

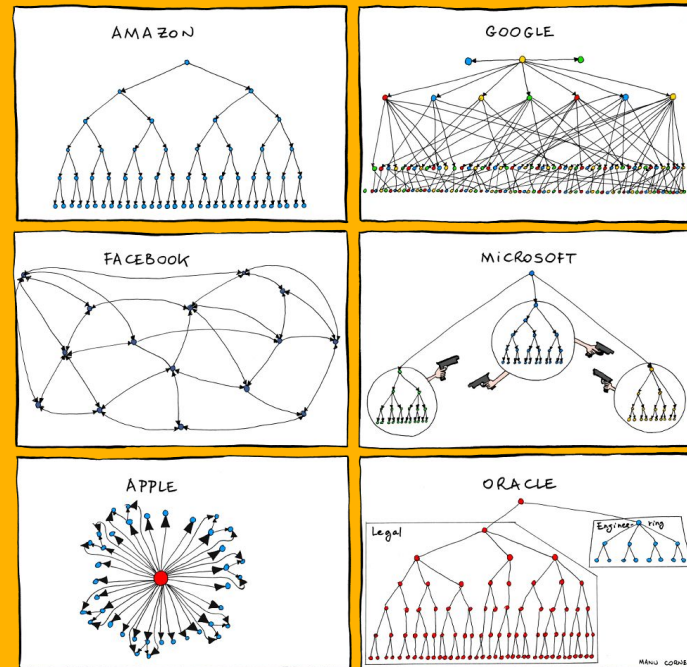
# Shades of Conway's Law

Thoughtful Thierry *(they/them)*



*"Organisations which design systems are constrained to produce designs which are copies of the communication structures of these organisations."*

- Melvin Conway  
How Do Committees Invent?, 1968



Similar observations  
in different industries  
around the same time.

The Mirroring Principle

- Organization Design
  - 1967: Thompson, Organizations in Action
  - 1974: Galbraith, Organization Design: An Information Processing View

“... differences in technical functions, or technologies, cause significant differences among organisations ...”

– Thompson, Organizations in Action, 1967, p12

“... the greater the amount of information that must be processed among decision makers during task execution ...

This is a **problem of organisation design** ...

... the organisation must adopt integrating mechanisms ...”

- Galbraith, Organization Design:  
An Information Processing View, 1974

Similar observations  
in different industries  
around the same time.

The Mirroring Principle

- Organization Design
  - 1967: Thompson, Organizations in Action
  - 1974: Galbraith, Organization Design: An Information Processing View
- Product Design
  - 1964: Alexander, Notes on the Synthesis of Form
  - 1972: Parnas, On the criteria to be used in Decomposing Systems into Modules

“... separate groups would work on each module with little need for communication ...”

– Parnas

On the criteria to be used in Decomposing Systems into Modules, 1972

### Software Industry Top-10 papers

1. **On the criteria to be used in decomposing systems into modules**
2. A note on distributed computing
3. The next 700 Programming Languages
4. Can programming be liberated from the von Neumann style
5. Reflections on trusting trust
6. Lisp, Good news bad news how to win big
7. An experimental evaluation of the assumption of independence in multi-version programming
8. Arguments and results
9. A Laboratory For Teaching object-oriented thinking
10. Programming as an experience, the inspiration for self

## Organisation Design & Product Design

take their inspiration from

1962: Simon,  
The Architecture of  
Complex Systems

“My thesis has been that one path to the construction of a non- trivial theory of complex systems is by way of a theory of hierarchy.

...

In their dynamics, hierarchies have a property, near-decomposability, ... [It] simplifies the description of a complex system, and makes it easier to understand ...”



## The **Mirroring Principle**

a structural correspondence between two networks, one technical and one organisational

# The Mirroring Hypothesis

*“The mirroring hypothesis predicts that organizational ties within a project, firm, or group of firms (e.g., communication, collocation, employment) will correspond to the technical dependencies in the work being performed.”*

- The Mirroring Hypothesis  
Baldwin and Colfer, 2016

*"... we provide empirical evidence that product ambiguity exists, and it is more likely to be present across organizational and system boundaries"*

- The Misalignment of Product Architecture and Organizational Structure in Complex Product Development, Sosa et al, 2004

*"**Congruence** between coordination requirements and coordination activities shortened development time."*

- Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools, Cataldo et al, 2006

*"Our results suggest that misalignment of the design organization with the product architecture negatively affects product quality"*

- The Impact of Misalignment of Organization Structure and Product Architecture on Quality in Complex Product Development, Gokpinar et al, 2007



# First empirical evidence for the software industry

“We find strong evidence to support the hypothesis that a product’s architecture tends to mirror the structure of the organization in which it is developed.”

- Exploring the Duality between Product and Organizational Architectures: A Test of the “Mirroring” Hypothesis  
Baldwin, MacCormack and Rusnak, 2012



Over the decades,  
many different people  
paraphrased Conway's Law in various ways.

=> uncovers new insights

sometimes they seem to contradict  
but all come to the same conclusion

# Systems are **isomorphic** to the Organisation

"The structure of any system designed by an organization is **isomorphic** to the structure of the organization."

– Edward Yourdon and Larry L. Constantine,  
Structured Design, 1979, p363

no single team can be responsible  
for more than one service

isomorphic *adjective*

1. corresponding or similar in form and relations.

source: Oxford Languages



## 2. MATHEMATICS

an isomorphism is a **structure-preserving mapping** between two structures of the same type that can be reversed by an inverse mapping.

source: Wikipedia

## 3. MATHEMATICS

an isomorphism is a **one-to-one** correspondence (mapping) between two sets that preserves binary relationships between elements of the sets.

source: Britannica





# Underlying Conway's Law is the **Homomorphic Force**

*“Speaking as a mathematician might, we would say that there is a **homomorphism** from the linear graph of a system to the linear graph of its design organization.”*

– Melvin Conway

How Do Committees Invent?, 1968

homomorphism *noun*

MATHEMATICS

*a structure-preserving map between two structures.*

*source: Wikipedia*

a single team can be responsible  
for one, two or more services

# Organisation and Systems are **congruent**

*"The organization of the software and the organization of the software team will be **congruent**."*

– Eric Raymond  
The New Hacker's Dictionary (3rd ed.), 1996, p124

congruent *adjective*

1. *in agreement or harmony.*

2. **GEOMETRY**

*identical in form; coinciding exactly when superimposed*

*source: Oxford Languages*

*"If you have 4 groups working on a compiler, you'll get a 4-pass compiler"*

– Eric Raymond



## Thousand Module Effect

*"an informal observation that if 1,000 programmers are assigned to develop a system before a structural design has been completed, then there will be at least 1,000 modules in the resulting system."*

- Edward Yourdon and Larry L. Constantine, Structured Design, 1979, p363

## Mealy's Law (example of the Mythical Man-Month)

*"There is an incremental person who, when added to a project, consumes more energy [...] than [they] make available. Thus, beyond a certain point, adding [people] slows progress in addition to increasing the cost."*

- Edward Yourdon and Larry L. Constantine, Structured Design, 1979, p363

# The organisation must be **compatible** with the system

*"If the parts of an organization (e.g., teams, departments, or subdivisions) do not closely reflect the essential parts of the product, or if the relationships between organizations do not reflect the relationships between product parts, then the project will be in trouble ...*

*Therefore: **Make sure the organization is compatible with the product architecture.**"*

– James Coplien & Neil Harrison  
Organisational patterns of agile software development, 2004, p246

compatible *adjective*

*(of two things) able to exist or occur together without problems or conflict.*

*source: Oxford Languages*





*"If the architecture of the system and the architecture of the organization are at odds, **the architecture of the organization wins.**"*

– Ruth Malan, Conway's Law, Feb 13, 2008

*"the very act of organizing a design team means that certain design decisions have already been made"*

– Melvin Conway

How Do Committees Invent?, 1968



Brings us to **Reverse Conway's Law**

*Organisations with long-lived systems will adopt a structure modelled on the system.*

– Allan Kelly, Continuous Delivery and Conway's Law

*We reorganised,  
but the system didn't get the memo* 🧑🏻

– a CTO,  
from Conway's Law Doesn't Apply to Rigid Designs,  
Mathias Verraes



*"... organizations should evolve their team and organizational structure to achieve the desired architecture."*

– Nicole Forsgren, PhD and friends  
Accelerate, 2018

*"Dysfunctional organizations tend to create dysfunctional applications. [...] In what could be termed an “**inverse Conway maneuver**”, you may want to begin by breaking down silos that constrain the team’s ability to collaborate effectively"*

– Jonny LeRoy and Matt Simons  
Dealing with creaky legacy platforms, 2010



# Organisational design is system design!

– Allan Kelly's Corollary

*"the very act of organizing a design team means that certain design decisions have already been made, explicitly or otherwise."*

– Melvin Conway  
How Do Committees Invent?, 1968

*"Team assignments are the first draft of the architecture."*

– Michael Nygard, Release It!, 2007

- if we want separate modules  
we need separate teams
- collective code ownership  
leads to more integrated teams  
and code



Drafting a system  
architecture is already  
designing an organisation.

*"Conway's Law also kicks in if we take an initial guess at the system decomposition, allocate subsystems to teams, and sally forth—the team boundaries will tend to become boundaries within the system."*

– Ruth Malan, Conway's Law, Feb 13, 2008



# Organizational flexibility is important to effective design

- Conway's Corollary  
as defined by Jeff Sussna

*“The initial design of a system is never the best. The system may need to change. Therefore it requires flexibility of the organisation to design effectively.”*

- Melvin Conway  
How Do Committees Invent?, 1968



# Incremental software development impacts the organisation

*"They [system and organization] will co-evolve, because if they don't, Conway's Law warns us that the **organization form will trump intended designs** that go "cross-grain" to the organization warp."*

– Ruth Malan, Conway's Law, Feb 13, 2008



Is there a better design  
that is not available to us  
because of our organization?

Can we change the organization  
if a better design is found?



*"The importance of the principle ... is ... that **your design organization is keeping you from designing some things that perhaps you should be building.**"*

- Melvin Conway,  
Toward Simplifying Application Development in a Dozen Lessons, 2017





System architecture is a source for archaeological research on past enterprise decisions.

*"You can read the history of an enterprise's political struggles in its system architecture."*

– Michael Nygard (@mtnygard) on Twitter  
May 9, 2013



*"the state of a system reflects ... the organisational history and the flow of people through those organisations over the long term."*

— Rob Smallshire  
Good with Computers, 2014

## Software Engineer Half-Life

*"... the turnover of developers can be modelled as if developers have a half-life within organizations [or a codebase]."*

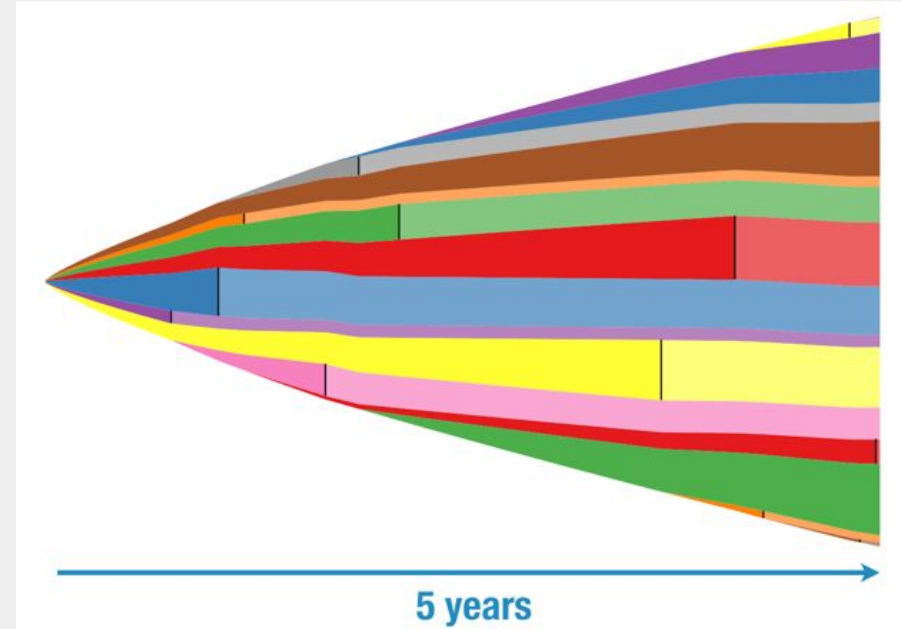
— Rob Smallshire  
Good with Computers, 2014

Industry average of 3.2 years

Simulate turn-over in a team of 7 engineers working on a code base during 5 years.

Churned through 19 engineers.

37% of code written by people still present at the end of 5 years.





*"A large organization similarly, stretches across time and space. Conway's Law ties the two. How we organize defines how we think collectively, and thus what we make collectively."*

- Pieter Hintjes  
Sex in Title, and other Stories, 2013



*"Your org structure isn't solving your problem.  
It's an artifact of how you've solved it before."*

– Adam Jacob  
(we assume the one from Chef)

*"you always ship your organization, so  
design your organization well"*

– Michael Feathers at CraftConf 2014

Managers are architects.

*"Another implication of Conway's Law is that if we have managers deciding on teams, and deciding which services will be built, by which teams, we implicitly **have managers deciding on the system architecture.**"*

– Ruth Malan, Conway's Law, 2008



Architects are managers.

*"When I think of the really good technical people I know ... **to solve technical problems requires them to work outside of the technical domain**"*

– Allan Kelly, Return to Conway's Law, 2006





## Allan Kelly's Advice

*"Grow the team with the system. Small teams, small systems, piecemeal growth.*

***Start as small as you can and grow as you need too.***  
*Don't start thinking big."*



# Start Small!

*Probably slightly smaller than we think we need* 🙌

# Hello, I am Thierry de Pauw

*fancies dark chocolate, black coffee & peated whisky*

## Acknowledgments:

Els, the one I love.

Ruth Malan ([@RuthMalan@mastodon.social](mailto:@RuthMalan@mastodon.social)) for the many conversations and insights.

Laphroaig for the creative support.

## The Article:

<https://thinkinglabs.io/articles/2021/05/07/shades-of-conways-law.html>





# Bibliography

[The Architecture of Complexity](#), Simon, 1962

[Notes on the Synthesis of Form](#), Alexander, 1964

[Organizations in Action: Social Science Bases of Administrative Theory](#), Thompson, 1967

[How Do Committees Invent?](#), Melvin Conway, 1968

[On the Criteria To Be Used in Decomposing Systems into Modules](#), Parnas, 1972

[Organization Design: An Information Processing View](#), Galbraith, 1974

[Structured Design](#), Edward Yourdon and Larry L. Constantine, 1979

[The New Hacker's Dictionary \(3rd ed.\)](#), Eric Raymond, 1996

[Organisational patterns of agile software development](#), James Coplien & Neil Harrison, 2004

[The Misalignment of Product Architecture and Organizational Structure in Complex Product Development](#), Sosa et al, 2004



# Bibliography

[Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools](#), Cataldo 2006

[The Impact of Misalignment of Organization Structure and Product Architecture on Quality in Complex Product Development](#), Gokpinar et al, 2007

[Return to Conway's Law](#), Allan Kelly, 2006

[Release It!](#), Michael Nygard, 2007

[Conway's Law](#), Ruth Malan, 2008

[Dealing with creaky legacy platforms](#), Jonny LeRoy and Matt Simons, 2010

[Exploring the Duality between Product and Organizational Architecture: A Test of the "Mirroring" Hypothesis](#), Baldwin, MacCormack, Rusnak, 2012

[Architecture without an end state](#), Michael Nygard, 2012

[Sex in Title, and Other Stories](#), Pieter Hintjes, 2013

# Bibliography

[Continuous Delivery and Conway's Law](#), Allan Kelly, 2014

[Good with Computers](#), Rob Smallshire, 2014

[Conway's Law: The DevOps Skeleton](#), Dan Slimmon, 2014

[Toward Simplifying Application Development in a Dozen Lessons](#), Mel Conway, 2016

[The Mirroring Hypothesis](#), Colfer and Baldwin, 2016

[Accelerate](#), Nicole Forsgren, PhD and friends, 2018

[Accidental Architects: How HR Designs Software Systems](#), Matthew Skelton

[Conway's Law Doesn't Apply to Rigid Designs](#), Mathias Verraes, 2022

[Mastodon conversation with Ruth Malan about Conway's Law](#)

[Isomorphism vs Homomorphism](#), Michael McCliment