

Simple is not easy

Rich Hickey's Strange Loop 2011 talk on why untangling software matters more than making it familiar.



● KEY IDEAS

Five ideas from the talk worth keeping

01

Simple

From sim-plex: one fold or braid. The opposite of complex, which means braided together.

02

Easy

From the root of adjacent: near at hand, familiar. Easy is relative to you.

03

Complect

To interleave or braid things together. Hickey's word for what makes code hard.

04

Reasoning

Every bug in the field passed the type checker and all the tests.

05

Speed

Ignore complexity and you will slow down, sprint after sprint.



- THE CORE QUESTION

Are you choosing a
tool because it is
simple,
or only because it is
familiar to you?



Tests are **guard rails**, not a steering wheel

Hickey calls it **guard rail programming**: changing code because the tests will catch you. Guard rails do not point the car anywhere.

To change or debug software you must **reason about it**, informally, and every braid you add makes that harder.



Swap complexing constructs for **simpler ones**

State and objects become **values**.
Methods become functions and namespaces. Inheritance and switch become polymorphism a la carte.

Syntax becomes **data**, loops become set functions, actors become queues, and ORM becomes declarative data manipulation.



END OF TALK NOTES

Thanks. Untangle one module.

SOURCE: SIMPLE MADE EASY, RICH HICKEY,
STRANGE LOOP 2011

Made with MagicSlides