

# THE ARCHITECT?

@ARIALDOMARTINI

@NICOLABALDI

even by our industry's  
standards, "architect" and  
"architecture" are terribly  
overloaded words.

MARTIN FOWLER

THOUGHTWORKS

CHIEF TECHNOLOGY SCIENTIST

**“THE HIGHEST LEVEL  
CONCEPT OF A SYSTEM  
IN ITS ENVIRONMENT”**



**IEEE**

**“THE HIGHEST LEVEL  
CONCEPT OF A SYSTEM  
IN ITS ENVIRONMENT”**



**IEEE**

**DEVELOPERS WHO  
TRANSCENDED THEIR  
OWN NATURE**



A  
**SINGLE MIND**

IS NEEDED TO  
ENSURE A SYSTEM'S

**CONCEPTUAL  
INTEGRITY**

# A **SINGLE MIND**

## IS NEEDED TO ENSURE A SYSTEM'S

# **CONCEPTUAL INTEGRITY**



jasongorman  
@jasongorman



+  
Follow

We measure the quality of a cake by tasting it. Conformance to the recipe is a poor metric in that respect.

Reply  Classic RT  Retweeted  Favorited ... More

Building a large, complex, enterprise-wide information system without an enterprise architect is like trying to build a city **without a city planner.**

Can you build a city without a city planner?  
**Probably.**

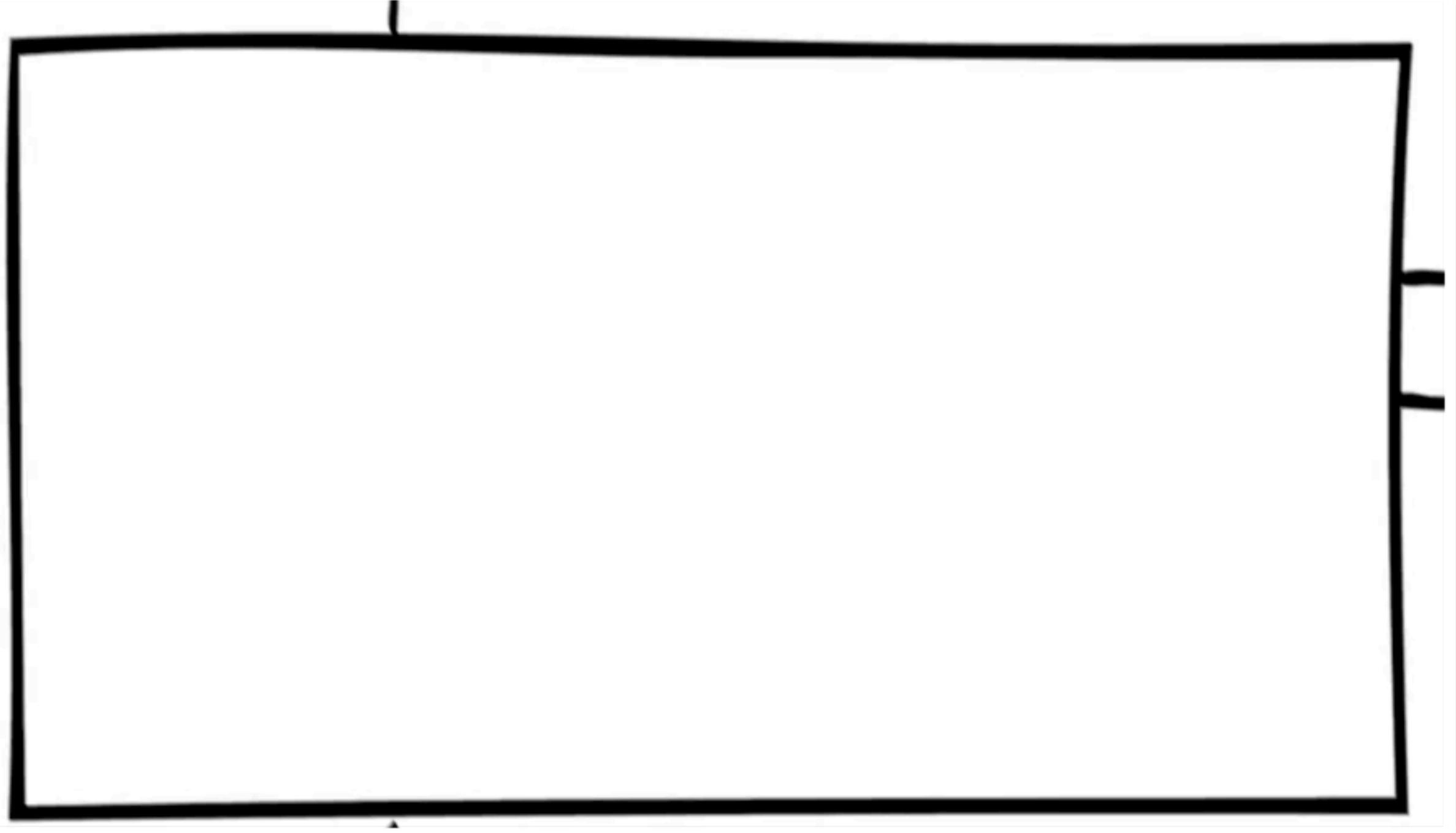
Would you want to live in such a city?  
**Probably not.**

Building a large, complex, enterprise-wide information system without an enterprise architect is like trying to build a city **without a city planner.**

Can you build a city without a city planner?  
Probably.

Would you want to live in such a city?  
Probably not.

**FACEBOOK**



@ARIALDOMARTINI

@NICOLABALDI

Around the world introducing an Enterprise Architecture has been an initiative for most financial institutions for the last five years or so, and it is not over. I have been working with such companies and helped some of them to avoid making the worst mistakes.

Most EA initiatives failed.

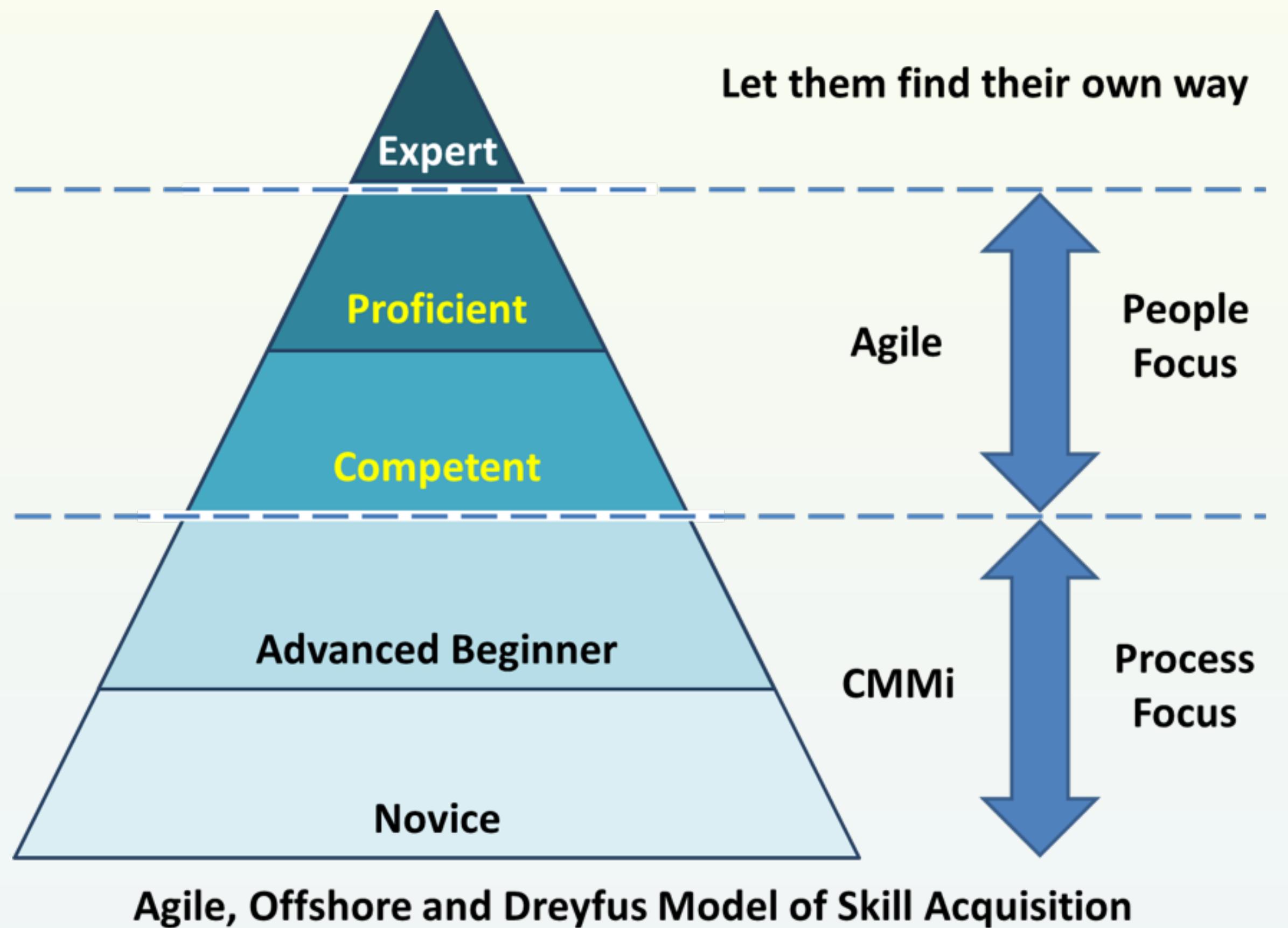
My guess is that more than 90% never really resulted in anything useful.

**IVAR JACOBSON**

There is no highest level concept of a system. Customers have a different concept than developers. Customers do not care at all about the structure of significant components. So, perhaps an architecture is the highest level concept that developers have of a system in its environment. Let's forget the developers who just understand their little piece. Architecture is the highest level concept of the expert developers. What makes a component significant? It is significant because the expert developers say so.

There is no highest level concept of a system. Customers have a different concept than developers. Customers do not care at all about the structure of significant components. So, perhaps an architecture is the highest level concept that developers have of a system in its environment. Let's forget the developers who just understand their little piece. Architecture is the highest level concept of the expert developers. What makes a component significant? It is significant because the expert developers say so.

# DREYFUS MODEL



# DREYFUS MODEL

```
balance = database.read("account-number")  
  
new_balance = balance - amount  
  
database.write("account-number", new_balance)  
  
dispense_cash(amount)
```

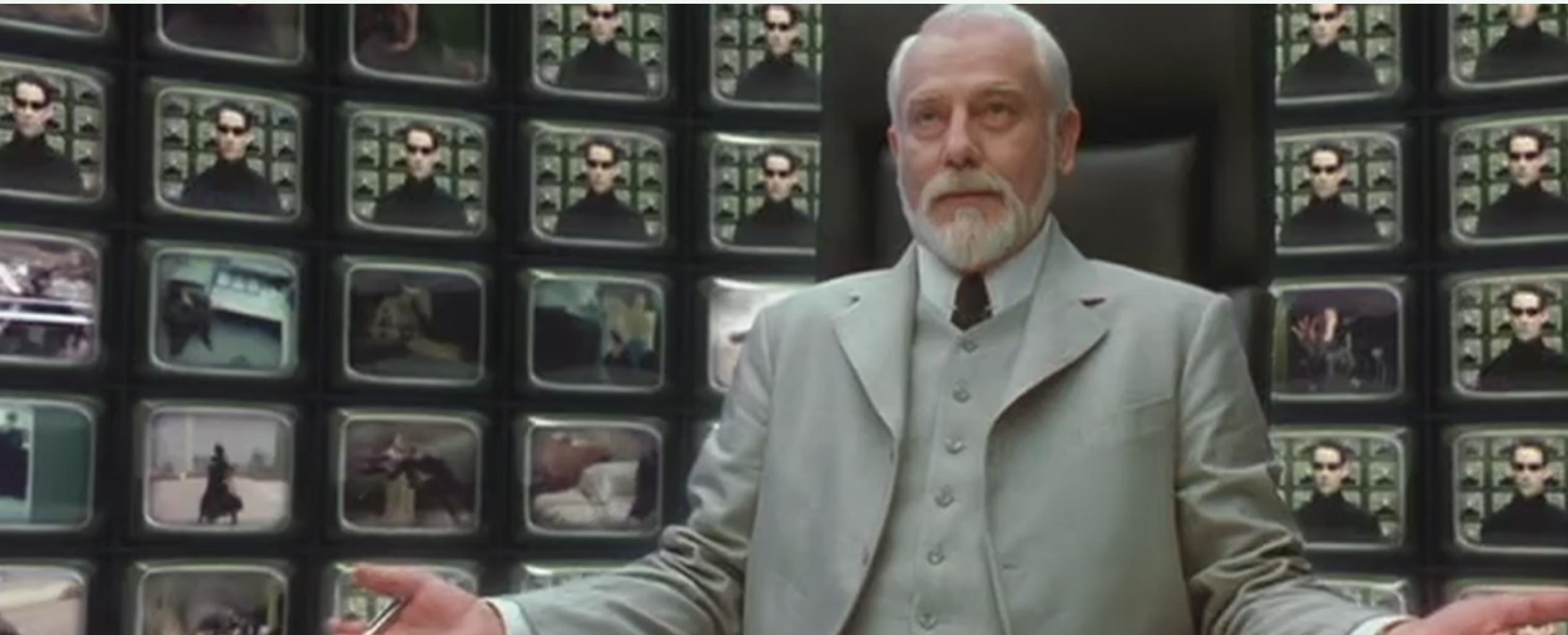
# ARCHITECTUS **RELOADUS**

IS THE PERSON WHO  
MAKES ALL THE  
IMPORTANT  
DECISIONS



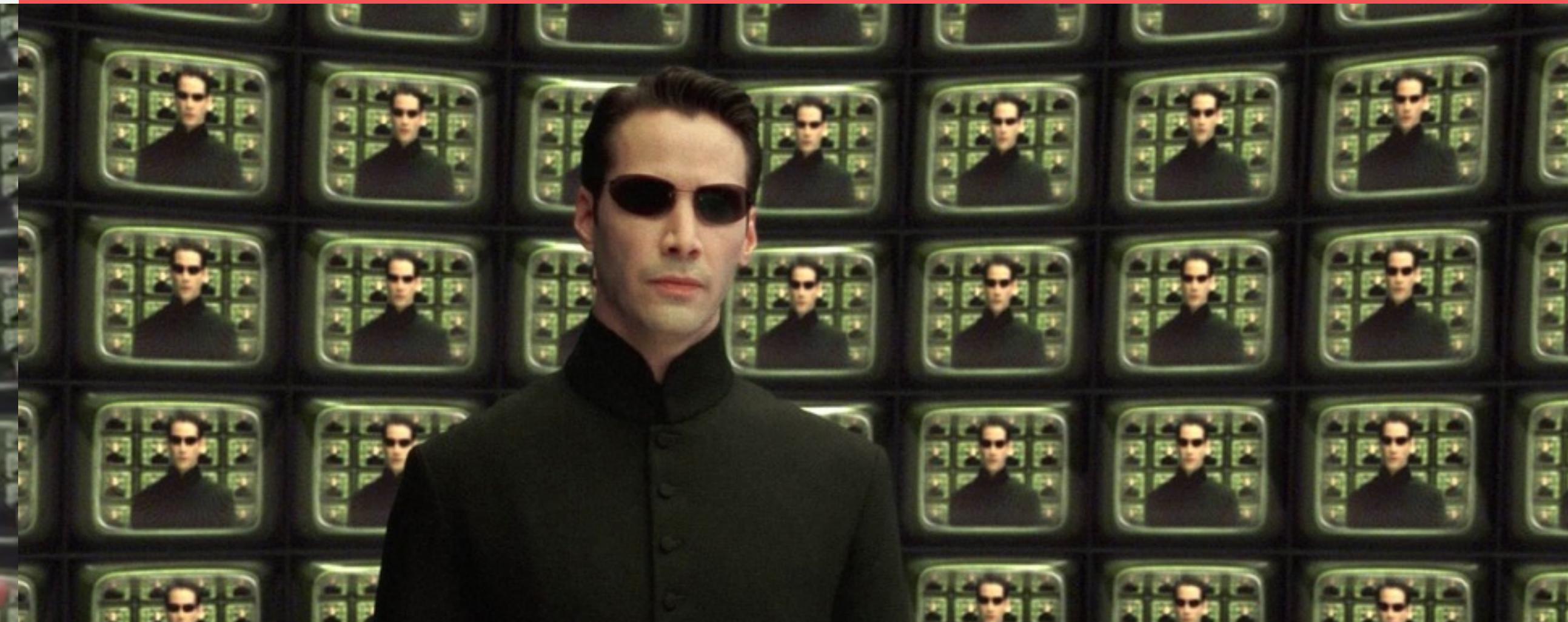
# ARCHITECTUS **RELOADUS**

IS THE PERSON WHO  
MAKES ALL THE  
IMPORTANT  
DECISIONS



# ARCHITECTUS **OZYRUS**

IS THE PERSON WHO  
IS AWARE OF  
WHAT'S GOING ON



The most important activity of Architectus Oryzus is to mentor the development team, to raise their level so that they can take on more complex issues.

# A GUIDE



**AN ARCHITECT'S  
VALUE IS**

**INVERSELY  
PROPORTIONAL**

**TO THE NUMBER  
OF DECISIONS HE  
OR SHE MAKES.**

**WHAT'S BETTER  
THAN AN  
ARCHITECT?**

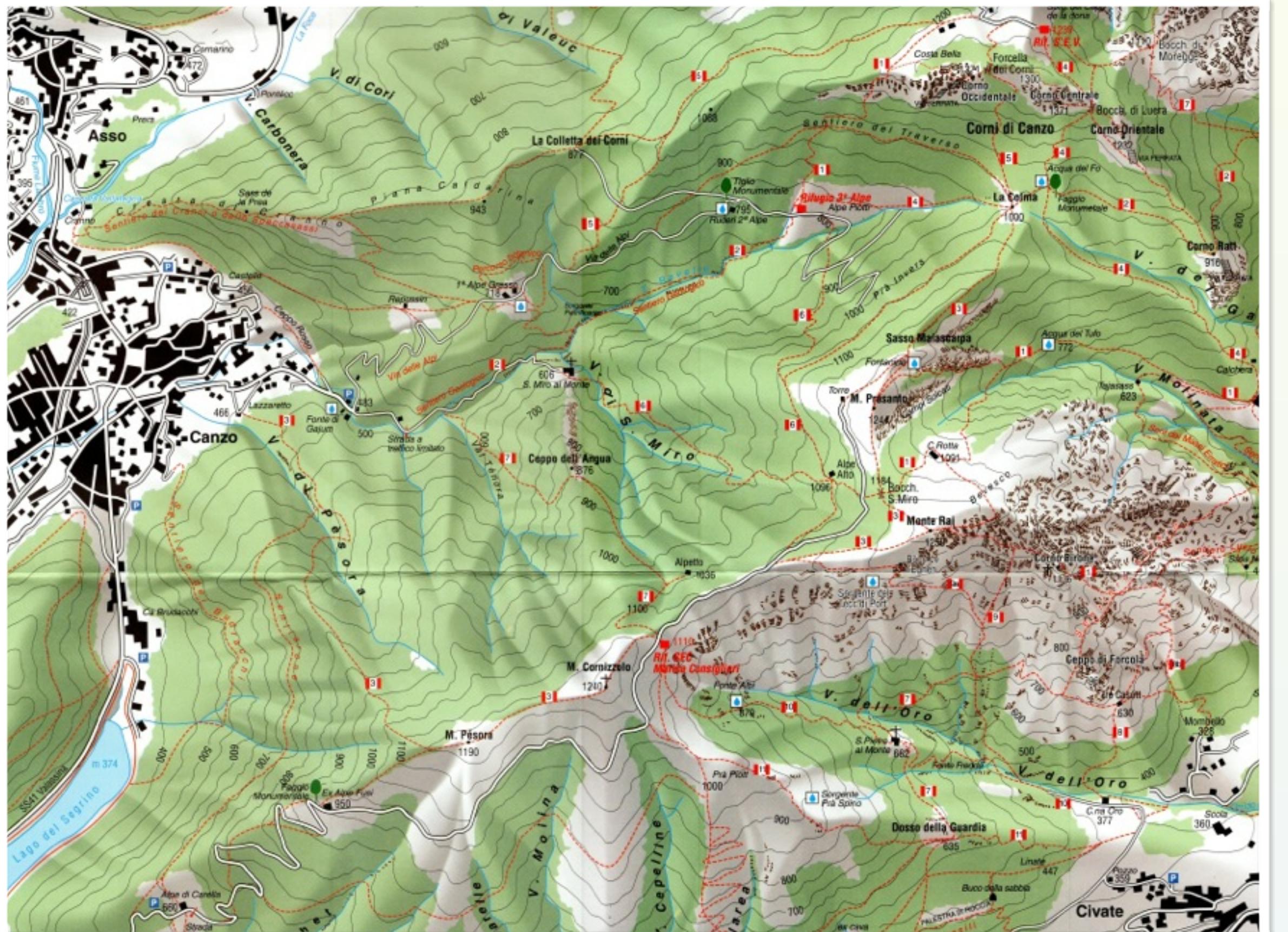
**NO ARCHITECTS**

**AN ARCHITECT'S  
VALUE IS**

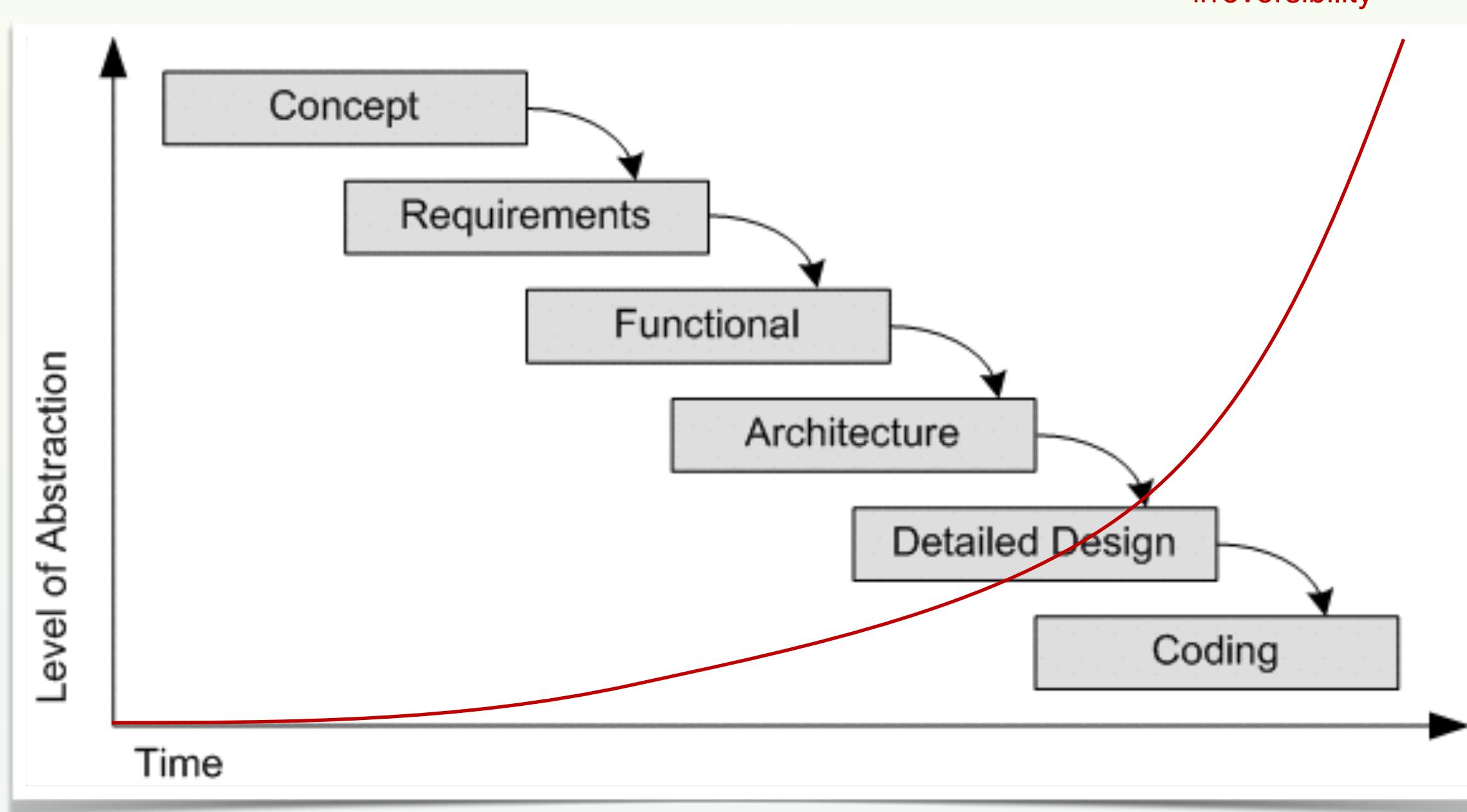
**INVERSELY  
PROPORTIONAL**

**TO THE NUMBER  
OF DECISIONS HE  
OR SHE MAKES.**

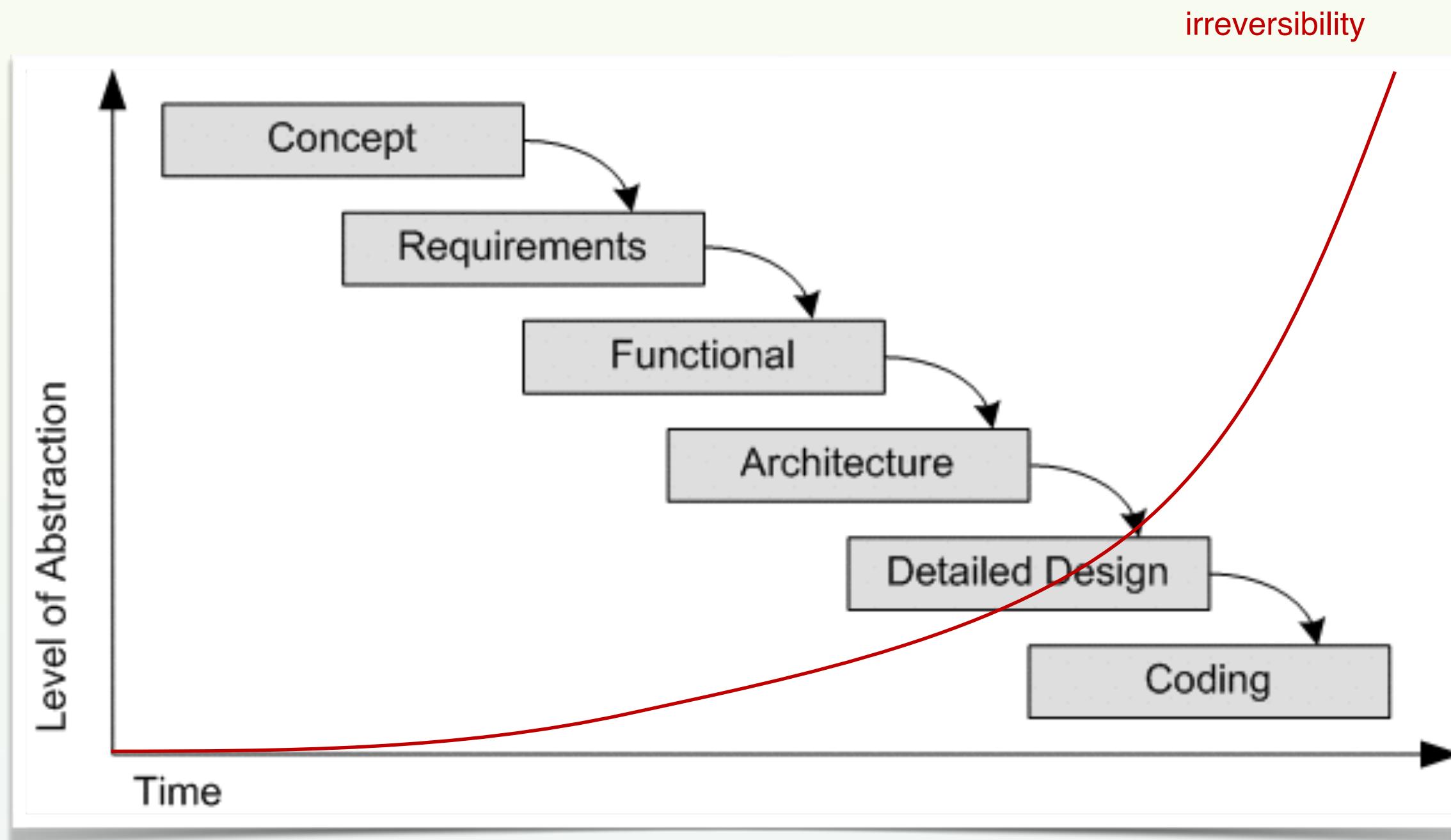
**AN ARCHITECT'S  
VALUE IS  
INVERSELY  
PROPORTIONAL  
TO THE NUMBER  
OF DECISIONS HE  
OR SHE MAKES.**



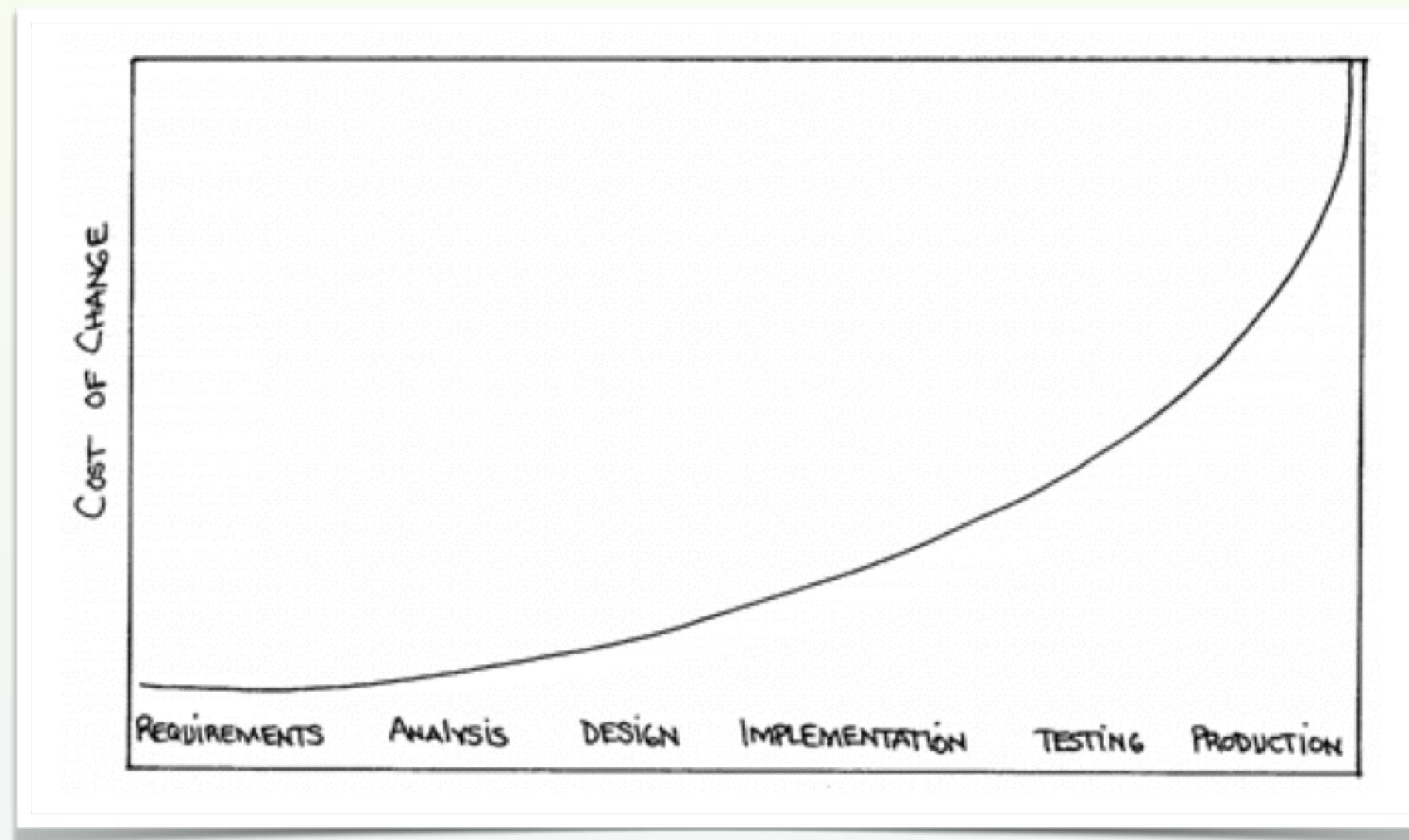
# ABSTRACTION IRREVERSIBILITY



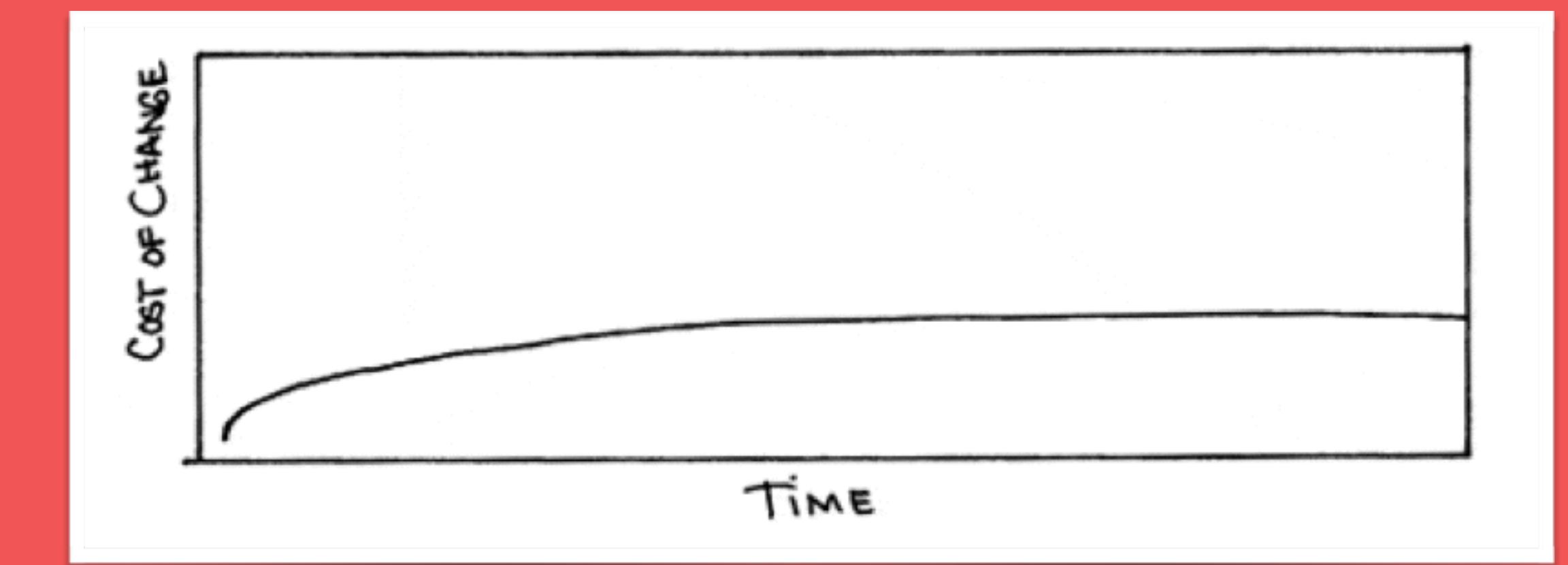
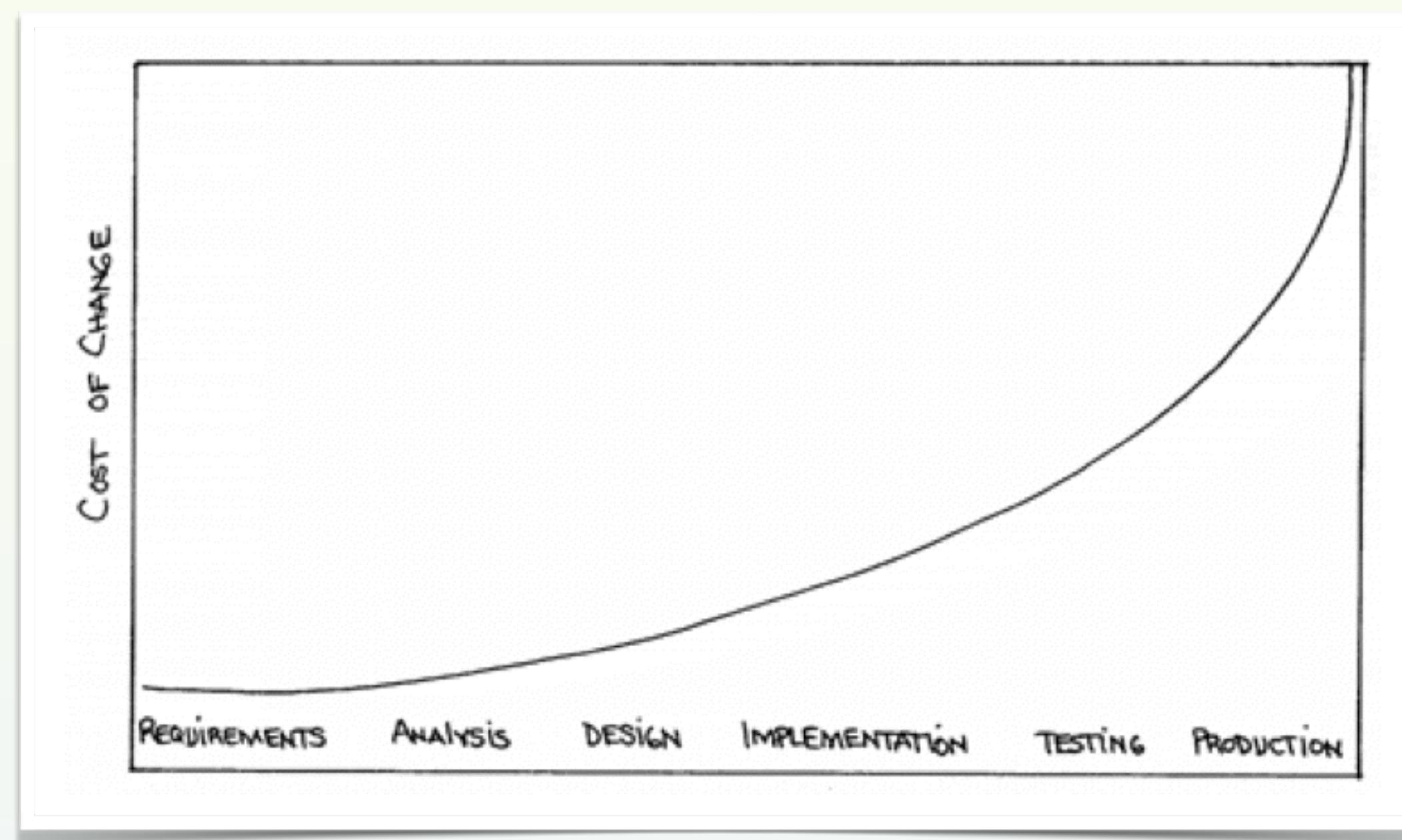
# ABSTRACTION IRREVERSIBILITY



# COST OF CHANGE



# COST OF CHANGE



# PROBLEM

BUILD + DEBUG + MAINTAIN

LARGE

COMPLICATED

# SOLUTION



Experts, usually an **AUTHORITY**, will architect the system. They won't be bothered with technical implementation details because **THEIR TIME IS VALUABLE**.

The team will be required to **FOLLOW** the architecture so that the good qualities of the system are assured.

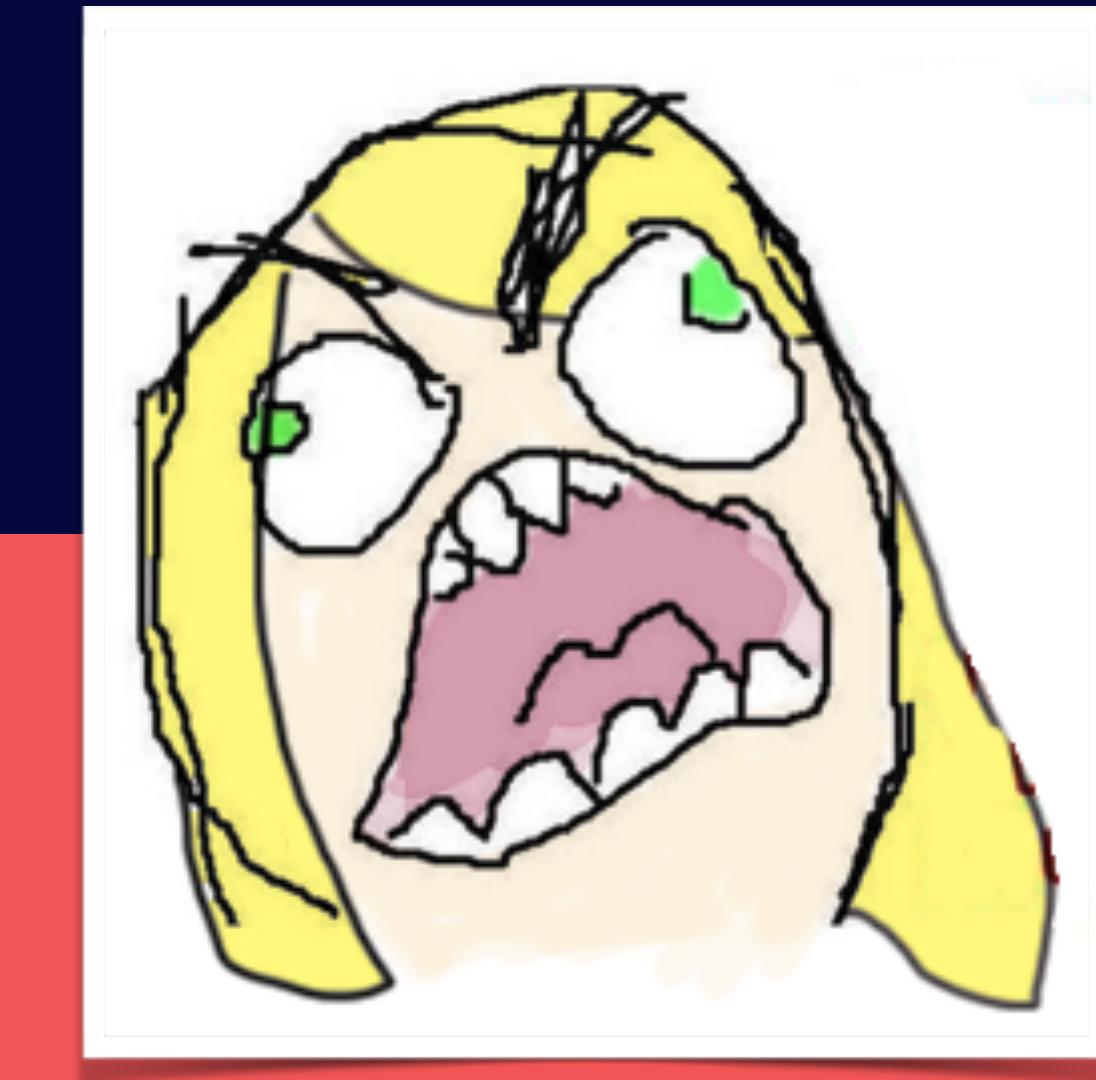
# SOLUTION



Experts, usually an **AUTHORITY**, will architect the system. They won't be bothered with technical implementation details because **THEIR TIME IS VALUABLE**.

The team will be required to **FOLLOW** the architecture so that the good qualities of the system are assured.

# RESULT



Implementation lags and finger pointings.

Whoever has the most **AUTHORITY** wins

The programmers deemed not to be "team players" because of arguing with the architect.

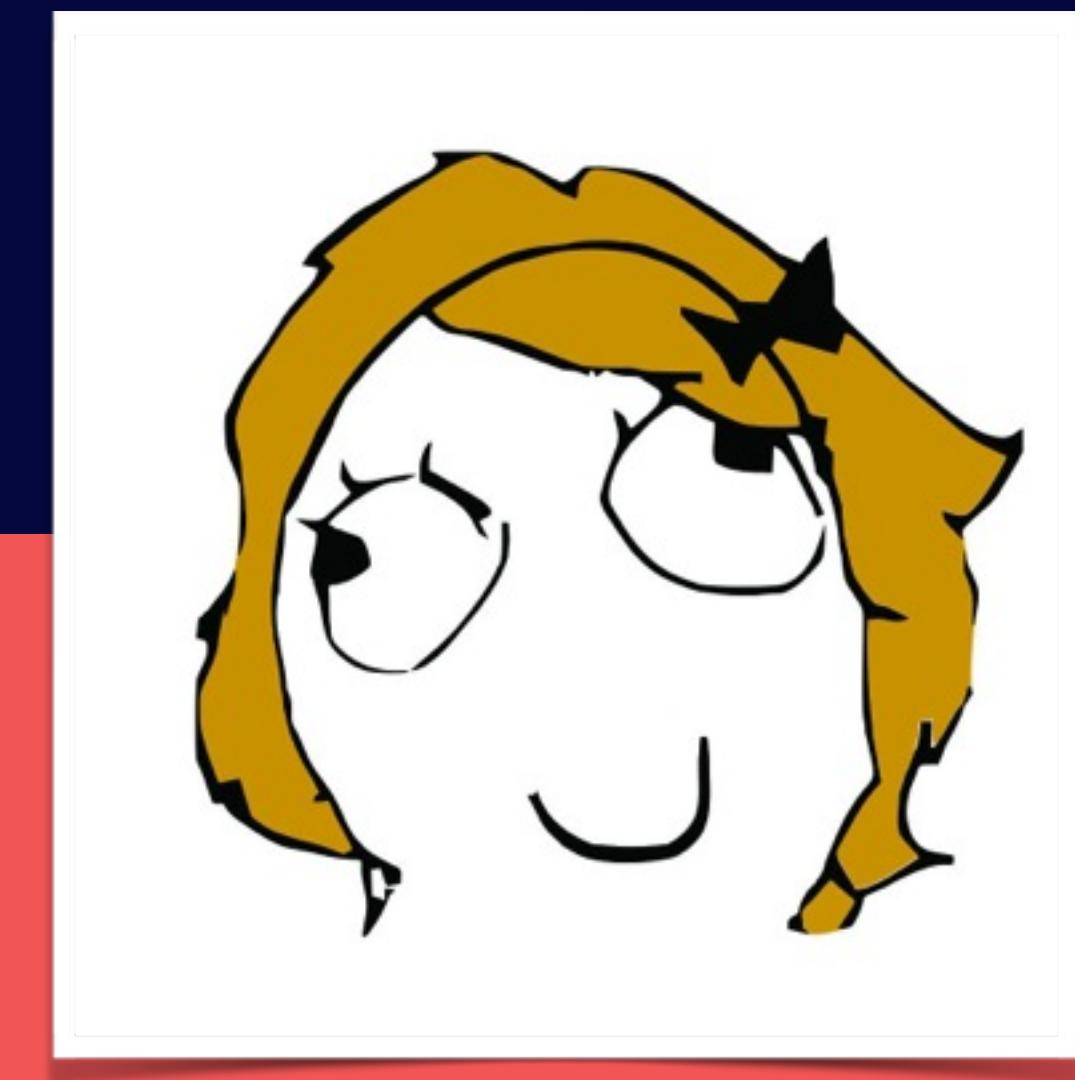
# SOLUTION



Experts, usually an **AUTHORITY**, will architect the system. They won't be bothered with technical implementation details because **THEIR TIME IS VALUABLE**.

The team will be required to **FOLLOW** the architecture so that the good qualities of the system are assured.

# SOLUTION



Get architects

INVOLVED AT AN  
IMPLEMENTATION LEVEL.

They need to be **AWARE** of how changes in their design affect the project.

# DEDICATED ARCHITECT



SINGLE POINT OF  
RESPONSIBILITY

# DEDICATED ARCHITECT



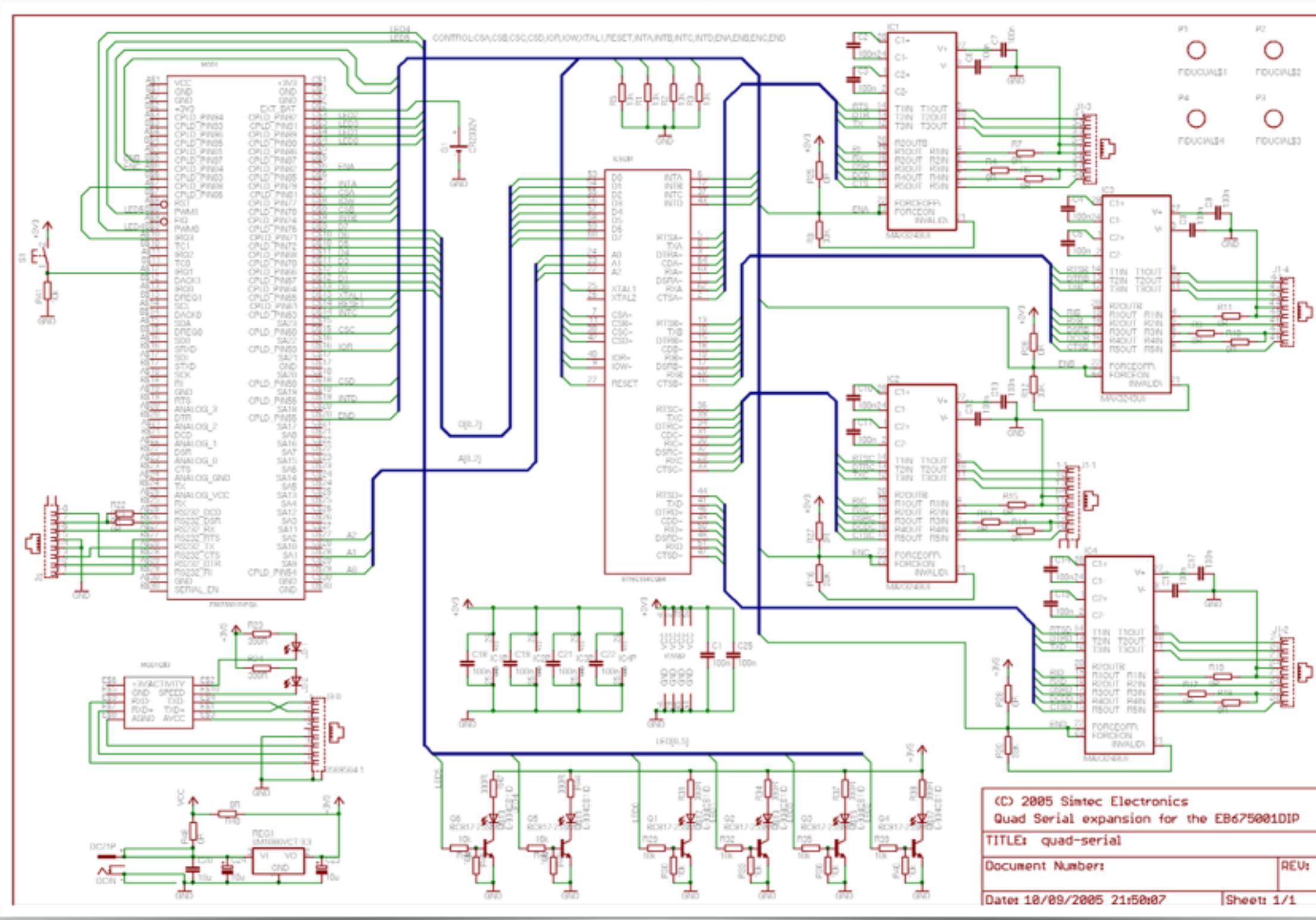
SINGLE POINT OF  
RESPONSIBILITY

# EVERYBODY IS AN ARCHITECT



JOINT  
RESPONSIBILITY

# CONFFLICT IN PROCESS



UP-FRONT DESIGN

```
(defun repl-fun (noprint)
  (/show0 "entering REPL").
  (loop
    ( unwind-protect
      (progn
        (scrub-control-stack).
        (sb!thread::get-foreground).
        (unless noprint.
          (flush-standard-output-streams).
          (funcall *repl-prompt-fun* *standard-output*).
          (force-output *standard-output*)).
        (let* ((form (funcall *repl-read-form-fun*
                               *standard-input*.
                               *standard-output*))
               (results (multiple-value-list (interactive-eval form))).
               (unless noprint.
                 (dolist (result results).
                   (fresh-line).
                   (prin1 result))))).
      (enable-stepping))).
  (disable-stepping)))).
```

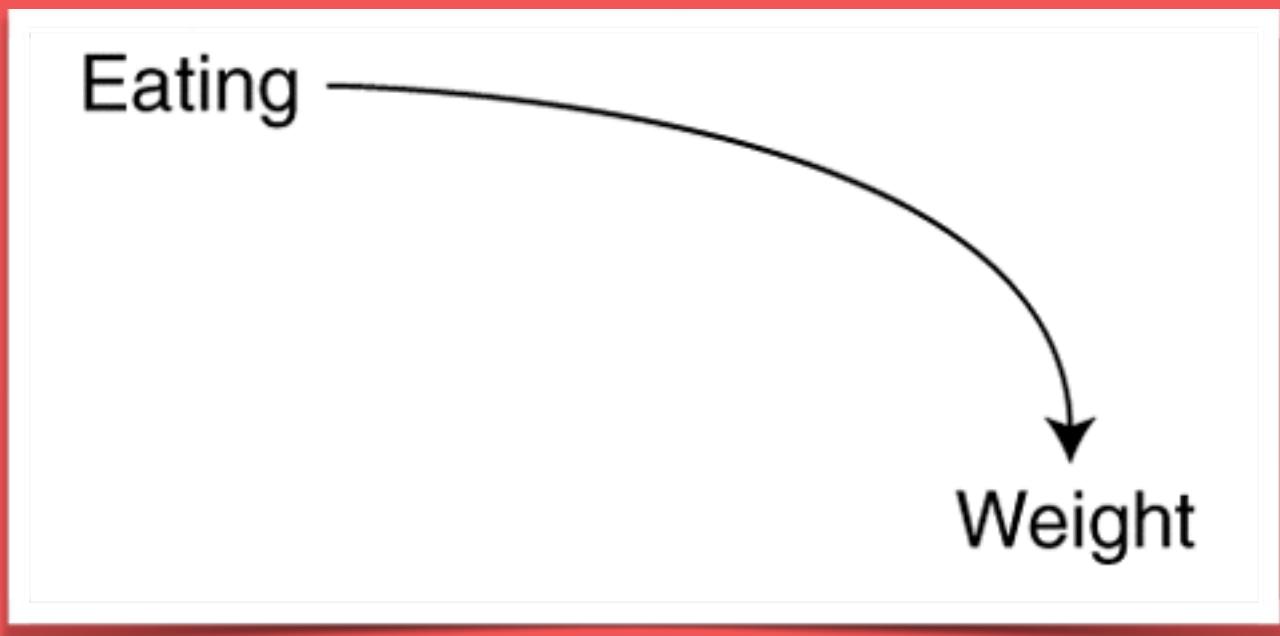
EVOLUTIONARY ARCHITECTURE

# CONFLICT IN PROCESS

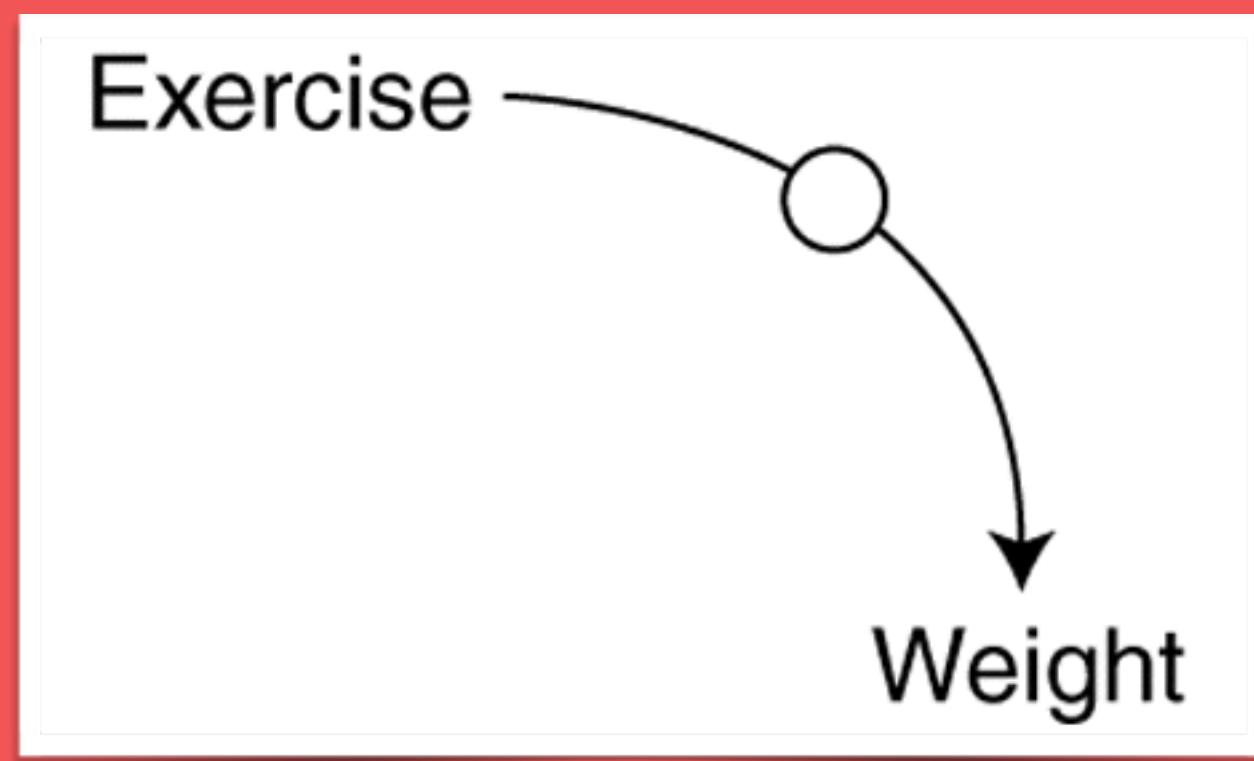
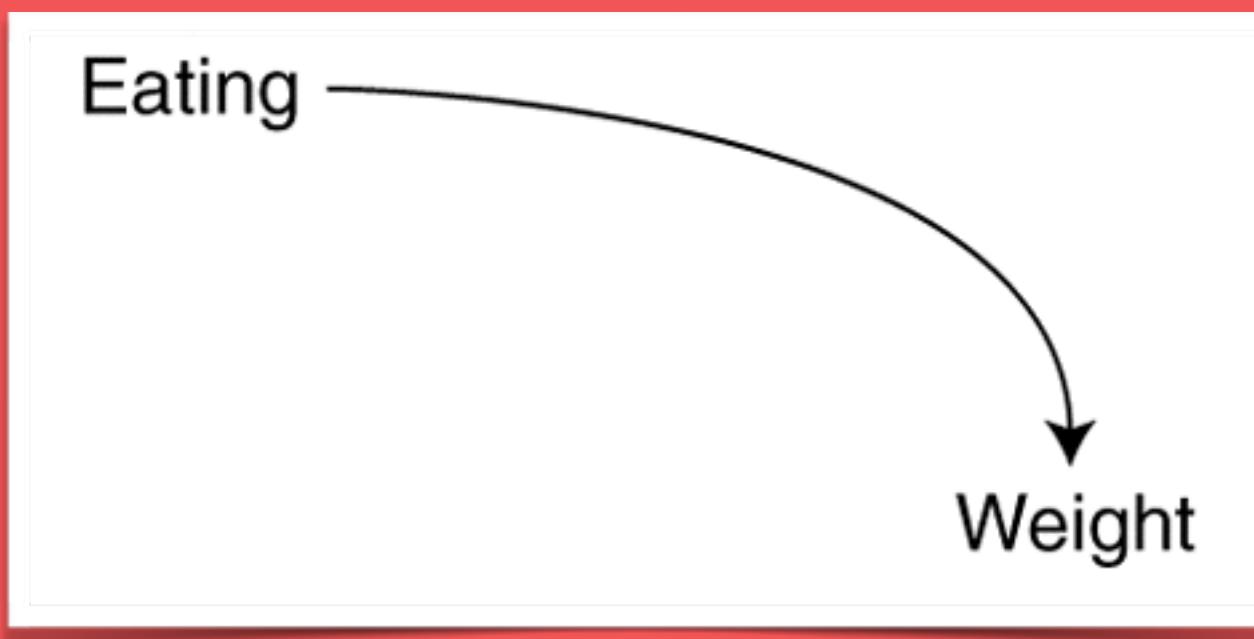
COMPREHENSIVE DOCUMENTATION

WORKING SOFTWARE

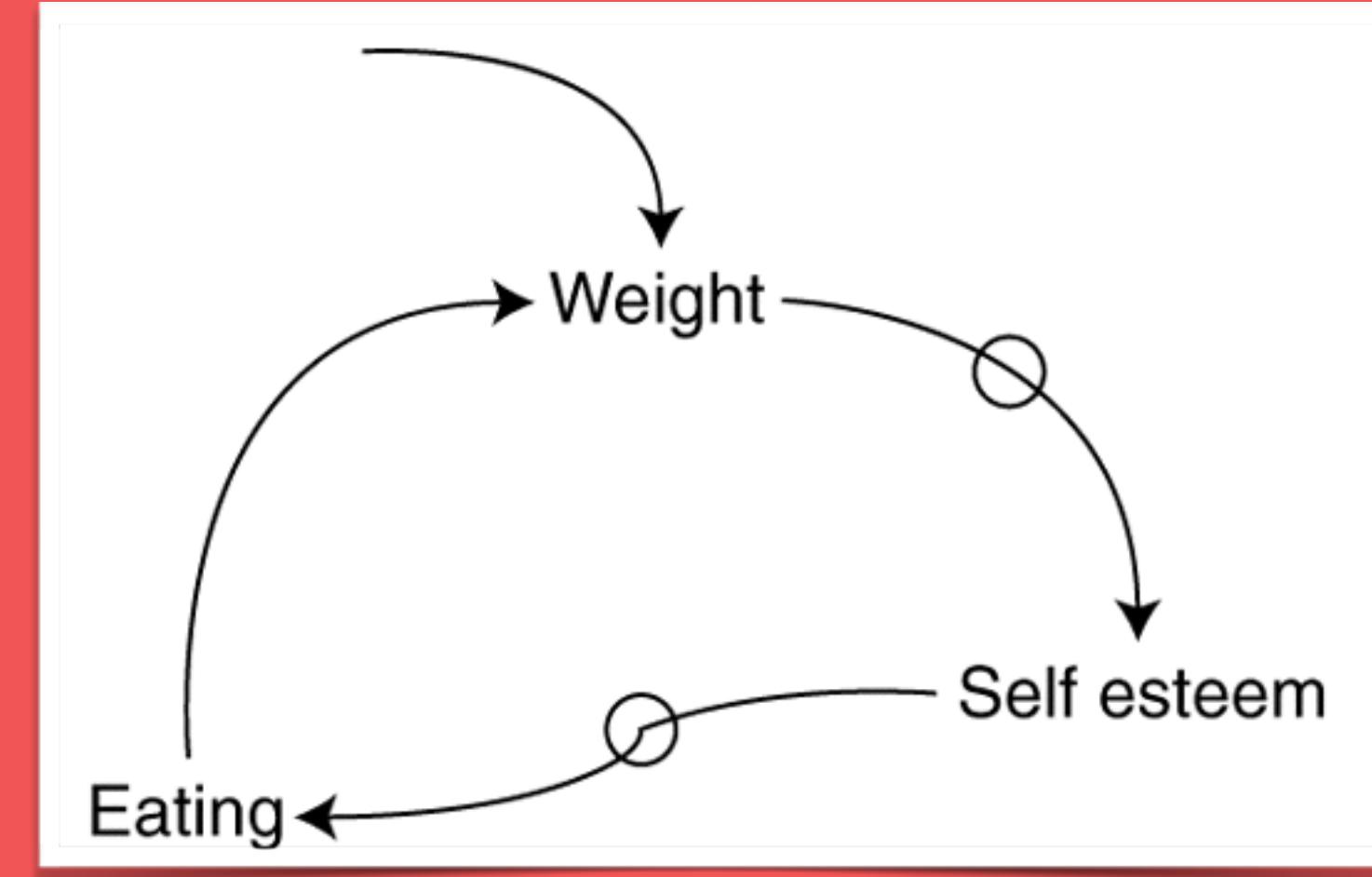
# INFLUENCEDIAGRAMS



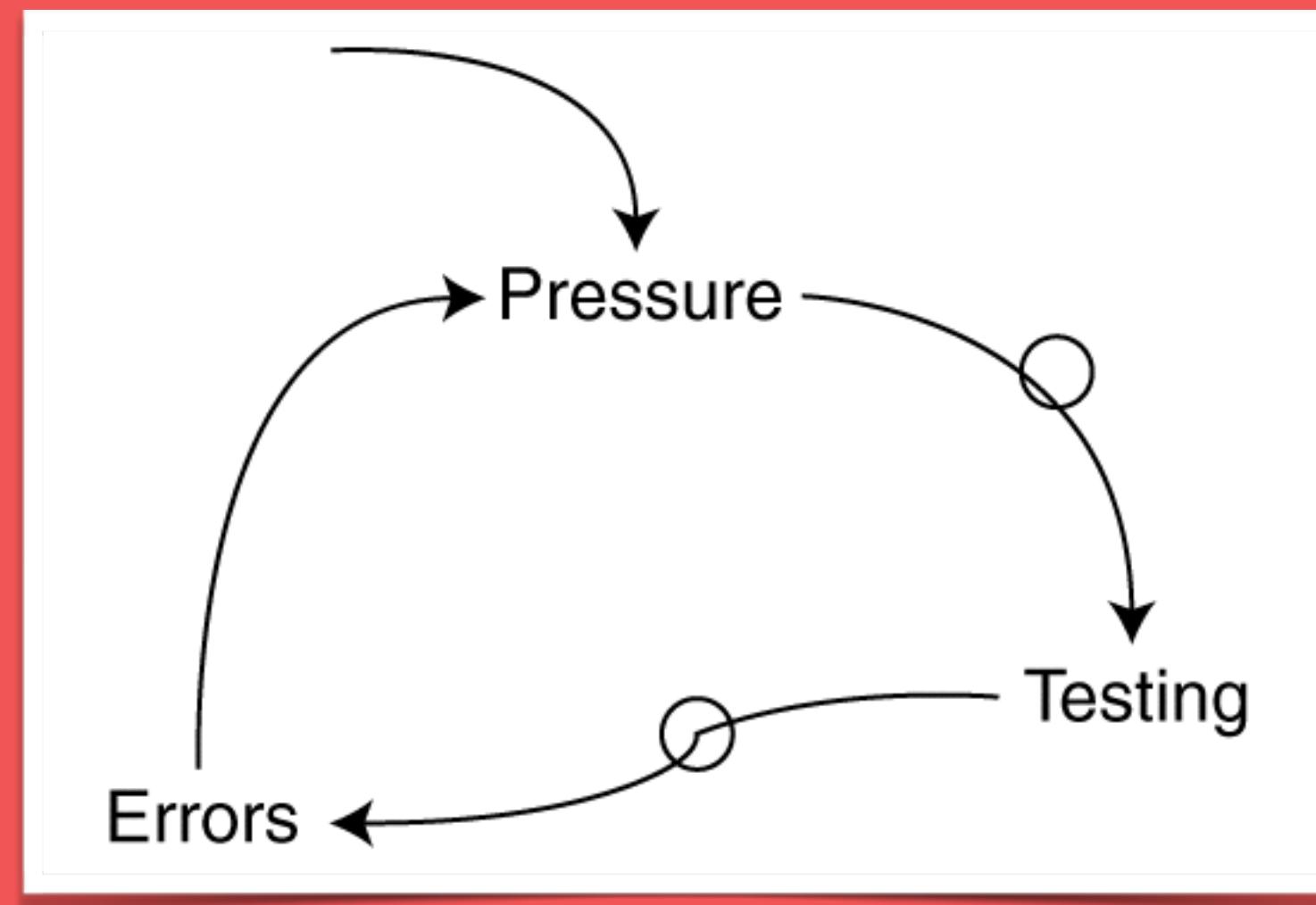
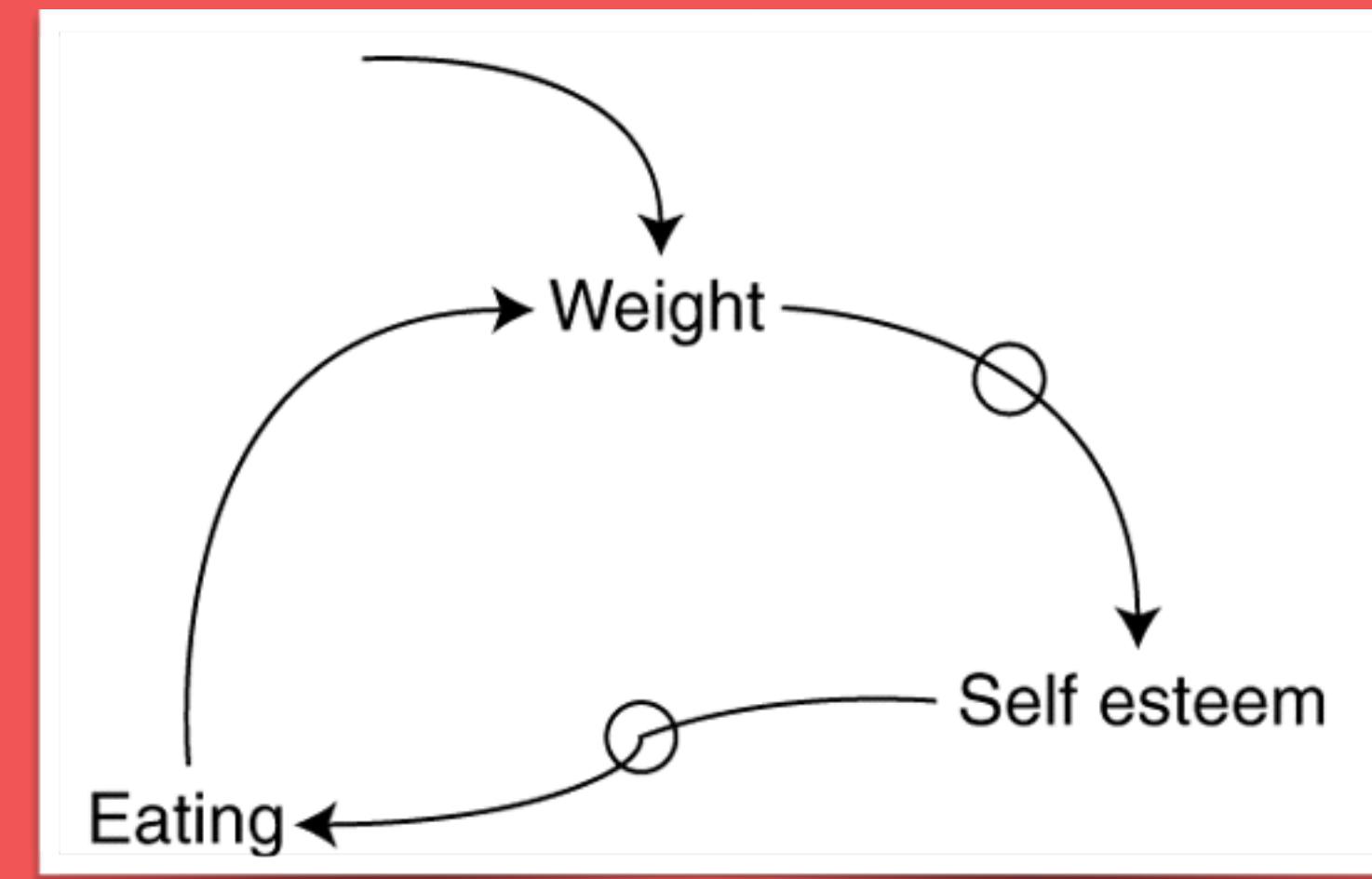
# INFLUENCEDIAGRAMS



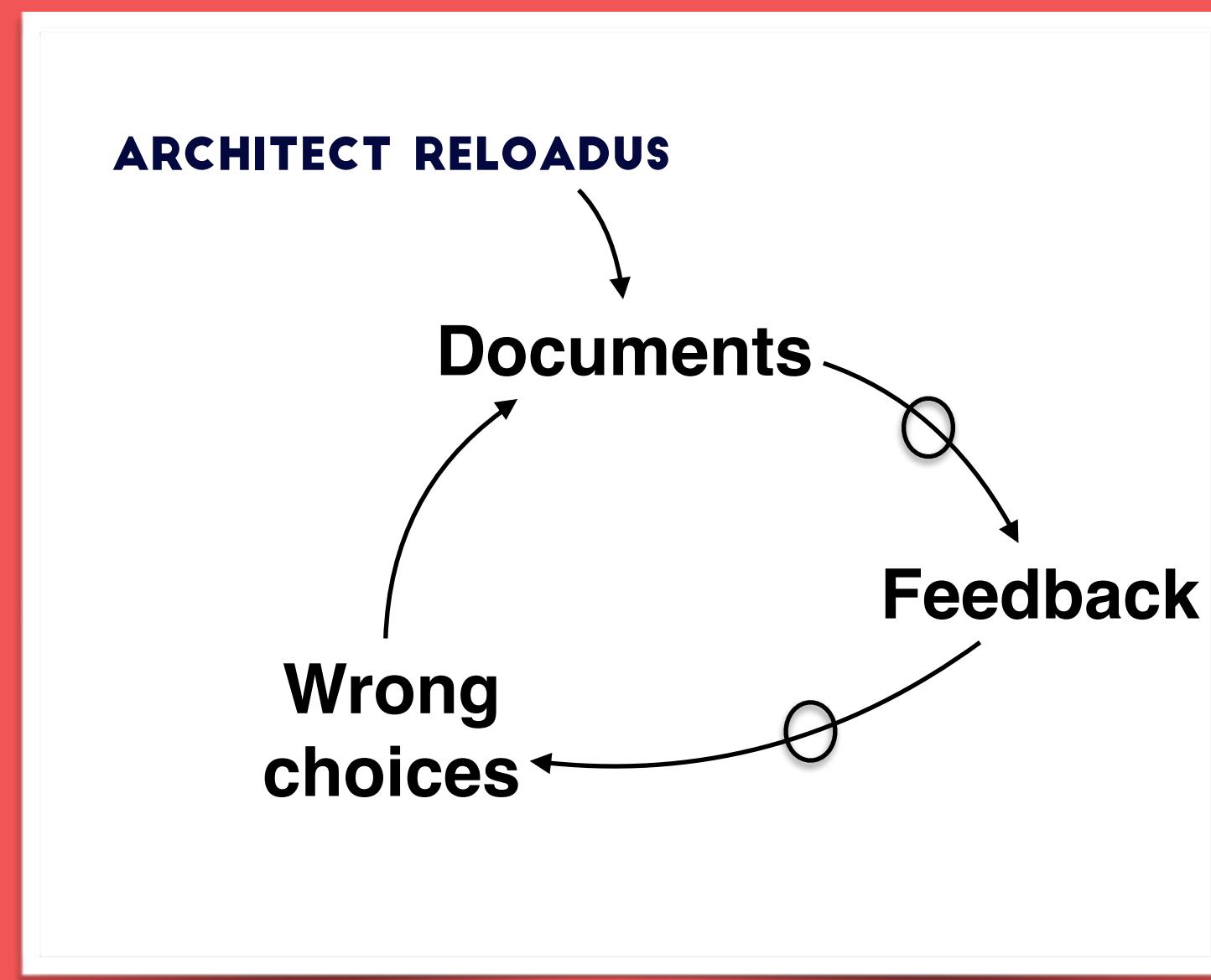
# INFLUENCEDIAGRAMS



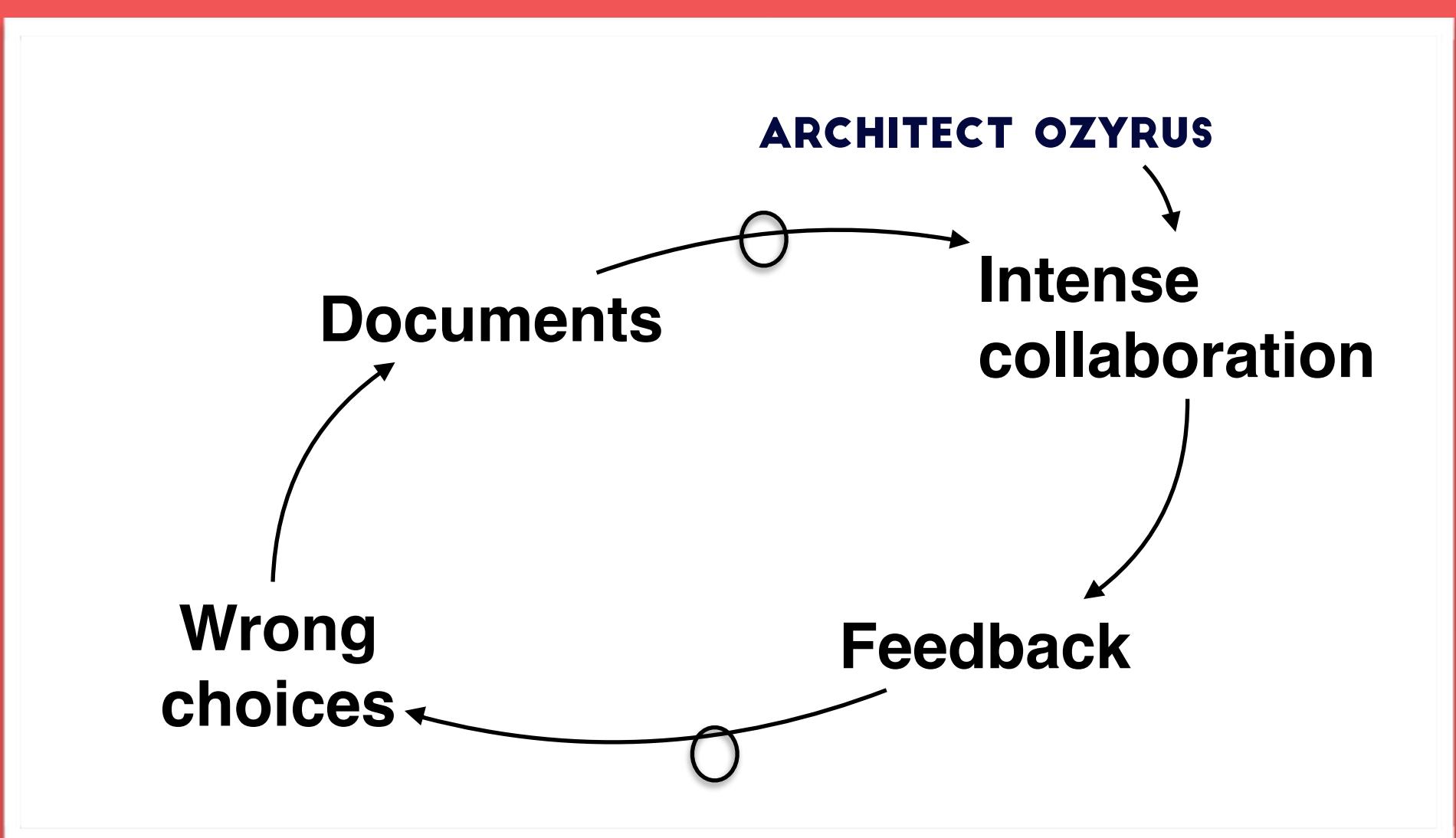
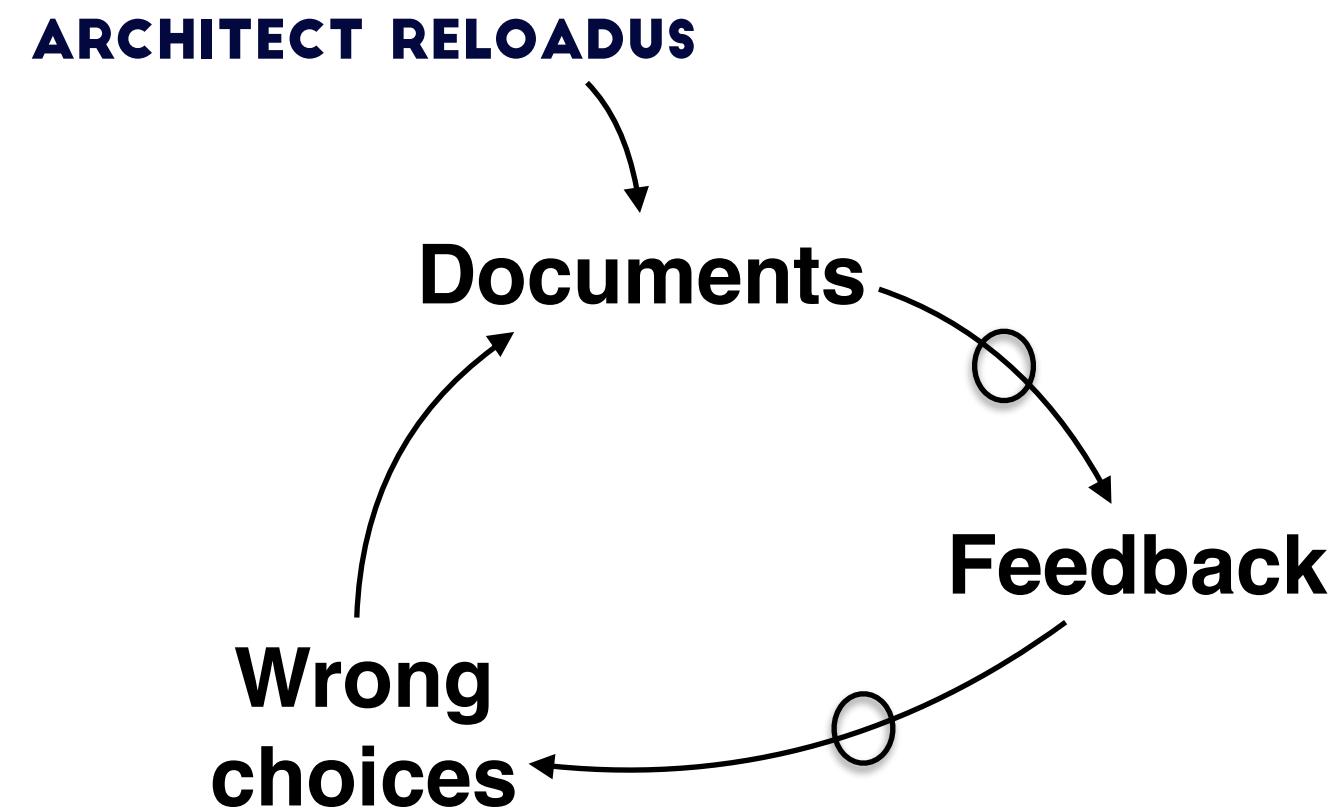
# INFLUENCEDIAGRAMS



# INFLUENCE DIAGRAMS



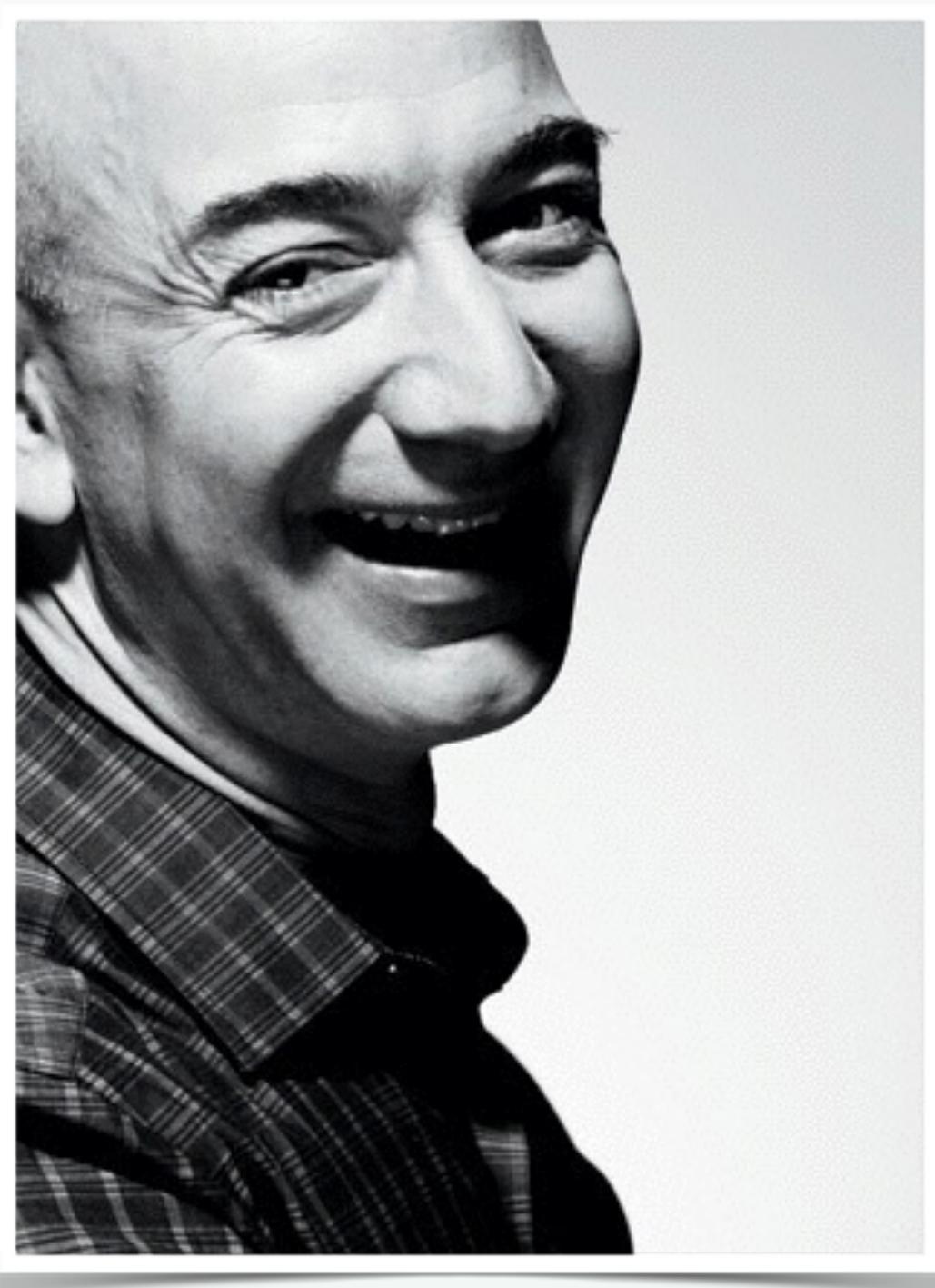
# INFLUENCEDIAGRAMS



# MICROSERVICE



# MICROSERVICE



# THE ARCHITECT?

@ARIALDOMARTINI

@NICOLABALDI