

# Architectural Skills

*CSSE6400*

Richard Thomas

May 18, 2026

*Quote*

Architecture is the stuff you can't Google.

– Mark Richards *[Richards and Ford, 2020]*

*Quote*

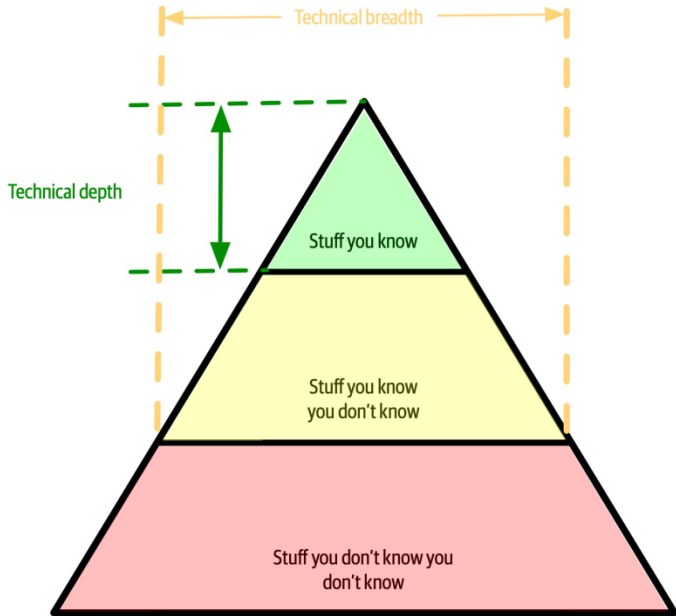
There are no right or wrong answers in architecture—only trade-offs.

– Neal Ford *[Richards and Ford, 2020]*

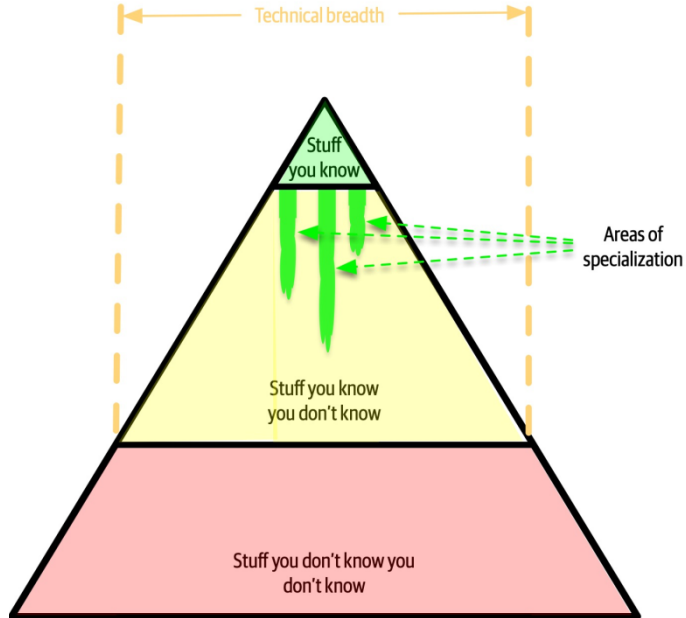
## *Architectural Design*

Architects use knowledge and experience to analyse trade-offs to design architectures appropriate to the system context.

# Developers – Technical Depth *[Richards and Ford, 2020]*



# Architects – Technical Breadth *[Richards and Ford, 2020]*



### *Definition 0.* Conway's Law

Organisations design systems whose structure is inevitably a copy of the organisation's communication structure *[Conway, 1968] [MacCormack et al., 2012]*.

## *Conway's Law Consequences*

- Business Process Management
- Microservices to reflect organisation structure
- Teams formed around services



## *Conway's Law Consequences*

Team insularity – more loyal to team than organisation

## Conway's Law Issues

- Cross-cutting concerns
  - e.g. Security
- Organisation structure should align with market structure
- Physical location of teams

*Definition 0.* Peopleware<sup>[Neumann, 1977]</sup>

People involved in development of systems.

– Peter G. Neumann

## Stakeholders *[Coplien and Bjørnvig, 2010]*

- End Users
- Organisation
- Customers / Sponsors
- Domain Experts
- Developers

# Communication

- Written
  - For those who are not there
- Oral
  - Immediate & interactive

## Circumventing Conway's Law *[Woods, 2017]*

- Cloud Platforms
- Microservices
- APIs
- *Culture*

## *Evidenced-Based Software Engineering*

Don't follow fads, seek evidence for good practice.

*Let's hear from an expert*

## Software Engineering's Greatest Hits

what we actually know about software development  
and why we believe it's true



**Greg Wilson**

<http://third-bit.com/talks/greatest-hits/>



1 / 47



## References

- [Conway, 1968] Conway, M. E. (1968).  
How do committees invent?  
*Datamation*.
- [Coplien and Bjørnvig, 2010] Coplien, J. O. and Bjørnvig, G. (2010).  
*Learn Architecture for Agile Software Development*.  
Wiley.  
[https://www.amazon.com/  
Lean-Architecture-Agile-Software-Development/dp/0470684208](https://www.amazon.com/Lean-Architecture-Agile-Software-Development/dp/0470684208).
- [MacCormack et al., 2012] MacCormack, A., Baldwin, C., and Rusnak, J. (2012).  
Exploring the duality between product and organizational architectures: A test  
of the “mirroring” hypothesis.  
*Research Policy*, 41(8):1309–1324.

[Neumann, 1977] Neumann, P. G. (1977).

Peopleware in systems.

*Peopleware in Systems*, pages 15–18.

[Richards and Ford, 2020] Richards, M. and Ford, N. (2020).

*Fundamentals of Software Architecture: An Engineering Approach*.

O'Reilly Media, Inc.

[Woods, 2017] Woods, D. (2017).

How platforms are neutralizing conway's law.

<https://www.forbes.com/sites/danwoods/2017/08/15/how-platforms-are-neutralizing-conways-law/>.