

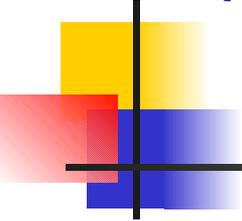
# Mechanics of Open Innovation: The case of Free Software

---

Creative Capital Conference  
Amsterdam, March 17, 2005

Rishab Aiyer Ghosh  
rishab@dxm.org

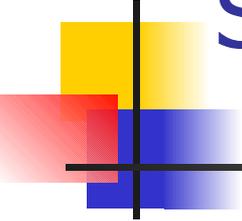
This presentation is based on work conducted at MERIT, University of Maastricht, in the projects CALIBRE and FLOSSPOLLS, funded by the European Union's FP6 IST programme



# “Protection” of what?

---

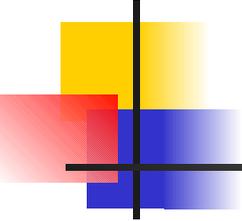
- Truly protecting “works of the human spirit” [WIPO] would mean protecting the ability – the freedom – to create
- Protection of “Intellectual property” on the other hand, may often be a restriction on “Intellectual creativity”.



# Spontaneous creativity?

---

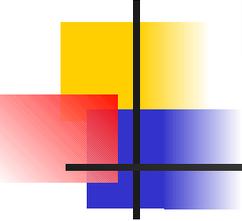
- Do we create and innovate spontaneously?
  - YES: as a natural urge
  - NO: only with economic incentives
- A mix of both, for different occasions?
- Policy focusses on the NO variant
- YES-innovation should be supported too, or at least not hampered



# “Protection” of what?

---

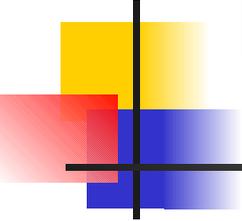
- Truly protecting “works of the human spirit” [WIPO] would mean protecting the ability – the freedom – to create
- Protection of “Intellectual property” on the other hand, may often be a restriction on “Intellectual creativity”.
- It protects a specific model of creation, not creativity



# The protection racket

---

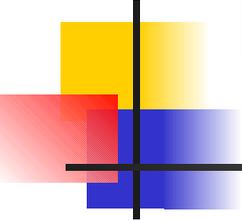
- Protecting creativity is a balance between protecting existing creations and enabling future creations
- The balance is weighted unfairly...
- ... and assumes that enabling future creations that build upon the past reduces the incentive to create.



# The protection racket

---

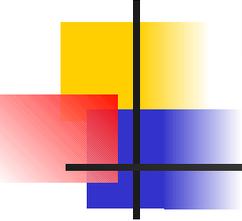
- This unfair, incorrect assumption results from treating information works as static “property” rather than as dynamic “innovation”
- Innovation requires the protection of a process of creation rather than the preservation of a state of ownership



# The protection racket

---

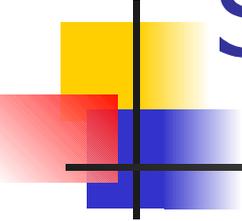
- Newton stood “upon the shoulders of giants” – this is dynamic innovation, rather than static property
- Today, Newton would have to pay a license fee just to see how tall those shoulders were... forget about standing upon them!



# The worth of free software

---

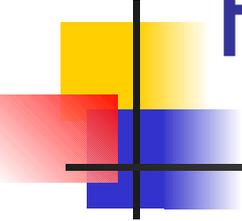
- Debian 2.2 GNU/Linux (2001):
  - Source lines of code: 55,201,526 (of which Linux forms under 6% – Stallman is correct!)
- If Debian was written in a software company:
  - Estimated effort: 14,005 person years
  - Estimated schedule: 6.04 years (team of 2,318!)
  - Development cost: US\$ 1,891,990,000
- Price of access to the public – and businesses!
  - Usage: Free; Copying: Free; Editing: Free



# Standing on shoulders

---

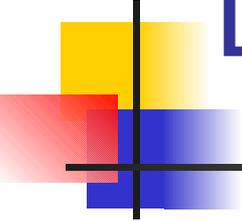
- Debian – the largest distribution of GNU/Linux – builds upon the work of over 10 000 individual programmers.
- Each of them build upon the work of others, without barriers to collaboration
- Result: a complex interaction of creators and a rapid, continuous form of innovation
- Shoulders can be, and are stood upon



# How is FLOSS developed?

---

- Bottom-up collaboration
- Collaboration in small, distributed pieces
- Joined together through:
  - Law (licences) reduce entry barriers
  - Technology (Net, modularity) to integrate
- Most individual contributions are small
- Motives are diverse, this doesn't matter

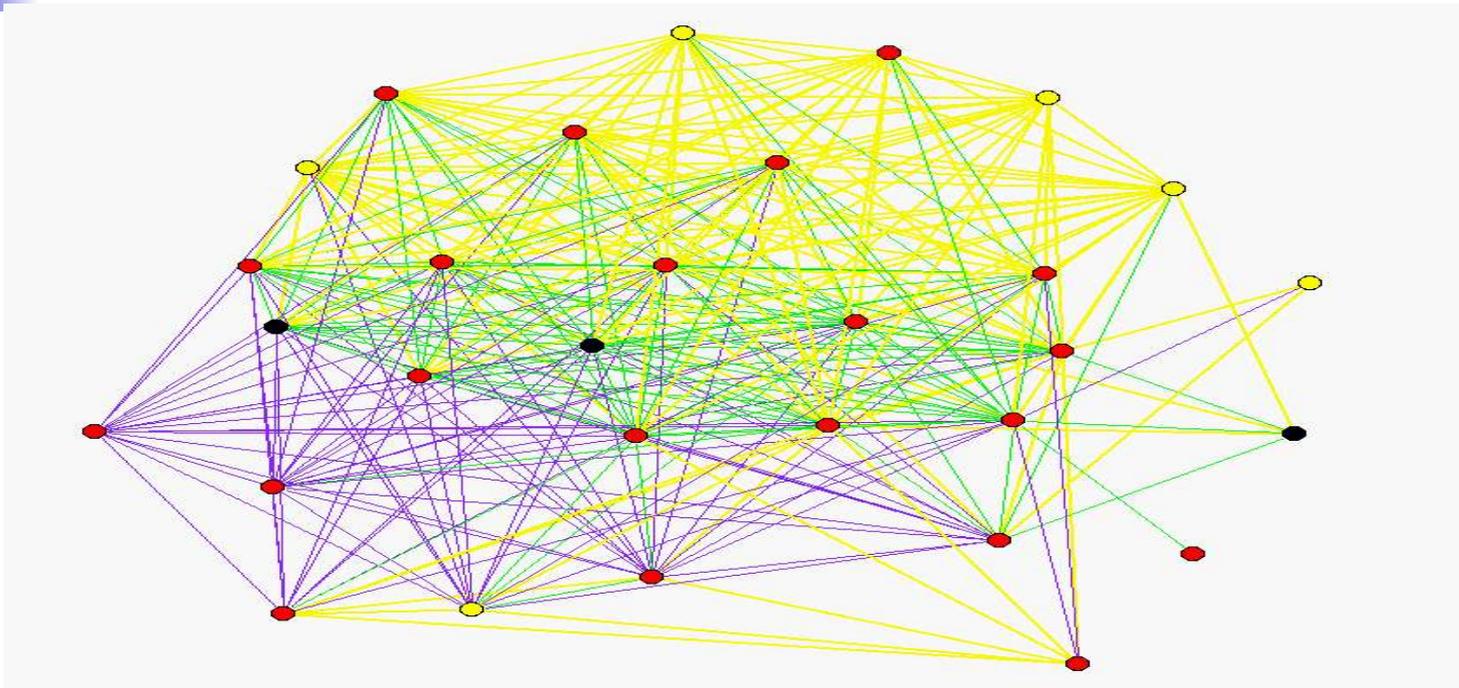


# Development: example Linux

---

- The Linux kernel was first written by Linus Torvalds, a student in Finland in 1990, and released over the Internet under the General Public Licence (GPL)
- GPL says: this provides the 4 freedoms, but if you modify it, your modified version must be free software too
- GPL + Internet allowed many authors...

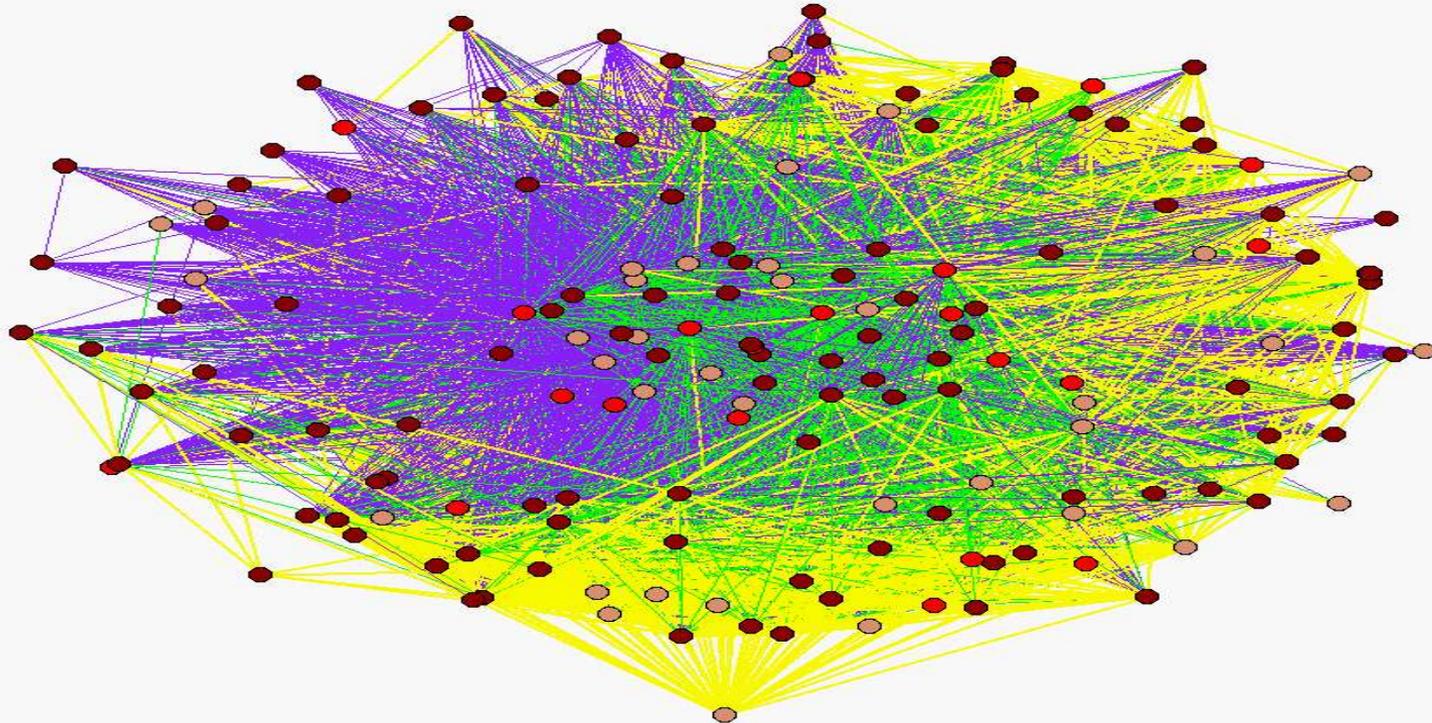
# Linux kernel v1.0: 158 authors



1994. Nodes are 30 modules. Arcs represent **common authors**, **code dependencies**, or **both**

(Source: "Nature and composition... ", Ghosh & David)

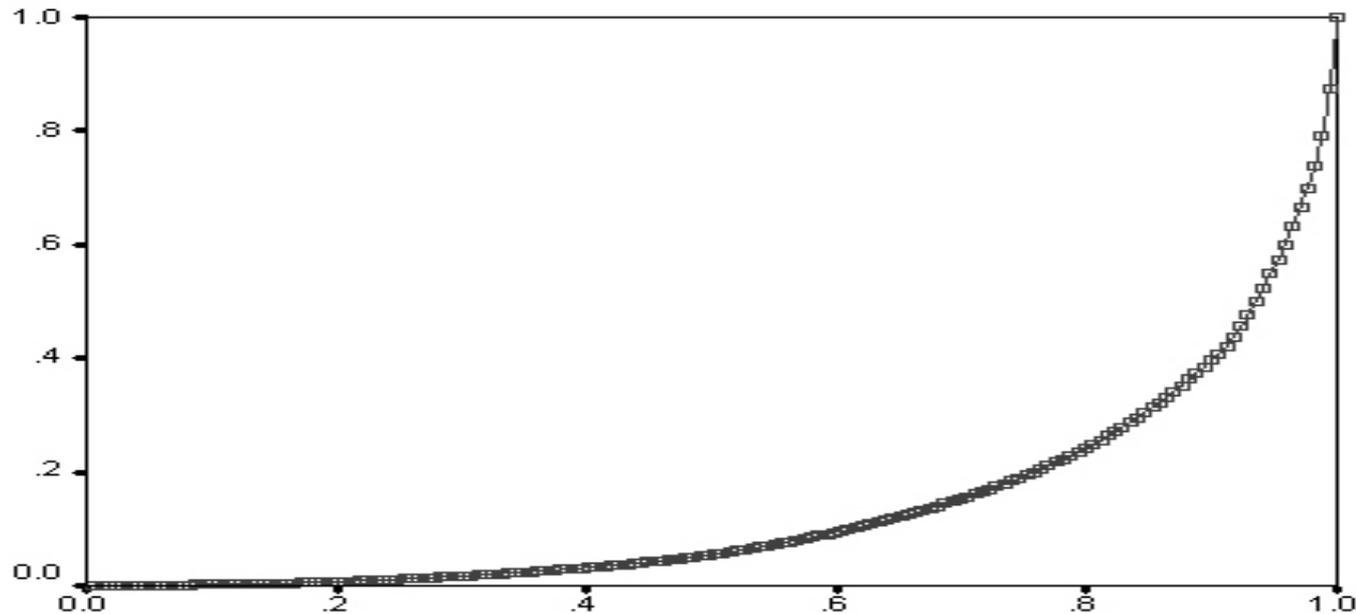
# Linux kernel v2.5.25: 2263 authors



2002. Nodes are 169 modules. Arcs represent **common authors**, **code dependencies**, or **both**

(Source: "Nature and composition... ", Ghosh & David)

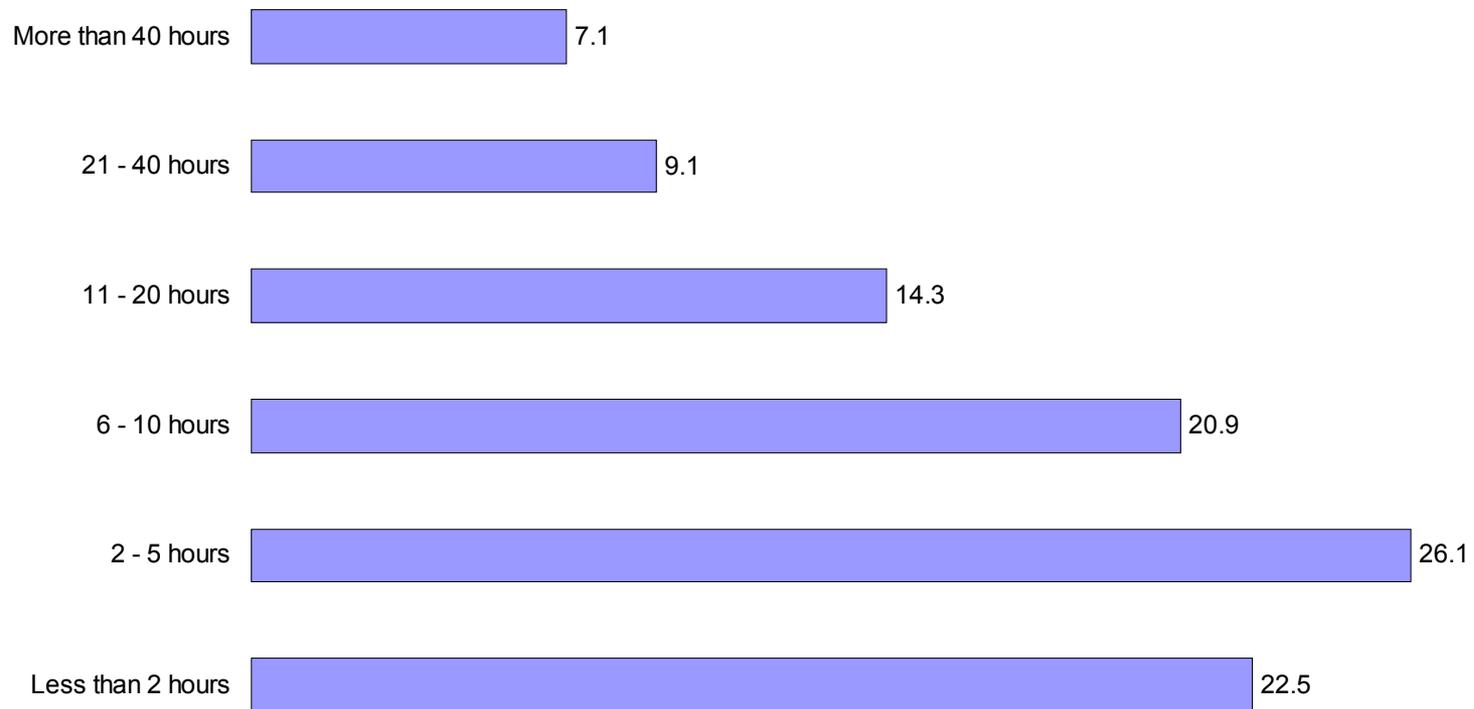
# Linux kernel v2.5.25: 2263 authors



Share of authors (X) vs share of code contributed (Y).  
Bottom 80% contributes under 30% of code.

(Source: "Nature and composition...", Ghosh & David)

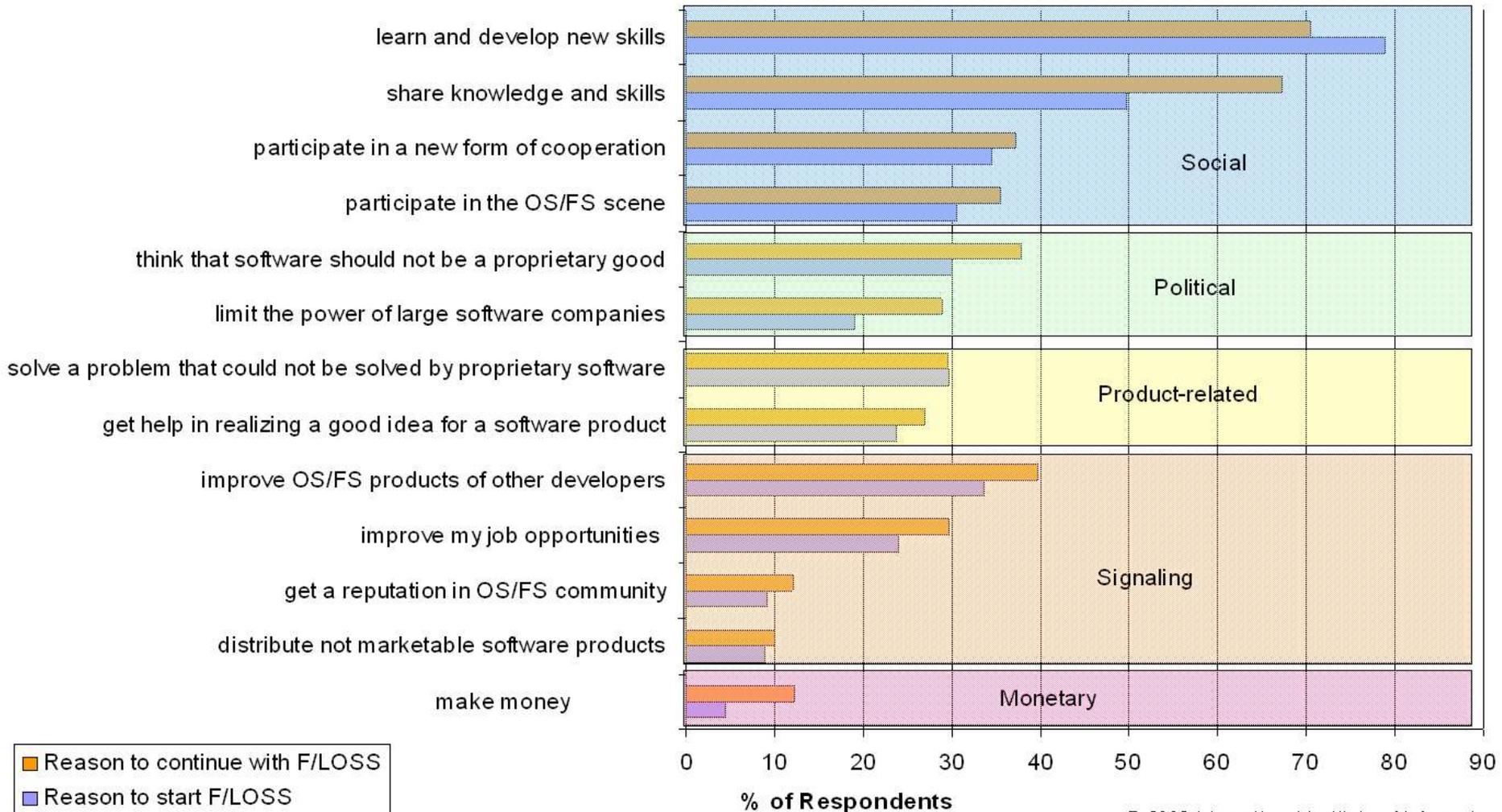
# Weekly hours per developer, %



© 2002 International Institute of Infonomics

(Across a range of FLOSS projects, 2002. Source: "FLOSS Final Report, Developer Survey", Ghosh et al)

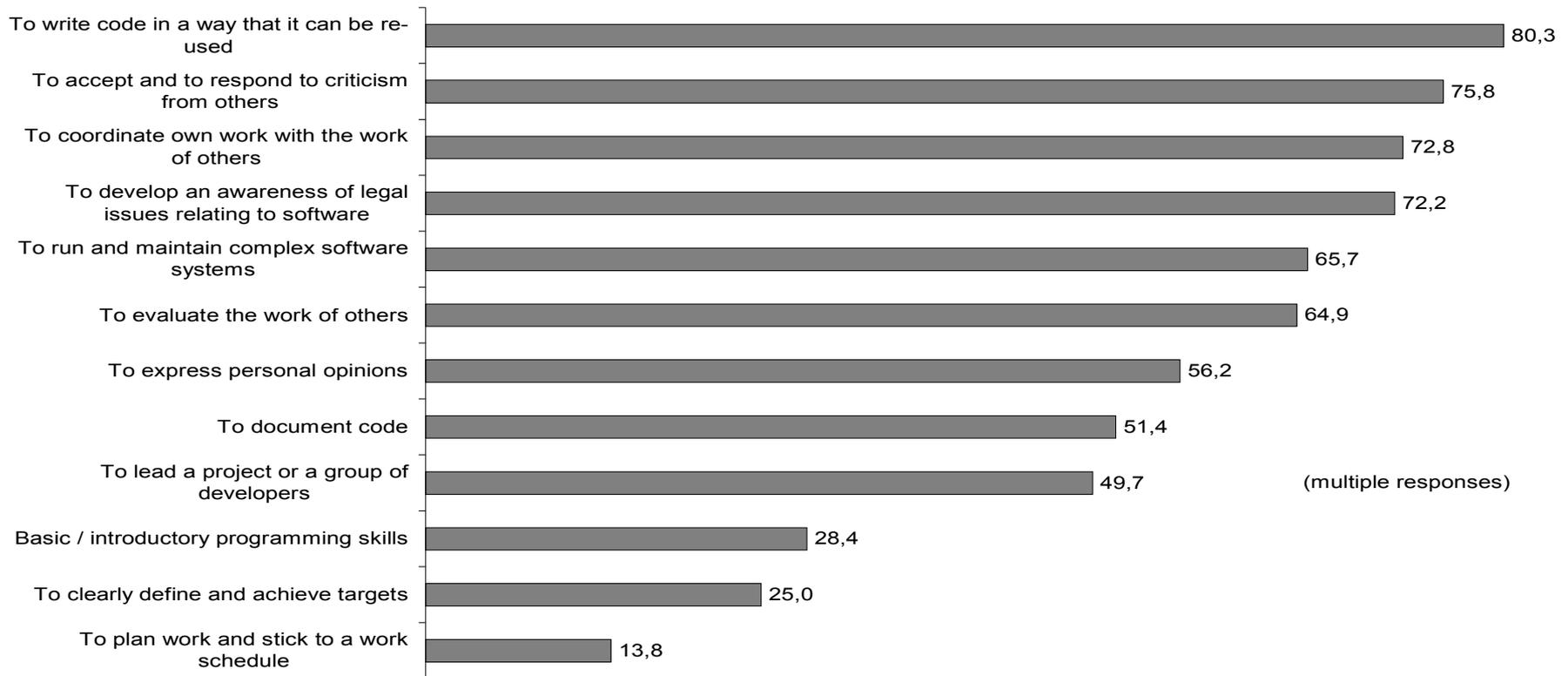
# Why develop FLOSS?



(Source: "FLOSS Final report", Ghosh et al)

# Participation transfers skills...

"Which of the following skills can be better learnt within the FLOSS community as compared to a formal computer science course?"



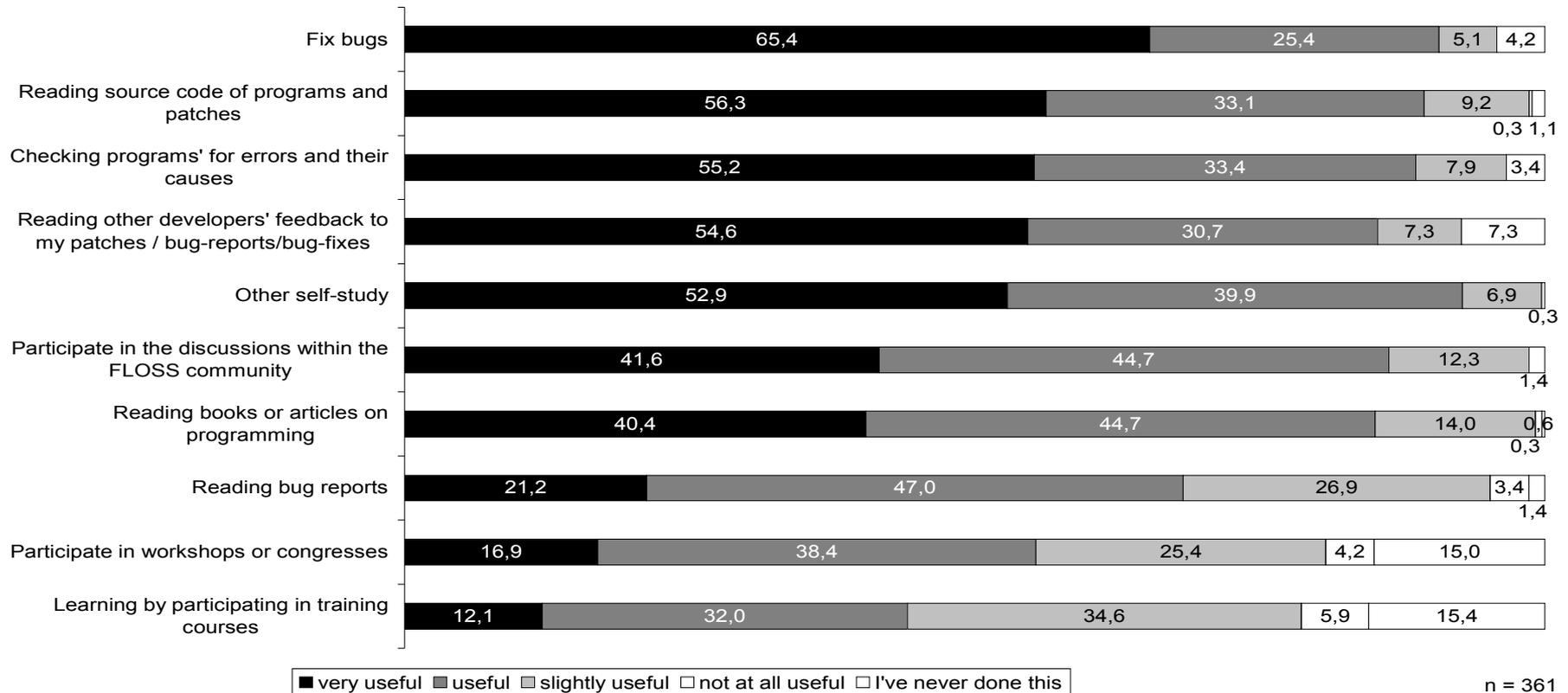
(multiple responses)

n = 361

(Source: FLOSSPOLs skills survey, 2004-2005, Ghosh et al)

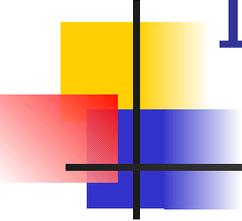
# Participation transfers skills...

"How useful do you personally consider the following activities as ways of learning technical skills?"



n = 361

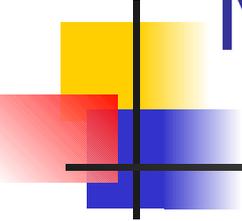
(Source: FLOSSPOLs skills survey, 2004-2005, Ghosh et al)



## In conclusion

---

- Innovation results from incentives OR from the innate urge to create
- Policy focuses on proprietary innovation
- Open innovation should be supported, not ignored or hindered
- FLOSS: exemplar of successful open innovation
- FLOSS provides a model for collaborative innovation in other sectors e.g. biotech



## More information

---

CALIBRE: <http://www.calibre.ie>

FLOSSPOLS: <http://flosspols.org>

References:

- FLOSS report: <http://flossproject.org/report/>
- “Nature and composition...” (Ghosh & David):  
<http://dxm.org/papers/licks1/>
- “Counting Potatoes” (Gonzalez-Barahona et al):  
<http://libresoft.dat.escet.urjc.es>