

Simplicity in Practice



“Words, words, words.”

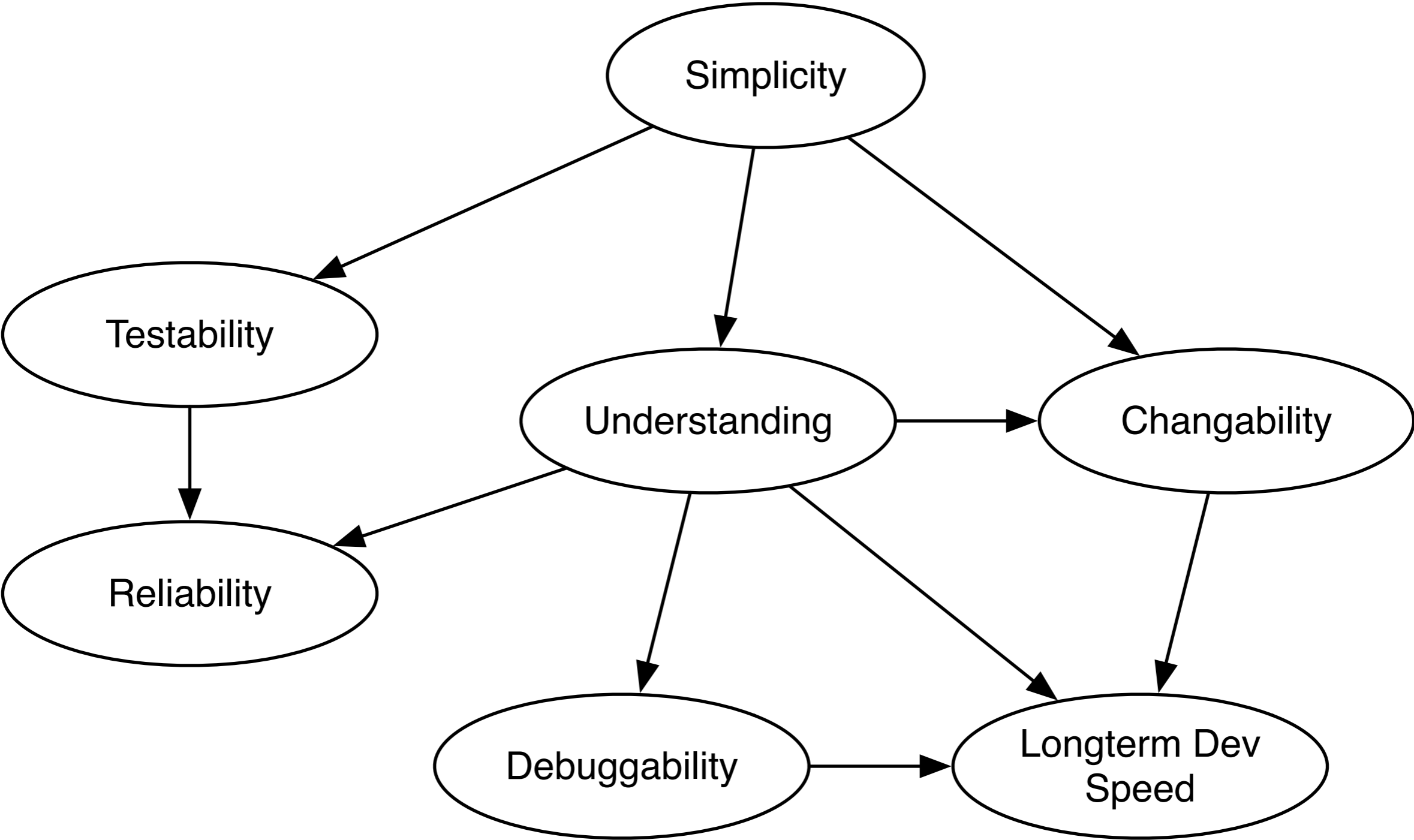
– Hamlet, Act 2 Scene 2

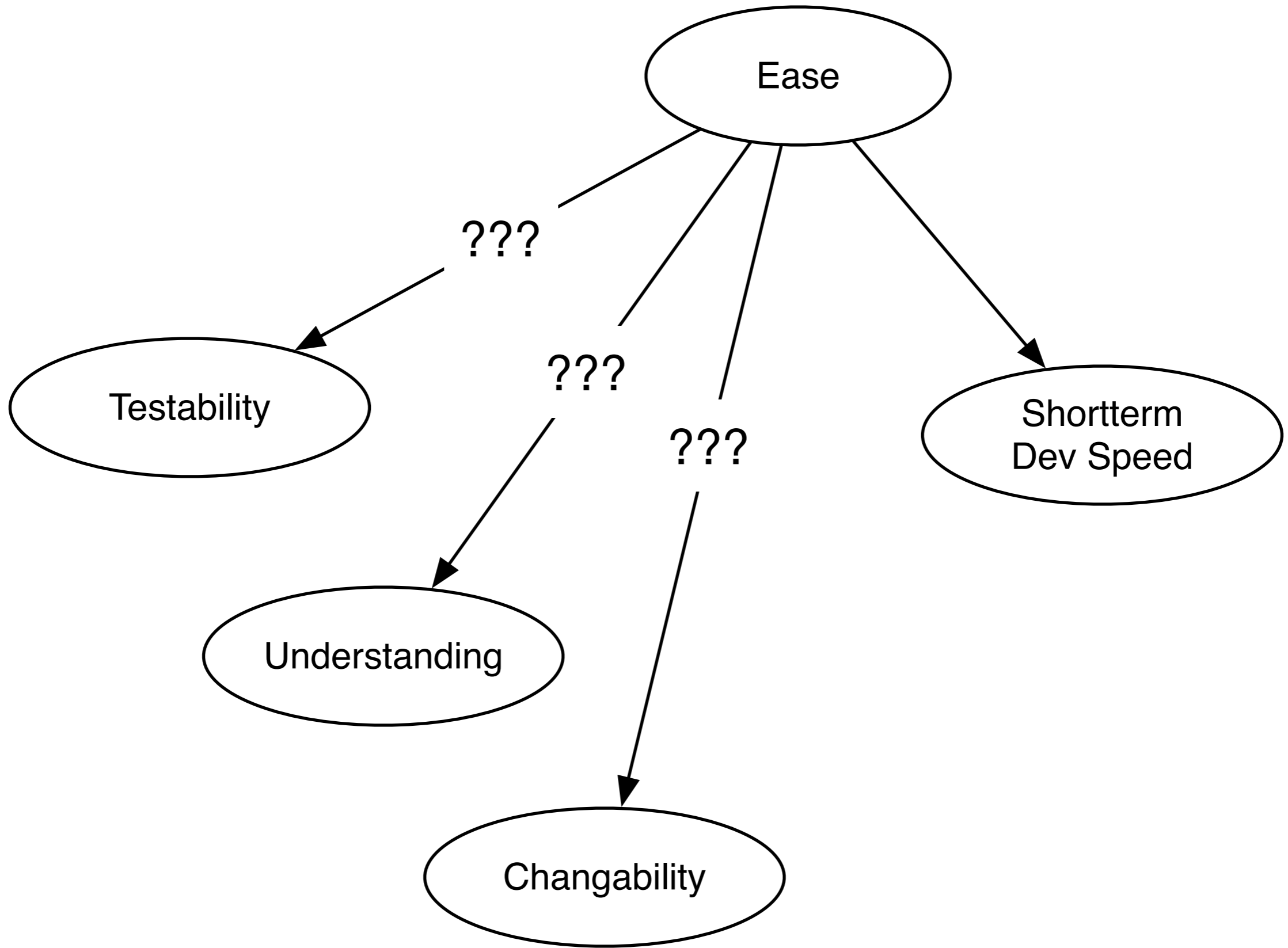
simple, adj.

- one fold/braid
- one role, task, concept, dimension
- antonym: complex, interleaved, of many parts
- objective measure

easy, adj.

- near at hand
- familiar
- antonym: hard, requiring effort
- relative measure



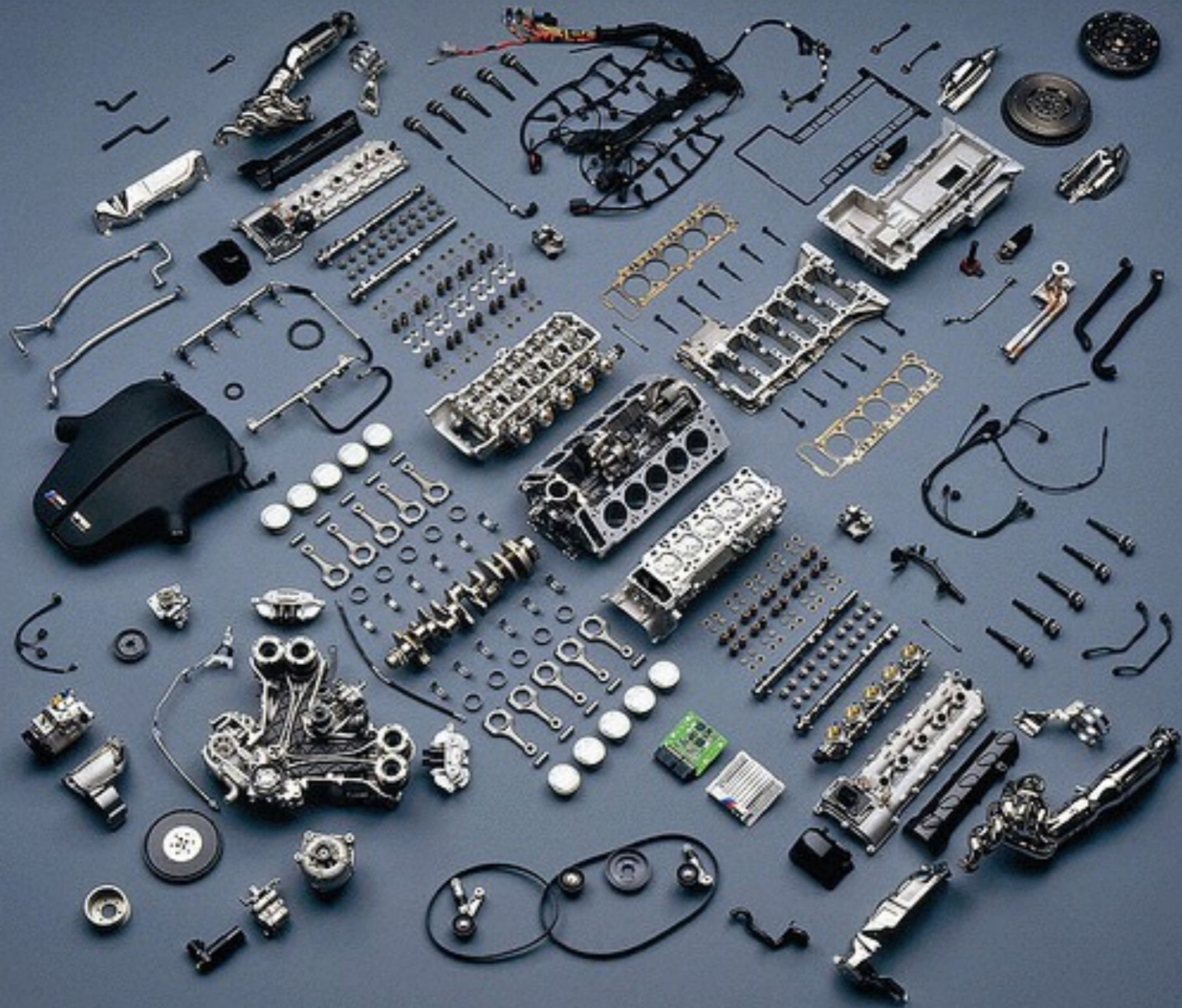


(< easy simple)

simplicity is necessary

simplicity isn't
sufficient





a small example

“Never define global, stateful vars”

– ~~Wise~~ Opinionated Clojure Programmers



```
(def uri "datomic:dev://localhost:4334/my-db")  
(def conn (d/connect uri))
```

- complects:
 - namespaces with runtime application state
 - compilation with app initialization

“What do I do instead?”

“Just create a single object representing your application’s state, which you construct explicitly at initialization time and pass around to the rest of your program”

by @jrecursive




```
(defrecord DB [uri]
  component/Lifecycle
  (start [:config] (d/connect uri)))

(defn system [config]
  (component/system-map
    :config config
    :db (->DB (-> config :db :uri))))
```

<https://github.com/stuartsierra/component>



“We're writers.”

–Bruce Eckel, 2009

code

- functions
- data
- execution flow

stories

- setting
- characters
- plot

interest

clarity

comprehension

collaboration

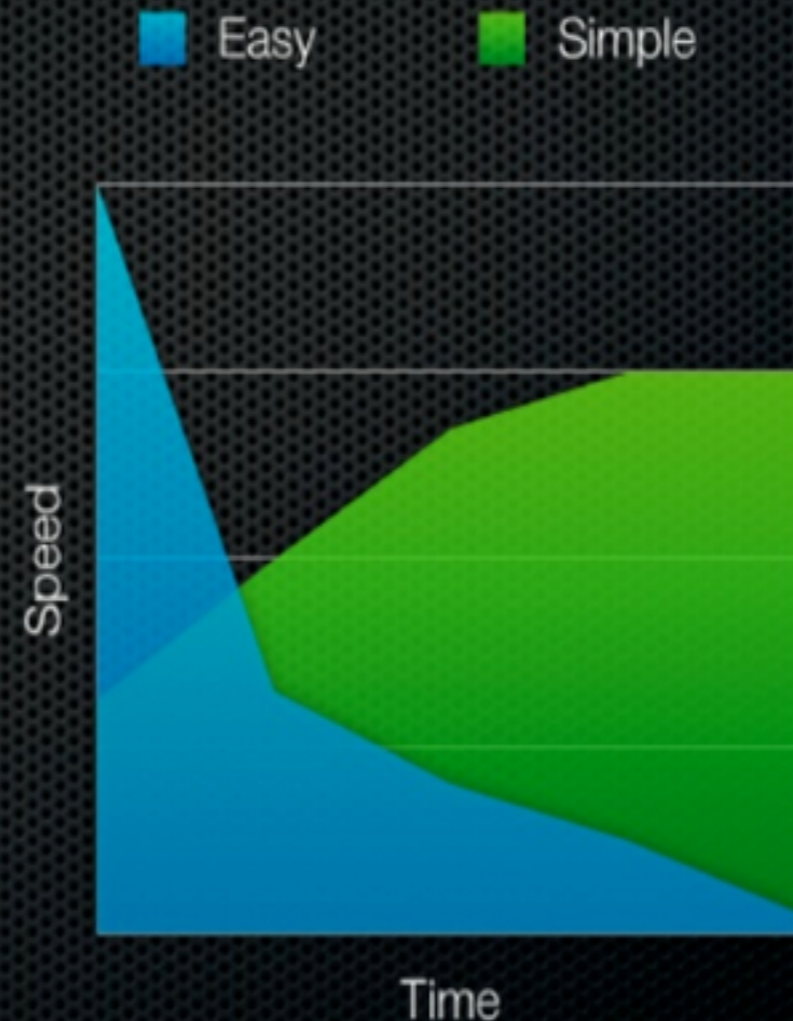


revisiting terms

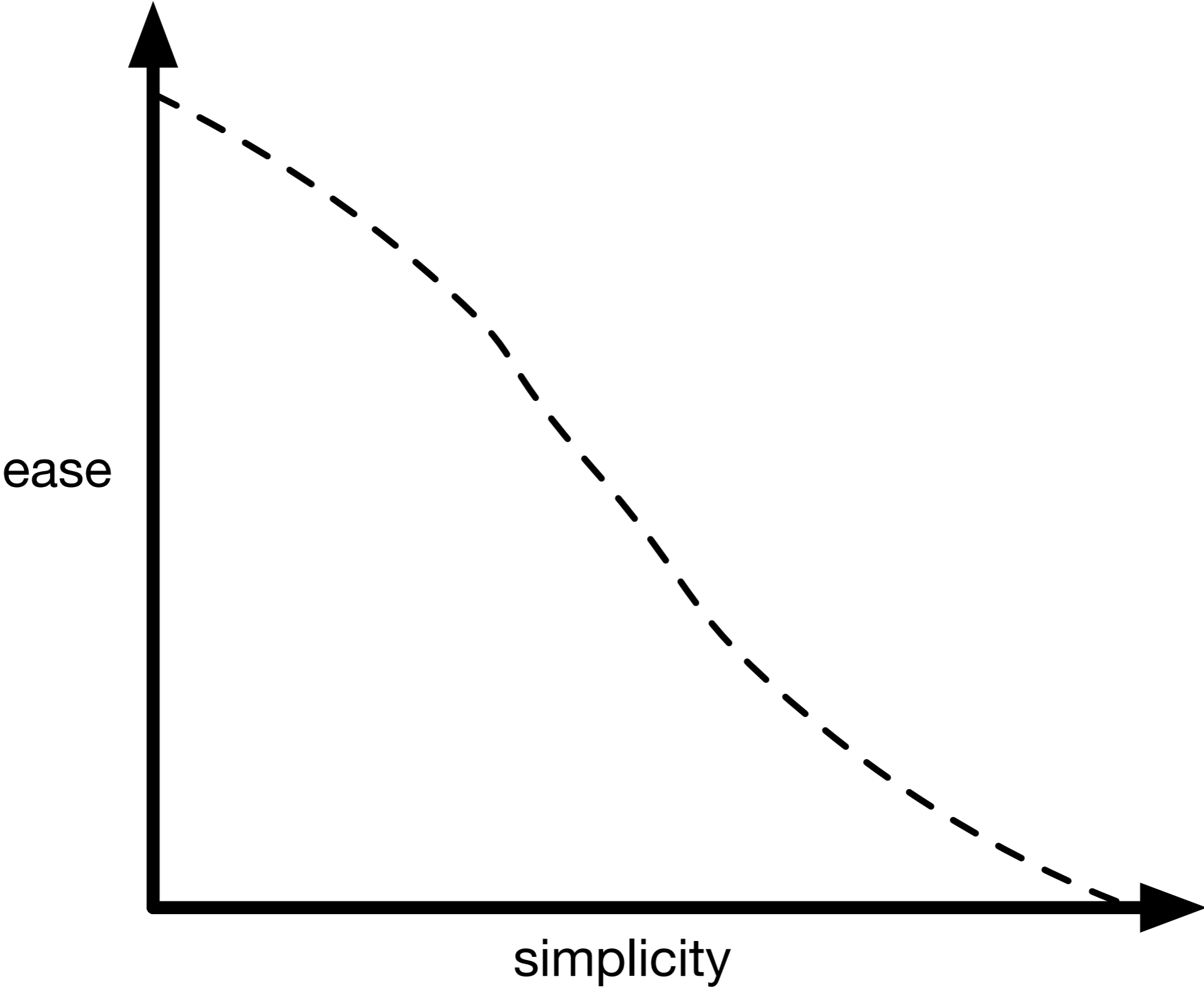
$f(\text{easy, simple})$

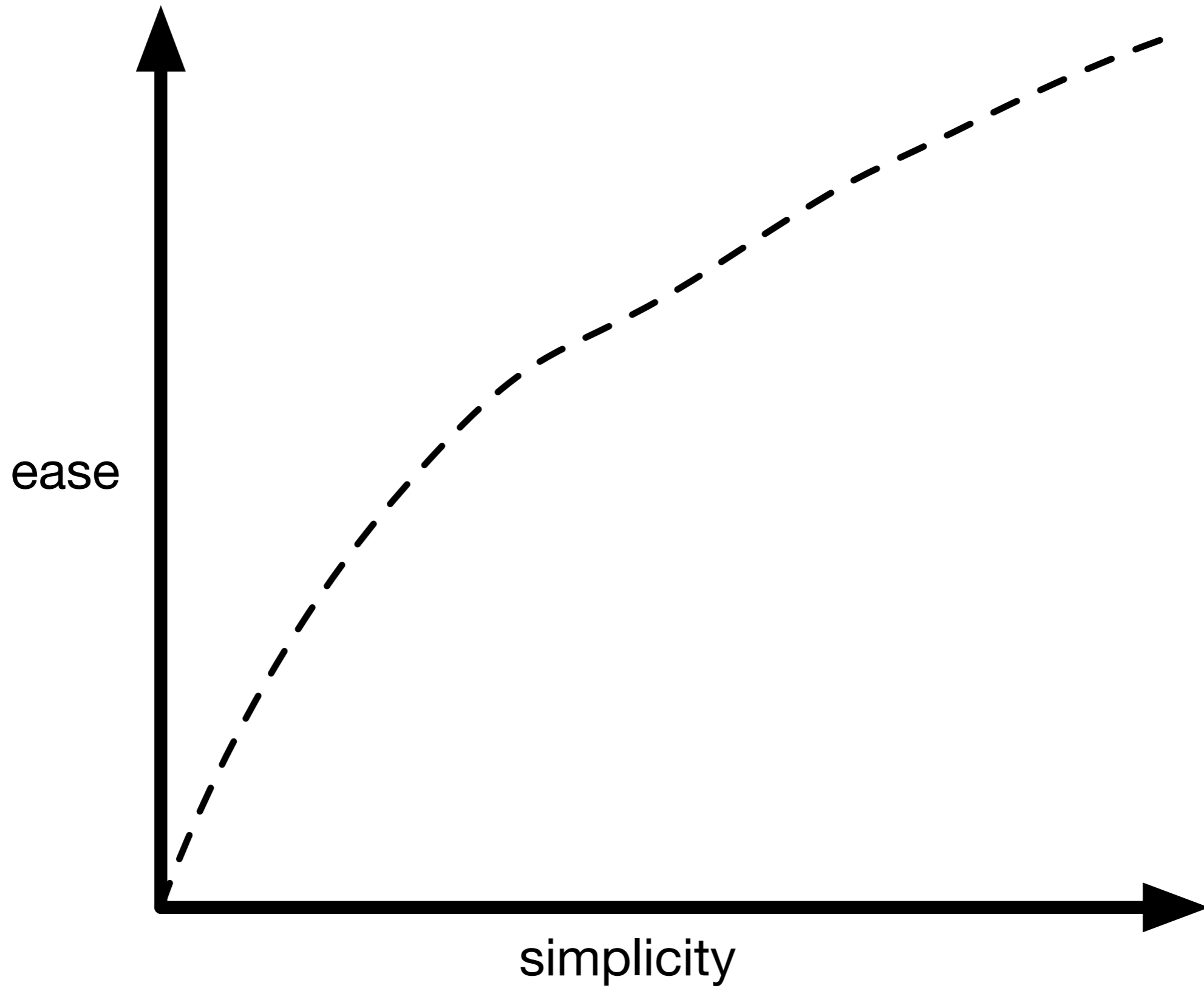
Development Speed

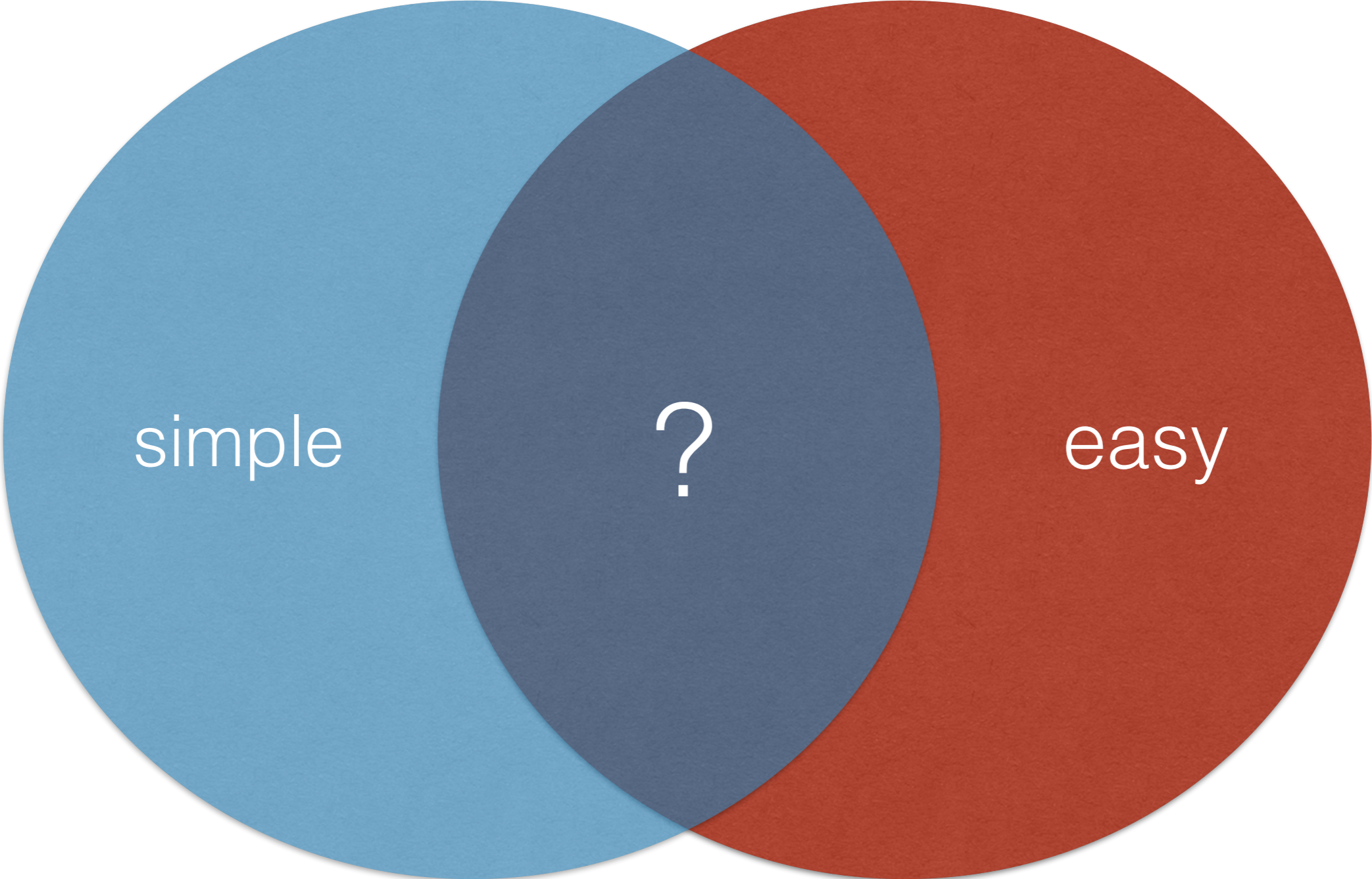
- Emphasizing ease gives early speed
- Ignoring complexity will slow you down over the long haul
- On throwaway or trivial projects, nothing much matters

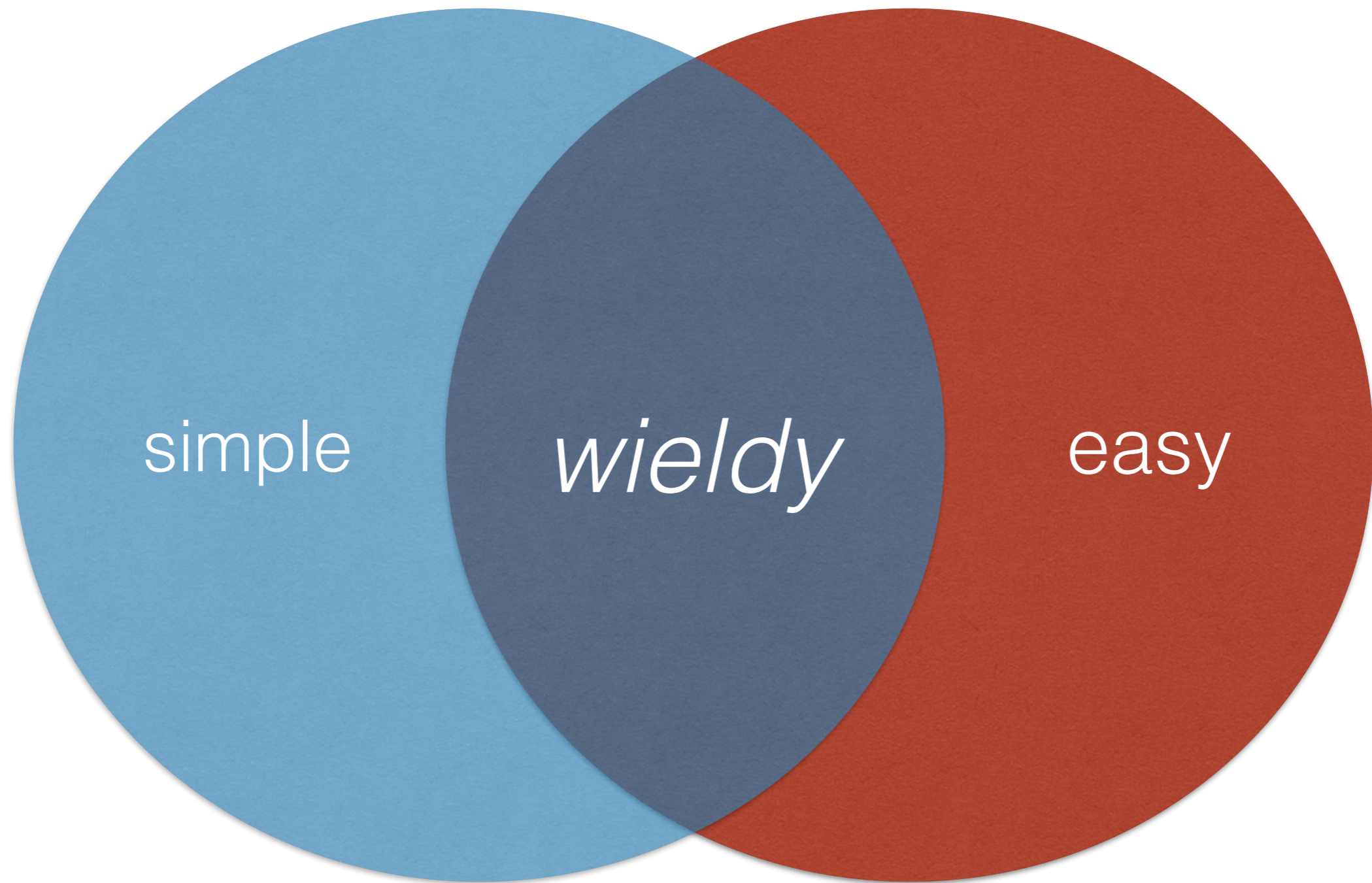


– Rich Hickey, Simple Made Easy









wieldy, adj.

formal definition

- capable of easily 'wielding' one's body or limbs, or a weapon, etc.; vigorous, active, agile, nimble.
- easily wielded, controlled, or handled; manageable; handy.
- requiring strength to wield, heavy or bulky; wielded with force, powerful.

Oxford English Dictionary

wieldy, adj.

programmer definition

- manageable, handy, powerful
 - easy \cap simple
- pertaining to developer experience
 - powerful and fun
- relational measure

benefits

- flexibility
- codebase size
- development speed
- developer morale
- developer improvement





“I should be sorry if I
only entertained
them, I wish to make
them better.”

–George Frideric Handel

Wielyd
Code



simple, unwieldy

```
[ { :db/id #db/id [ :db.part/db ]  
  :db/ident :artist/name  
  :db/valueType :db.type/string  
  :db/cardinality :db.cardinality/one  
  :db/fulltext true  
  :db/index true  
  :db/doc "The artist's name"  
  :db.install/_attribute :db.part/db }
```

```
{ :db/id #db/id [ :db.part/db ]  
  :db/ident :artist/country  
  :db/valueType :db.type/ref  
  :db/cardinality :db.cardinality/one  
  :db/doc "The artist's country of origin"  
  :db.install/_attribute :db.part/db }
```

easy, unwieldy

```
(jdbc/with-connection db
  (jdbc/insert-records :language_authors
    {:first_name "Rich" :last_name "Hickey"}
    {:first_name "Larry" :last_name "Wall"}))
```

recontextualization

- To reinterpret or refresh (an idea, work of art, etc.) by placing it in a new or different context.

three kinds of
documentation

reference	→	simple
tutorial	→	easy
conceptual	→	wieldy

reference style

- terse
- complete
- no context

sorted-set-by

function

Usage: `(sorted-set-by comparator & keys)`

Returns a new sorted set with supplied keys, using the supplied comparator. Any equal keys are handled as if by repeated uses of `conj`.

Added in Clojure version 1.1

[Source](#)

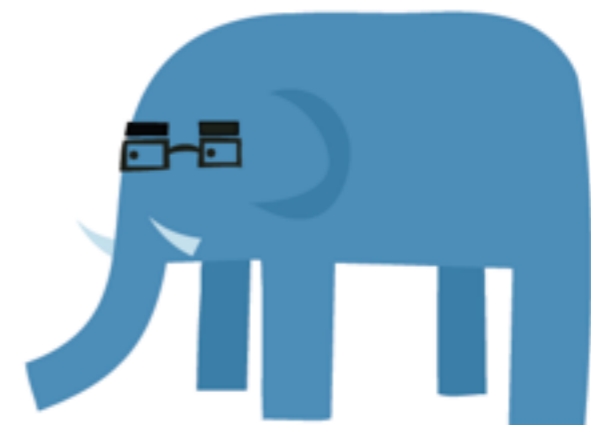
tutorial style

- task-oriented
- longform
- concepts are doled out on demand



Learn You a Haskell for Great Good!

A Beginner's Guide



Miran Lipovača



conceptual style

1. problem statement
2. concepts
3. common usage examples
4. pointers to reference documentation
5. roughly 1 page

i.e. telling the story

so I know a new word

what now?