

How Free Software Develops Skills and Local Economic Growth

Wizards of OS 3

Berlin, June 10, 2004

Rishab Aiyer Ghosh

rishab@dxm.org

MERIT/Infonomics, University of Maastricht

This research is part of the FLOSSPOLs project supported by the EU 6th Framework IST Programme

Background: The FLOSS study

- **FLOSS project: Free/Libre/Open Source Software Study**
- **Largest, most comprehensive survey of developers worldwide (2002-2004)**
 - ~2800 responses to original EU-funded survey
 - ~1500 responses to FLOSS-US survey (Stanford/SIEPR)
 - ~650 responses to FLOSS-JP/FLOSS-Asia (Mitsubishi Research)
- **FLOSSPOLS project: Policy support, supported by the European Commission**

Software in society

“Access [to ICTs] is not enough, it is the ability to create, to add value, that is important”

Felipe Gonzalez
former Spanish Prime Minister,
Speaking at Open Source conference in Málaga, Spain, 18/2/04

Skills and economic growth

- **Skills development: “the ability to create”**
FLOSS is a training environment that increases the earning capacity of community participants without any explicit investment in training: a novel form of technology transfer
- **Economic growth: “ability to add value”**
FLOSS allows local entrepreneurs to provide a greater share of total value added, thus retaining a greater share of profits within the local economy

FLOSS develops local skills

- **Not skills *to use* FLOSS applications, but skills *learned from* participating in the FLOSS community**
- **Skills learnt through participating in the FLOSS community:**
 - programming
 - copyright law and licenses
 - teamwork and team management

FLOSS develops local skills

- FLOSS encourages not only passive “use” but active participation in the creative process
- FLOSS provides a very low barrier to entry for creativity – you don’t *have* to be creative but if you want to, you *easily can*

Learning skills – then sharing!

- **78%** of developers *join* the FLOSS community “**to learn and develop new skills**” (70% *continue* for this reason)
- **67%** of developers *continue* their participation in the FLOSS community “**to share ... knowledge and skills**”

Source: Free/Libre/Open Source Software (FLOSS) Study of Developers

These skills have economic value

- **30%** of developers participate in the FLOSS community **“to improve ... job opportunities”**
- **Over 30%** of developers **derive income directly** through their FLOSS work
- **A further 20%** **derive indirect income** as a result of their FLOSS work
- **18%** **got job because of FLOSS experience**

Source: Free/Libre/Open Source Software (FLOSS) Study of Developers

Employers appreciate this...

- **36%** of organisations “totally” or “somewhat” agree that employees can work on FLOSS projects on employer time
- These are not software companies! **16% of low IT-intensity companies** (retail, automobiles, tourism, construction) “totally agree” with this

Source: Free/Libre/Open Source Software (FLOSS) Study of Users

...but don't pay for it.

- **FLOSS communities are like informal apprenticeships – but apprentice/students and master/teachers contribute their own time for free**
- **Any cost (time, effort) is borne voluntarily by the participants themselves and not paid for directly by those who benefit (employers, society at large)**

“To each according to need...”

- **Everyone can benefit equally from this training, though not everyone invests equally in it – many “teachers” may have been formally trained at university or at work (which is explicitly paid for)**
- **In the larger perspective, this training system represents a subsidy – or technology transfer – from those who pay for formal training to those who don’t (or can’t)**

“To each according to need...”

- **Within countries, this represents a technology transfer from big companies to SMEs, who can less afford formal training**
- **Globally, this represents a technology transfer from economies who can afford formal training, to those who cannot**

“To each according to need...”

- **Sectoral benefits: poor countries may have formal computer training during computer science degree courses, but perhaps not in other subjects (biology)**
- **FLOSS usage provides students of other subjects to *informally* learn computer skills, programming skills and enhance their competence in their formal training**

But do we all want to program?

- How will we know, unless we can try?
- HTML is a programming language – the web only took off because it was open, so people could learn to write their own sites just by copying and changing other sites
- “Programming” covers a very broad range of skills from HTML to C; FLOSS allows entry at any degree with little investment in time or effort

But do we all want to program?

- In a proprietary environment, you have to decide to be a programmer, then buy development software, then spend lots of time and effort – all of which is a risk and entry barrier
- With FLOSS, you can *tinker*. You don't need to buy tools. You can use them to the extent you choose.

But do we all want to program?

- Learning skills in FLOSS, you risk losing only your time and effort
- However, since the barrier to entry is low (HTML!) you can control the degree of your investment – paddle at the shallow end or dive in deeper.
- In proprietary environments, the dividing line between user and developer is much sharper – the pool has only a deep end, you have to dive in or stay out altogether.

Skills and economic growth

- **Skills development: “the ability to create”**
FLOSS is a training environment that increases the earning capacity of community participants without any explicit investment in training: a novel form of technology transfer
- **Economic growth: “ability to add value”**
FLOSS allows local entrepreneurs to provide a greater share of total value added, thus retaining a greater share of profits within the local economy

Local value addition: proprietary

- **Building over a platform**

This applies equally to any platform, which is simply used as a (non-modifiable) base on which new services or software are built: 100% of the added value is local

- **Sales commissions**

Something which is rarely possible with free software, but also represents little value. Only the commission is retained locally, which is a small part of the total value.

- **Support, integration, customisation...**

Local value addition limited, as “deep” (high-value) services require “deep” access – only the proprietor has it.

Local value addition: FLOSS

- **Building over a platform**

As with proprietary software, free software platforms can be used as a (*modifiable!*) base on which new services or software are built: 100% of the added value is local

- **Sales commissions**

Rarely possible with free software, but also represents little value. However, the entire “sale price” can be retained locally, as no proprietor has to be paid a royalty or licence.

- **Support, integration, customisation...**

Local value addition extensive, as “deep” access is available. 100% of such services can be provided locally, retaining 100% of the value locally.

The importance of customisation...

- Custom or in-house software represents about 67% of total software produced (in the US; probably more elsewhere)
- If based on free software, custom solutions greatly benefit the solutions provider who captures 100% of the total value, not just the value added locally – no royalties/licences paid

Code re-use, higher service levels

- Free software allows providers to reuse code rather than build from scratch, and to reuse a huge base of code written by others
- Re-using (and modifying) allows the creation of much better end-user solutions for the same effort than writing from scratch
- Put together, this provides better value for money for customers and better profit margins for local service providers

“Deep” support, more local value

- Local companies are limited in the integration and support services they can provide for proprietary software
- Deep support: fixing software bugs, customising it to user requirements, or integrating extensively with other software requires deep access.

“Deep” support, more local value

- **Deep access to proprietary software is controlled by the proprietor (limits access or requires royalties, diminishing value retained locally)**
- **Deep access to free software is available to anyone – limited only by their skills. This allows every provider to potentially provide deep support services, and retain 100% of the value.**

Skills and economic growth

- **Skills development: “the ability to create”**
FLOSS is a training environment that increases the earning capacity of community participants without any explicit investment in training: a novel form of technology transfer
- **Economic growth: “ability to add value”**
FLOSS allows local entrepreneurs to provide a greater share of total value added, thus retaining a greater share of profits within the local economy

• Building *local* ICT competencies

- Be passive users of “black-box” software or active participants in global ICT?
- Being active requires being able to create, locally – and choose with the least barriers the level of creativity
- Skills development requires access to the ability to create – you don’t have to be a programmer, but you *should* have the choice.
- Relative local value addition is much higher with free software, as compared to proprietary (where the vendor controls and provides the most value)

More information...

The FLOSSPOLS project

<http://flosspols.org>

Licence fees & GDP/capita paper:

http://firstmonday.org/issues/issue8_12/ghosh/

European Union Open Source Observatory

<http://europa.eu.int/ISPO/ida/osol>

Paper on FLOSS as Official Development Aid,

By Jordi Carrasco Munoz

[http://www.i-today.com.vn/itoday/open_source/baocao/
jordi_europeancommunity.ppt](http://www.i-today.com.vn/itoday/open_source/baocao/jordi_europeancommunity.ppt)