSE 2.0 (AL 2.0)	MOZILLA PUBLIC LICENSE 2.0 (MPL 2.0)	GNU LESSER GENERAL PUBLIC LICENSE v3.0 (LGPL)	GNU GENERAL PUBLIC LICENSE v3.0 (GPL)	GNU AFFERO GENERAL PUBLIC LICENSE v3.0 (AGPL)
PERMISSIONS						
Commercial use for licensed material and derivatives	■	■	■	■	■	■
Distribution	■	■	■	■	■	■
Modification	■	■	■	■	■	■
Private use	■	■	■	■	■	■
Express grant of Patent rights from contributor to recipient		■	■	■	■	■
CONDITIONS						
Disclosure of source code when distributing the software			■	■	■	■
Copy of license and copyright notice	■	■	■	■	■	■

90. GitHub, Inc. 'Licenses' (*Choose a License*) <https://choosealicense.com/licenses/> accessed 23 January 2025.

91. Carnegie Mellon University CTTEC, 'Open Source License Comparison Grid' <https://www.cmu.edu/cttec/forms/opensourcelicensegridv1.pdf> accessed 3 March 2025.

	MIT	AL 2.0	MPL 2.0	LGPL	GPL	AGPL
Users who interact with the software via network are given the right to receive a copy of the corresponding source code						■
Modification to be released under the same license; in some cases similar or related licenses may be used			■ (Modification of files)	■ (This condition may not apply to works that use the licensed material as a library)	■	■
Indicate changes made to the code		■		■	■	■
LIMITATIONS						
Software without warranty, and no liability for damages	■	■	■	■	■	■
Explicitly states no grant of trademark rights		■	■			

However, using licenses is not without its challenges. Some of the specific challenges in the FOSS context are:[92]

1. License proliferation (excessive number of open source licenses) leading to compliance issues, confusion and incompatibility between licenses hindering collaboration and code reuse.[93]
2. The possibility of different jurisdictions interpreting licenses inconsistently, leading to compliance challenges.
3. Enforcement costs, particularly for smaller organisations.

'The more liberal the licensing, the easier the choice.'
Abhishek Jain, CPTO, Swasth

92. See Vasudeva (n 66); Amanda Brock (ed), *Open Source Law, Policy and Practice* (2nd edn, Oxford University Press 2022) <https://academic.oup.com/book/44727> accessed 22 January 2025; Lindberg (n 85) ch 7; Noam Shemtov and Ian Walden (eds), *Free and Open Source Software: Policy, Law, and Practice* (1st edn, Oxford University Press 2013).

93. Robert Gomulkiewicz, 'Open Source License Proliferation: Helpful Diversity or Hopeless Confusion?' (2009) 30 Washington University Journal of Law & Policy 261.

Our case studies also show a strong preference for permissive licenses like Apache and MIT. It needs to be added that many of the organisations explicitly expressed that they avoid GPL licenses, citing concerns about its incompatibility with their licenses and its restrictive licensing conditions. One organisation clarified by adding that the viral nature of licenses such as the GNU GPL is a challenge for their FOSS adoption.

Additionally, out of the 12 organisations in our sample who released the software they developed as open source, we observe that there is roughly 50:50 split between permissive and restrictive licensing.[94]

'Software should be open so that others can build on top of what I have done, and I can build on what others have done.'
Shuvam Misra, Founder-Chairman, Remiges

During our interactions, it was observed that licensing conditions play a role in the decision-making process and six organisations reported instances wherein licensing conditions led them to reject an FOSS component. While five of them mentioned restrictive licensing conditions as the reason, one organisation reported ambiguity in license terms, which can lead to increased overhead, as a legal team is required to provide clarity.

It may also be highlighted here that one healthcare organisation emphasised prioritising quality and suitability of the solution for patient care over a sub-optimal solution to avoid licensing fees, indicating an instance wherein operational priorities outweigh licensing concerns.

Shuvam Misra of Remiges mentioned a notable example of the effect of viral license, MySQL's transition of its client libraries from LGPL to GPL following its acquisition by Oracle.

With this change, Oracle database client libraries, which are essential for connecting applications to databases, became subject to GPL terms, requiring developers to release their application as derivative works under GPL. This shift created significant challenges for enterprises relying on MySQL, as it pressured them to purchase commercial support agreements to avoid open sourcing their proprietary software.

Distinction between GPL and LGPL: While GPL applies its licensing requirements to all derivative works, LGPL typically applies to shared libraries without extending those requirements to application code. This differentiation has historically made LGPL the preferred licensing for client libraries.

Vendor-driven FOSS projects are increasingly facing competition from SaaS providers. It is reported that these organisations engage in free riding i.e., the use of OSS without contributing anything in return. This has forced OSS vendors such as MongoDB, Elastic, and Redis Labs to modify their licenses in such a manner that restrict the use of the software by third parties, or require them to pay fees or share their modifications in turn, thereby making the project less open.[95]

94. While some of them mentioned the licenses used, for others, the same has been taken from their individual projects as released on GitHub.

95. John Walsh, 'What's Driving Changes in Open Source Licensing?' *(DevOps.com*, 8 March 2024) <https://devops.com/whats-driving-changes-in-open-source-licensing/> accessed 26 January 2025.

Most organisations in our study are aware of potential licensing risks that may arise in the future due to changes in licensing of a software project. Seven organisations outlined specific risk mitigation strategies they have adopted, with some employing more than one method. The outlined strategies include issuing advisories and alternative suggestions, adopting older versions, using the latest and best fork of the project, implementing modular architectures, and conducting regular audits using SBOM.

In a nutshell, despite minimal direct conflicts with IP and licensing, organisations studied as part of this report exercise consistent caution and adopt risk management practices. Some of them also anticipate a rapid change in licensing-related challenges due to AI-driven code development.

'As the world moves to AI-driven software development, testing or code review and documentation, intellectual property [rights] for software [will] become obsolete in the next three to five years.'
Dilip Asbe, MD, CEO, NPCI

Software Stack

Eight of the organisations analysed in this report have provided their software stack, stating the extent to which different categories of software are used in their organisation. The details regarding the same can be accessed in Table 5. To maintain confidentiality, we have removed the names of the organisations and sector details.

Table 5. Software Stack Usage Across Organisations

CATEGORY	STARTUP			STARTUP		
ORGANISATION	S1			S2		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
Operating Systems	30%	70%	0%	0%	100%	0%
Web Servers	0%	100%	0%	0%	100%	0%
Middleware				50%	50%	0%
Cloud Native Software	100%	0%	0%	70%	30%	0%
Development Framework	0%	100%	0%	0%	100%	0%
Programming Languages	0%	100%	0%	0%	100%	0%
Database Management System	100%	0%	0%	0%	100%	0%
Data Visualisation						
Messaging and Queueing						
Infrastructure Automation				50%	50%	0%
Observability & Monitoring	100%	0%	0%	0%	100%	0%
Access Control	100%	0%	0%			
Networking	100%	0%	0%			

ORGANISATION	S1			S2		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
CI/ CD	0%	100%	0%	50%	50%	0%
AI/ ML						
Security Tools				0%	100%	0%
ERP						
CRM	100%	0%	0%	0%	0%	100%
CMS				0%	0%	100%
Ticketing/ Workflow Management System	100%	0%	0%	0%	0%	100%
MIS						
LMS				100%	0%	0%
Accounting & Finance	100%	0%	0%	100%	0%	0%
HR and Payroll	100%	0%	0%	100%	0%	0%
Project Management	100%	0%	0%	100%	0%	0%
Communication (Email/ Instant Messaging or Email/ Office Suite)	100%	0%	0%			
API Integration	100%	0%	0%			

CATEGORY	STARTUP			STARTUP		
ORGANISATION	S3			S4		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
Operating Systems	10%	90%	0%	0%	100%	0%
Web Servers	0%	90%	10%	0%	85%	15%
Middleware	10%	45%	45%	0%	20%	80%
Cloud Native Software	80%	0%	20%	0%	30%	70%
Development Framework	10%	80%	10%	0%	100%	0%
Programming Languages	0%	100%	0%	0%	100%	0%
Database Management System	30%	70%	0%	0%	100%	0%
Data Visualisation						
Messaging and Queueing						
Infrastructure Automation	0%	50%	50%	0%	80%	20%
Observability & Monitoring	50%	50%	0%	0%	100%	0%
Access Control	0%	100%	0%	0%	100%	0%
Networking	50%	50%	0%	30%	60%	10%

ORGANISATION	S3			S4		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
CI/ CD	50%	50%	0%	0%	80%	20%
AI/ ML	0%	70%	30%	0%	90%	10%
Security Tools	0%	50%	50%	0%	100%	0%
ERP	100%	0%	0%			
CRM	50%	50%	0%	100%	0%	0%
CMS	0%	50%	50%	0%	100%	0%
Ticketing/ Workflow Management System	0%	100%	0%	100%	0%	0%
MIS	0%	0%	100%	100%	0%	0%
LMS	0%	0%	100%			
Accounting & Finance	100%	0%	0%	100%	0%	0%
HR and Payroll	100%	0%	0%	100%	0%	0%
Project Management	30%	50%	20%	0%	100%	0%
Communication (Email/ Instant Messaging or Email/ Office Suite)	100%	0%	0%	0%	100%	0%
API Integration						

CATEGORY	LARGE			NON-PROFIT		
ORGANISATION	L1			NP1		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
Operating Systems	1%	99%	0%	0%	100%	0%
Web Servers	0%	100%	0%	0%	100%	0%
Middleware	0%	90%	10%			
Cloud Native Software	0%	100%	0%	0%	100%	0%
Development Framework	0%	100%	0%	0%	100%	0%
Programming Languages	0%	100%	0%	0%	100%	0%
Database Management System	0%	100%	0%	0%	100%	0%
Data Visualisation				0%	100%	0%
Messaging and Queueing				0%	100%	0%
Infrastructure Automation	0%	100%	0%	0%	100%	0%
Observability & Monitoring	0%	100%	0%	0%	100%	0%
Access Control				0%	100%	0%
Networking				0%	100%	0%

ORGANISATION	L1			NP1		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
CI/ CD	0%	100%	0%	100%	0%	0%
AI/ ML				0%	100%	0%
Security Tools	0%	100%	0%			
ERP	0%	100%	0%			
CRM	0%	50%	50%			
CMS	0%	100%	0%			
Ticketing/ Workflow Management System	0%	90%	10%	100%	0%	0%
MIS						
LMS						
Accounting & Finance				100%	0%	0%
HR and Payroll				100%	0%	0%
Project Management				100%	0%	0%
Communication (Email/ Instant Messaging or Email/ Office Suite)	70%	30%	0%	100%	0%	0%
API Integration						

CATEGORY	NON-PROFIT			MEDIUM		
ORGANISATION	NP2			M1		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
Operating Systems	30%	70%	0%	5%	95%	0%
Web Servers				5%	95%	0%
Middleware				5%	95%	0%
Cloud Native Software						
Development Framework	0%	100%	0%	5%	95%	0%
Programming Languages	0%	100%	0%	5%	95%	0%
Database Management System				20%	80%	0%
Data Visualisation	0%	100%	0%			
Messaging and Queueing						
Infrastructure Automation				5%	95%	0%
Observability & Monitoring				20%	80%	0%
Access Control				30%	70%	0%
Networking				0%	100%	0%

ORGANISTION	NP2			M1		
	Proprietary software	FOSS	In-house developed software	Proprietary software	FOSS	In-house developed software
CI/ CD				10%	90%	0%
AI/ ML				20%	80%	0%
Security Tools				0%	100%	0%
ERP	0%	100%	0%			
CRM	0%	100%	0%	100%	0%	0%
CMS				0%	100%	0%
Ticketing/ Workflow Management System	0%	100%	0%	0%	100%	0%
MIS				100%	0%	0%
LMS						
Accounting & Finance	100%	0%	0%	100%	0%	0%
HR and Payroll				100%	0%	0%
Project Management	0%	100%	0%	0%	100%	0%
Communication (Email/ Instant Messaging or Email/ Office Suite)	100%	0%	0%	90%	10%	0%
API Integration						

	Data was not provided by the organisation/ The stack is not used by the organisation

FOSS Policies and Industry Expectations

The union government and various state governments in India have formulated and implemented policies to encourage the development and adoption of FOSS. These can be divided into the following domains:

1. **Public procurement:** The 'Policy on Adoption of Open Source Software for the Government of India', released in 2015, advocated a preference for FOSS. It required government organisations implementing e-governance systems to develop their RFPs (Request for Proposals) such that respondents were required to consider FOSS alongside proprietary software, and that preference be given to the former.

   MeitY also published the 'Framework for Adoption of Open Source Software In e-Governance Systems' to operationalise the above policy. It provides guidelines for government departments to adopt and develop IT systems using FOSS.

2. **Open sourcing software:** The government also released the 'Policy on Collaborative Application Development by Opening the Source Code of Government Applications'[96] in 2015. This provides a framework for archiving the government's custom-developed source code in repositories and opening these repositories for promoting reuse, sharing and remixing.

   In line with this, the code-hosting platform OpenForge[97] was developed for the collaborative development of e-governance applications. As of January 2025, 3,189 projects are listed on the platform. However, government agencies have complete authority to control access to repositories and decide whether to permit contributions to these projects.

Among state governments, Kerala has been the pioneer of FOSS adoption in India. The state's 2001 IT policy sought to encourage FOSS, while its 2007 policy sought to preferentially adopt FOSS in all government ICT projects. Key initiatives include various state institutions and departments moving their critical IT operations to FOSS systems, setting up ICFOSS (International Centre for Free and Open Source Solutions) to promote R&D in FOSS, and implementing the IT@School project (refer to KITE case study, discussed earlier, for more details).

Alongside Kerala, other states such as Andhra Pradesh,[98] Assam, Goa, Maharashtra and Tamil Nadu[99] have also initiated steps towards adoption of FOSS.

'Open source is, at the end of the day, nothing but collaborative innovation.'
Abhishek Jain, CPTO, Swasth Alliance

96. Department of Electronics and Information Technology (DeitY), Ministry of Communications and Information Technology, Government of India, 'Policy on Collaborative Application Development by Opening the Source Code of Government Applications' <https://www.meity.gov.in/static/uploads/2024/03/Policy-Document.pdf> accessed 4 March 2025.

97. 'About OpenForge' (*OpenForge*) <https://openforge.gov.in/openforge/about.php> accessed 3 February 2025.

98. ePDS in Andhra Pradesh.

99. Has seen adoption of BOSS (Bharat Operating System).

Organisations with whom we interacted as part of this study expressed varied views on policy measures relating to FOSS. Some of them categorically expressed the view that all software developed by and for the government, i.e., funded through taxpayer funds, must be open source. Some advocated for better implementation of existing government policies, particularly those relating to procurement. Despite favourable guidelines, they felt that FOSS has not been sufficiently adopted due to the optional character of the provisions, rather than strong mandates for FOSS usage.

In addition, some expressed the view that the government should actively develop FOSS-based software solutions, including DPGs. The reasons cited for this included new market creation (as in the case of UPI), alleviating data privacy-related challenges, and addressing security issues arising from the use of software developed by foreign organisations. They also pointed out that this may reduce duplication of efforts.

Some organisations also advocated for financial support from the government for FOSS projects. This could be in the form of grants for contributing code or maintaining projects. Some also suggested allowing organisations to use their CSR funds for the same.

A counter view also emerged during some of our conversations, wherein some organisations argued that government support was not necessary for FOSS. In their view, the adoption of FOSS and support for FOSS must be driven by the industry and the community.

Some other important recommendations shared by organisations include:

1. Creating awareness about FOSS and its benefits, and addressing misconceptions about open source, such as it being less secure or compromising data privacy.
2. Modification of procurement guidelines such that respondents to RFPs be mandated to use FOSS.
3. Inclusion of FOSS in education curriculum across the country, in addition to supporting (particularly public) universities like IITs to contribute to FOSS.

'If FOSS is the sport of cricket, we need the equivalent of IPL.'
Pramod Verma, CTO, EkStep Foundation

The sport of cricket has witnessed a transformation with the advent of IPL. In the past, aspiring players would be reluctant to invest their time and resources on practice since the chances of selection to the national cricket team were slim. There was only one team with twelve positions.

A format like IPL has expanded opportunities and allowed new talent to bubble up. Creating an IPL-like common branded platform for FOSS would offer a 'playground' for developers, similar to competitive platforms in sports, to contribute to open source, develop their skills and attract and nurture talent, all while significantly expanding contribution to FOSS.

DIFFICULTY IN QUANTIFICATION OF THE BENEFITS OF OPEN SOURCE

Existing literature and insights from this study show that the benefits of open source outweigh its costs. While quantification in economic terms of the value addition of FOSS in the Indian context was one of the intended goals of our study at the beginning of the project, a major finding from the study has been the challenges regarding the same. It is difficult, if not impossible, to economically quantify all the benefits from FOSS.

Apart from the fact that most the Indian organisations do not keep comparisons of the total (economic) costs of proprietary software vis-a-vis FOSS, there are some additional factors that we would like to highlight in this regard.

Lessons from Utility Theory in Economics

Utility represents customer satisfaction, but its measurement is inherently abstract and subjective. Consequently, economists often rely on ordinal utility, which ranks preferences qualitatively, instead of cardinal utility, which assigns precise numerical values.

1. **Non-economic motivations:** While cost savings are one of the primary reasons for adopting FOSS over other forms of software, it is important to emphasise that it is not the only reason. As is evident from the case studies in this report, organisations choose to adopt FOSS for a variety of reasons including its role in enabling learning, building their technical capacity, avoiding vendor lock-in, in incremental development, and building differentiated products and solutions. It is impossible to assign numerical value to those important benefits.
2. **Lack of alternatives:** As is evident from the discussions we had with developers from different sectors, for certain categories of software there may not be a closed-source equivalent. This makes comparisons impossible and thereby the quantification of advantages also impossible.
3. **Difficulty in comparison with proprietary software:** This study uses the concept of TCO to compare costs of different types of software, to quantify EVA (Economic Value Addition). Respondents have pointed out the complexities in calculating TCO for software, and it is also found that this is not an exercise most organisations engage in as part of their software evaluation process.
4. **Pricing mechanism of commercial proprietary solutions:** Vendors of commercial proprietary (specifically B2B) software often employ convoluted pricing mechanisms. Organisations purchasing the software are unaware of the rationale of vendor quotations and lack any frame of reference. This makes comparison with different types of software even more difficult.
5. **Quantification of time value:** FOSS provides readily available functionality that organisations can incrementally develop on. This helps in maintaining a faster software development cycle. Quantification of the value of time saved and the benefits of having their product or service earlier in the market is difficult, if not impossible.

Convincing Organisations to Adopt FOSS: Insights from an Interview

While convincing policy-makers and organisations to adopt FOSS in the absence of showing economic benefits in numbers is difficult, over a period of time, people have devised different approaches to address the same. During an interview with us, Kishan Parekh, consulting CTO at TechPartner which majorly deals with SMEs, had some important insights on this complexity and how he addresses it. The discussion with Kishan revolved around two main challenges:

1. The difficulty of convincing an organisation to adopt open source.
2. The complexities involved in calculating the commercial implications.

For SMEs, the challenge often lies in convincing them to transition to open source. Many SMEs are accustomed to using familiar operating systems and software that is well-known. When considering open source, they often express concerns about whether there is a skilled talent pool available to support it, whether the solution will work reliably, and if skilled personnel can be found to maintain the system.

According to Kishan, these concerns are not uncommon. Whenever he recommends to SMEs that they adopt open source solutions, the initial response is often scepticism. Despite having some awareness of open source, SMEs still hesitate to take decisive action, and even before discussing the cost reduction in using open source, the directors or owners first seek assurance for their concerns.

Kishan notes that parallels can be drawn to a period of time 15-20 years ago, when the BFSI sector faced similar scepticism on the introduction of Linux. At that time, convincing stock exchanges, depositories and brokers to use Linux was challenging. However, today these organisations are not only comfortable with Linux but have entire teams proficient in using it. This shift has made it significantly easier to highlight additional benefits of open source to BFSI organisations, because they are already convinced of its value. In contrast, SMEs remain cautious.

Kishan detailed his approach to the task of demonstrating the cost and value of open source to clients. He starts by understanding the organisation's specific requirements and the business problems they aim to solve. For most SME owners or directors, reducing costs is important but it is secondary to solving their core business challenges. It is important to identify their goals and create a comprehensive list of necessary software components. Once this groundwork is done, he explains how the cost implications of open source are not limited to immediate savings but also include considerations for future growth. For instance, if a business grows 1.5

times, the licensing costs of proprietary software can increase almost five times. With open source, scaling can often be done at a much lower cost.

Another important aspect of his approach is identifying the components that are non-negotiable. In some cases, open source alternatives may not be mature enough or suitable for certain critical functions. In such situations, Kishan ensures that existing software and trained teams are left undisturbed, while focusing on implementing open source in areas where it can be effective.

One of Kishan's clients produces chip-based cards used by banks. This client needed a solution to ensure permanent deletion of data i.e., that data once deleted should not be recoverable by any means. This would be verified by auditors who used specialised software and tools to check for data recovery, that bypassed the operating system layer and interfaced directly with the hard drive. Proprietary solutions were unsuitable because they contained layers of closed-source software, and even with commercial licenses their working was not known, and hence could not be trusted to guarantee permanent erasure. Using open source, Kishan's team devised a solution that allowed for the inspection and configuration of all layers such as the operating system and kernel, and achieved the desired functionality.

The result was a diskless solution that ran on smaller machines. There was no disk, and the machine could be just switched off and restarted to ensure data erasure. This reduced not just hardware but also personnel costs, as non-technical operators could be employed to manage the system. Additionally, this solution reduced the power and cooling required for operations, thereby minimising overall maintenance costs. This flexibility, transparency and cost efficiency were not achievable with the proprietary software.

In another example, a company could scale its systems easily by replicating small, cost-effective units across locations, managed by semi-skilled workers rather than specialised IT staff. This is crucial for a business whose core operations are not IT-focused, such as printing.

In some instances, Kishan builds environments using open source to demonstrate their feasibility. These environments allow clients to compare costs directly with their current setups and better understand the potential savings. Although initial savings may seem small for businesses with smaller setups, they become more apparent as operations scale.

RECOMMENDATIONS FOR POLICY MAKERS

Based on the insights from the current study, we specifically recommend the adoption of the following suggestions:

1. **Modification of procurement guidelines:** Existing guidelines regarding IT procurement as per the 'Framework For Adoption of Open Source Software In e-Governance Systems'[100] require government agencies to give preference to FOSS over comparable closed-source solutions. However, it provides for exceptions where vendors can provide justification for not using FOSS in response to RFPs.

   Since the publication of this framework a decade ago, the use of FOSS in industry has soared exponentially and reliable FOSS-based solutions are available for almost every type of software requirement. Hence, it is recommended that the guidelines be modified to make procurement of FOSS solutions mandatory when they are available, and the closed-source software should be permitted for specialised or exceptional cases wherein FOSS alternatives are not available.

2. **Constitution of a committee to review the working of existing guidelines:** In addition, we also recommend the constitution of a committee comprising open source experts, industry representatives and law-makers, among others. This committee may study and review the implementation of all existing government policies that interface with FOSS and develop recommendations to improve the effectiveness of the existing policies. The committee may also recommend new policies to the government.

3. **Grants for FOSS contribution:** Developers contributing to or maintaining FOSS projects largely do so voluntarily and do not receive any compensation. The sustainability of these contributions is uncertain in the long run and organisations have flagged the lack of financial support as one of the challenges they face with FOSS.

   We recommend that the government provide grants to encourage contributions to specific FOSS projects. These can take a variety of forms, such as directly funding universities. It is worth noting that popular FOSS projects like the BSD Unix and PostgreSQL database were developed at universities. Grants could also be distributed based on certain metrics. 'FLOSS/fund', set up by Zerodha,[101] is a private sector initiative that has committed to provide $1 million annually to support FOSS projects. There could be similar government initiatives by both the central government and state governments.

---

100. Department of Electronics and Information Technology (DeitY), Ministry of Communications and Information Technology, Government of India, 'Framework for Adoption of Open Source Software in E-Governance Systems' (April 2015) <https://egovstandards.gov.in/sites/default/files/2021-07/Framework%20for%20Adoption%20of%20Open%20Source%20Software%20in%20e-Governance%20Systems.pdf> accessed 3 March 2025 .

101. Kailash Nadh (n 25).

4. **FOSS in education:** The CBSE and state-level school education boards are recommended to incorporate FOSS into their curriculum. FOSS can not only be leveraged to teach software, but also other subjects in a cost-effective manner, as seen from the case of Kerala. It can aid in improving student learning, contribute to digital capacity-building, and help in developing a pipeline of young talent skilled in FOSS technologies.

   As in projects like the Atal Tinkering Labs (ATL), established as part of the Atal Innovation Mission, students should be provided with FOSS and hardware. They must also be familiarised with using open source collaboration tools like GitHub and GitLab through innovative pedagogical approaches. Additionally, they should be encouraged to and sensitised on the importance of contributing to the community in return.

5. **Incentivising contributions from educational and research institutions:** Contribution to FOSS projects, including adding features or fixing bugs, could become part of student assignments in such institutions. If such contributions can be linked to their formal evaluations, more students would be incentivised to contribute.

   The framework for adoption of OSS[102] had recommended engaging academia by incentivising students and faculty developing and managing FOSS projects. However, it did not provide details regarding implementation possibilities. For example, while deciding on recruitments and promotions at the universities, FOSS-related contributions of faculty and researchers can also be taken into consideration, apart from the current focus on publications and patents. Such changes can encourage and recognise contributors' efforts.

6. **State ICT policies:** Some states, such as Kerala, have formulated and effectively implemented ICT policies that provide preference for FOSS. Its use, particularly in education, has set in motion a positive feedback loop: as more students are being skilled in FOSS, there is a higher likelihood that they will continue FOSS contribution when they enter the workforce.

   We recommend that other state governments in India adopt and implement similar provisions for a preference for FOSS in all their ICT requirements. Additionally, local governments largely use proprietary software and require assistance, through awareness and capacity-building, to adopt FOSS.

7. **Establishment of OSPO:** An Open Source Program Office (OSPO) is an institutional construct that can support and accelerate the consumption, creation and application of FOSS.[103] While several private sector organisations now have initiatives like OSPO, most public sector organisations in India do not have.[104] It is important to observe here that governments have also started adopting them,

102. 'Framework for Adoption of Open Source Software in E-Governance Systems' (n 100).

103.'The OSPO - A New Tool for Digital Government' (*OpenForum Europe*, 2022) <https://openforumeurope.org/publications/the-ospo-a-new-tool-for-digital-government/> accessed 3 February 2025.

104. 'TODO Members' (*TODO Group*) <https://todogroup.org/about/members/> accessed 3 February 2025.

105. 'EC Open Source Programme Office' (*Interoperable Europe*) <https://interoperable-europe.ec.europa.eu/collection/ec-ospo> accessed 3 February 2025.

and this includes the European Commission establishing an OSPO in 2020.[105] The rationale for having OSPOs internal to an organisation is to improve their effectiveness as they better understand the requirements and challenges faced by that organisation or department.

It is recommended that the Government of India also establish an OSPO. This can serve as a nodal agency and assist various ministries, government departments, state governments and PSUs in setting up their own OSPOs. These OSPOs would advise, assist and lend technical expertise with FOSS, alongside fostering engagement with FOSS communities in the relevant area.

8. **Improve awareness and accessibility of OpenForge:** Despite the majority of organisations surveyed demanding that government-developed solutions be open source, none mentioned OpenForge, which indicates either limited awareness about the platform among the community or a lack of quality projects on the platform. We recommend that the committee suggested as part of our second recommendation also review the working of OpenForge. If it is found to not be suitable for the promotion of FOSS projects, necessary changes may be made to convert it into an active code-hosting platform or to discontinue the support for the platform. The government may also explore other alternatives, including the use of community-owned platforms for sharing code.

9. **Open source government works:** The 2015 Policy on Collaborative Application Development[106] grants the government full rights to the custom-built software source code, ensuring that it retains IP ownership for government-funded software development. However, while the policy emphasises making source code public and permits code changes, it does not explicitly mandate the use of standard open licenses (e.g., MIT License or Apache License v2.0) for public distribution. This creates a gap between the policy's intent and its legal framework, as the absence of a standard license leaves the terms of reuse, modification and redistribution unclear. This gap is clearly visible on OpenForge. While the platform lists 3,189 projects, a significant majority do not specify a license, and others use non-standard or restrictive licenses.[107]

   Cases like *B.N. Firos v. State of Kerala and Others*[108] illustrate the government's ability to retain control over software developed with public resources, but also highlight the need for clear frameworks to ensure such software benefits the public. In this case, the Court ruled that the government has the power to declare a computer system as a 'protected system' under the provisions of the IT Act, 2000, if that system is a 'government work' as per Sec. 17(d) of the Copyright Act, 1957. However, this case had a specific scenario wherein an MoU between the

106. 'Policy on Collaborative Application Development by Opening the Source Code of Government Applications' (n 96).

107. 'Public Projects Map' (*OpenForge*) <https://openforge.gov.in/softwaremap/trove_list.php?form_cat=78> accessed 13 February 2025.

108. *B.N. Firos v State of Kerala and Others* [2018] 9 SCC 220.

TSP for e-governance and the Government of Kerala explicitly vested IP ownership with the Government.

In the light of these aspects, we recommend that, as emphasised in our second point above, there should be strict enforcement of the 2015 policy across all government-funded projects, ensuring that the government retains IP ownership, and that source code is made publicly available. We also recommended that since the IP ownership would be vested with the government in all cases where public funds are used, all such government works be released under an open license that provides sufficient freedom for learning and modifications.

This practice is followed by countries such as the UK, wherein the Home Office and the Government Digital Service make everything open by default, licensed under MIT.[109] While the MIT License is preferred,[110] the UK government also allows flexibility in choosing an appropriate open source license that meets OSI standards, ensuring adaptability for different software projects.[111]

Bulgaria also follows a similar framework. After the 2016 amendment to the law on electronic governance, requiring all new government software to be open source, Bulgaria has taken significant steps to implement this law, including making EUPL (European Union Public License) the preferred license.[112] EUPL was established and approved by the European Commission between 2005 and 2007. This license, approved by the OSI, is the basis of the development of various FOSS foundations, such as the Open Source Observatory and Repository for European public administration (OSOR.eu).[113]

Another method to open-source government works is to place them in the public domain. The United States is an example in this regard and specifically takes the position that there is no copyright in government works. While there are certain exceptions, software developed by federal employees is automatically placed in the public domain by default, making it open source.[114]

It is high time that India also takes an approach similar to the United States or the United Kingdom in this regard, so that more innovation in this area can be facilitated.

---

109. 'Government Digital Service' (*GitHub*) <https://github.com/alphagov> accessed 8 January 2025.

110. MIT is designed for that purpose, and is more widely recognised in the software world. James Stewart, 'Coding in the Open – Government Digital Service' (*Government Digital Service*, 2012) <https://gds.blog.gov.uk/2012/10/12/coding-in-the-open/#comment-2388> accessed 8 January 2025.

111. 'Open Source Licensing' (*Home Office, Engineering Guidance and Standards*, 2023) <https://engineering.homeoffice.gov.uk/standards/open-source-licensing/> accessed 8 January 2025.

112. 'Open Source Software Country Intelligence Report Bulgaria 2023' (*OSOR*, 2023) <https://interoperable-europe.ec.europa.eu/sites/default/files/inline-files/Open%20Source%20Software%20Country%20Intelligence%20Report%20-%20Bulgaria.pdf> accessed 8 January 2025.

113. Tuesday Bwalya, Dr Akakandelwa Akakandelwa, and Dr Milena Dobreva-McPherson, 'Adoption and Use of Free and Open Source Software (FOSS) Globally: An Overview and Analysis of Selected Countries' (2019) 3 Zambia Journal of Library & Information Science 48.

114. US General Services Administration, 'Code' (*Code.gov*) <https://code.gov/> accessed 3 February 2025.

ANNEXURE

Questionnaire

Section I – Organisation profile

Internal note: Attributes of organisation including but not limited to sector, size (financial and employee strength) etc.

Questions

1. Which among the below categories best describes your organisation?

   a. Corporate/ Large enterprise.

   b. Start-up/ SME.

   c. Non-profit/ Think tank.

   d. Government/ Public sector organisation.

2. Which sector/ industry the organisation operates in?

   a. Software and IT services.

   b. Finance.

   c. Education.

   d. Healthcare.

   e. Manufacturing.

   f. Consulting.

   g. Non-profit (development sector).

   h. Non-profit (think tank).

   i. Non-profit (research foundation).

   j. Others (please specify).

3. Which year was the organisation founded?

4. Please provide the approximate number of employees in your organisation: _____

5. What is the size of the organisation based on annual turnover?

   a. < ₹5 crore

   b. ₹6–₹50 crore

   c. ₹51 crore–₹250 crore

   d. ₹251 crore–₹1000 crore

   e. > ₹1000 crore

6. What was the average annual revenue earned by the organisation in the last 2 years? (if applicable): _____

7. How crucial is IT and/ or software for your core business/ operations?

   < 5 point Likert scale > (1 – Not very crucial, 5 – Very crucial)

8. Please elaborate on the role played by IT in your organisation.

Section II – FOSS usage

Internal note: This section tries to understand the areas where the surveyed organisation adopts FOSS. This will be captured in 2 ways:

1. Technology wise: What FOSS technology/ component/ product is adopted?

2. Application wise: Where is FOSS used? (Within the firm's product or service, internal use, experimentation for future offerings, etc).

By Free and Open Source Software (FOSS), we mean software that provides users the freedom to run, make copies, study, change and improve and distribute original or modified versions of the software.

Questions

9. Are you using any software that can be classified as FOSS?

   Please share the organisation level software technology stack (if available). Example: https:// zerodha.tech/stack/

   a. Yes, in production environment i.e. as or as part of software that is used live by the intended users.

   b. Yes, in non-production environment i.e. as or as part of software that is under development or testing (Skip to Q12).

   c. No (Skip to Section VII).

10. In which year did you first adopt any form of FOSS in a production environment? (unless using since inception).

11. Where is FOSS used in your organisation? (You may select more than one option).

    a. Embedded in software-based products or services used by clients/ customers/ beneficiaries (External-facing).

    b. Applications used internally by employees (Internal-facing)

    c. To experiment with new features/ technologies/ R&D.

12. Consider the following types of software: proprietary software (off-the-shelf commercial software), FOSS, and software developed within the organisation (software developed in-house for use by firm). To what extent each of these categories of software are used in your organisation.

    Illustration: If you use both Windows and Linux operating systems within your organisation on 70% and 30% of systems respectively, the share would be

    Proprietary software - 70%, FOSS - 30%, In-house software - 0%

	PROPRIETARY SOFTWARE	FOSS	IN-HOUSE DEVELOPED SOFTWARE
Operating Systems			
Web Servers			
Middleware			
Cloud Native Software			
Development Framework			
Programming Languages			
Database Management System			
Infrastructure Automation			
CI/ CD			
AI/ ML			
Security Tools			
ERP			
CRM			
MIS			
LMS			
Accounting			
HR and Payroll			
Any Other (please specify)			

Section III – Relevant considerations in software evaluation

Internal note: The factors considered for FOSS adoption (identified in new literature). These can be of 2 types:

1. Technical.

2. Organisational.

Here we would attempt to capture both pre-adoption assessment and post-adoption results.

Questions

13. Whenever the requirement for a software solution arises, do you perform a comparative assessment to decide whether you should adopt proprietary software, FOSS or go for in-house development? (Note: For cloud hosted solutions or managed services consider the underlying technology stack.)

    a. Yes (If yes, please elaborate: _____)

    b. No (Skip to Section IV and post-completion skip to Section VI).

14. Post such evaluations, have you in any case taken the decision to use FOSS components?

a) Yes.

b) No (Skip to Q19).

15. Which among the following technical factors, considered as part of your evaluation, favoured/ positively influenced the adoption of FOSS components?

    a. Fulfilment of functional requirements.

    b. Compatibility with existing systems.

    c. Integration feasibility and adaptation required for integration.

    d. Usability/ ease of usage.

    e. Maintainability of the component and system it is integrated into.

    f. Compatibility of licencing terms (if used in a product) of the component with that of the product.

    g. Extent of customisation possible.

    h. Adherence to standards.

    i. Security.

    j. Others (please specify).

16. Which among the following technical factors, considered as part of your evaluation, hindered/ negatively influenced the adoption of FOSS components?

    a. Fulfilment of functional requirements.

    b. Compatibility with existing systems.

    c. Integration feasibility and adaptation required for integration.

    d. Usability/ ease of usage.

    e. Maintainability of the component and system it is integrated into.

f. Compatibility of licencing terms (if used in a product) of the component with that of the product.

g. Extent of customisation possible .

h. Adherence to standards.

i. Security.

j. Others (please specify).

17. Which among the following organisational or business factors, considered as part of your evaluation, favoured/ positively influenced the adoption of FOSS components?

a. Economic/ financial cost effectiveness/ Total cost of ownership.

b. Time required for adoption.

c. Extent of business process re-engineering required.

d. Availability of support (community support or as a service).

e. Availability of documentation.

f. Availability of training (in case deemed necessary).

g. Ownership and governance of the component's parent project.

h. Extent of use in the industry.

i. Presence under active development.

j. Others (please specify).

18. Which among the following organisational or business factors, considered as part of your evaluation, hindered/ negatively influenced the adoption of FOSS components?

a. Economic/ financial cost effectiveness/ Total cost of ownership.

b. Time required for adoption.

c. Extent of business process re-engineering required.

d. Availability of support (community support or as a service).

e. Availability of documentation.

f. Availability of training (in case required).

g. Ownership and governance of the component's parent project.

h. Extent of use in the industry.

i. Presence under active development.

j. Others (please specify).

19. If you chose not to adopt FOSS post-evaluation, what alternative did you choose to adopt?

    a. Proprietary software: Off-the-shelf product.

    b. In-house development.

20. Why did you choose this alternative? Select among factors mentioned in Q15 and Q17.

    a. Technical.

    b. Organisational/ Business.

Section IV – Experienced benefits and challenges

Internal note: The drawbacks and benefits (both tangible and intangible) of FOSS adoption (validate and enhance findings of literature) with focus on newer technologies like cloud computing, CI-CD, etc.

Questions

21. What benefits did you experience by virtue of adopting FOSS?

    a. Lower TCO (Total Cost of Ownership)/ Improved cost effectiveness through savings on licencing fees.

    b. Enhanced revenue from new products or services.

    c. Improved development velocity resulting from readily available functionality.

    d. Faster time to market (deriving from c).

    e. Faster rollout of enhancements, releases and patches.

    f. Open standards and interoperability.

    g. Freedom from vendor lock-in and associated costs (deriving from f).

    h. Stability derived from community support.

    i. Reliability from reduced errors and bugs.

    j. Enhanced security.

    k. Impetus to innovation.

22. What challenges did you face by virtue of adopting FOSS?

    a. Security gaps and maintaining security policies.

    b. Maintaining end-of-life versions.

    c. Tracking and applying patches and updates.

    d. Lack of commercial and high-level technical support.

    e. Revenue loss due to free-riding competitors.

    f. Legal uncertainty regarding licencing.

    g. Availability of personnel with skills, experience and proficiency in OSS technologies.

    h. Availability of training and associated costs.

    i. Testing required due to lack of certification for OSS.

j. Switching costs from proprietary solutions.

23. How would you describe your overall experience of FOSS adoption?

a. The drawbacks were greater than the benefits by small margin.

b. The drawbacks were greater than the benefits by large margin.

c. The benefits were greater than the drawbacks by small margin.

d. The benefits were greater than the drawbacks by large margin.

Section V - Economic value addition

Internal note: The impact of FOSS adoption on the firm's financials like revenue, profits etc.

Questions

24. Please provide the details of the costs (whichever applicable) for a particular requirement where different software components were evaluated based on their type. (You may choose to provide the total cost as well. The component breakup is not mandatory and is only for reference.) Details of the requirement:

_____

COMPONENT-WISE COSTS	PROPRIETARY/ OFF-THE-SHELF	FOSS	IN-HOUSE DEVELOPMENT
Licensing and Subsequent Renewal			
Support			
Training			
Development			
Documentation			
Hosting			
Hardware			
Integration and Customisation			
Maintenance and Upgrades			
Costs of Associated Components			
Business Process Re-engineering			
Time for Completion			
Any Other (please specify)			
Total			

Section VI – Innovation leverage

Internal note: Assess specifically the innovation leverage/ impetus provided by FOSS and gauge the level of innovativeness of the firm using some proxies.

Questions

25. What was your total expenditure on R&D that involved significant use of FOSS over the past two financial years?

This includes in-house efforts aimed at developing new or significantly improved products and services, and investments in or acquisitions of innovative start-ups.

26. Please provide the share of financing of your organisation (as a percentage of total) from each of the below sources.

SOURCE	SHARE OF TOTAL FINANCING (IN %)
Debt Financing (including loans)	
Private/ Public Equity	
Government Grant	
Venture Capital/ Angel Investment	
Crowdfunding	
Bootstrapping	
Others (please specify) _____	

27. How would you describe your engagement with FOSS projects?

    a. Consume FOSS for developing software solutions (Skip to Que 29).

    b. Contribute to FOSS projects.

    c. Engage with FOSS projects (say to manage dependencies or as part of tech strategy).

    d. Providing economic support (say for example, by way of sponsorship) for FOSS Projects.

    e. Others (please specify).

28. How is FOSS supported/ anchored in your organisation?

    a. Not supported, teams decide open source adoption individually.

    b. At the product level.

    c. A central FOSS support organisation (e.g., an OSPO or a central team).

29. Please provide an estimate of the total number of commits to GitHub (or any public code repository) on FOSS projects from your organisation in the past two financial years.

30. Has the organisation presented any technology or software solutions developed that makes significant use of FOSS at a technology conference? If yes, how many such solutions have been showcased in the past two years?

31. How many hackathons or coding challenges have you organised in the past two years?

32. As per your estimate, what percentage of code developed in the organisation is re-used internally within the organisation?

33. Please provide the following details about projects released under FOSS licence (if any) and their usage.

PROJECT NAME	NUMBER OF DOWNLOADS (ON GITHUB OR ANY OTHER PUBLIC CODE REPOSITORY)

Section VII - FOSS and IP

Internal note: This section focuses on the extent to which intellectual property acts as a hindrance towards FOSS adoption.

Questions

34. Whether intellectual property claims (for example, claims relating to violation of copyright, patents or trade secrets) has been an impediment in adoption of FOSS?

35. Whether lack of exclusivity (say for example, because of the mandate to use a Share-Alike licence) has ever been a consideration in adoption/ rejection of FOSS? If yes, please specify: _____

36. Do you perform Software Composition Analysis on the code base of your developed software? If yes, have any licensing conditions of FOSS components prevented you from adopting said component in your software?

Section VIII – Legal/ policy/ Govt. support related wish-list

Internal note: If the organisation is a firm that has adopted/ wishes to adopt FOSS, what kind of policy changes do they consider as relevant?

Questions

37. In your opinion, what are the ways in which the government can aid in better adoption of FOSS and more FOSS contributions to the society?

    a. Financial support for opensource projects.

    b. Measures to support the creation of and improving engagement with open source communities.

    c. Modifications in public procurement guidelines that favour FOSS solutions.

    d. Development of an industrial policy incentivising FOSS usage.

    e. Government organisations publishing developed software under FOSS licence.

    f. Creation of a nodal agency (similar to Open Source Program Offices in the EU) to guide and support firms in their FOSS adoption, contribution and community engagement.

    g. Integration of FOSS in ICT education curriculum.

    h. Better implementation of existing mandates.

    i. Others (please specify).

The report analyses the adoption of FOSS in India, primarily through case studies across four sectors (healthcare, education, finance, and software and IT services) and different types of organisations (start-ups, non-profits, medium, large, and public sector organisations). The study highlights both the benefits and challenges experienced by organisations using FOSS. The study illustrates that while organisations benefit from increased innovations, cost/ time savings, flexibility, and enhanced security, they also face challenges such as lack of enough skilled personnel and limited community support. Organisations are also seen taking a cautious approach to licensing, favouring permissive licenses over restrictive ones. Based on the diverse empirical findings, the report also recommends some policy reforms including mandating FOSS adoption for government bodies, integrating FOSS in ICT education, and updating procurement guidelines to mandate FOSS solutions when they are available.

DR. ARUL GEORGE SCARIA is a Professor of Law and Co-director of the Centre for Intellectual Property Research and Advocacy at NLSIU.

DR. SURYAPRAKASH MISHRA is an Associate Professor of Social Science and Chair Professor (I/C), Reliance Chair on Corporate Law & Governance at NLSIU.

MR. SHUBHAM SHINDE is a technology policy researcher interested in studying the impact of emerging technologies on markets and society.

MS. RASHI SINGHAL is an Academic Fellow at NLSIU, with research interests in Intellectual Property, Data Privacy, and AI.

NATIONAL LAW SCHOOL OF INDIA UNIVERSITY

WWW.NLS.AC.IN