# Code **Janitor**

nobody's dream, everyone's job
(and how Erlang can help)

# Who?

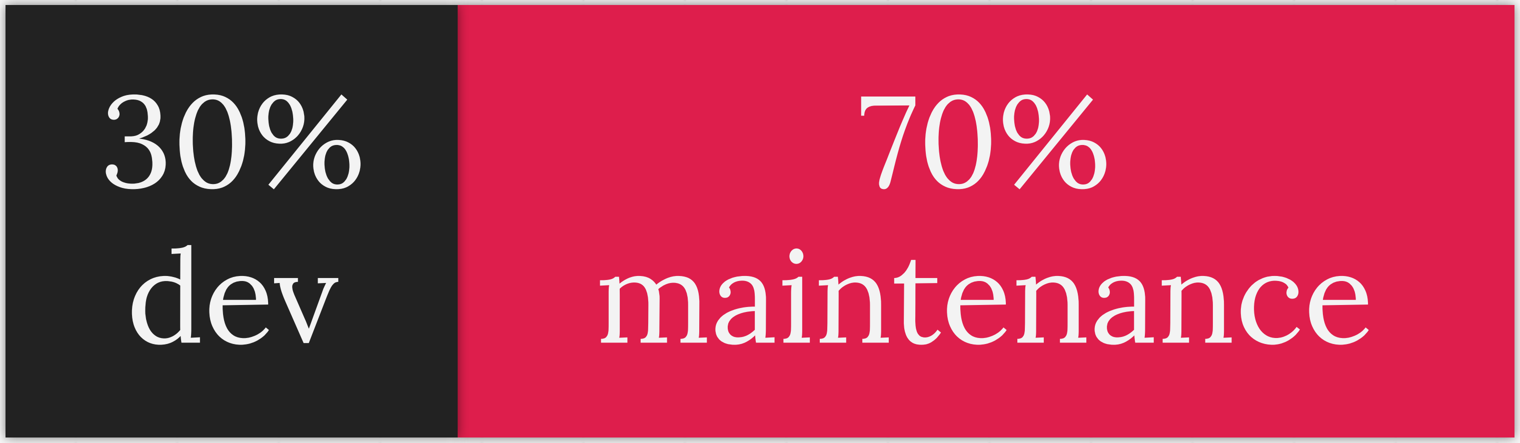
- Learn You Some Erlang
- Erlang Solutions Ltd.
- AdGear Technologies
- Erlang User of The Year 2012
- Erlang shell history
- Heroku

Fred Hebert

Twitter: @MononcQc
Blog: http://ferd.ca

Maintenance is the price to pay to earn the right to write new code.

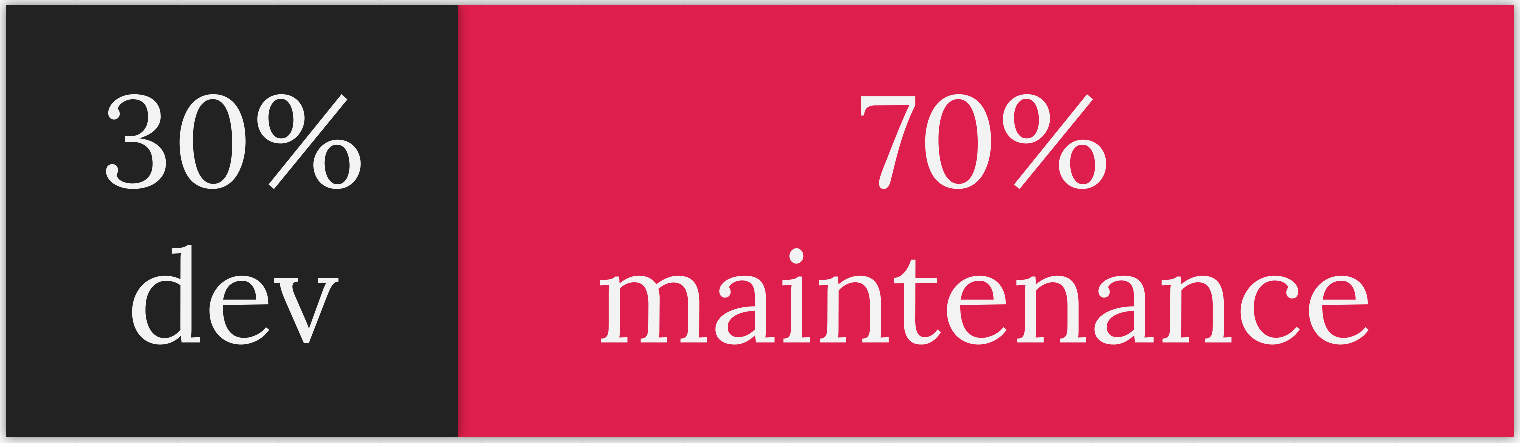
# *Time spent*



30% dev

70% maintenance

# Time *spent*



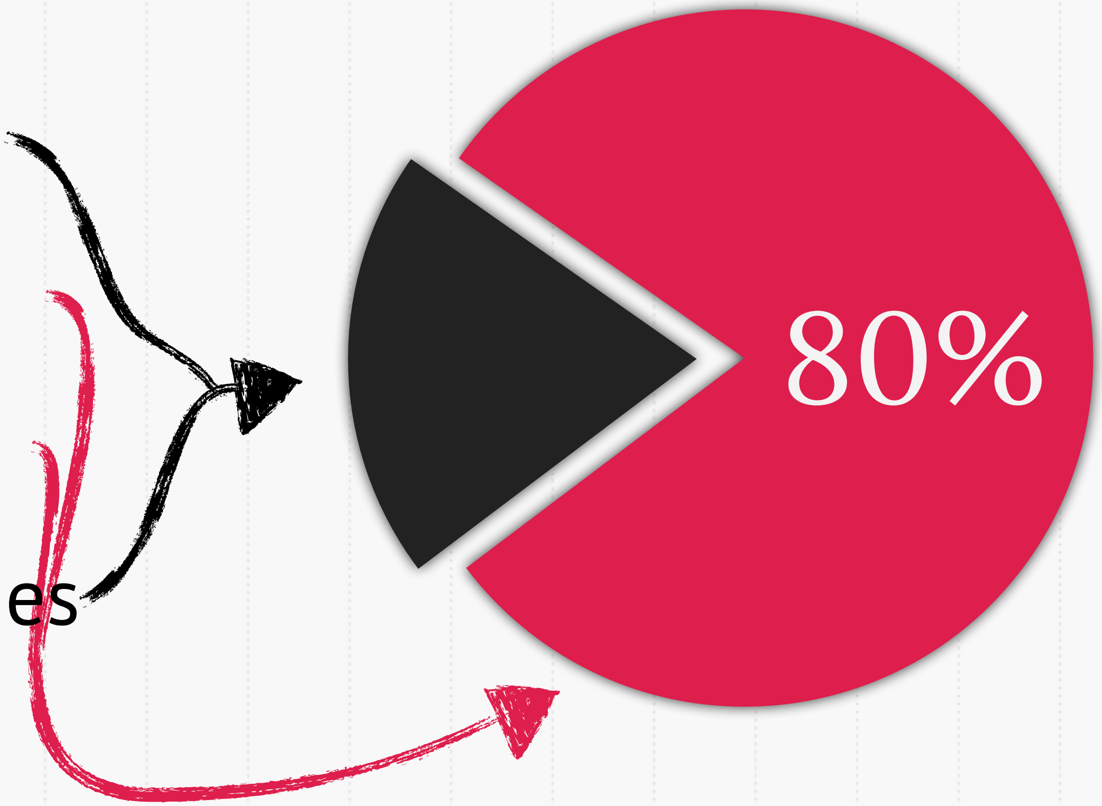
30% dev

70% maintenance

60% à 80% of costs

# Types of maintenance

- Corrective
- Adaptive
- Perfective
- Emergencies

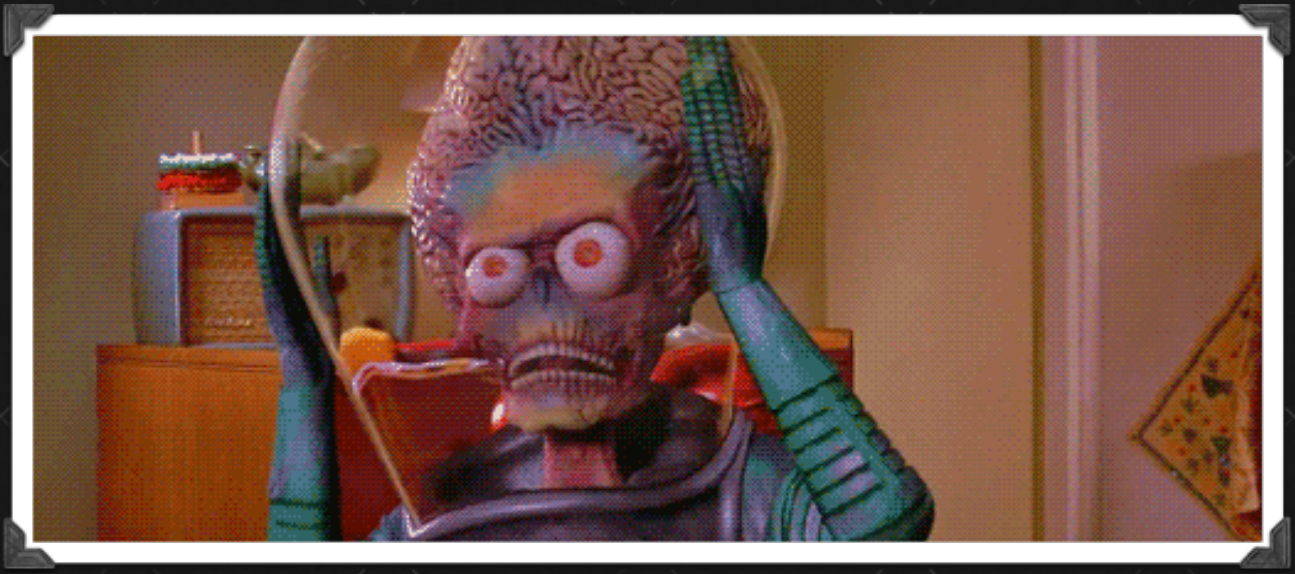


# Lehman's laws

- A program that is used and that as an implementation of its specification reflects some other reality, undergoes continual change or becomes progressively less useful. The change or decay process continues until it is judged more cost effective to replace the system with a recreated version.

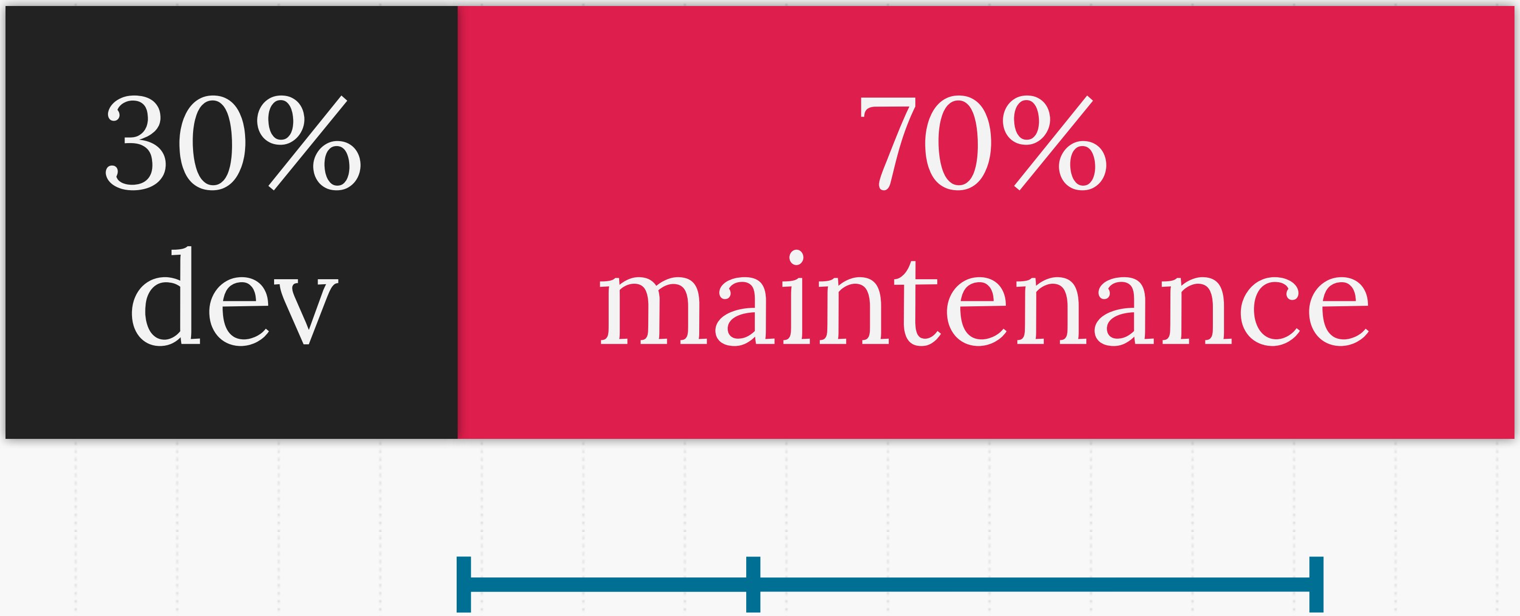
# Lehman's laws

- As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.



The problem with complexity is **purely human**.

# Consequences?



30%
dev

70%
maintenance

22.5% to 57.6% of any software project is spent trying to understand the system

# *Information* ***sources***

~ Code

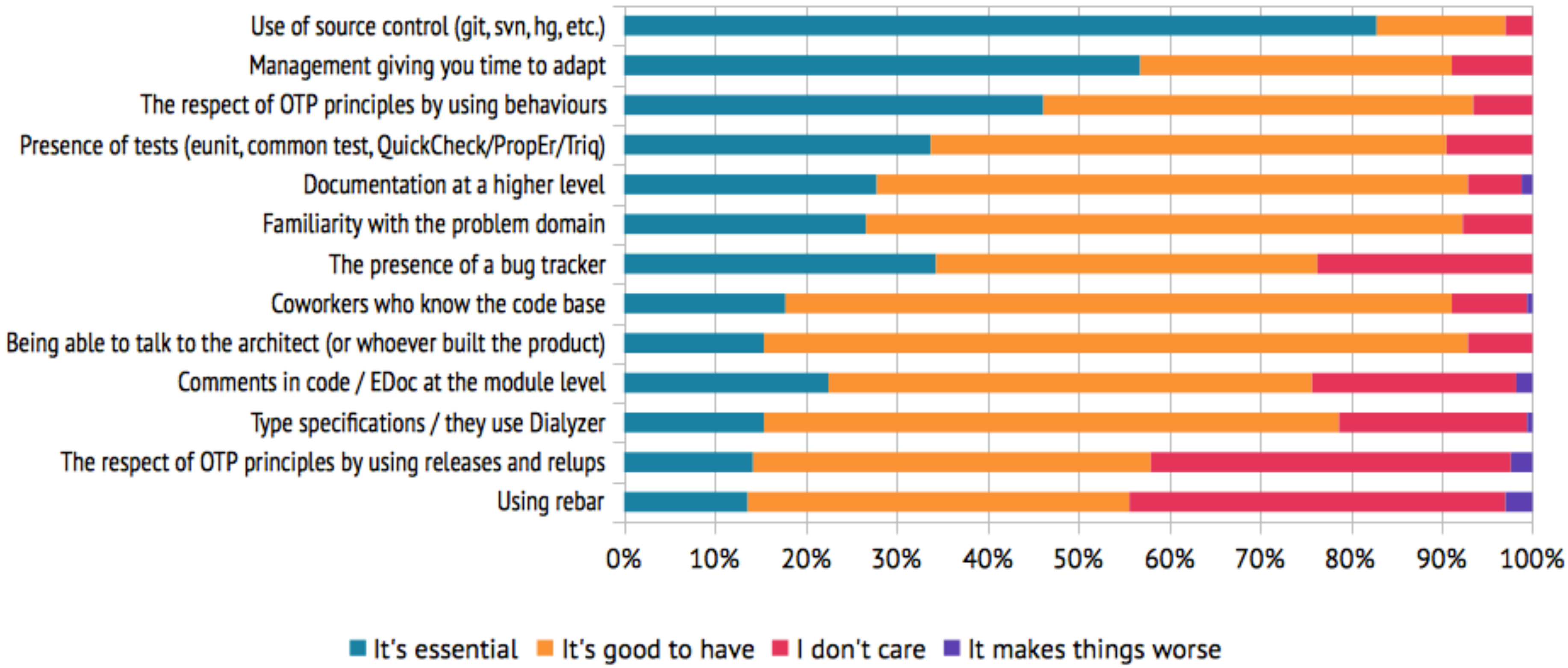
~ Coworkers

~ Tools, knowledge bases (bug trackers, etc.)

~ Documentation

Erlang can help

*You are hired to take over an existing Erlang code base. Based on your experience, what do you think is important for you to feel comfortable and take 'ownership' of the code?*



# Why OTP Matters

The importance of *behaviours*

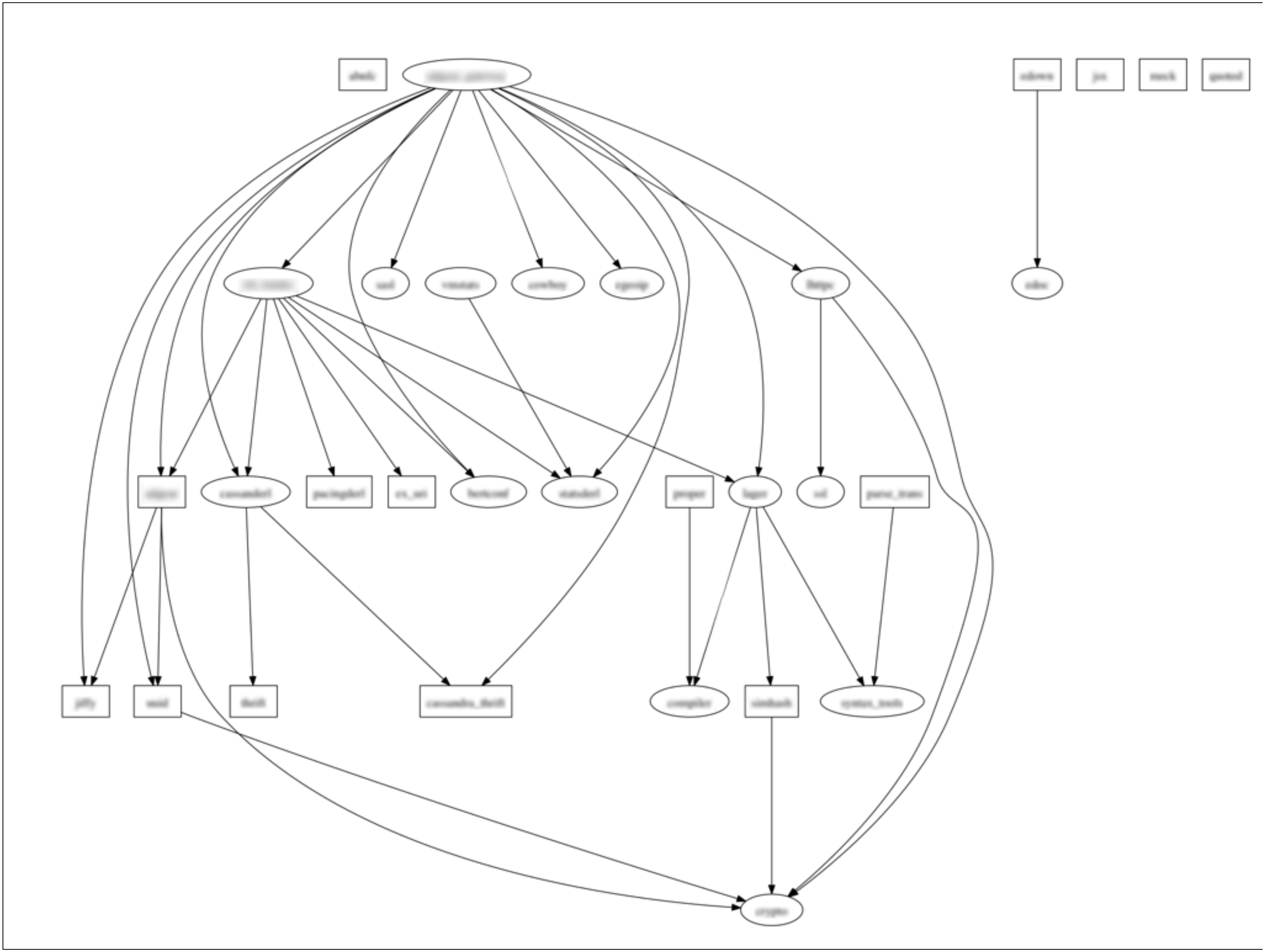
# Components

- Workers
- Supervisors
- Applications

# Turning Jenga into Lego

# Protocols

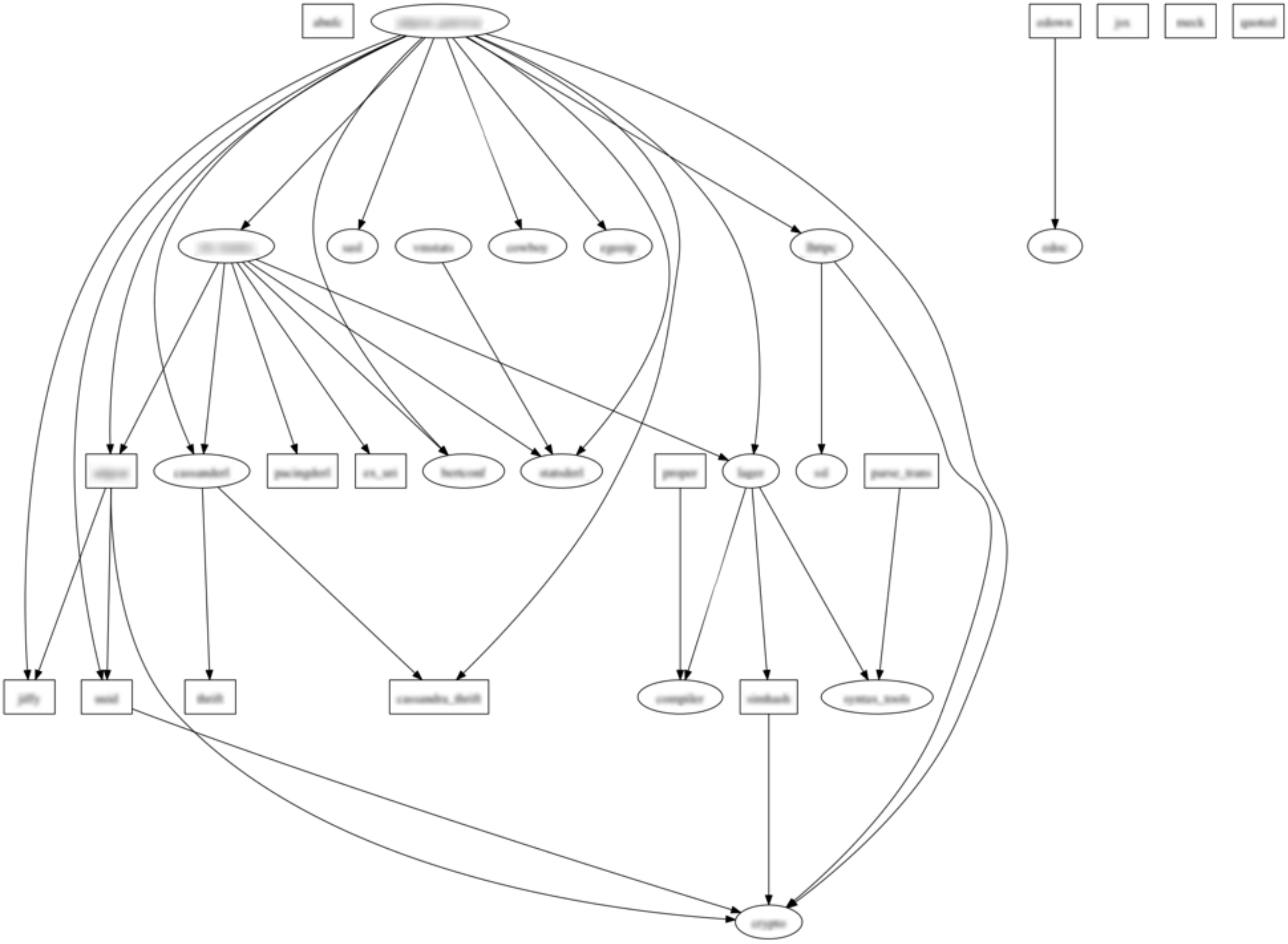
- Isolation
- Message passing
- Define standards in order to structure the abstract (ex: TCP/IP, BitTorrent Wire, HTTP)
- Applications and internal OTP protocols


abnfc
ssl
vmstats
cowboy
egeoip
lhttpc
edown
jsx
meck
quoted
edoc
cassanderl
pacingderl
ex_uri
bertconf
statsderl
proper
lager
ssl
parse_trans
jiffy
uuid
thrift
cassandra_thrift
compiler
simhash
syntax_tools
crypto

Modifications *without* Erlang/OTP

# *Understanding from afar*

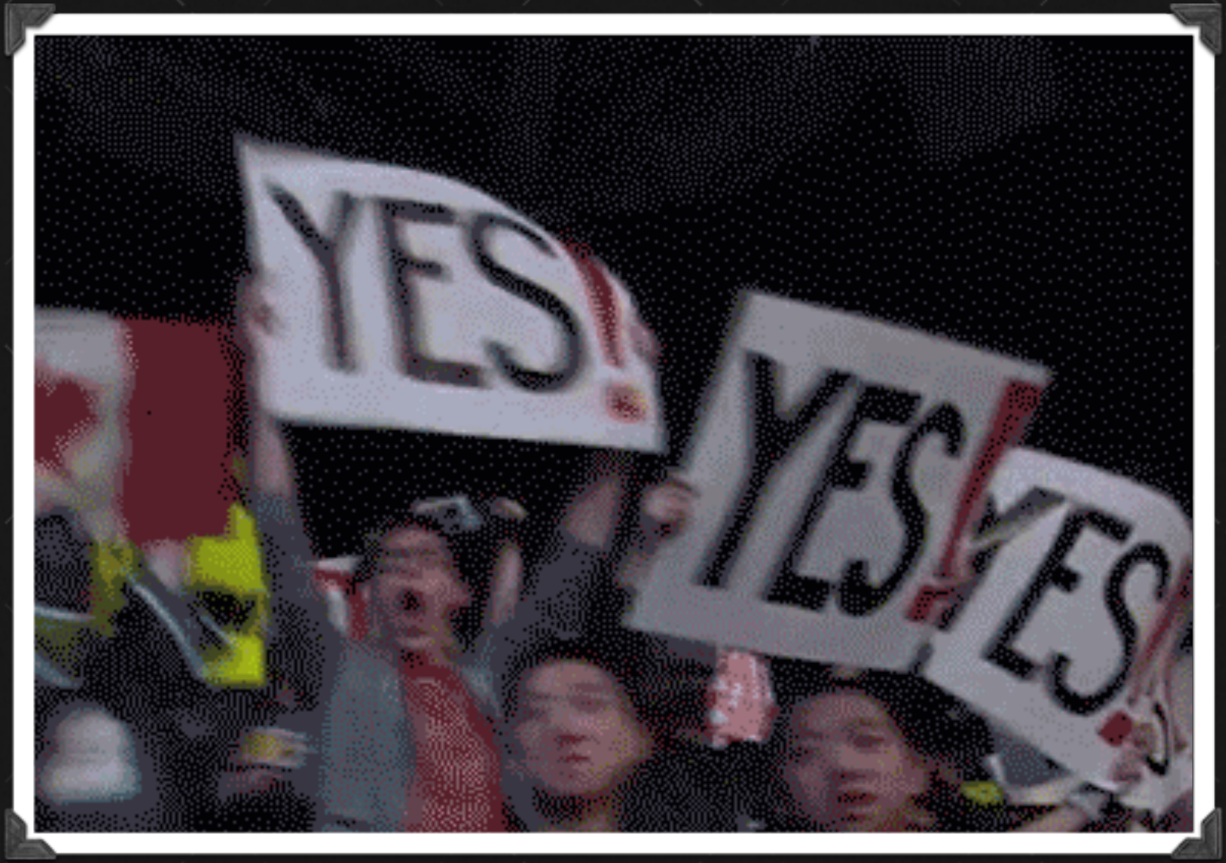
- Used in all domains
- Quick understanding of a system’s structure
- Application metadata


abnfc
edown
jsx
meck
quoted
sasl
vmstats
cowboy
egossip
lhttpc
edoc
cassanderl
pacingderl
ex_uri
bertconf
statsderl
proper
lager
ssl
parse_trans
jiffy
uuid
thrift
cassandra_thrift
compiler
simhash
syntax_tools
crypto

# Standard *patterns*

- Servers
- finite state machines (FSMs)
- event handlers
- supervisors

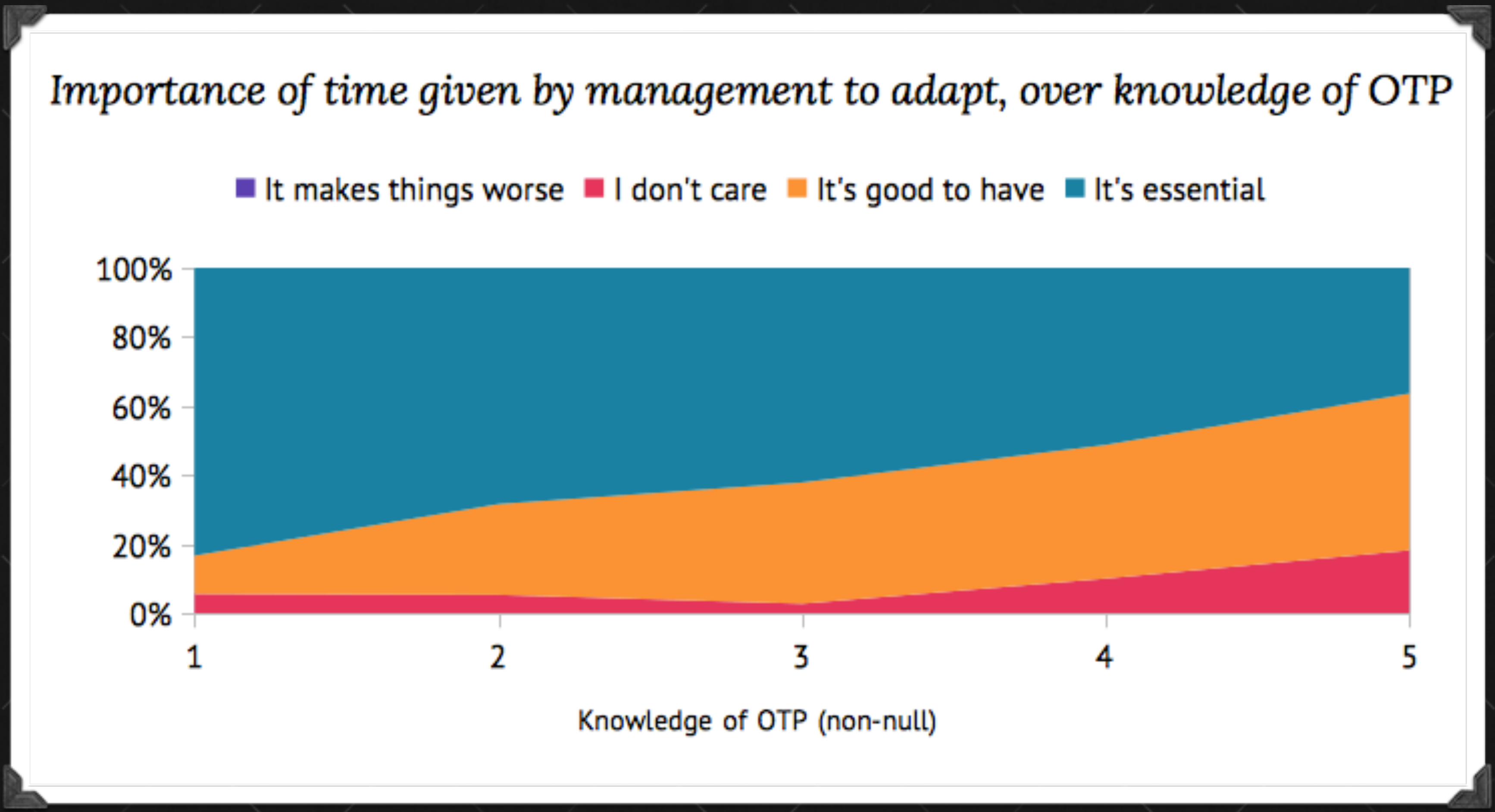
# Patterns in other langages



Everyone uses it!

# Reverse Engineering ++

- Tracing!
- sys:get_status(Worker)
- sys:trace(Worker, true)
- DBG



Importance of time given by management to adapt, over knowledge of OTP

# *Problems*

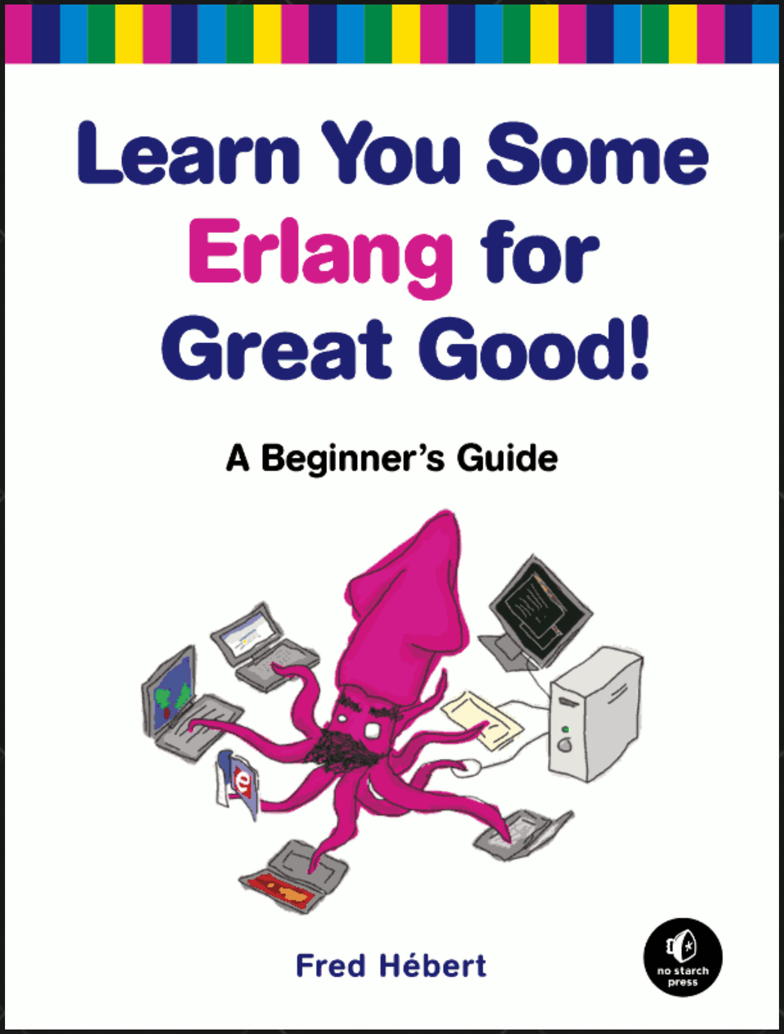
- Libraries and version clashes
- No namespaces
- Package management
- i18n, l10n

OTP systems are Solid

# Sources

- *Programs, Life Cycles, and Laws of Software Evolution*, MM. Lehman, 1980
- *Software Maintenance*, Gerardo Canfora and Aniello Cimitile, 2000
- *Software Maintenance* - clarityincode.com
- *Poll Results: Erlang & Maintenance* - ferd.ca

SHAMELESS



Learn You Some Erlang for Great Good!

A Beginner's Guide

Fred Hébert

no starch press

ADVERTISING

nostarch.com/erlang