

# Architectural Skills

*CSSE6400*

Richard Thomas

May 20, 2024

*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]*



- Architects need greater technical breadth than depth.
- Breadth allows better consideration of trade-offs.
- Avoid trying to become an expert across many areas – you'll fail.
- Don't stop learning – increase your breadth – don't let your knowledge become stale.

### *Definition 1. Conway's Law*

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

- First citation is original article.
- Second citation is one of several about MIT and Harvard research into the phenomenon, calling it the “mirroring hypothesis”.
  - Compared open source to commercial packages (e.g. Linux to Solaris).
- Elaborate on this point and Coplien's research into organisational sociology.
- Reasons
  - Governance structures constrain communication paths.
  - This constrains space in which to search for solutions, constraining problem solving approaches.

### *Conway's Law Consequences*

- Business Process Management
  - Microservices to reflect organisation structure
  - Teams formed around services
- BPM: Redesign organisation structure to reflect system you want.
  - Microservices: Design system to reflect your organisation.
  - Elaborate on benefits of both approaches.
  - Comment on benefits of small focused teams.



### *Conway's Law Consequences*

Team insularity – more loyal to team than organisation.

- Amazon example from week 11, negotiation difficulties with other teams.
- Need to ensure inter-team cooperation.
- Possibly move people between teams.
- Microservices can encourage insularity.
- Many teams can also encourage insularity.
- Intra-team communication becomes more difficult with large teams.

## Conway's Law Issues

- Cross-cutting concerns
  - e.g. Security
- Organisation structure should align with market structure
- Physical location of teams
  - Cross-cutting concerns span services, & consequently teams.
  - Can't have a "security" service. Needs to be part of every service.
  - Teams solely based around Conway's law and services may not deliver some cross-cutting concerns.
  - Cooperation, documentation and audits may be necessary.
  - Market structure may complement team structure to place teams closer to their end users.
  - Global development and outsourcing mean different teams are likely to be in different locations.
  - Requires additional overhead and documentation for cooperation between teams.

*Definition 2.* Peopleware

People involved in development of systems.

– Peter G. Neumann (1977)

## Stakeholders *[Coplien and Bjørnvig, 2011]*

- End users
  - Organisation
  - Customers / Sponsors
  - Domain Experts
  - Developers
- Organisation for whom system is being developed.
  - Customers / Sponsors are those who are responsible for the end user's work process (e.g. managers).
  - Domain Experts understand details of problem and/or solution domain.

## Communication

- Written
    - For those who are not there
  - Oral
    - Immediate & interactive
- Both apply to all types of stakeholders.
  - Time Dimension: Written is available in the future.
  - Extent of either depends on SE process.
  - Extent also depends on engagement of each type of stakeholder.
  - Technical Documentation guest lecture in week 6 only touched on written for developers.

## Circumventing Conway's Law <sup>[Woods, 2017]</sup>

- Cloud Platforms
  - Microservices
  - APIs
  - *Culture*
- Culture is most important
  - Easily deployed & accessible services with good APIs opens possibility of communication outside of organisation silos.

*Evidenced-Based Software Engineering*

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

Elaborate on finding reliable sources of information and confirming facts yourself.

*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, 2011] Coplien, J. O. and Bjørnvig, G. (2011).  
*Learn Architecture for Agile Software Development*.  
Wiley.  
<http://www.leansoftwarearchitecture.com/>.
- [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.
- [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/](https://www.forbes.com/sites/danwoods/2017/08/15/how-platforms-are-neutralizing-conways-law/).