Robert Virding

Principle Language Expert
at Erlang Solutions Ltd.

Erlang Solutions Ltd.

On Language Design
- A Philosophy

© 1999-2015 Erlang Solutions Ltd.

- Based on experiences of developing languages at 3 different levels
  - Erlang, base level
  - LFE, adapting a language
  - Luerl, Erlog, implementing an existing language

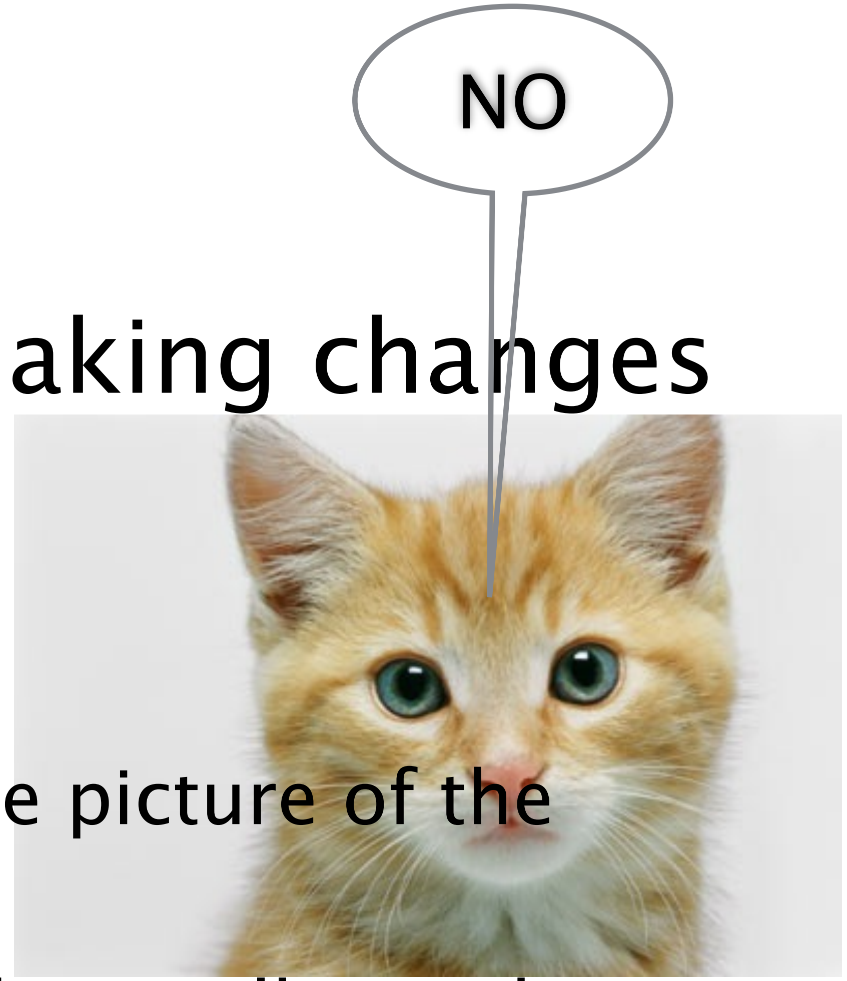
+

- Other languages I have used

Provide tools, not solutions

- Tools are what you need to build things
- Solutions tend to be specific or general
  - General solutions become complex to use
  - Adding specific solutions just means you will end up adding lots of them
- Put effort into make a good set of tools which allow users to build **THEIR** solutions

Don’t be nice to users!



“Be very careful when making changes suggested by users”

- They often don’t see the whole picture of the changes they suggest
- They often don’t know what they really need
- They often want help with a solution not solving a problem

THE QUESTION!

- What problem are you trying solve?

  Why are you implementing this language?

- Decide this and focus on that!
- Don’t lose focus and get side-tracked

FOCUS, FOCUS, FOCUS

© 1999-2015 Erlang Solutions Ltd.

Semantics is king

© 1999-2015 Erlang Solutions Ltd.

Syntax is irrelevant, but no it isn’t

- Going bonkers that a syntax doesn’t look like JXXX is ridiculous, RTFM
- But when designing a language, make a good syntax

- The syntax of a language should reflect the semantics
- “Borrowing” the syntax of a language with a different semantics can lead to problems

- Avoid providing alternate syntaxes for the same thing
  - One is enough and many just confuse the issue
  - Makes it harder for newbies as they have to learn them all

- Corollary: avoid adding syntax for “special cases”

© 1999-2015 Erlang Solutions Ltd.

Adding features

- Be very restrictive about adding features to a language

"Because there is no feature that is not a limitation on something else. TANSTAAFL."

"There comes a point where throwing new features at a language is a bad idea and it's time to stop and redesign from the beginning, possibly keeping the VM backwards compatible."

Richard O'Keefe

Advice

- It is much harder to change something once it has been released
  - Hindsight before you release

- Document **WHY** things are what they are, and not just what they are

Be consistent

- If it looks the same it should mean the same
- If it means the same it should look the same

© 1999-2015 Erlang Solutions Ltd.

Keep it simple

- Complexity never wins
- Keep the language simple
  - You might not get all the cool features but it will be easier to understand

Be explicit

- This makes it easy to see what is happening

Maintenance

- Those maintaining your code will love you if you do all this
- Or hate you if you don’t
- (75–80% of cost in maintenance phase)

FOCUS, FOCUS, FOCUS

(Have I said this before?)

DSLs

DSLs are great

– but –

- All of the above rules apply
- Some functions and a lax syntax hiding function calls is not a DSL, IMAO

FOCUS, FOCUS, FOCUS

Simplicity

Consistency

Explicitness

Thank you

Robert Virding: rvirding@gmail.com @rvirding