

# Estimating Development Effort in Free/Open Source Software Projects by MSR

## A Case Study of OpenStack

Gregorio Robles, Jesús M. González-Barahona, Carlos Cervigón,  
Andrea Capiluppi, Daniel Izquierdo-Cortázar

Universidad Rey Juan Carlos, Brunel University, Bitergia S.L.

MSR 2014, Hyderabad, June 1st 2014



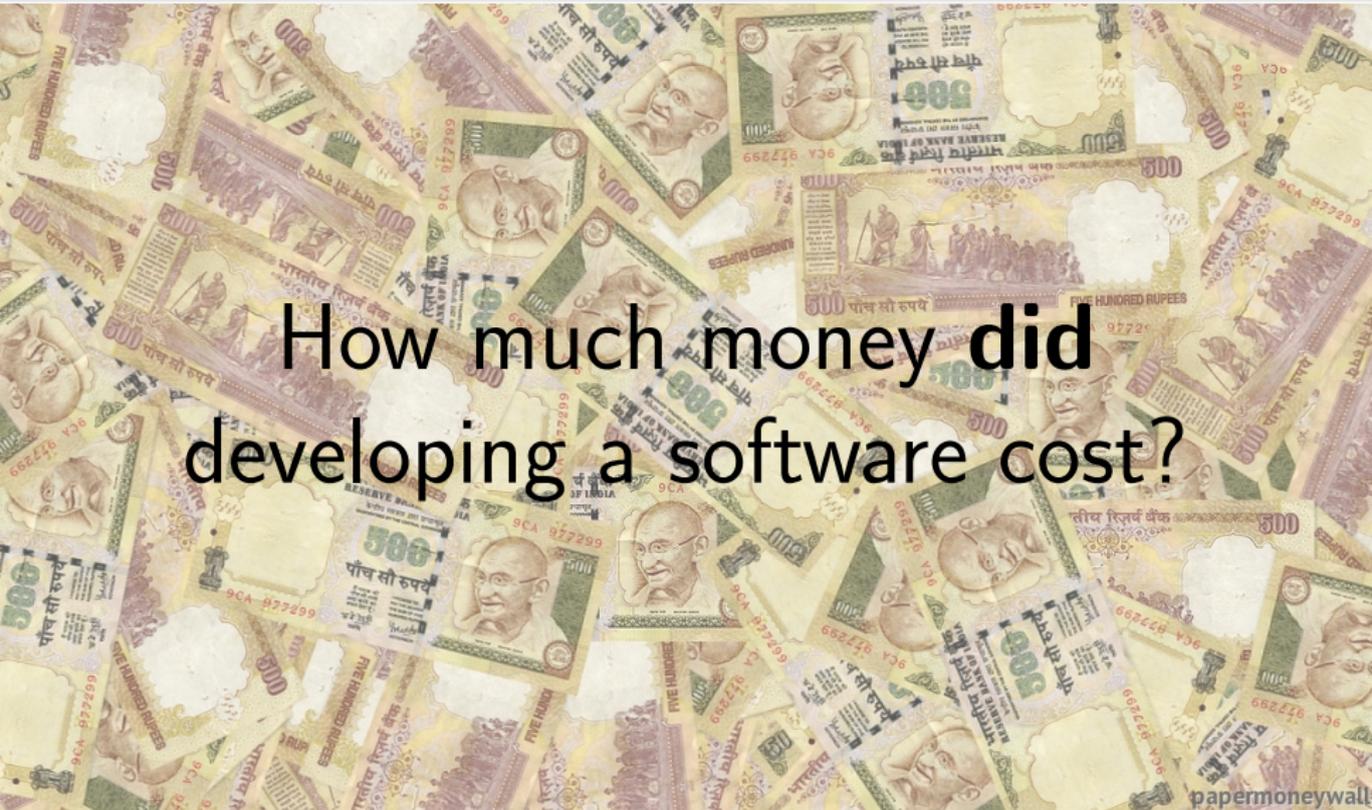


(cc) 2014 The authors  
Some rights reserved. This work licensed under Creative Commons  
Attribution-ShareAlike License. To view a copy of full license, see  
<http://creativecommons.org/licenses/by-sa/3.0/> or write to  
Creative Commons, 559 Nathan Abbott Way, Stanford,  
California 94305, USA.

Some of the figures have been taken from the Internet  
Source, and author and licence if known, is specified.

For those images, *fair use* applies.

The question is...

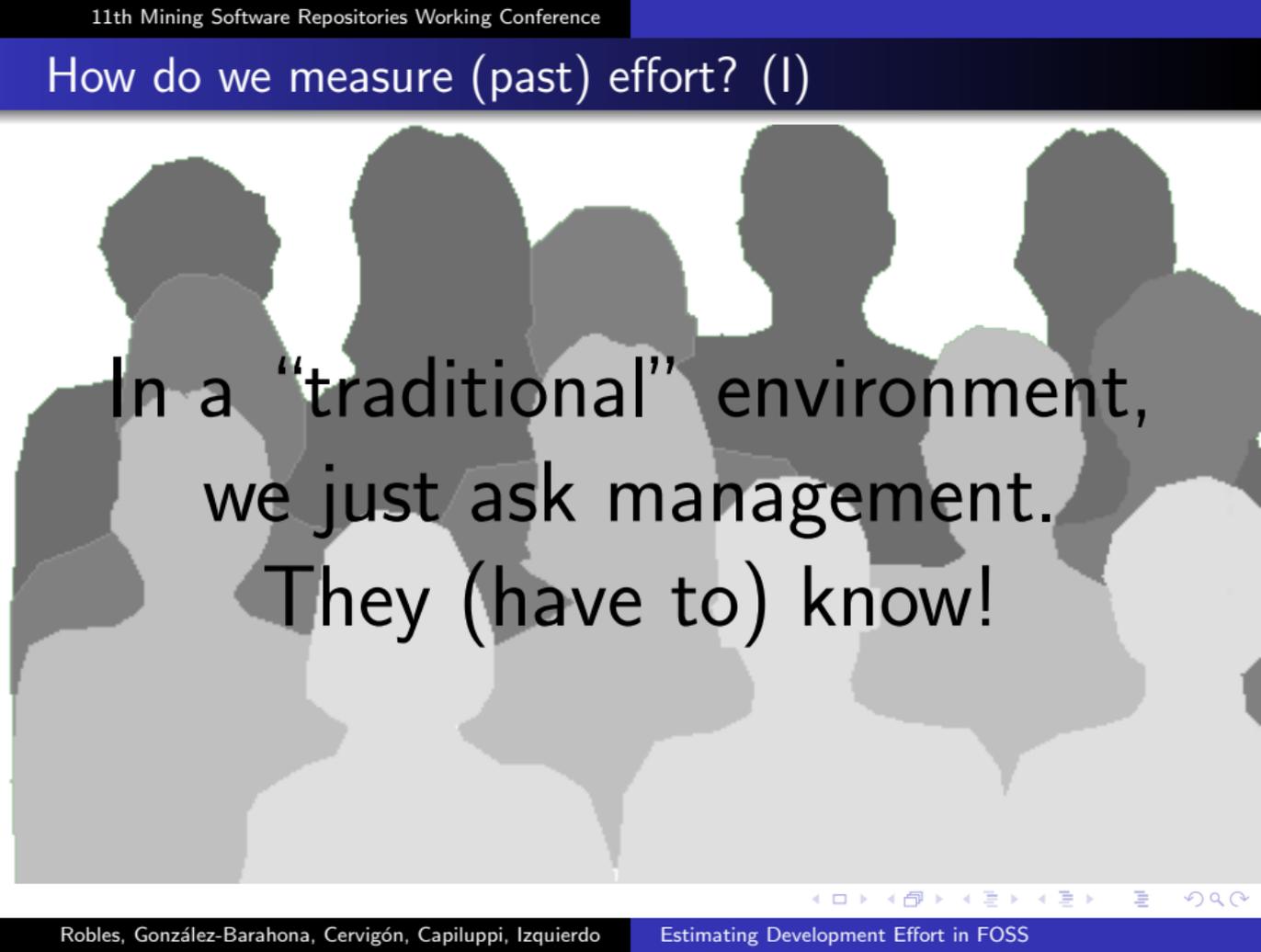


How much money **did**  
developing a software cost?

papermoneywall



## How do we measure (past) effort? (I)

The background of the slide features a group of approximately ten stylized, semi-transparent grey silhouettes of people's heads and shoulders, arranged in a loose cluster. The text is overlaid on this background.

In a “traditional” environment,  
we just ask management.  
They (have to) know!

## How do we measure (past) effort? (and II)



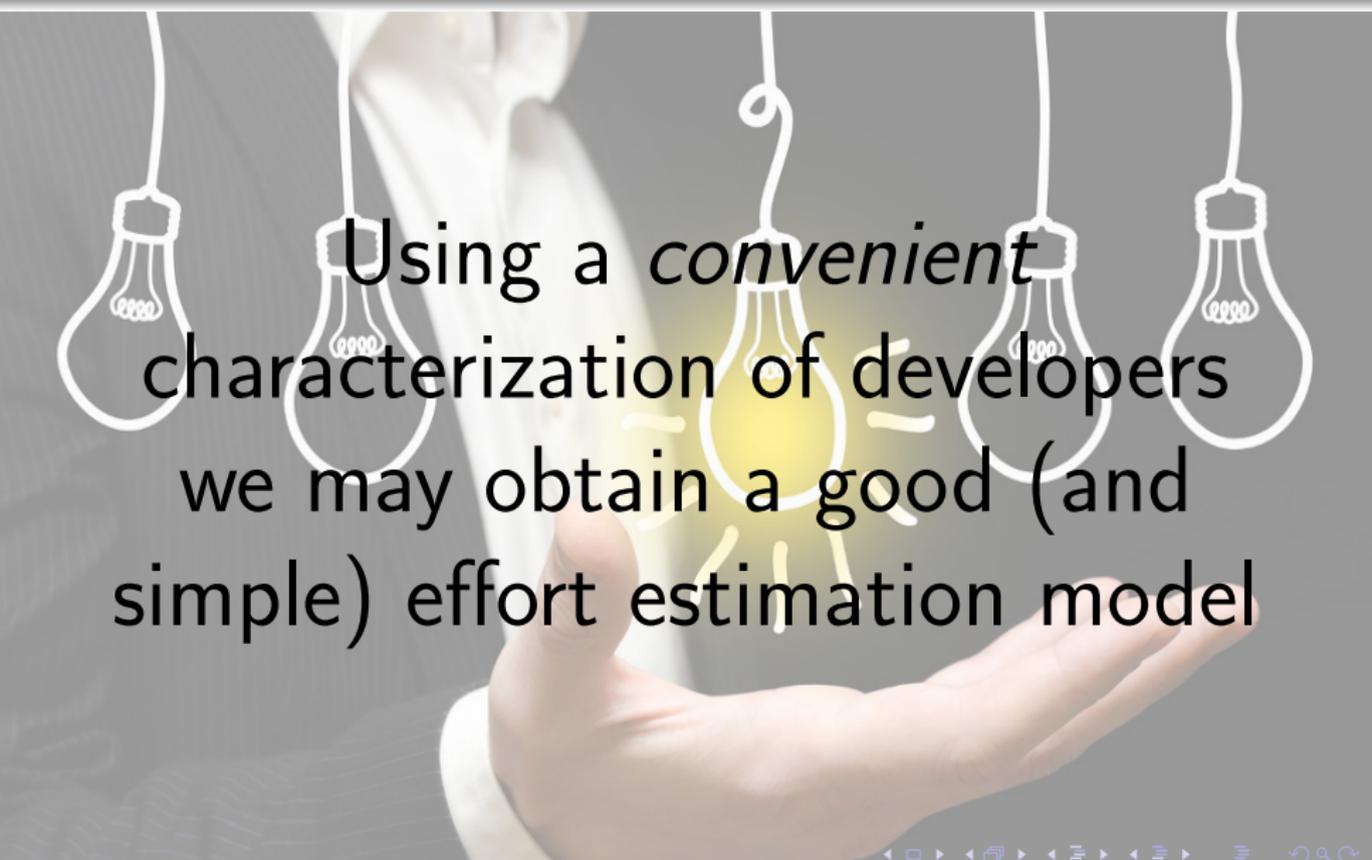
How about Free/Open Source  
Software?

## First (maybe bad) idea

We can track developers in all repositories and sum all their micro-contributions

The background of the slide features a stylized illustration. On the left, a large, dark blue silhouette of a person's head and shoulders is shown in profile, looking towards the right. A magnifying glass is held over the person's eye, with its handle extending downwards. In the lower right corner, a smaller silhouette of a person is seated at a desk, working on a computer. The entire scene is set against a bright yellow background.

## Our approach

A hand in a white shirt is shown holding a glowing yellow lightbulb. Several other lightbulbs are hanging from above, some of which are outlined in white. The background is a blurred image of a person in a white shirt.

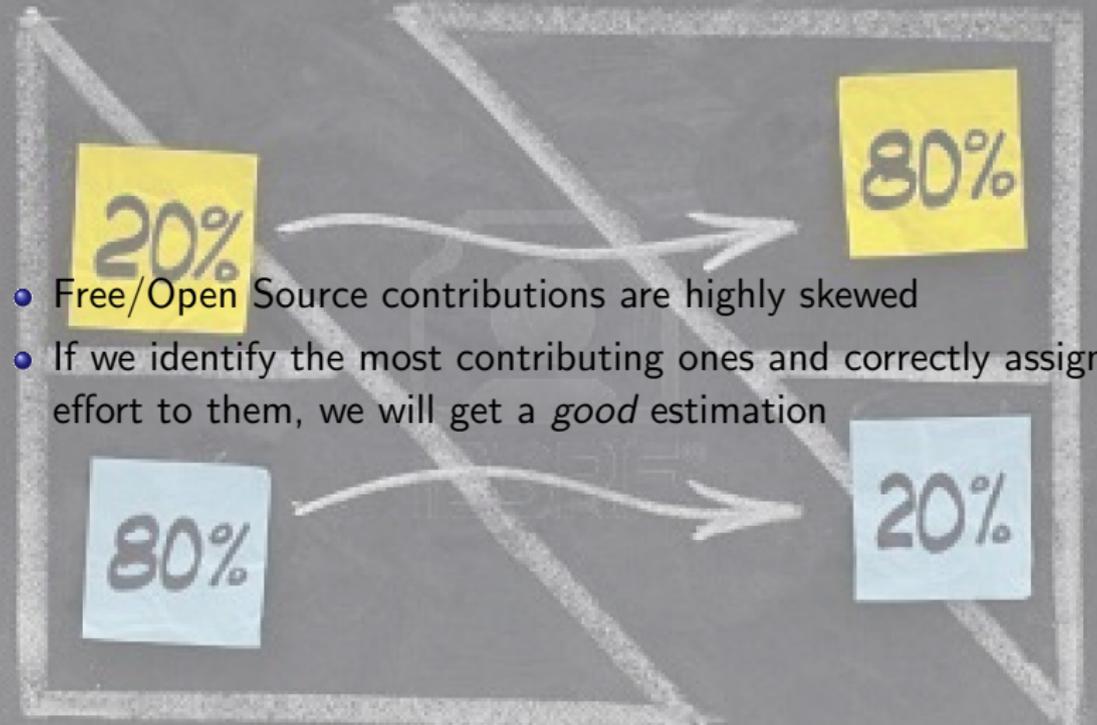
Using a *convenient* characterization of developers we may obtain a good (and simple) effort estimation model

Model: We divide developers in two groups



Full-time and non-full-time  
developers

# Why?

- 
- Free/Open Source contributions are highly skewed
  - If we identify the most contributing ones and correctly assign effort to them, we will get a *good* estimation

# The Model

- 1 We define a time period  $T$
- 2 Any developer with more commits than a threshold  $t$  in a time period  $T$  is considered a full-time developer:

$$\text{Effort (in person-months)} = \begin{cases} T & \text{if full-time} \\ T * x/t & \text{else, } x = \text{commits in } T \end{cases}$$

# An example

- 1 A project has 3 developers
- 2 Their number of commits in the last 3 months is 100, 10, 3
- 3 Let's assume timespan  $T = 3$  months and threshold  $t = 30$  commits
- 4 The effort spent in this project in the last three months is
$$3 * (1 + 1/3 + 3/30)$$
$$= 3 * 1.433 = 4.3 \text{ person-months}$$

# Case study: OpenStack



We know the project very well and we know people there.

# What we are looking for



What is the best value for the timespan  $T$ ?

What is the best value for the threshold  $t$ ?

# Naïve Approach

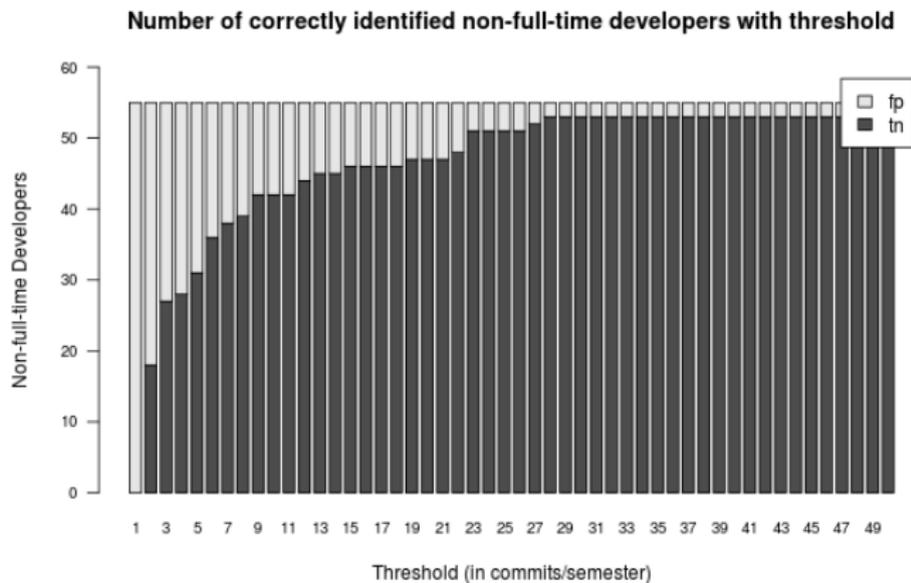
$T = 1$  release = 6 months

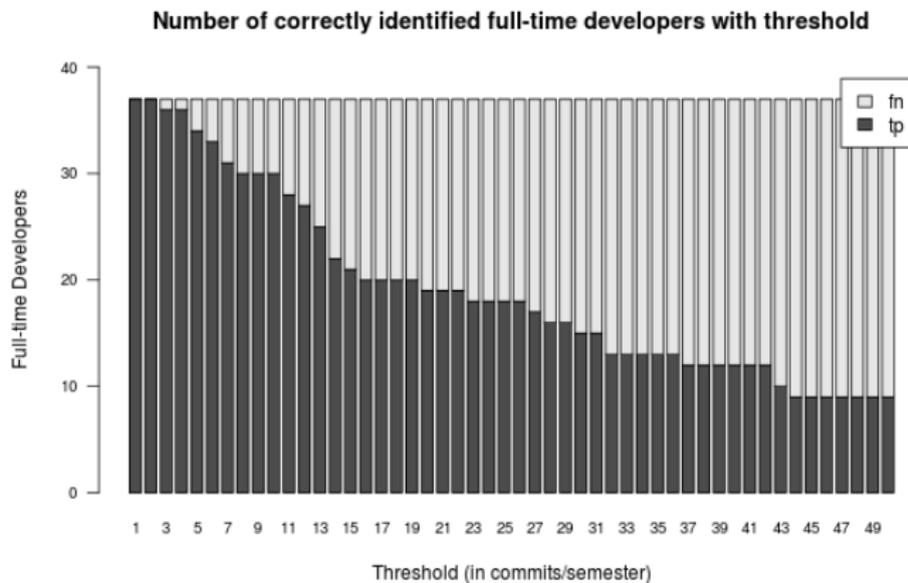
$t = 1$  commit

(used by the OpenStack community as a naïve estimation)

# How have we proceeded?

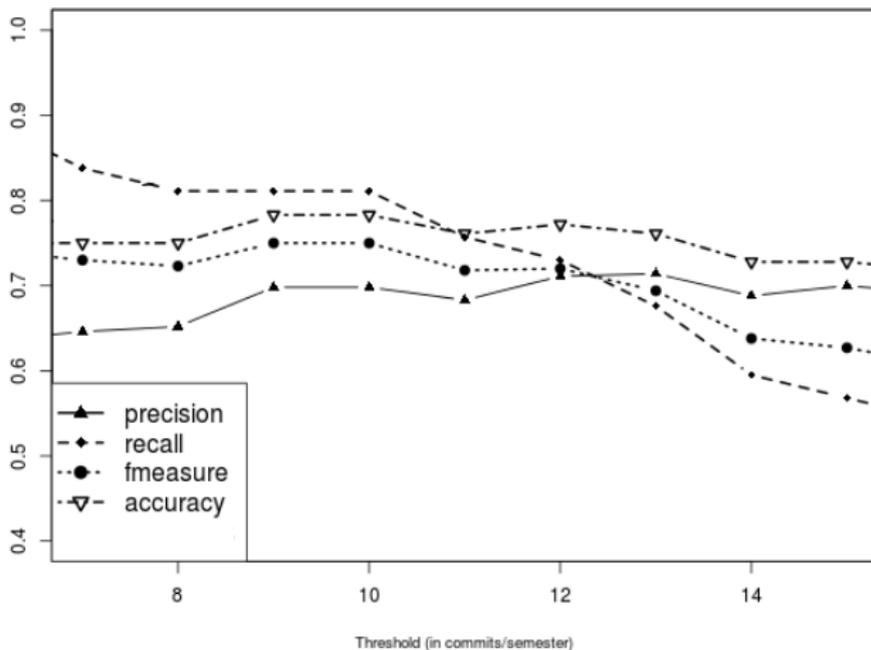
- We obtained data from their git repository
- Looking for the timespan  $T$ : 6 months (following OpenStack's release periodicity)
- Looking for the threshold  $t$ : We asked (active) developers in a survey on their status: are you full-time in OpenStack?
  - More than 100 responses obtained! ( $> 10\%$  of the OpenStack active developers, statistically representative of the entire population)



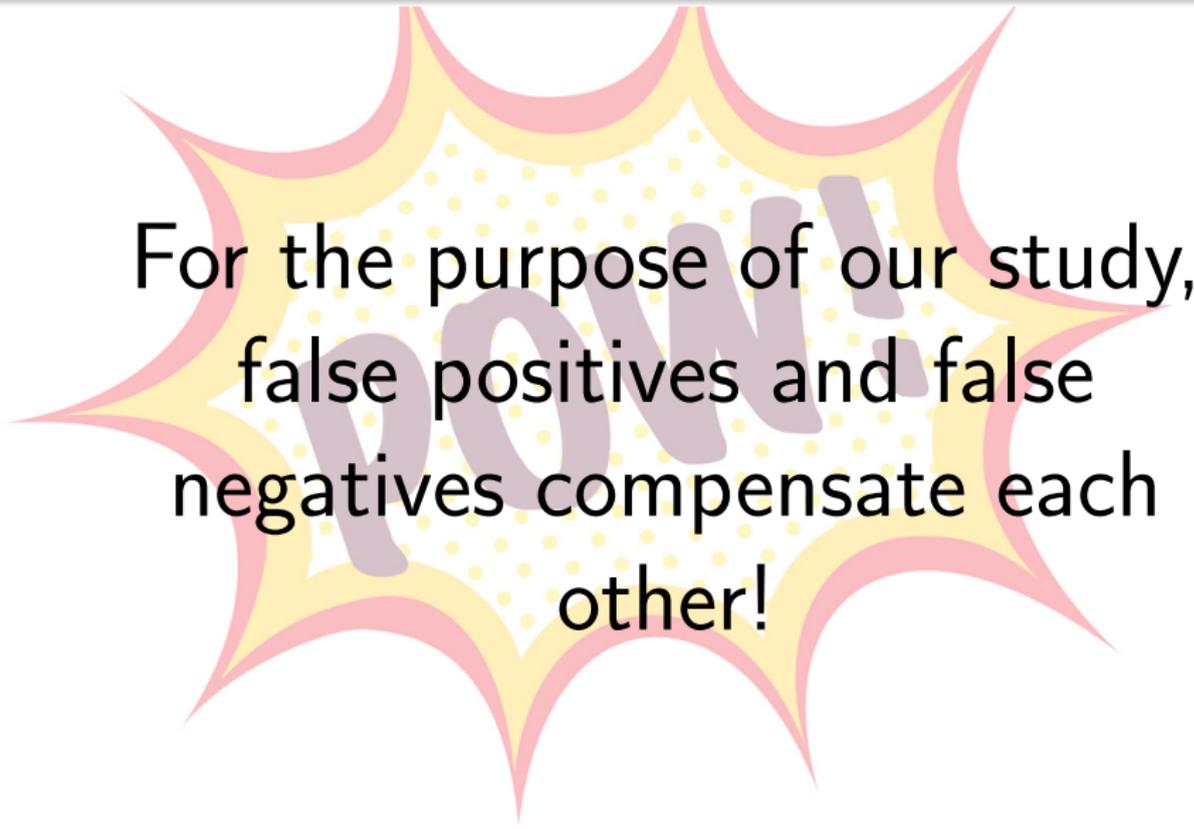


# Precision, recall and so on...

Relevance of results (zoom for best values)

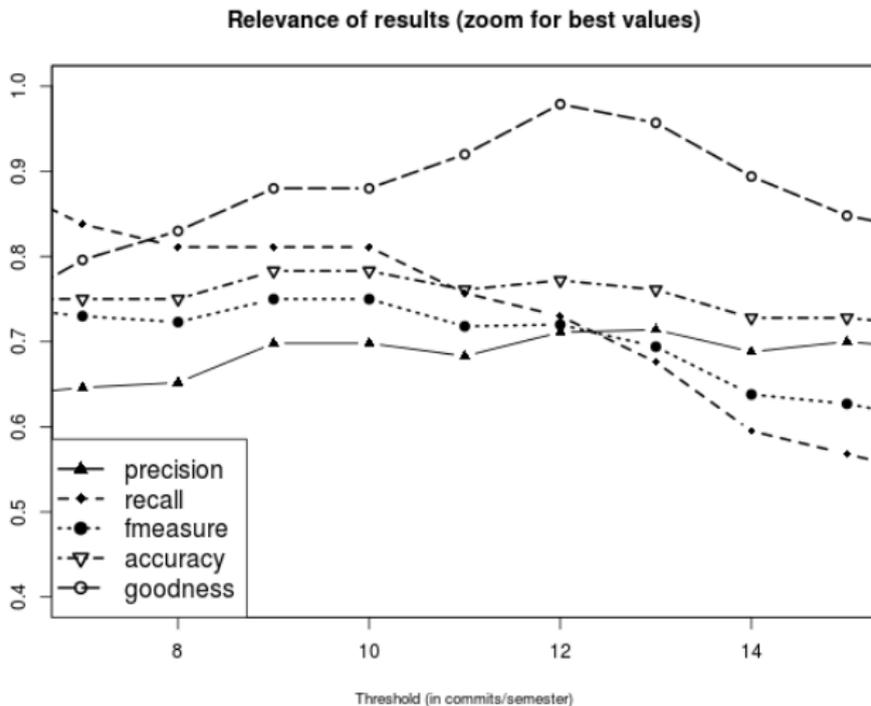


## Interestingly...

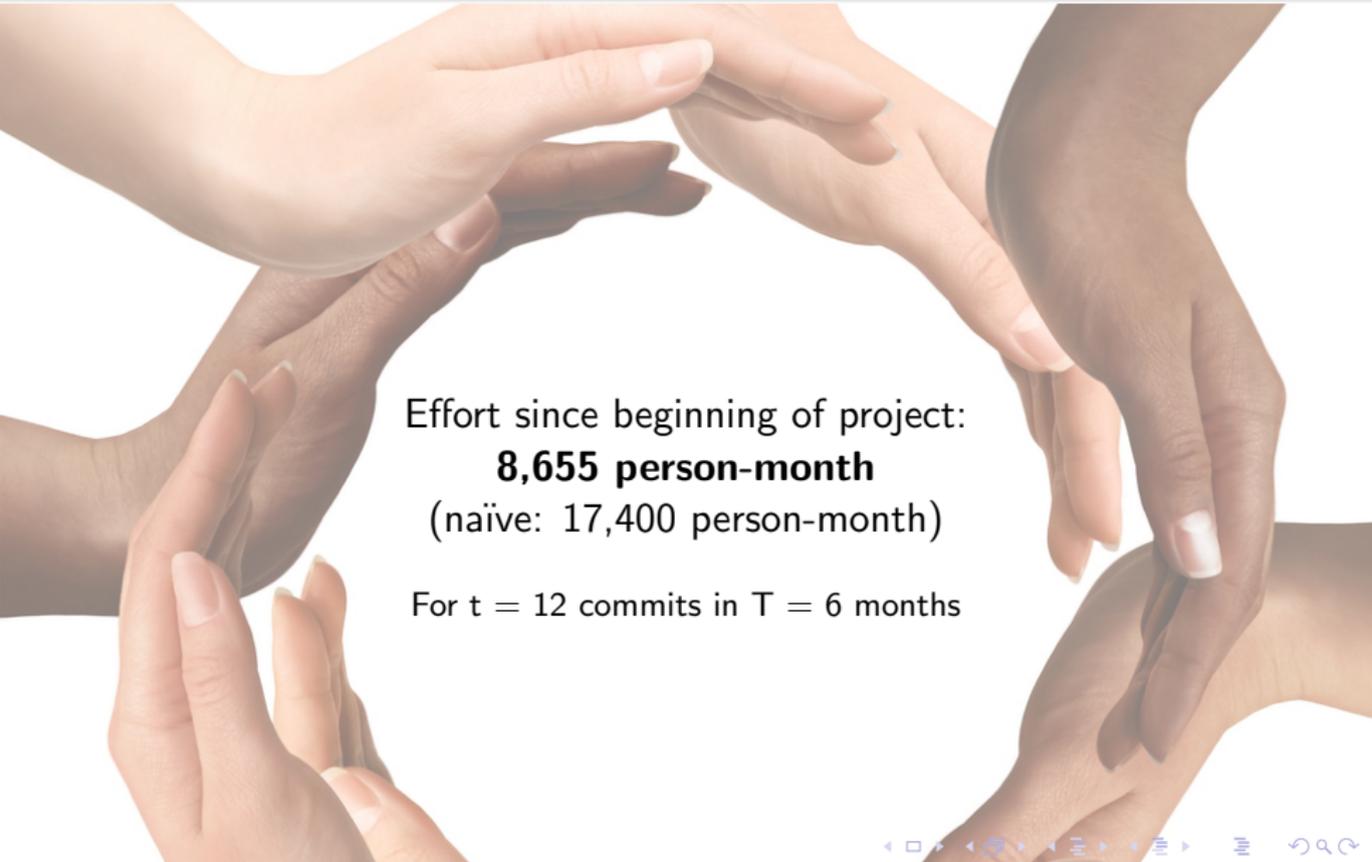


For the purpose of our study,  
false positives and false  
negatives compensate each  
other!

# False positives and false negatives compensate each other!



# Results (I)



Effort since beginning of project:  
**8,655 person-month**  
(naïve: 17,400 person-month)

For  $t = 12$  commits in  $T = 6$  months

## Results (and II)



Effort for the last release:  
**2,634 person-month**  
(naïve: 5,916 person-month)

440 person-months each month  
~250 professional developers work on  
OpenStack hired by companies

For  $t = 12$  commits in  $T = 6$  months

# Summary

Given a repository, we provide a **simple** way of estimating **past** development effort

Two parameters have to be determined:  
timespan  $T$  and threshold  $t$

A value of threshold  $t$  of 12  
has been obtained as the best for OpenStack

# Preview of current work

## Best threshold $t$ value for following projects

**Linux:** 18 commits every 6 months  
653 out of 3,665 (17.8%) developers responded

**MediaWiki:** 29 commits every 6 months  
95 out of 605 (15.7%) developers responded

**WebKit:** 17-25 commits every 6 months  
86 out of 690 (12.5%) developers responded

**Moodle:** 15 commits every 6 months  
43 out of 174 (24.7%) developers responded

# Estimating Development Effort in Free/Open Source Software Projects by MSR

## A Case Study of OpenStack

Gregorio Robles, Jesús M. González-Barahona, Carlos Cervigón,  
Andrea Capiluppi, Daniel Izquierdo-Cortázar

Universidad Rey Juan Carlos, Brunel University, Bitergia S.L.

MSR 2014, Hyderabad, June 1st 2014

