Legacy / long running projects

RubyConfLT 2012

def

"A legacy system is an old method, technology, computer system, or application program that continues to be used, typically because **it still functions for the users' needs**, even though newer technology or more efficient methods of performing a task are now available." *(Wikipedia)*

def

Long running: > 2 years

def

Reality: long running =~ legacy

important!

technology

+

team

important?

technology

+

team

management

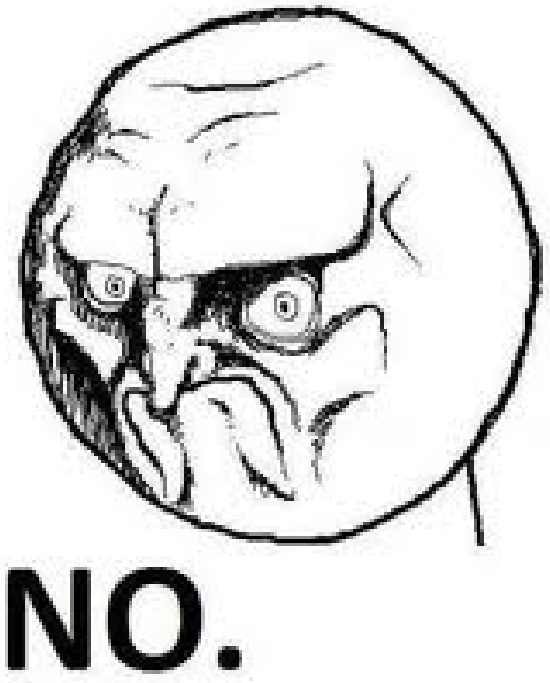
technology: upgrade

worth total rewrite?

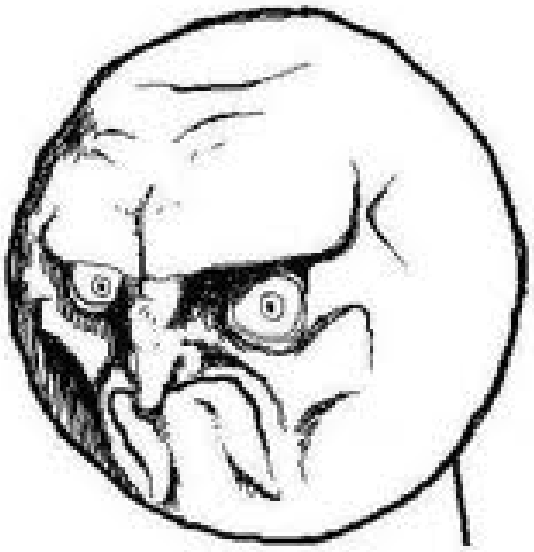
=~

to be or not to be

technology: rewrite



technology: rewrite



unless...

technology: rewrite

- 50% rule
- **clear** legacy system requirements
- new technology mature enough & team ready
- benefits/costs ratio
- customer understands what's going on (...)

technology: rewrite

Good (famous) examples:

- Mac OS -> OS X
- Minix -> Linux
- PHP 1 -> PHP 2

technology: rewrite

Good (famous) examples:

- Mac OS -> OS X
- Minix -> Linux
- ~~PHP 1 -> PHP 2~~
- UBB -> vBulletin
- Netscape -> Mozilla
- ...

technology: rewrite

- integration tests
- prioritize!
- be ready to support legacy system
- simplify
- migrate data early
- rewrite piece-by-piece
- usually takes longer than expected

technology: technical debt

"Like a financial debt, the technical debt **incurs interest payments**, which come in the form of the extra effort that we have to do in future development because of the quick and dirty design choice."[1]

[1] http://martinfowler.com/bliki/TechnicalDebt.html

technology: technical debt

Main diff from short period projects:

technical debt matters

technology: technical debt

Main diff from short period projects:

technical debt matters

A LOT

technology: technical debt

Impact on team & development speed

hard part: team

As always with HR: difficult to measure

team: motivation

- lack of "hot" tech stuff
- high level of sh*t (everything needs to be fixed!..)
- lack of challenge in well known field
- routine, routine, routine

team: motivation

side projects

IS-A-MUST

team: motivation

side projects

IS-A-MUST

even if forced

team: motivation

hackathons
startup weekends
conferences

team: motivation

scoped(freedom)
to choose tech

team: technical debt

pay your debt in time.

team: motivation

collective code ownership

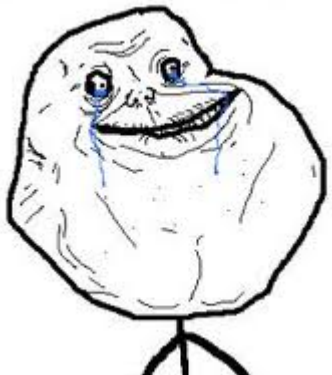
at least weak code ownership

team: motivation

adequate attention to non-tech problem solving (universal)

team: motivation

if a product has some fans (hopefully), forward/communicate love letters to the team



return

1. technology matters as much as it matters to the TEAM
2. changes are and must be inevitable
3. way of recharging the batteries must be developed to get rid of long run exhaustion
4. …
5. profit!

end

+Vidmantas Kabošis @vidmantas

vidmantas@kabosis.lt
vidmantas.kabosis.lt

slides: bit.ly/legacy-long-running-projects

Questions?