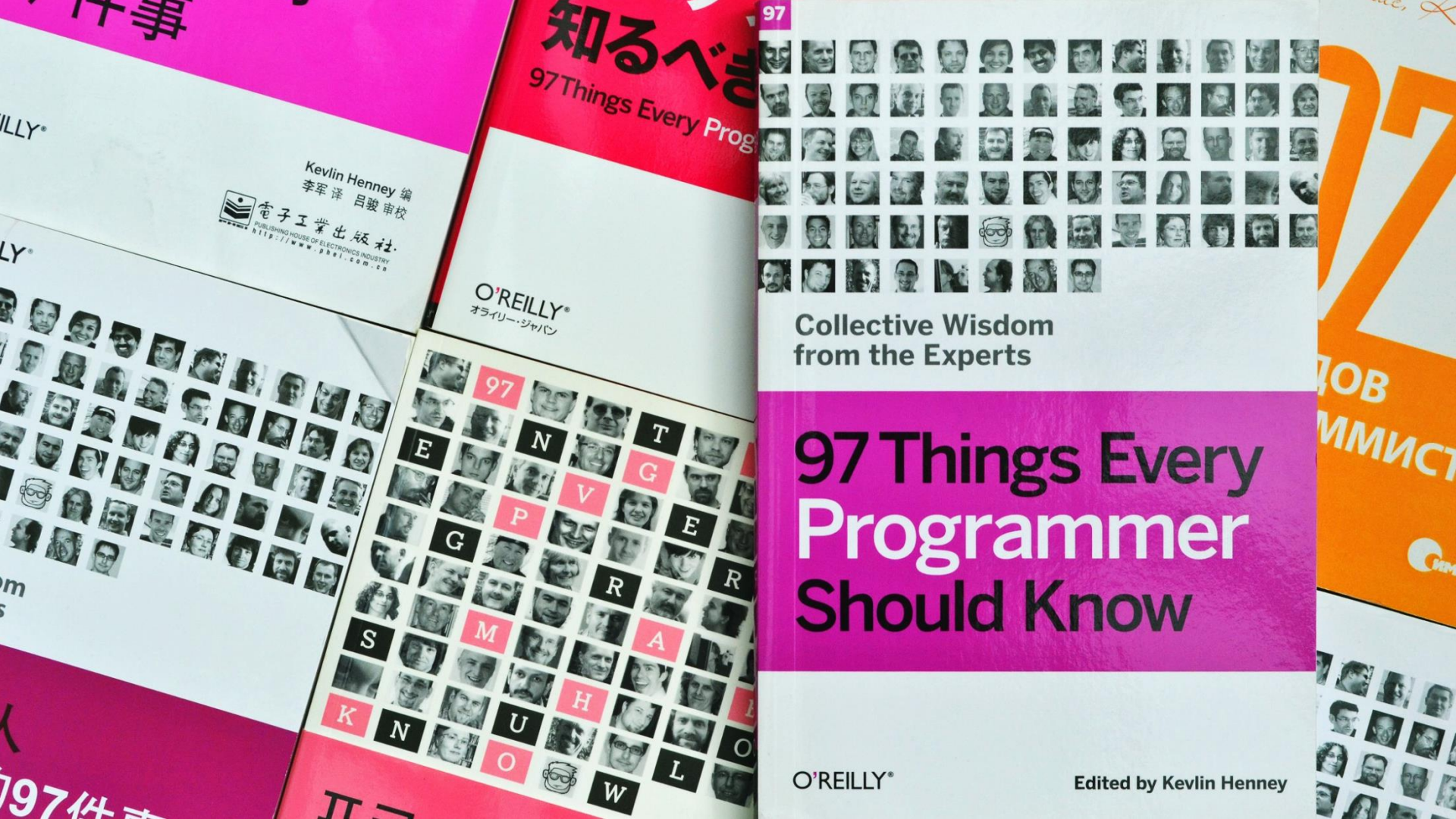


# Craft and Scale

## @KevlinHenney





知るべき  
97 Things Every Prog

Kevin Henney 編  
李军译 吕骏审校  
電子工業出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

O'REILLY®  
オライリー・ジャパン



Collective Wisdom  
from the Experts

# 97 Things Every Programmer Should Know

O'REILLY®

Edited by Kevlin Henney

97件事

97件事

JOB  
MMICT

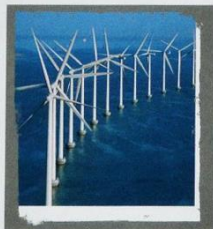




WILEY SERIES IN  
SOFTWARE DESIGN PATTERNS

# PATTERN-ORIENTED SOFTWARE ARCHITECTURE

A Pattern Language for  
Distributed Computing



**Volume 4**

Frank Buschmann  
Kevlin Henney  
Douglas C. Schmidt



WILEY SERIES IN  
SOFTWARE DESIGN PATTERNS

# PATTERN-ORIENTED SOFTWARE ARCHITECTURE

On Patterns and Pattern Languages



**Volume 5**

Frank Buschmann  
Kevlin Henney  
Douglas C. Schmidt





WILEY SERIES IN  
SOFTWARE DESIGN PATTERNS

# PATTERN-ORIENTED SOFTWARE ARCHITECTURE

On Patterns and Pattern Languages



**Volume 5**

Frank Buschmann  
Kevlin Henney  
Douglas C. Schmidt

**Art. Craft.  
Engineering.  
Science. These  
are the swirling  
muses of design  
patterns. Art and  
science are  
stories; craft and  
engineering are  
actions.**

*Wayne Cool*

# The Draft

Normally screws are so cheap and small and simple you think of them as unimportant. But now, as your Quality awareness becomes stronger, you realize that this one, individual, particular screw is neither cheap nor small nor unimportant.

Robert M Pirsig

*Zen and the Art of Motorcycle Maintenance*

Right now this screw is worth exactly the selling price of the whole motorcycle, because the motorcycle is actually valueless until you get the screw out.

With this reevaluation of the screw comes a willingness to expand your knowledge of it.

Robert M Pirsig

*Zen and the Art of Motorcycle Maintenance*



# More Programming Pearls

Confessions of a Coder

Jon Bentley





The image shows the front cover of the book 'More Programming Pearls: Confessions of a Coder' by Jon Bentley. The cover has a dark red upper half and a dark, textured lower half. A close-up of a vintage computer keyboard is visible in the lower-left corner, with keys like 'E', 'R', 'T', 'Y', 'D', 'F', 'N', 'V', '2', '3', '4', '5', '8', '9' visible. The title 'Details count.' is overlaid in large yellow font across the middle.

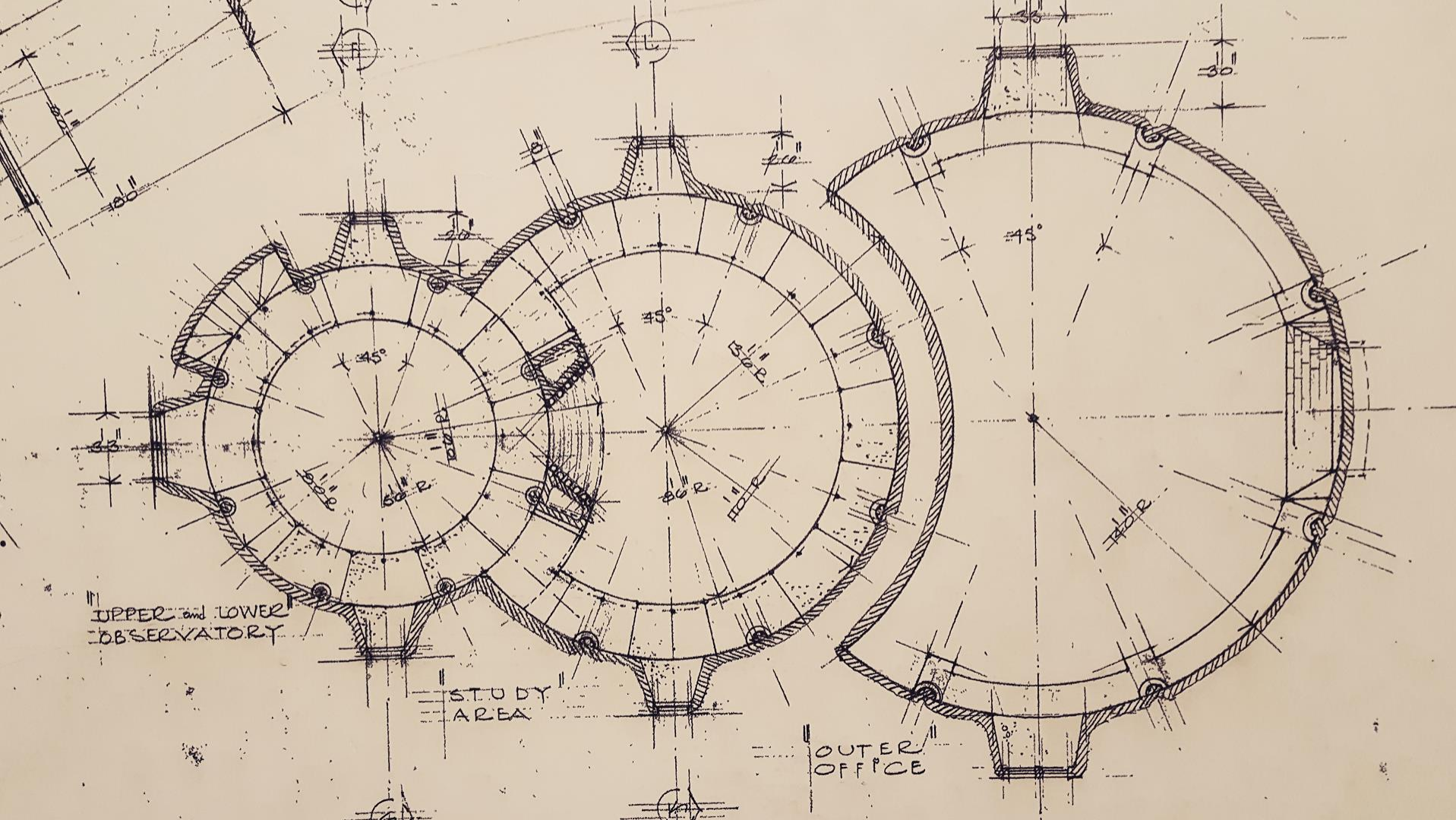
More Programming Pearls

Confessions of a Coder

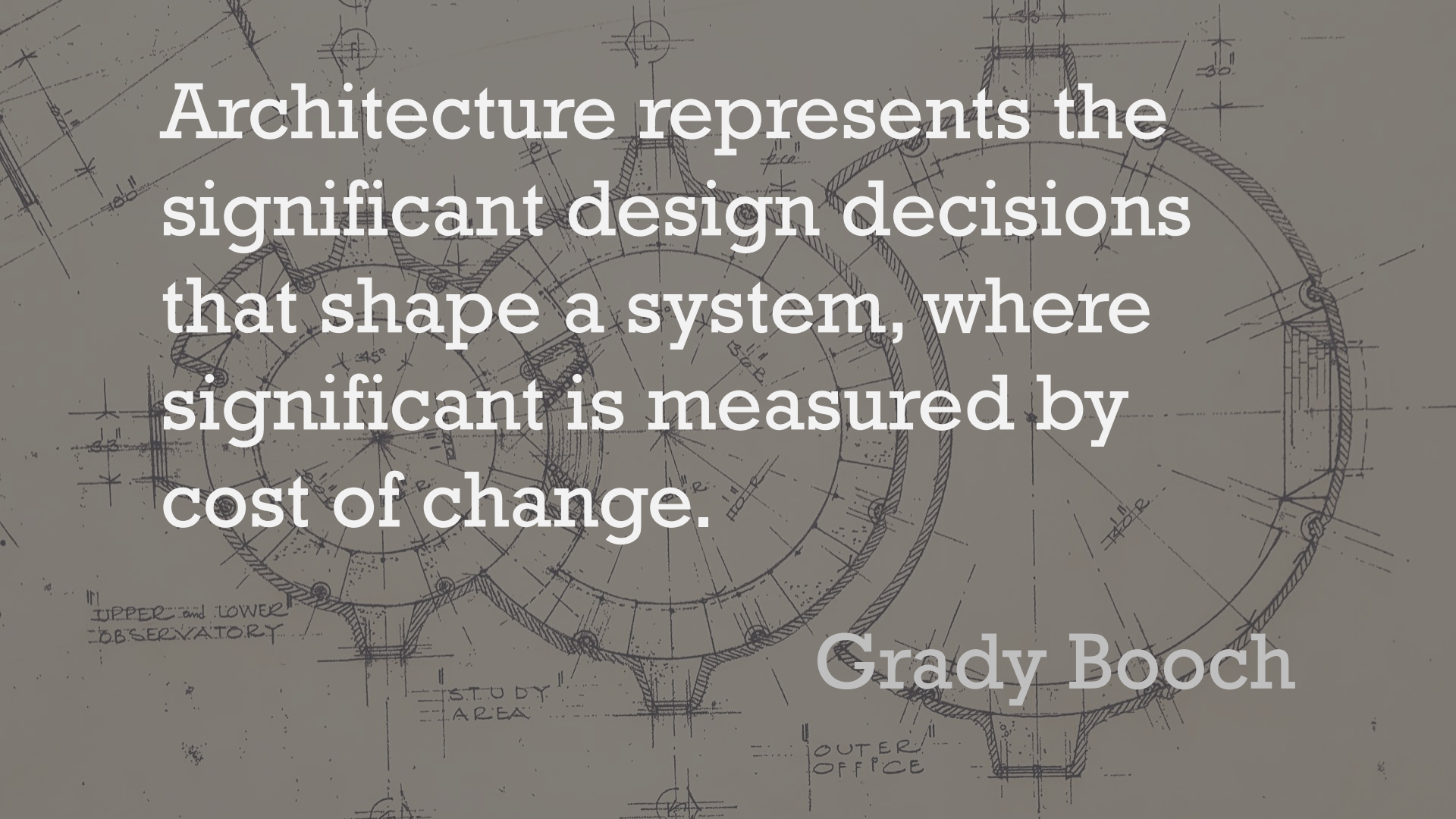
Jon Bentley

# Details count.

Peter Weinberger





A detailed architectural drawing of a circular building plan, likely a dome or observatory. The drawing features concentric circles, radial lines, and various dimensions in feet and inches. Handwritten labels include "UPPER and LOWER OBSERVATORY", "STUDY AREA", and "OUTER OFFICE". The text "Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change." is overlaid in white.

Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change.

Grady Booch

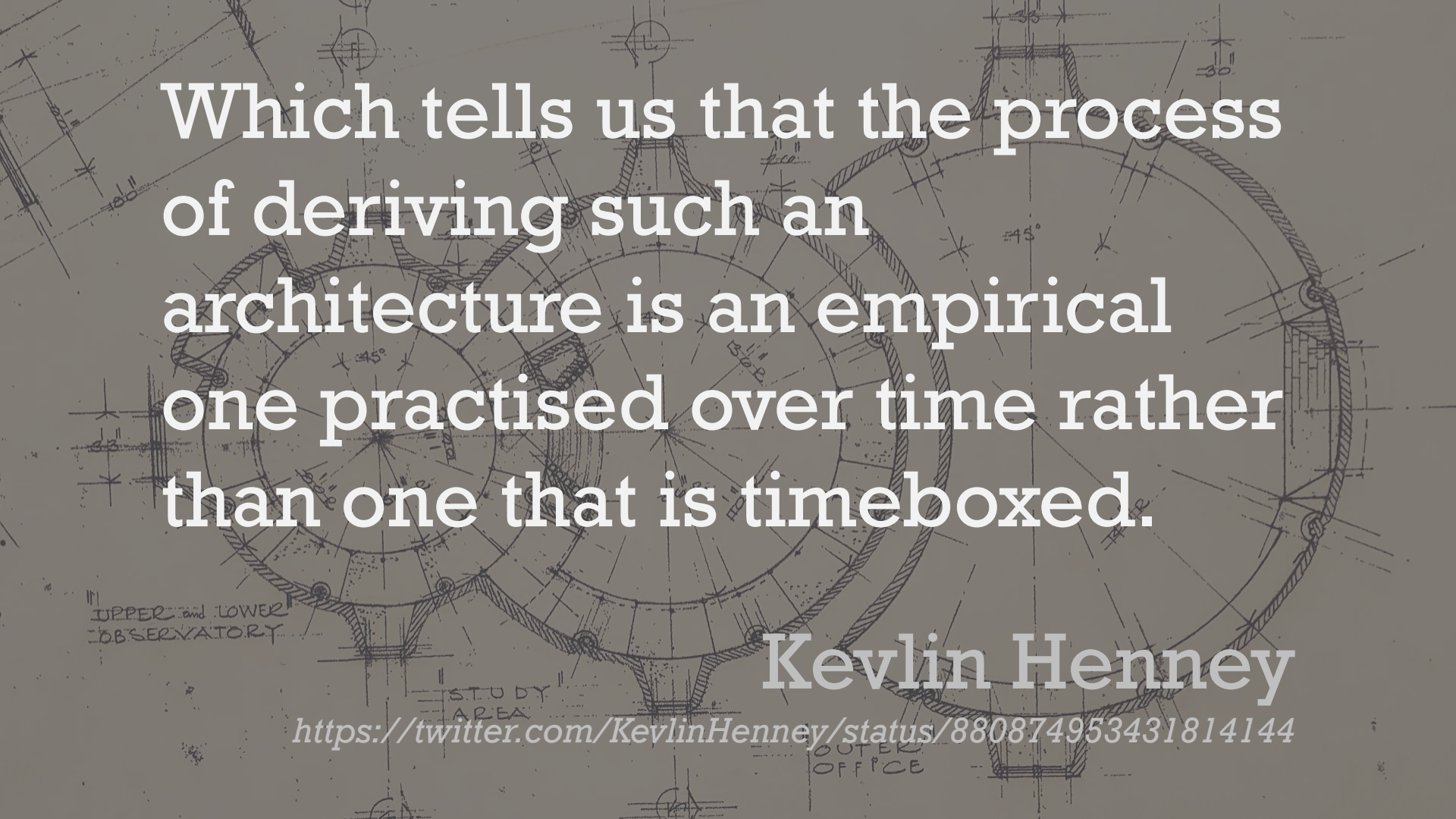
# Kevlin Henney

UPPER and LOWER  
OBSERVATORY

STUDY  
AREA

OUTER  
OFFICE



The background is a detailed architectural drawing of a circular structure, likely an observatory. It features concentric circles, radial lines, and various technical annotations. Labels include 'UPPER and LOWER OBSERVATORY' on the left, 'STUDY AREA' in the center, and 'OUTER OFFICE' at the bottom. Dimensions like '30"', '45°', and '150 R.' are visible. The drawing is rendered in a technical, hand-drawn style with fine lines and hatching.

Which tells us that the process  
of deriving such an  
architecture is an empirical  
one practised over time rather  
than one that is timeboxed.

Kevlin Henney

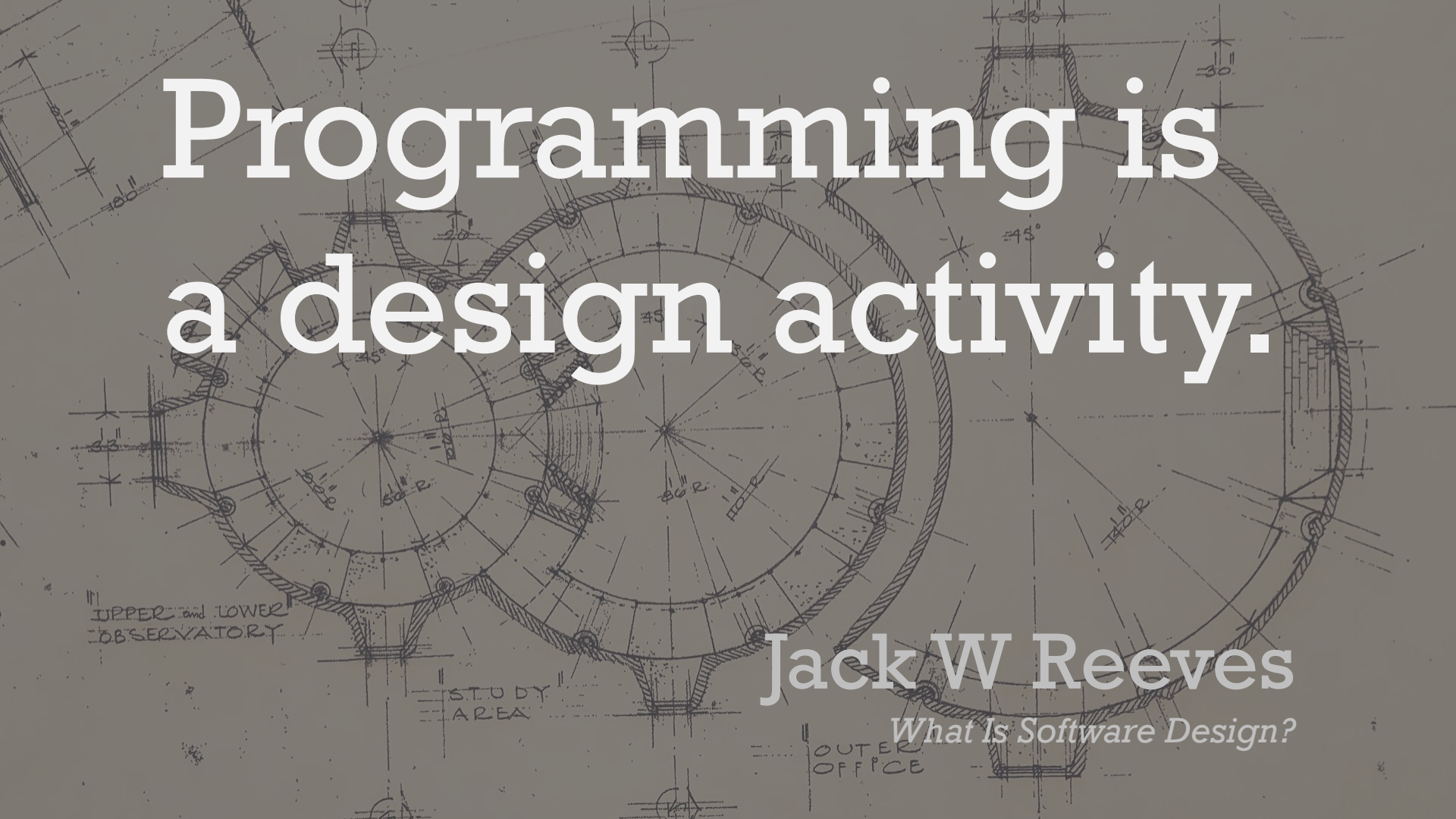
<https://twitter.com/KevlinHenney/status/880874953431814144>

A detailed technical drawing of a dome-shaped structure, possibly a building or observatory. The drawing shows a cross-section with various internal components and structural elements. Labels include "UPPER and LOWER OBSERVATORY", "STUDY AREA", and "OUTER OFFICE". Dimensions and angles are marked throughout the drawing.

It's expensive to  
know everything  
up front.

Kolton Andrus

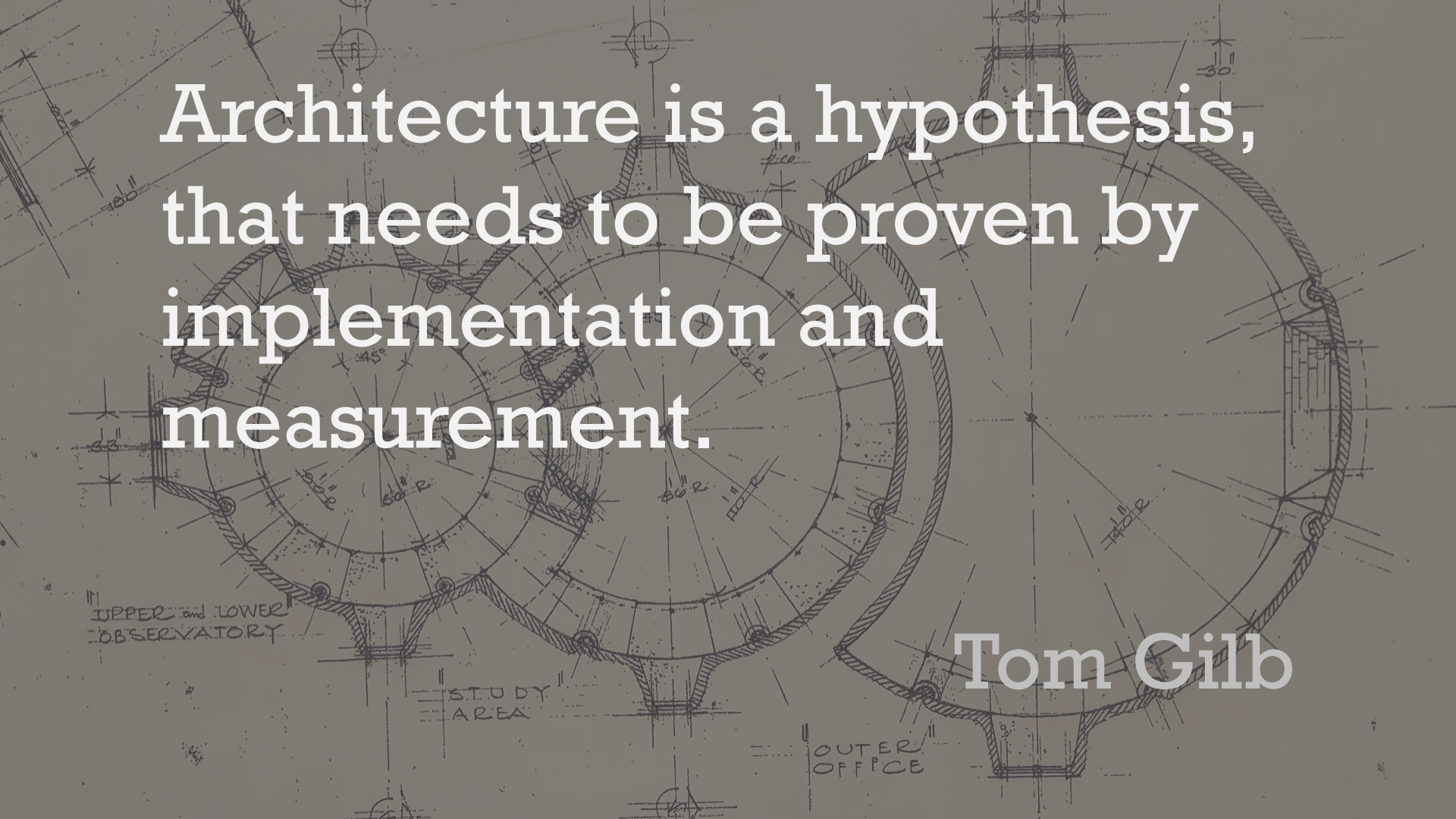


A detailed technical drawing of a dome-shaped structure, possibly a planetarium or observatory. The drawing shows concentric circles and radial lines, with various dimensions and labels. Labels include 'UPPER and LOWER OBSERVATORY' on the left, 'STUDY AREA' at the bottom center, and 'OUTER OFFICE' at the bottom right. The main title 'Programming is a design activity.' is overlaid in large white text.

# Programming is a design activity.

Jack W Reeves

*What Is Software Design?*

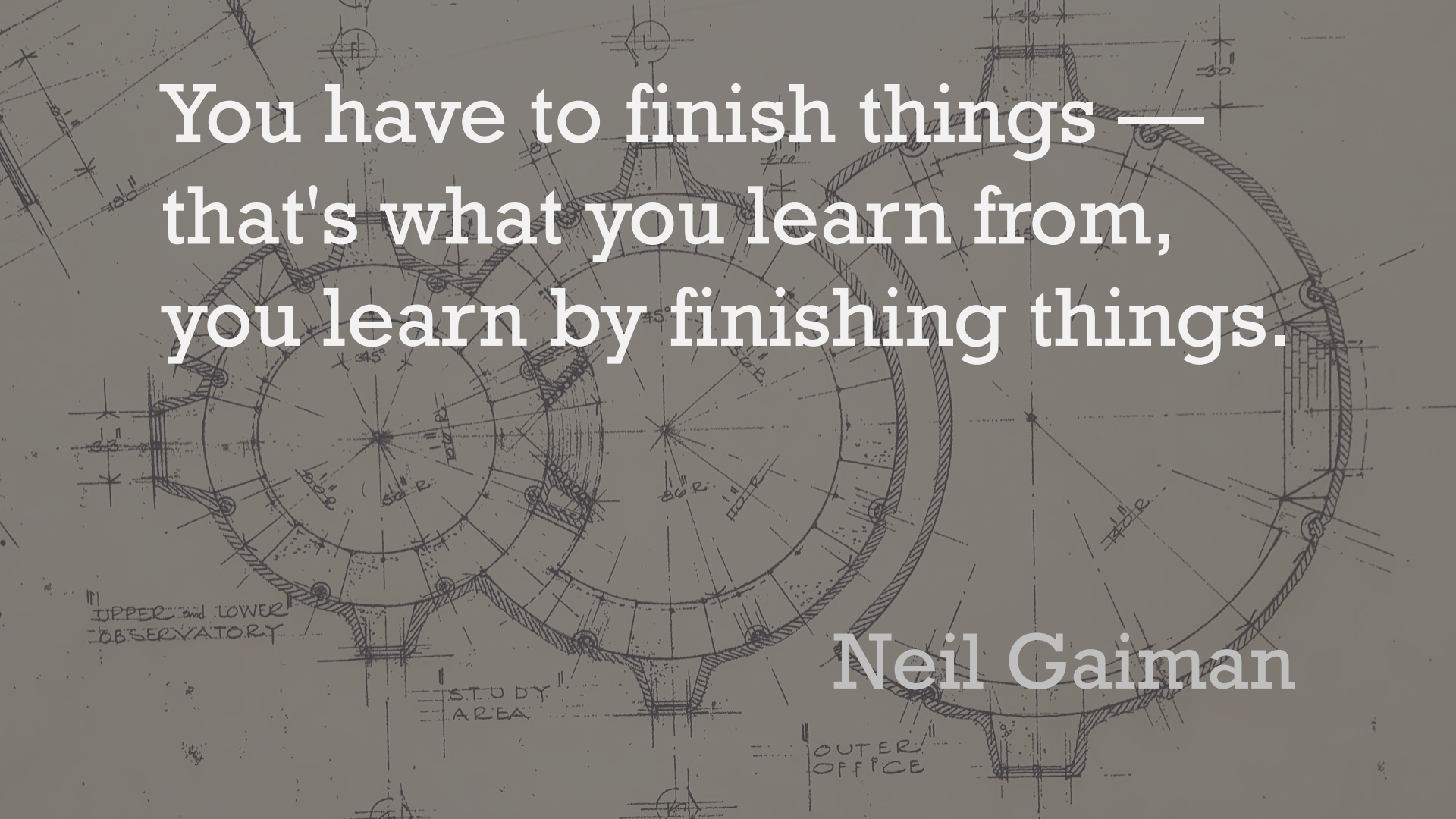


Architecture is a hypothesis,  
that needs to be proven by  
implementation and  
measurement.

The background is a detailed architectural drawing of a circular building plan. It features concentric circles and radial lines, with various dimensions and labels. The drawing is rendered in a technical, hand-drawn style with fine lines and hatching. The text is overlaid in a large, white, sans-serif font.

Tom Gilb



A detailed technical drawing of a watch movement, showing various gears, jewels, and components. The drawing is rendered in a dark, sketchy style with fine lines and annotations. The text is overlaid on the central part of the drawing.

You have to finish things —  
that's what you learn from,  
you learn by finishing things.

Neil Gaiman

code  
tests  
scripts



codified  
knowledge

knowledge  
acquisition



learning

communication

communication



social  
negotiation

model of  
participation



**The first rule of scaling agile is...  
don't do it!**

**Or rather, try to avoid it for as  
long as you possibly can.**

**Andreas Rowell**

*First Rule of Scaling Agile – DON'T!*

<https://www.linkedin.com/pulse/first-rule-scaling-agile-dont-andreas-rowell/>

**Software development does *not* have economies of scale.**

**Development has *diseconomies* of scale.**

**Allan Kelly**

*Beyond Projects*

<http://www.slideshare.net/allankellynet/no-projects-beyond-projects-refreshed-version>



$$t = t_1$$



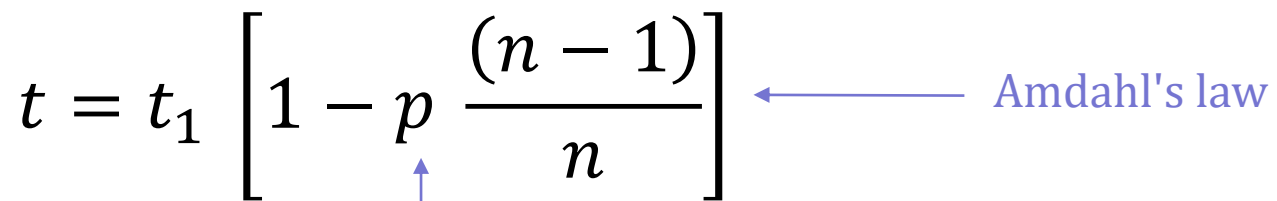
completion time  
for one person

$$t = \frac{t_1}{n}$$



division of  
labour



$$t = t_1 \left[ 1 - p \frac{(n - 1)}{n} \right]$$


Amdahl's law

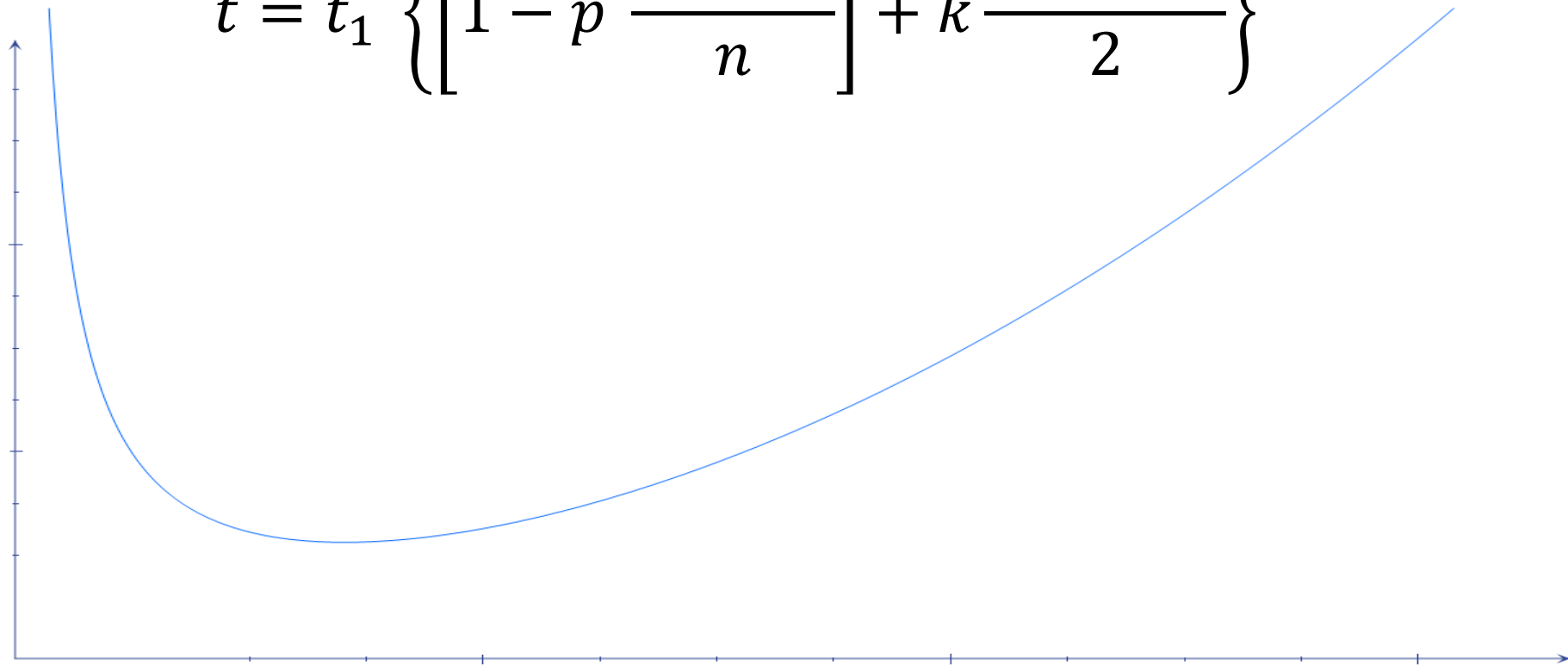
portion in  
parallel

$$t = t_1 \left\{ \left[ 1 - p \frac{(n-1)}{n} \right] + k \frac{n(n-1)}{2} \right\}$$

connections  
(worst case)

communication  
overhead  
(typical)

$$t = t_1 \left\{ \left[ 1 - p \frac{(n-1)}{n} \right] + k \frac{n(n-1)}{2} \right\}$$



# Man recreates TSA's \$47,400 "head this way" app in 10 minutes

TSA guards at airports had a new weapon in their arsenal in 2014: tablets that they held up to **randomly direct travelers into different lines**. According to the TSA's documentation, **they spent \$47k developing the app that did this**. In this YouTube video, Chris Pacia develops a clone of the app in a few minutes.



I have yet to see any problem,  
however complicated, which,  
when you looked at it in the  
right way, did not become still  
more complicated.

Anderson's Law

The Facebook iOS app has over 18,000 Objective-C classes, and in a single week 429 people contributing to it.

*Facebook's code quality problem*

Graham King

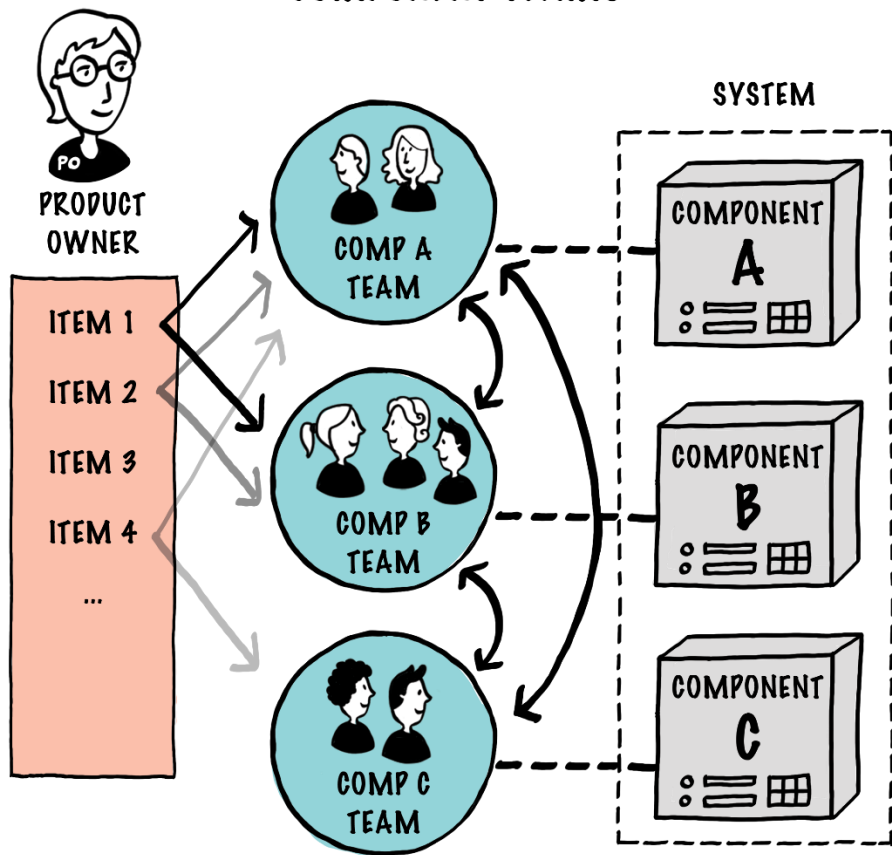
<http://www.darkcoding.net/software/facebooks-code-quality-problem/>

E.F. Schumacher  
**Small  
is Beautiful**

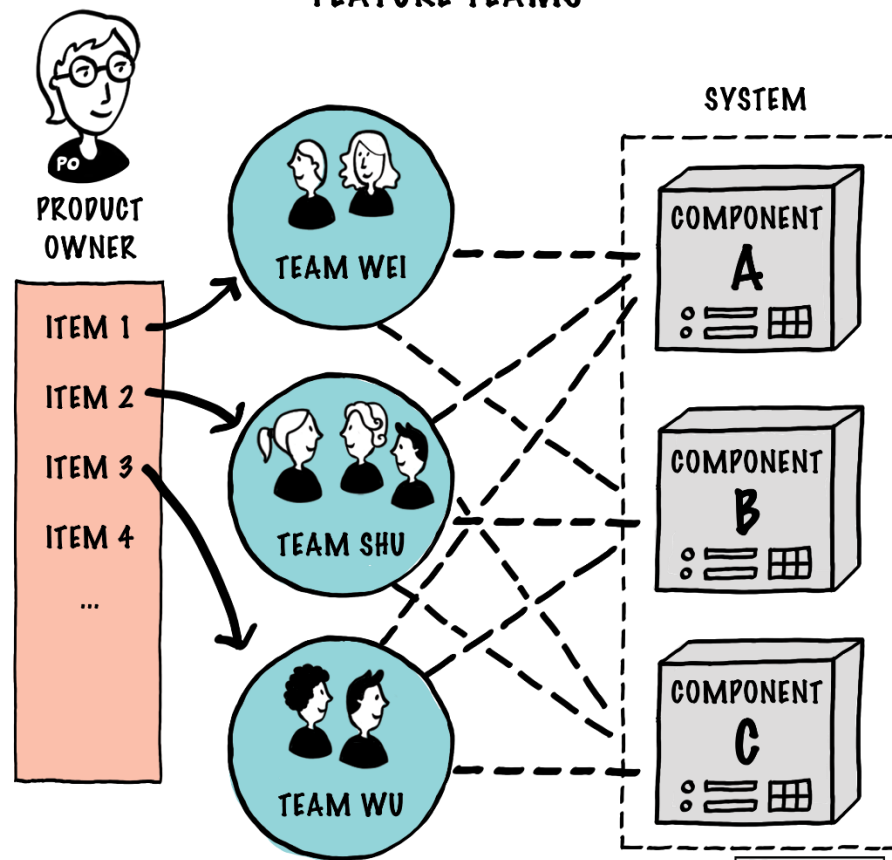


For every activity  
there is a certain  
appropriate scale.

## COMPONENT TEAMS



## FEATURE TEAMS





Good fences make good  
neighbours.

*Josh:* "Good fences make good neighbors." Did he talk about that?

*Donna:* Yeah.

*Josh:* What did he say?

*Donna:* Basically, that if you stay within your personal space, you'll end up getting along with everyone.

*The West Wing*

*Josh:* You had to study modern poetry.

*Donna:* Yes.

*Josh:* Is that what Frost meant?

*Donna:* No, he meant that boundaries are what alienate us from each other.

*Josh:* Why did he say "Good fences make good neighbors?"

*Donna:* He was being ironic.

*The West Wing*

Something there is that  
doesn't love a wall.

Robert Frost

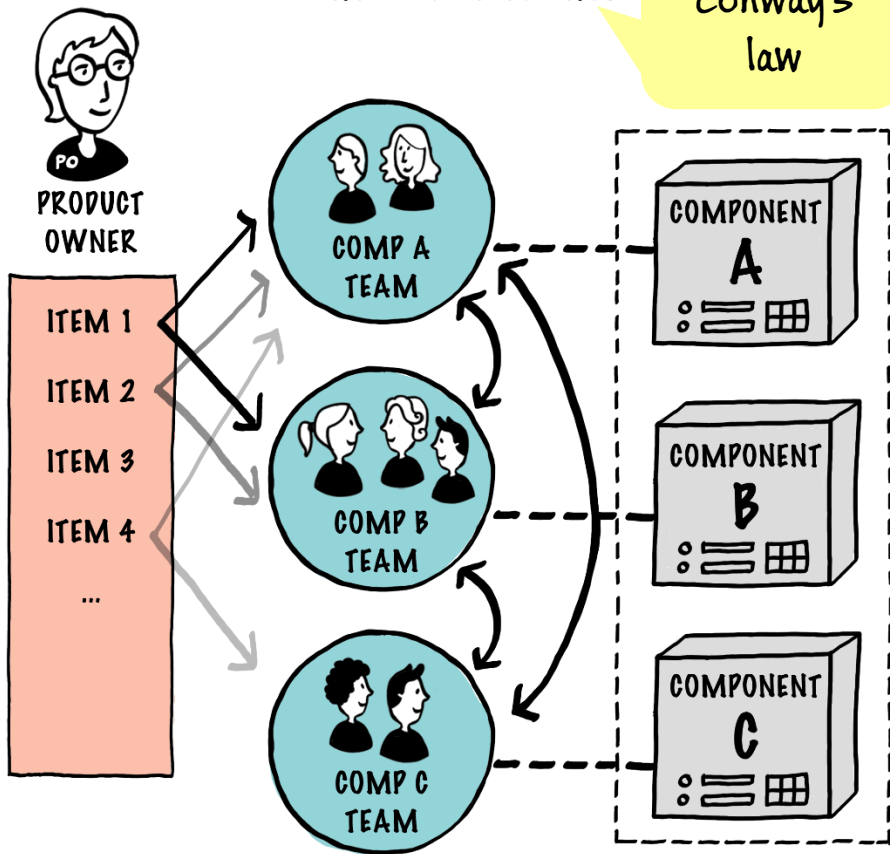


Be conservative in what  
you do, be liberal in  
what you accept from  
others.

Jon Postel

## COMPONENT TEAMS

Follows  
Conway's  
law



## FEATURE TEAMS

Avoids  
Conway's  
law

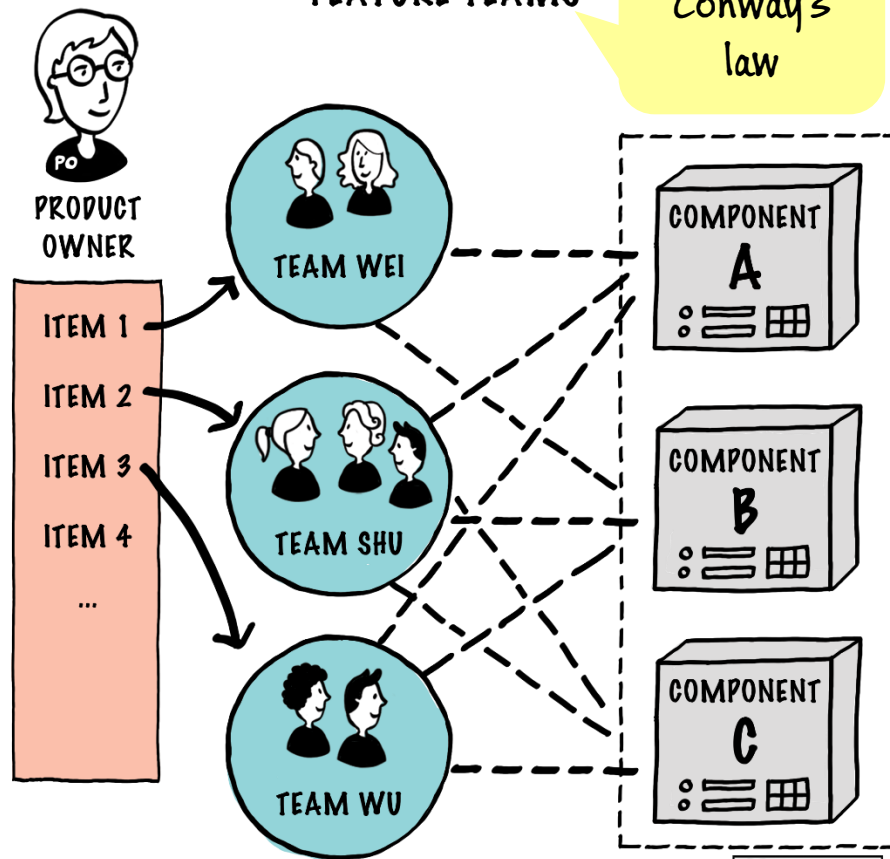
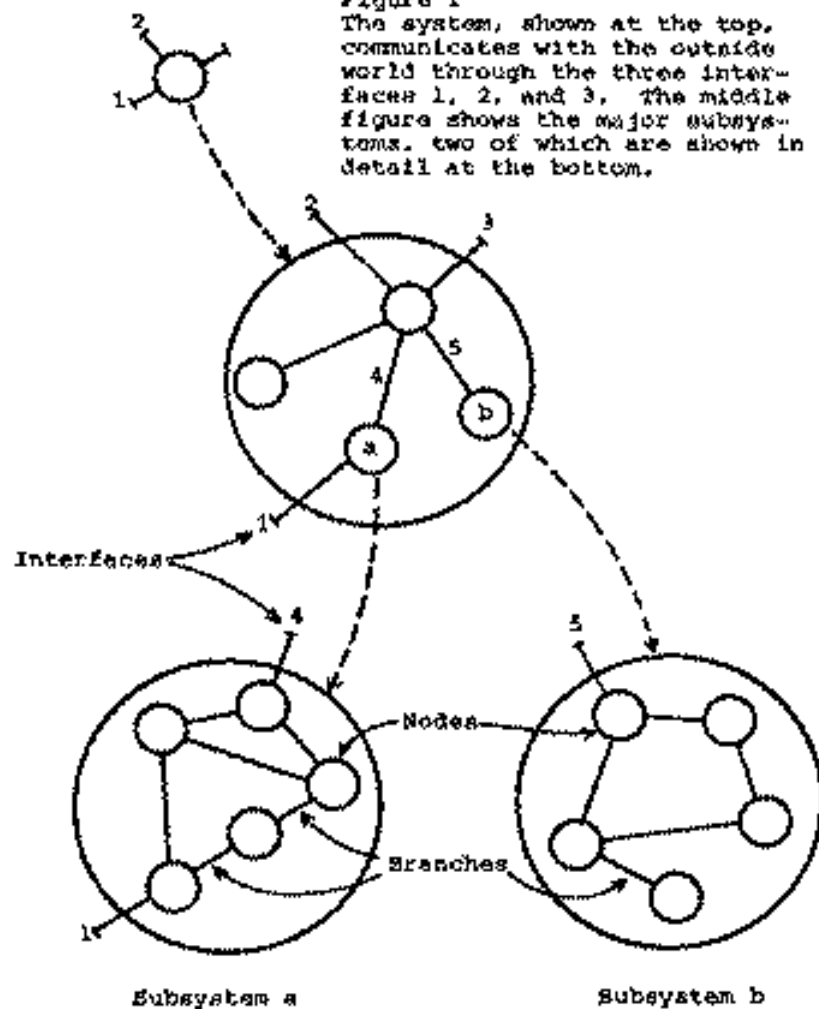
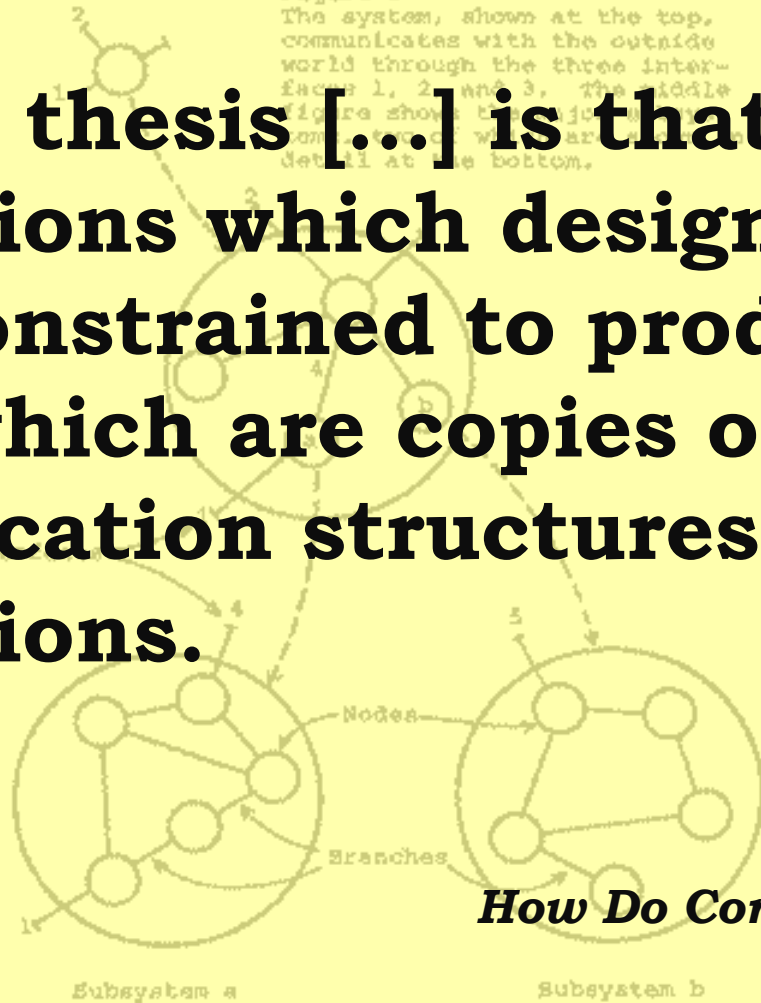


Figure 1  
The system, shown at the top, communicates with the outside world through the three interfaces 1, 2, and 3. The middle figure shows the major subsystems, two of which are shown in detail at the bottom.



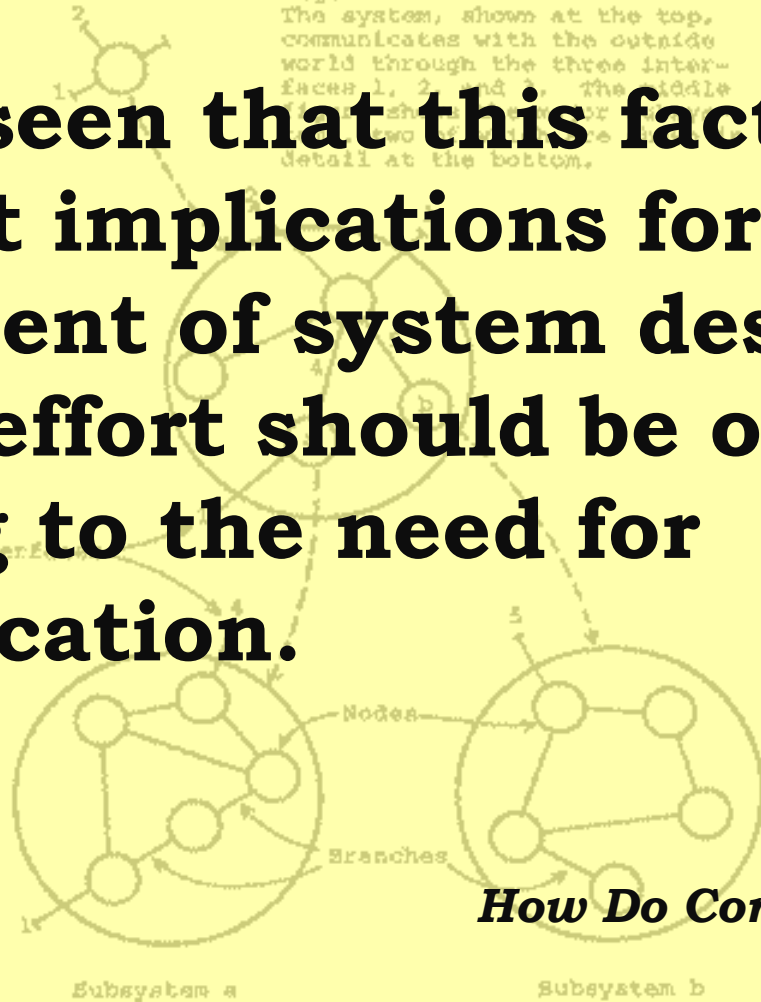
**The basic thesis [...] is that organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations.**



**Melvin Conway**  
***How Do Committees Invent?***



**We have seen that this fact has important implications for the management of system design. [...] A design effort should be organized according to the need for communication.**



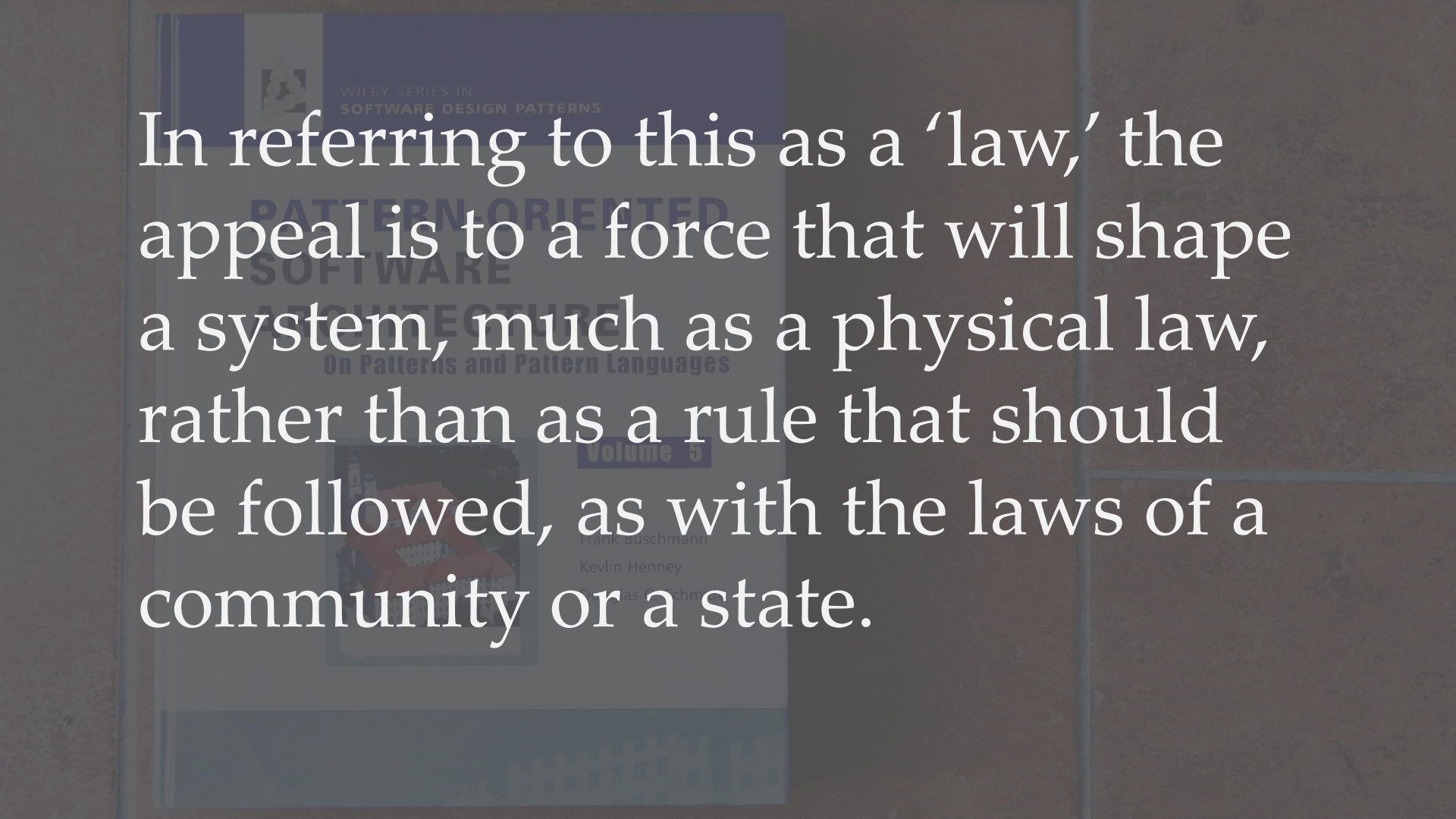
**Melvin Conway**  
***How Do Committees Invent?***

Informal communication is important in developing software. If a barrier to informal communication is created (e.g. outsourcing or off-shoring) it is necessary to compensate for this barrier. Failure to do so will affect the architecture.

Lise Hvatum & Allan Kelly  
*What do I think about Conway's law now?*

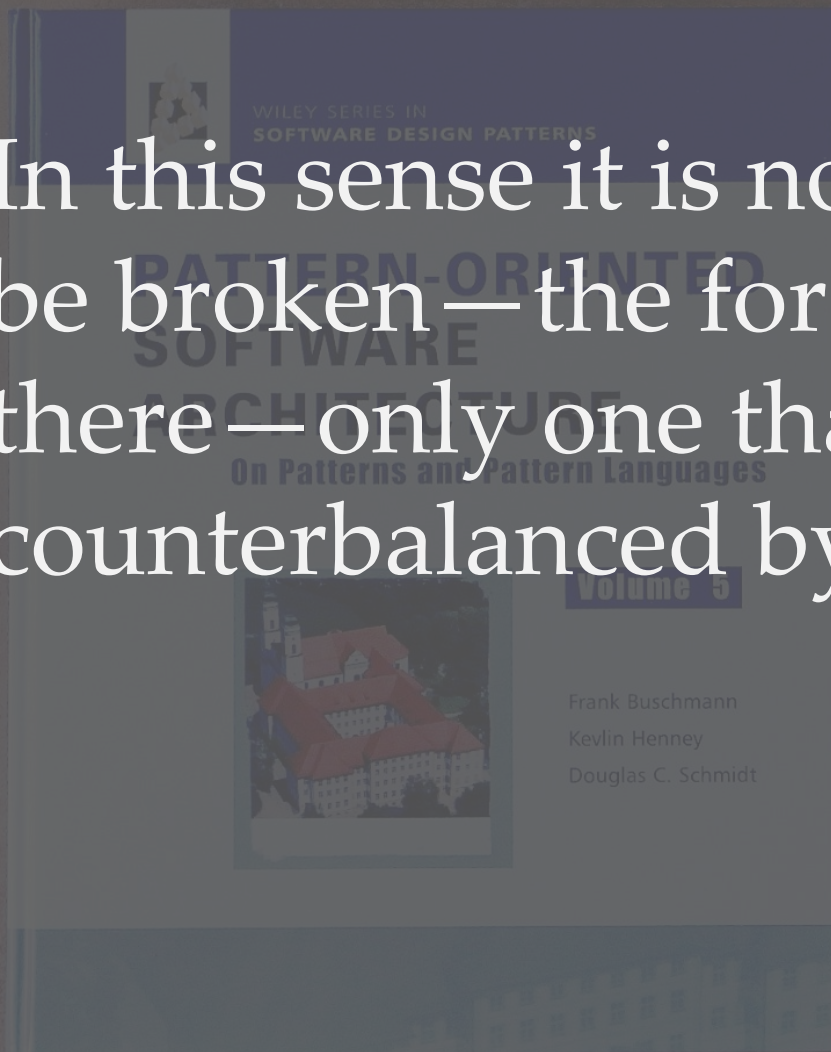
We are not powerless [...].  
Removing barriers requires  
an active intervention.

Lise Hvatum & Allan Kelly  
*What do I think about Conway's law now?*

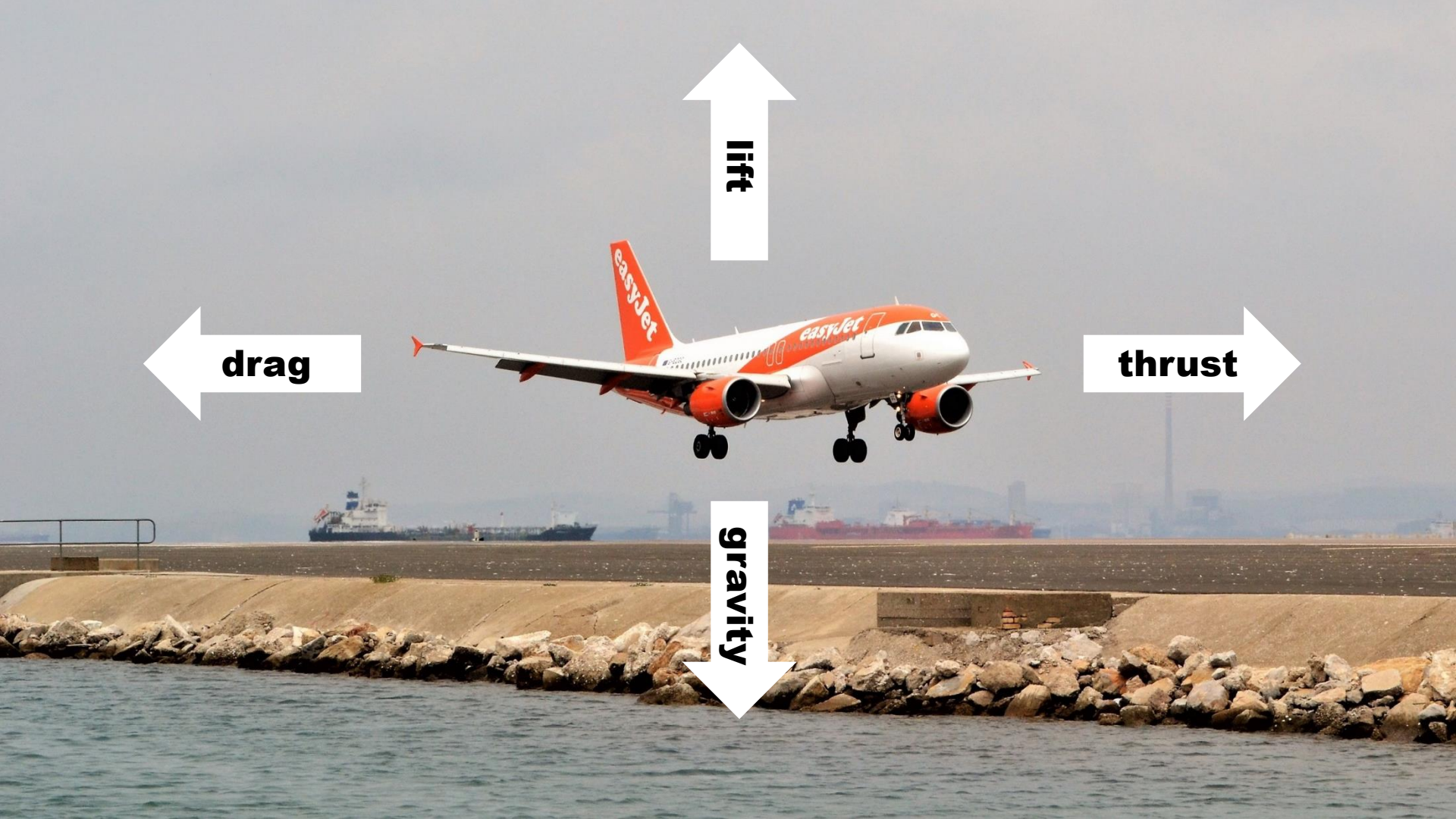


In referring to this as a ‘law,’ the appeal is to a force that will shape a system, much as a physical law, rather than as a rule that should be followed, as with the laws of a community or a state.

In this sense it is not a law that can be broken—the force is always there—only one that can be counterbalanced by other forces.





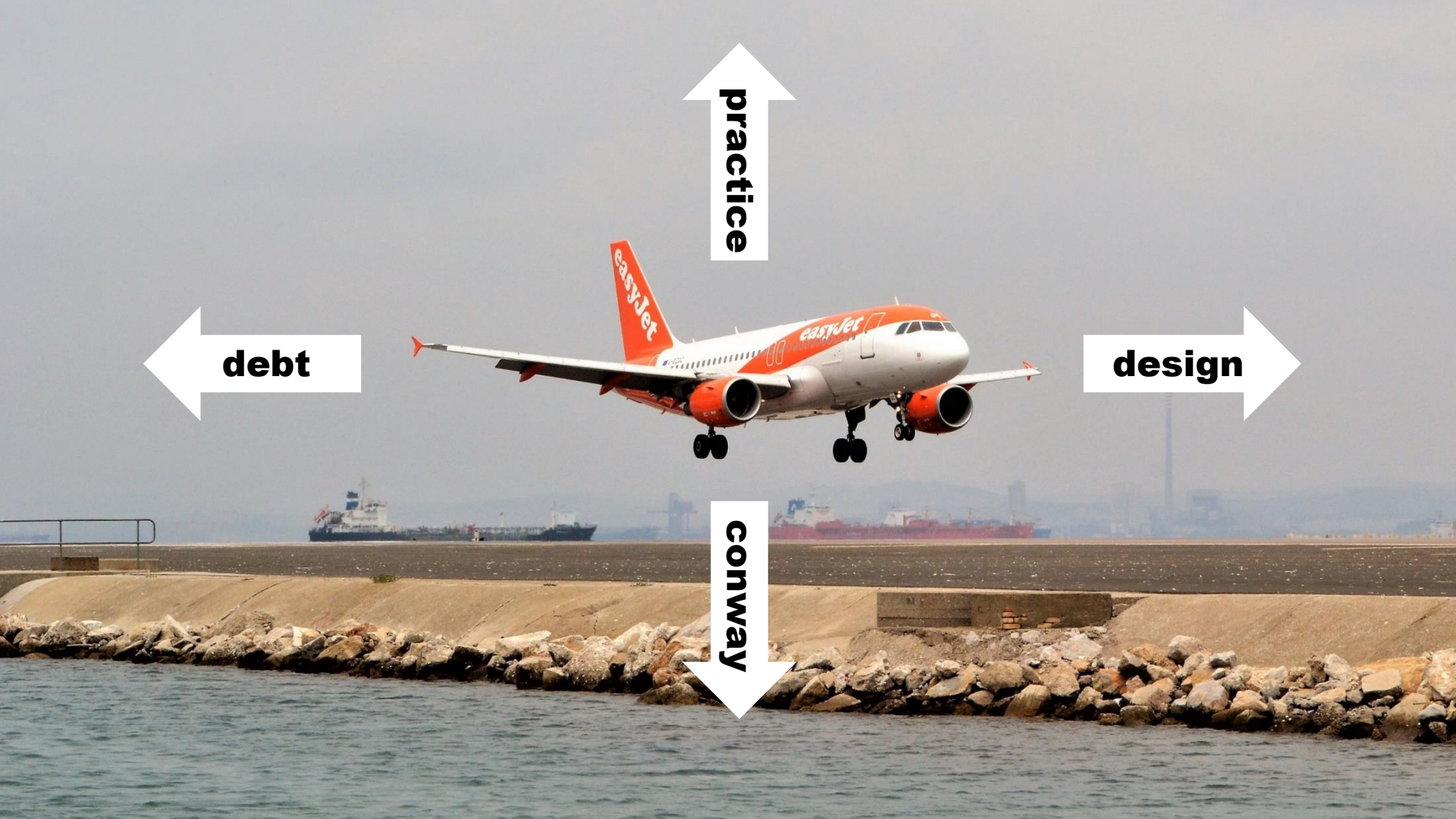


**lift**

**drag**

**thrust**

**gravity**

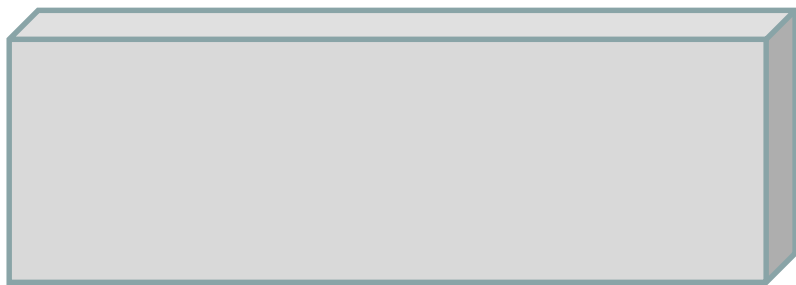


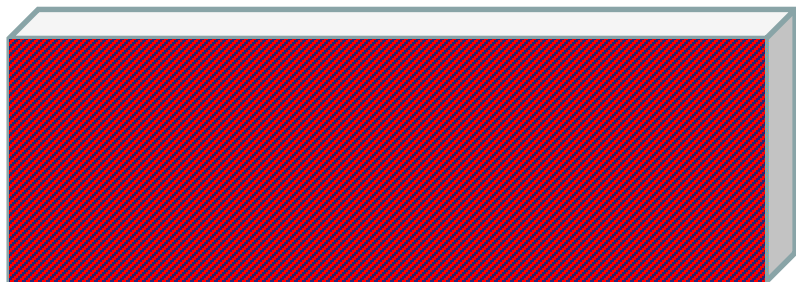
**practice**

**debt**

**design**

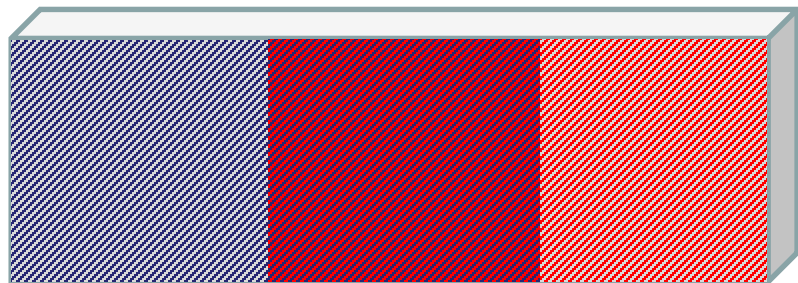
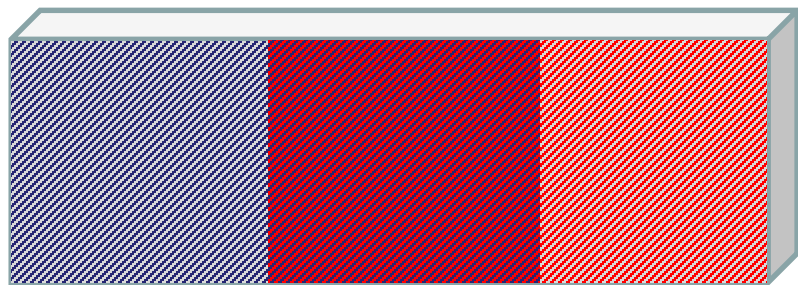
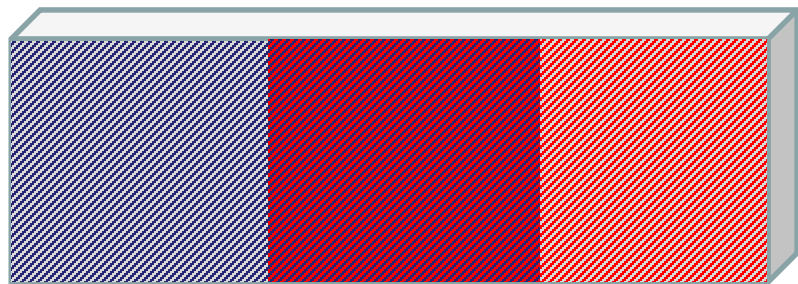
**conway**

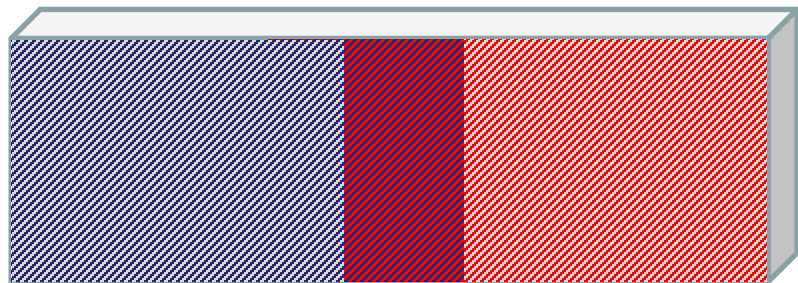
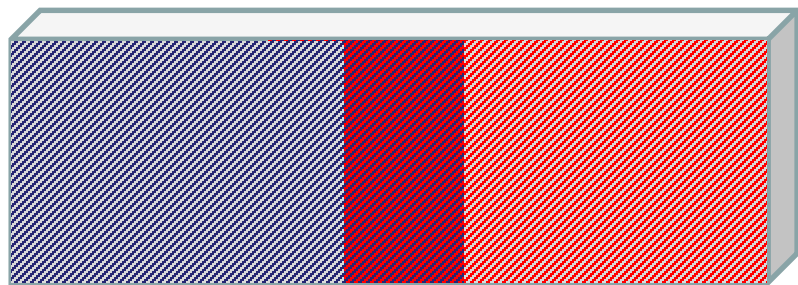
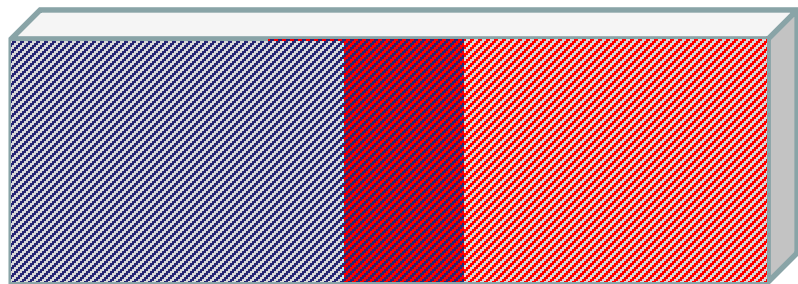


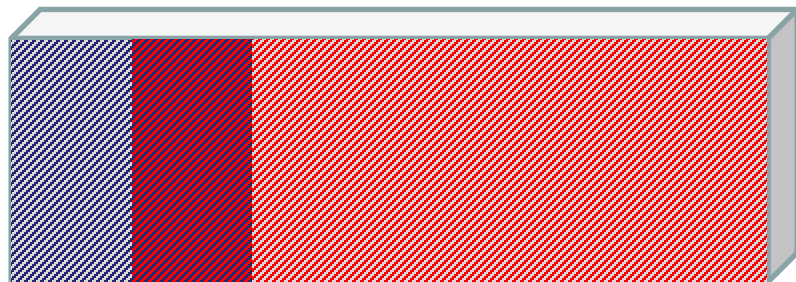
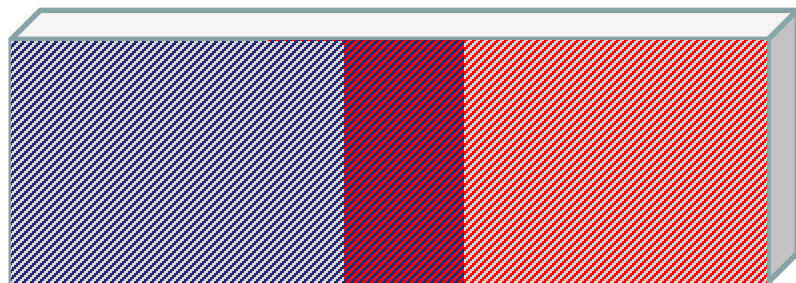




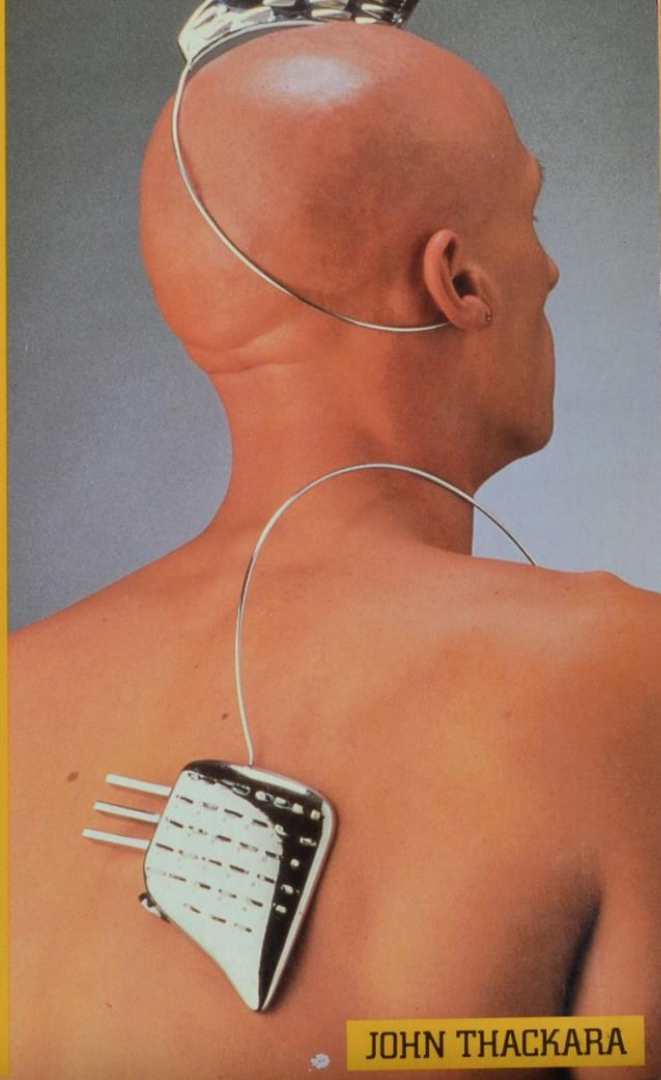








DESIGN AFTER MODERNISM



JOHN THACKARA

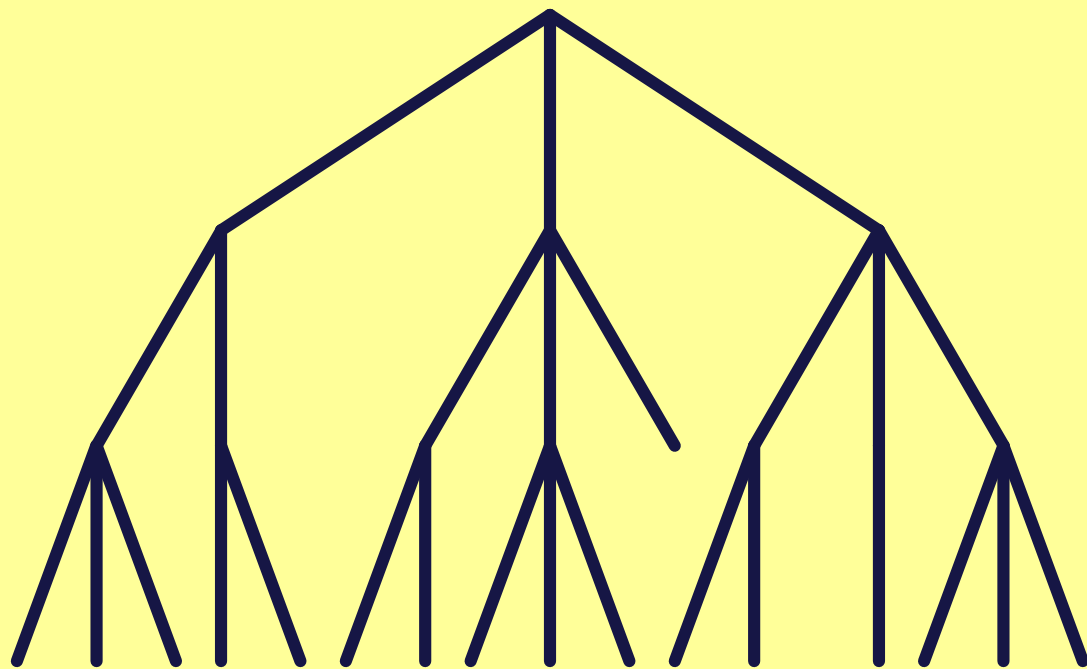
A city is not a tree

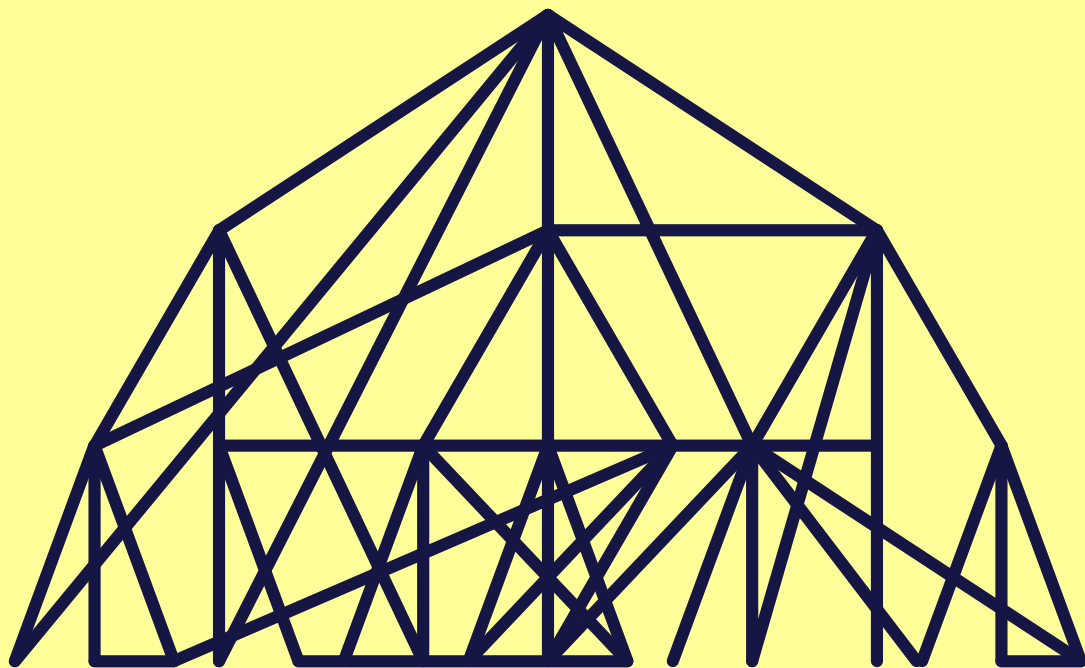
Christopher Alexander

The tree of my title is not a green tree with leaves. It is the name of an abstract structure.

Both the tree and the semilattice are ways of thinking about how a large collection of many small systems goes to make up a large and complex system.

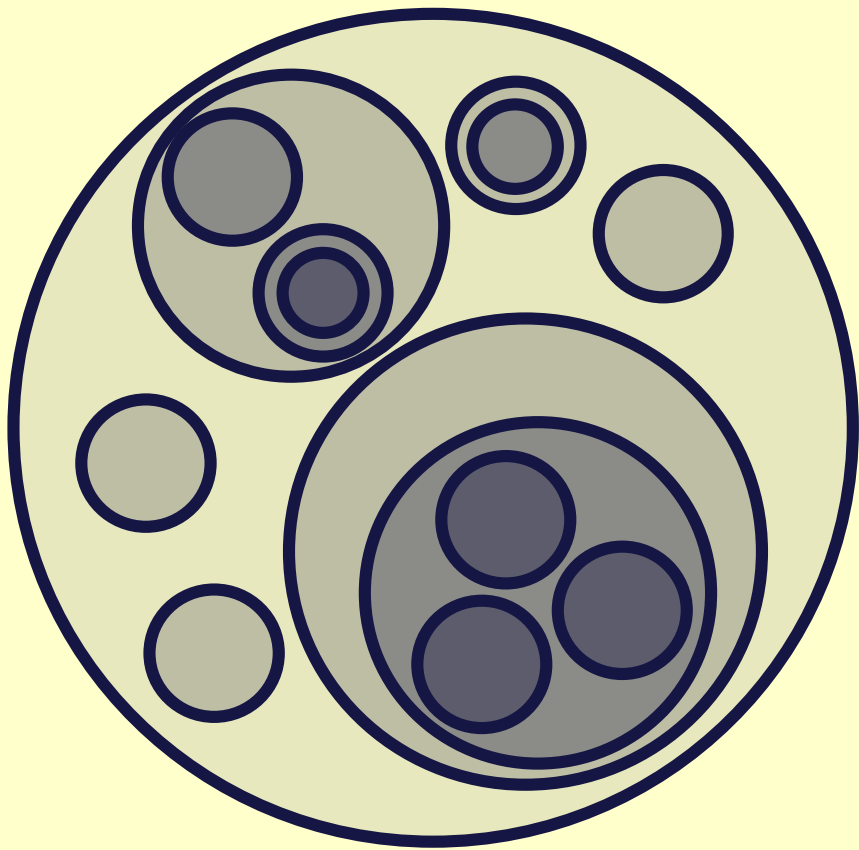


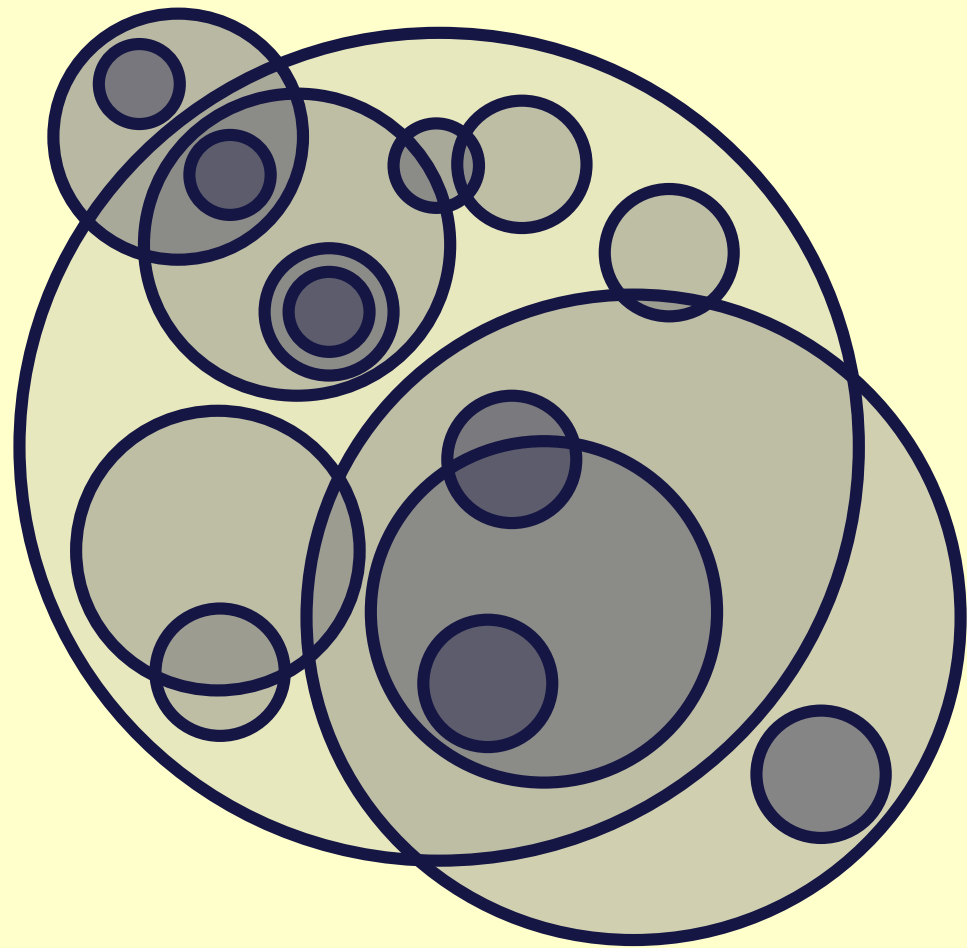




**The semilattice is potentially a much more complex and subtle structure than a tree.**

**The reality of today's social structure is thick with overlap — the systems of friends and acquaintances form a semilattice, not a tree.**

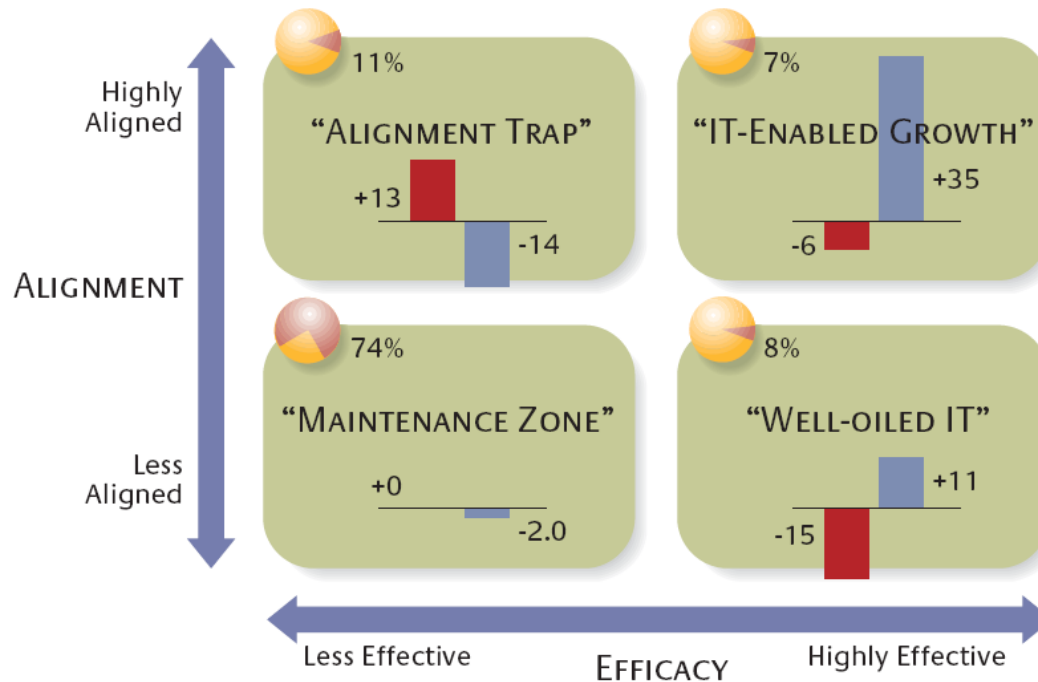






**IT's effort to satisfy its various (and sometimes conflicting) business constituencies created a set of Byzantine, overlapping systems that might satisfy individual units for a while but did not advance the company's business as a whole.**

David Schpilberg, Steve Berez, Rudy Puryear and Sachin Shah  
"Avoiding the Alignment Trap in Information Technology"  
*MIT Sloan Management Review*  
<http://sloanreview.mit.edu/article/avoiding-the-alignment-trap-in-it/>



% of Respondents  
(n=504)

Differences in Percentage  
compared to overall averages



IT  
Spending



3-Year Sales  
Compound  
Annual  
Growth Rate

**Aligning a poorly performing IT organization to the right business objectives still won't get the objectives accomplished.**

**Richard F Connell**

David Schpilberg, Steve Berez, Rudy Puryear and Sachin Shah  
"Avoiding the Alignment Trap in Information Technology"  
*MIT Sloan Management Review*  
<http://sloanreview.mit.edu/article/avoiding-the-alignment-trap-in-it/>

**The biggest advantage of  
autonomously working teams is  
risk reduction through increased  
group intelligence.**

Kevlin Henney

<https://jaxlondon.com/blog/java-core-languages/the-error-of-our-ways-kevin-henney/>

There's little correlation between a group's collective intelligence and the IQs of its individual members. But if a group includes more women, its collective intelligence rises.

*"What Makes a Team Smarter? More Women"*

Anita Woolley & Thomas W Malone

<http://hbr.org/2011/06/defend-your-research-what-makes-a-team-smarter-more-women/>





THE  
WISDOM OF CROWDS

Why the Many Are Smarter Than the Few

JAMES SUROWIECKI

'Dazzling . . . the most brilliant  
book on business, society and  
everyday life that I've read in years'

MALCOLM GLADWELL  
AUTHOR OF THE TIPPING POINT



The four conditions  
that characterize  
wise crowds:  
diversity of opinion,  
independence,  
decentralization,  
and aggregation.

**We're talking about a very  
specific mindset that is crucial  
when it comes to the act of  
creating.**

Will Gompertz  
*Think Like an Artist*

**It is an attitude that can be  
encapsulated in a simple but  
demanding rule: always think  
both big picture and fine detail.**

Will Gompertz  
*Think Like an Artist*