15th October, 2023

To,

The Controller General of Patents,
Government of India.

Subject: FOSS United's Comments on the Patent Manual

Dear Sir,

We sincerely thank your office for initiating this timely discussion on the Patent Manual.
I am hereby submitting comments on behalf of FOSS United, an organisation that works to grow the Free and Open Source Software (FOSS) ecosystem in India. As an organisation, we are deeply concerned about developer freedoms. We therefore hope that our comments are taken seriously and responded to. We would also request an opportunity to meet with you to explain our viewpoint on the strategic importance of FOSS for India.

With warm regards,

Venkatesh Hariharan
Public Policy Director
FOSS United
Mobile: 9004011970
Email: venkyh@gmail.com

FOSS United's Comments on Section 3(k) of the Patent Manual

FOSS United is an organisation that aims to grow the Free and Open Source Software (FOSS) community in India. We are deeply concerned about *developer freedoms*. Therefore, we have been closely involved in the long and tortuous debates around Section 3(k) which states that "a mathematical or business method or a computer programme per se or algorithms;" are not patentable subject matter. Drawing on that experience, we would like to state that the current status where software patents are being granted is leading India on the road to unfreedom. We also explain how software patents are like medicines that were never effective in the first place, are way past their expiry date and are now positively harmful. We believe software patenting will create serious negative consequences for Atmanirbhar Bharat, Startup India and Digital India initiatives, for decades to come. It could also enable the digital colonisation of India by large patent holding firms.

Consider the issue of software patents from the perspective of a young software developer. The developer spends days and months writing a piece of software that becomes extremely popular. Revenues grow rapidly and newspaper articles and venture capital funding follow. However, this also attracts the attention of a patent holder who files a case and gets an injunction preventing our developer from selling their products. Revenues crash, the court case drags on for years and costs crores of rupees, and the business eventually goes out of business.

There are a number of lessons that can be derived from looking at software patents from the developer's perspective:

1. **In a regime that allows for software patents, developers have no way to avoid violating software patents:** Every piece of software ever written combines hundreds and thousands of programming methods. Even if a software developer wishes to be law abiding, and has an army of lawyers doing prior art searches, it is impossible to avoid violating software patents. This is because software development is an abstract art, and it is impossible to draw clear boundaries around abstract ideas. This lack of clear boundaries makes software patents a fertile ground for litigation. To understand this, we highly recommend reading Chapter 3 of the book *Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk* by Boston University professors, James Bessen and Michael Meurer. The chapter is titled, *If You Can't Tell the Boundaries, Then It Ain't Property* and can be accessed at http://www.researchoninnovation.org/dopatentswork/dopat3.pdf. Chapter 9 titled, *Abstract Patents and Software* is also a useful read and can be accessed at http://www.researchoninnovation.org/dopatentswork/dopat9.pdf. In this chapter, the authors demonstrate that because of its abstract nature, 30 percent of patent litigation is around software and business method patents. The introductory chapter of the book can be accessed at http://press.princeton.edu/chapters/s8634.pdf.
2. **Independent invention is not a defence in a software patent lawsuit:** Software is an applied form of maths and logic. Globally, patents are not granted on maths and logic because their unfettered use is considered essential for the progress of humanity. In sharp contrast, software patents are state granted monopolies that give the patent

holder the right to exclude others from using programming methods that are patented. This is a perverse system because it punishes developers for using mathematics and logic. Software is already protected by copyrights and trade secrets. With copyrights, even if two people write something similar, defendants can protect themselves if they can prove that they arrived at their work independently, without copying the plaintiff's works. With software patents, no such defence exists.

3. **The "disclosure" in software patents has no value for developers:** Any developer who has read through a few software patents knows that the disclosure in software patents are as clear as mud. Developers who have multiple patents in their name have reported that they could not recognize their own works once lawyers have done their obfuscatory magic. Combine this with the fact that patent offices often have overworked staff members who do not get sufficient time to uncover prior art and evaluate patent claims, and you have a system that is heavily loaded against software developers.

The past, present and future of software innovation

In this section, we explain how software patents are like medicines that were never effective in the first place, are way past their expiry date and are now positively harmful. The social contract between the inventor and the state is that inventors disclose their inventions, in return for a state granted monopoly for a limited period of time. After this limited period expires, the invention passes into the commons, where everyone can benefit from it.

The Past

The theory is fantastic, but the practice has been very different. The most fundamental inventions of the computer age, from algorithms that underpin the AI models in use today, to compilers, programming languages, spreadsheets and many others were created well before software patenting took off in the eighties. This proves that patents were never effective in terms of promoting innovation.

The Present

The word patent originates from the Latin word, *patere which means* "to lay open." Today, the vast majority of software innovation is happening in the world of FOSS, where software is open and built in the commons by default. 15 years ago, proprietary, closed source software was the norm, and FOSS was the exception. Today, the tables have been turned and the FOSS model of Collaborative Innovation is the norm, while proprietary software is the exception. The three pillars of FOSS are collaboration, community and the shared ownership of knowledge. This has created a massive global commons worth hundreds of billions of dollars. FOSS has created world class software that can be used, shared and modified by anyone. This has enabled innovation by reducing barriers to entry. The proof lies in the fact that FOSS comprises 60-90 percent of the technology stacks of most companies. If FOSS was absent, and companies had to rely on proprietary software, their IT costs would go up between 10x to 100x. For example, a seed stage startup might raise Rs 2-3 crores in the

beginning and spend 40-60 percent of that on their IT infrastructure. Without FOSS, they might have to raise much larger seed rounds of to pay for proprietary software.

Cost is just one aspect of the Collaborative Innovation model of FOSS. FOSS licences enable people around the world to collaborate, create communities of practice and rapidly innovate new technologies. Most innovation in software is happening in the collaborative world of FOSS where people *voluntarily open up the source code*. In other words, innovation is happening in the open, and every important emerging technology - from AI, ML, IoT, cloud computing and many others - is being built in the commons. Thanks to FOSS, billions of dollars worth of technologies are available to Indians free of cost. When so much innovation is happening in the open, what is the need to hand out state granted monopolies (software patents) as incentives? For an emerging economy, FOSS is a resource to be cherished and protected at all costs.

GitHub, a leading FOSS collaboration platform, reports that 94 million software developers on its platform have added 413 million software contributions in 2022, taking the total to 3.5 billion. The Linux Foundation, a leading FOSS organisation that hosts 300+ FOSS projects, estimates that its projects have created 1.15 billion lines of code worth $54 billion. Similarly, the Apache Software Foundation estimates that the 350+ projects it hosts have created FOSS worth $22 billion. These projects cover the most fundamental technologies from cloud computing, distributed computing, big data and analytics, blockchain technologies and many others. Linux, which is the leading example of FOSS runs everything from supercomputers to smartphones to stock exchanges to search engines and even the Mars Rover.

The Future

GitHub reports that the number of developers on its platform grew 27% year on year and that 90% of companies use FOSS. Most emerging technologies will be built as FOSS because the Collaborative Innovation model of FOSS enables rapid innovation that cannot be matched by proprietary software development practices. Therefore, the FOSS model of innovation is now the dominant model of software development. This has important implications for public policy.

While FOSS is a public good, freely available to all, software patents are a private good available only to an elite few who have the time, money and legal firepower required to file patents. India has benefited hugely from FOSS. As an organisation deeply focussed on developer freedoms, we would like to state that software patents are loaded against software developers, wastes the precious time of industry and patent examiners, and benefits only patent lawyers and monopolistic rent seekers. No wonder some experts call software patents a “welfare scheme for lawyers.” Therefore, for pragmatic as well as principled reasons, FOSS United recommends that India must protect FOSS and keep software out of the realm of patentable subject matter.

We also note with dismay, that the number of software patents being granted in India, has been increasing. A study commissioned by FOSS United, and researched by Software Freedom Law Center found that the IPO has been granting patents on software, with most of

these patent grants going to MNCs. This contravenes what the then Minister, Mr Kamal Nath had said in 2005 about Section 3(k) being modified, “on the ground that this may give rise to monopoly of multinationals.”

The no. of software patents granted per year in violation of the Patents Act, 1970

Patents	FY 2014-2015	FY 2015-2016	FY 2016-2017	FY 2017-2018	FY 2018-2019	FY 2019-2020	FY 2020-21	FY 2021-22
Patent applications (as per govt.)	42763	46904	45444	47854	50659	56284	58502	66440
Granted patents (as per govt.)	5978	6326	9847	13045	15283	24936	28391	30074
Granted software patents as per our inputs	325	259	550	664	1054	877	813	462
Foreign	295	232	492	641	975	815	637	344
Indian	30	27	58	23	79	62	176	118

Source: Ministry of Commerce & Industry; SFLC.in

The increase in software patents is correlated to the fact that the excellent three part test in the Computer Related guidelines of 2015 were removed. The three part test provided excellent clarity to patent examiners until they were overturned and we request that it be reinstated immediately. FOSS United examined a number of software patents granted in India and found that many of them would not be granted even in the US, which is the least discerning jurisdiction when it comes to granting software patents. Post the Bilski and Alice judgements, the bar for obtaining software patents has become higher, even in the US.

Conclusion

Granting software patents goes over and above our obligations under TRIPS. Giving up a right that we have is an extremely serious decision that should be taken at the highest levels of government. As the CRI Guidelines states, in Section 1.5, “However, these guidelines do not constitute rule making. In case of any conflict between these guidelines and the provisions of the Patents Act, 1970 or the Rules made thereunder, the said provisions of the Act and Rules will prevail over these guidelines. The guidelines are subject to revision from time to time based on interpretations by Courts of law, statutory amendments and valuable inputs from the stakeholders.” The net effect of granting software patents will be increased litigation, as patent holders take to the courts.

There is also a strong geopolitical dimension to patents because countries that have a dominant position in patents use it to exert control over global supply chains, and capture the bulk of the value in their industries. Therefore allowing software patents at a time when we do not have a dominant position in software technology, will be a decision that is extremely hard to reverse, and will take decades to recover from.

While we must adopt best practices from the rest of the world, we must keep India's interests, current geo-political realities, the radically transformed landscape of innovation due to FOSS in mind when implementing policies. We request the Indian Patent Office to take a strict stance against software patents as their proliferation will lead to increased cost of doing business, legal uncertainty, challenges in fund raising for startups and diversion of time and money from productive activities. For a country that aims to be a software superpower, and has the world's largest base of developers, this is a self-goal that is highly avoidable.

FOSS United's recommendations on the Guidelines for Examination of Computer Related Inventions

Section	Suggestion	Reason
CRI Guidelines	Reinstate the three part test from the 2015 CRI guidelines (copied below). **Tests/Indicators to determine Patentability of CRIs:** Examiners may rely on the following three stage test in examining CRI applications: 1. Properly construe the claim and identify the actual contribution; 2. If the contribution lies only in mathematical method, business method or algorithm, deny the claim; 3. If the contribution lies in the field of computer programme, check whether it is claimed in conjunction with a novel hardware and proceed to other steps to determine patentability with respect to the invention. The computer programme in itself is never patentable. If the contribution lies solely in the computer programme, deny the claim. If the contribution lies in both the computer	The three part test provided a clear and simple way for patent examiners to evaluate patent eligibility.

	programme as well as hardware, proceed to other steps of patentability.	
4.5.1 Claims directed as "Mathematical Method": Mathematical methods are a particular example of the principle that purely abstract or intellectual methods are not patentable. Mathematical methods like method of calculation, formulation of equations, finding square roots, cube roots and all other similar acts of mental skill are therefore, not patentable. Similarly mere manipulations of abstract idea or solving purely mathematical problem/equations without specifying a practical application also attract the exclusion under this category. However, mere presence of a mathematical formula in a claim, to clearly specify the scope of protection being sought in an invention, may not necessarily render it to be a "mathematical method" claim. *Also, such exclusions may not apply to inventions that include mathematical formulae and resulting in systems for encoding, reducing noise in communications/ electrical/electronic systems or encrypting/ decrypting electronic communications*.	We suggest that the last sentence be dropped.	Systems for encoding, reducing noise in communications/ electrical/electronic systems or encrypting/ decrypting electronic communications are *applied forms of mathematics* and should not be granted patents, especially when implemented on general purpose hardware.