



Conway's Law? I'm Tired of It!

INNOQ



Eberhard Wolff
Fellow @ewolff
@ewolff@mastodon.social

Why this Talk?

- Conway's law is important.
- But: far too many simplifications & misunderstandings.

Project manager defines
the architecture.

Good idea?

Conway's Law
(Sorry 😞)



Mel Conway

@conways_law

Access to tools. Radical simplification.
Systems thinking applied to human
activity.

📍 Beverly, Massachusetts, USA

🔗 melconway.com

📅 Joined September 2012

Tweet to

Message

HOW DO COMMITTEES INVENT?

by MELVIN E. CONWAY

That kind of intellectual activity which creates a useful whole from its diverse parts may be called the *design* of a *system*. Whether the particular activity is the creation of specifications for a major weapon system, the formation of a recommendation to meet a social challenge, or the programming of a computer, the general activity is largely the same.

Typically, the objective of a design organization is the creation and assembly of a document containing a coherently structured body of information. We may name this information the *system design*. It is typically produced for a sponsor who usually desires to carry out some activity guided by the system design. For example, a public official may wish to propose legislation to avert a recurrence of a recent disaster, so he appoints a team to explain the catastrophe. Or a manufacturer needs a new product and designates a product planning activity to specify what should be introduced.

The design organization may or may not be involved in the construction of the system it designs. Frequently, in public affairs, there are policies which discourage a group's acting upon its own recommendations, whereas, in private industry, quite the opposite situation often prevails.

It seems reasonable to suppose that the knowledge that one will have to carry out one's own recommendations or that this task will fall to others, probably affects some design choices which the individual designer is called upon to make. Most design activity requires continually making choices. Many of these choices may be more than design decisions; they may also be personal decisions the designer makes about his own future. As we shall see later, the incentives which exist in a conventional management environment can motivate choices which subvert the intent of the sponsor.¹

stages of design

The initial stages of a design effort are concerned more with structuring of the design activity than with the system itself.² The full-blown design activity cannot proceed until certain preliminary milestones are passed. These include:

1. Understanding of the boundaries, both on the design activity and on the system to be designed, placed by the sponsor and by the world's realities.
2. Achievement of a preliminary notion of the system's organization so that design task groups can be meaningfully assigned.

We shall see in detail later that the very act of organiz-

¹ A related, but much more comprehensive discussion of the behavior of system-designing organizations is found in John Kenneth Galbraith's, *The New Industrial State* (Boston, Houghton Mifflin, 1967). See especially Chapter VI, "The Technostructure."

² For a discussion of the problems which may arise when the design activity takes the form of a project in a functional environment, see C. J. Middleton, "How to Set Up a Project Organization," *Harvard Business Review*, March-April, 1967, p. 73.

design organization criteria

ing a design team means that certain design decisions have already been made, explicitly or otherwise. Given any design team organization, there is a class of design alternatives which cannot be effectively pursued by such an organization because the necessary communication paths do not exist. Therefore, there is no such thing as a design group which is both organized and unbiased.

Once the organization of the design team is chosen, it is possible to delegate activities to the subgroups of the organization. Every time a delegation is made and somebody's scope of inquiry is narrowed, the class of design alternatives which can be effectively pursued is also narrowed.

Once scopes of activity are defined, a coordination problem is created. Coordination among task groups, although it appears to lower the productivity of the individual in the small group, provides the only possibility that the separate task groups will be able to consolidate their efforts into a unified system design.

Thus the life cycle of a system design effort proceeds through the following general stages:

1. Drawing of boundaries according to the ground rules.
2. Choice of a preliminary system concept.
3. Organization of the design activity and delegation of tasks according to that concept.
4. Coordination among delegated tasks.
5. Consolidation of subdesigns into a single design.

It is possible that a given design activity will not proceed straight through this list. It might conceivably reorganize upon discovery of a new, and obviously superior, design concept; but such an appearance of uncertainty is unflattering, and the very act of voluntarily abandoning a creation is painful and expensive. Of course, from the



Dr. Conway is manager, peripheral systems research, at Sperry Rand's Univac Div., where he is working on recognition of continuous speech. He has previously been a research associate at Case Western Reserve Univ., and a software consultant. He has an MS in physics from CalTech and a PhD in math from Case.

Conway's Law

Architecture

copies

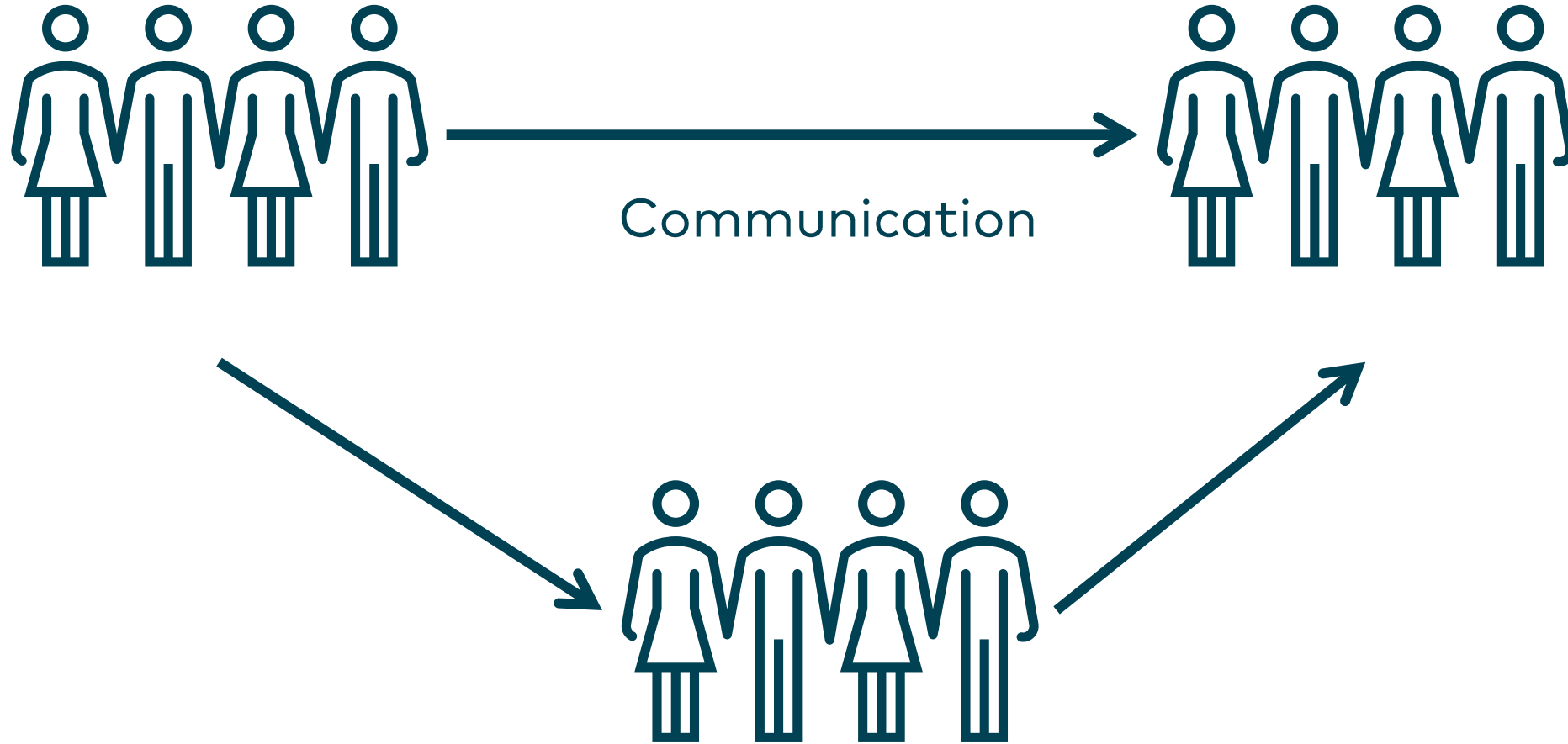
communication structures

of the organization

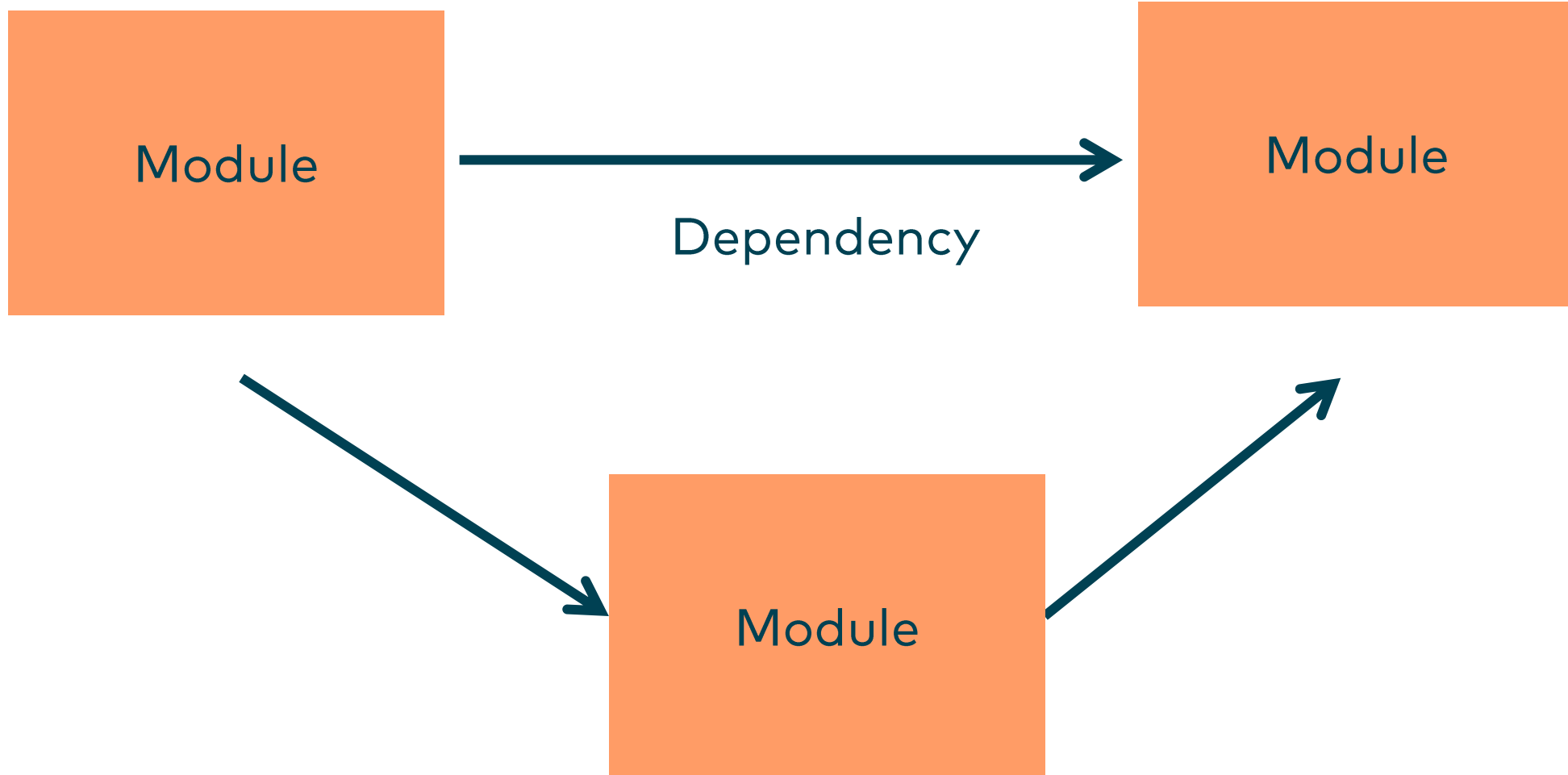
Conway's Law: Misunderstanding

- Conway's law talks about the organization that builds the system.
- Not any other organization.
- Not users, customers ...

Conway's Law: Organization



Conway's Law: Organization



Conway's Law: Short

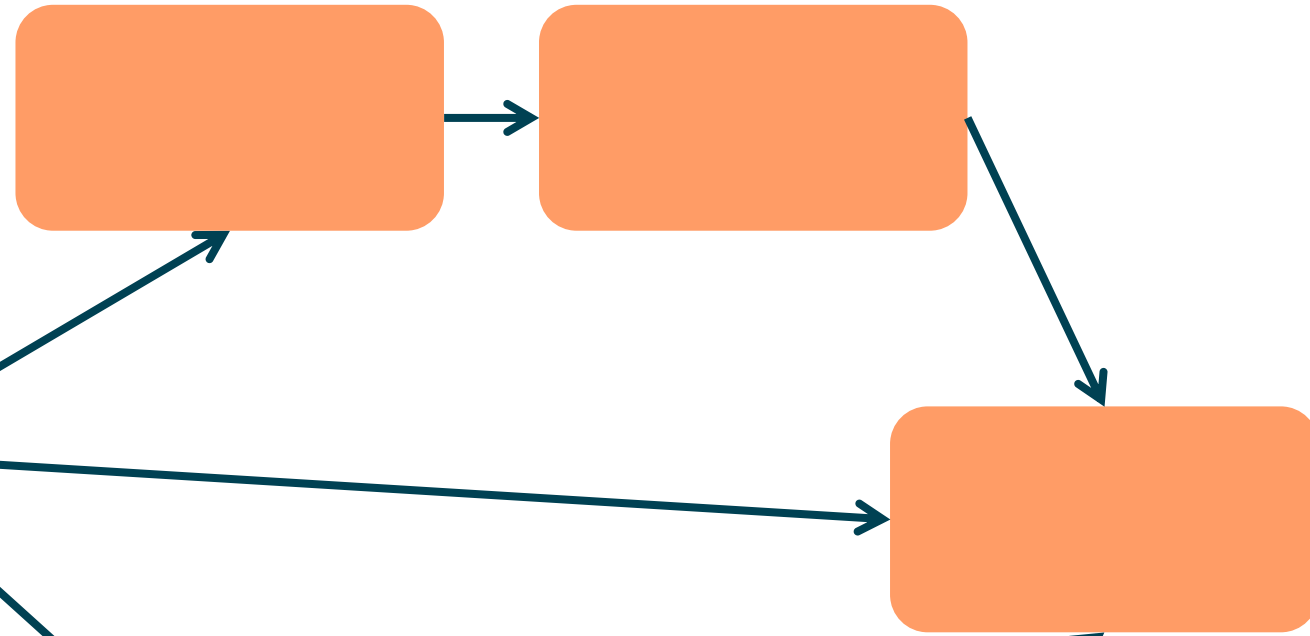
Communication -> Architecture

Modules

What are modules
and why would we care?

Modularization: Limit Knowledge

Understand
modules'
collaboration



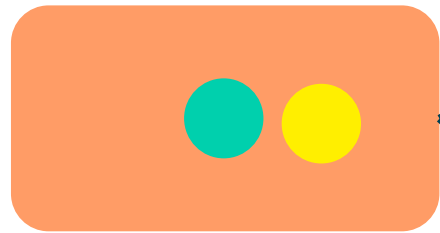
Understand single
module in detail



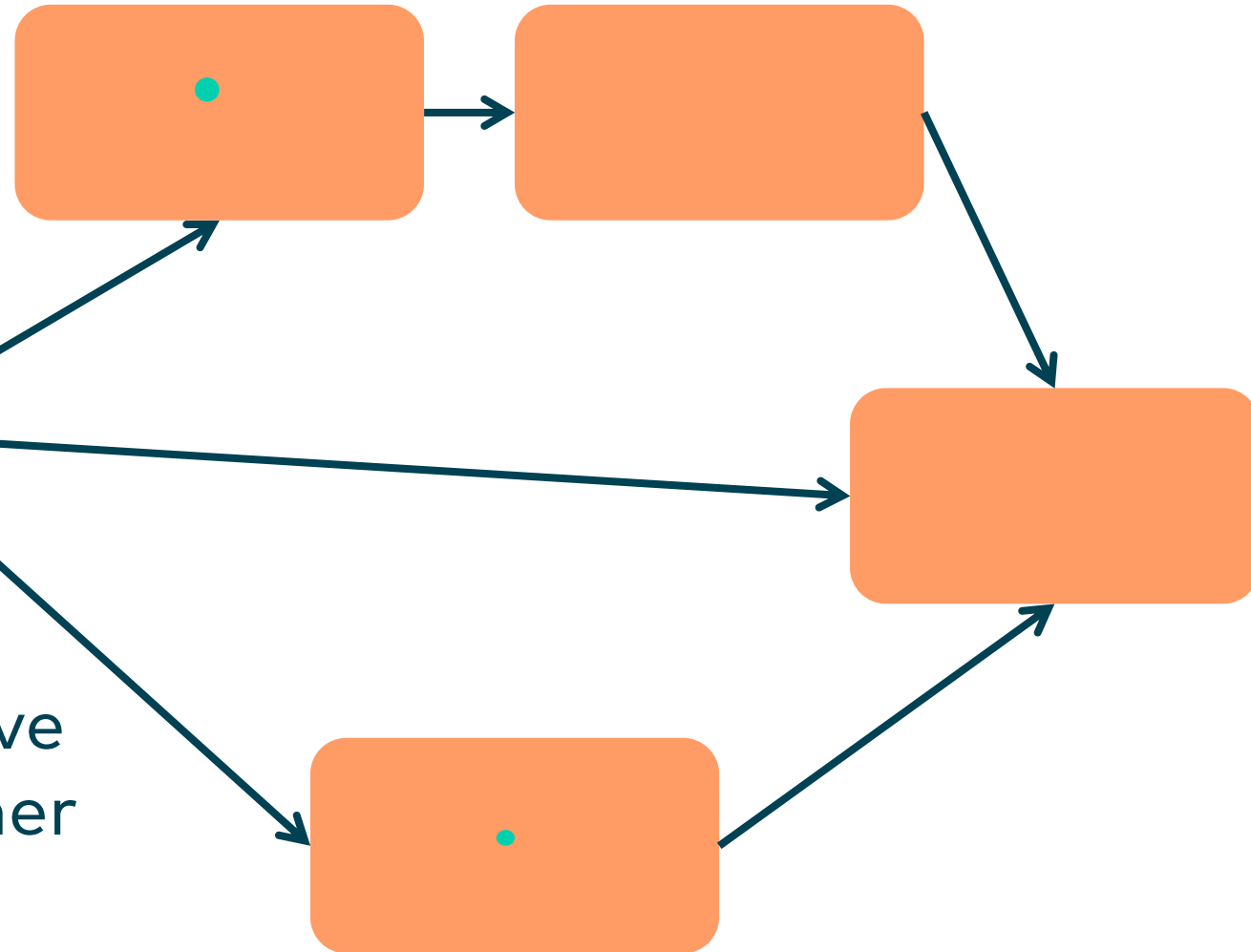
Modularization: Loose Coupling

Modules have dependencies

They are parts of a whole



Changes to a module should have little effect on other modules.



Modularization: Why?

- Limit knowledge on modules
- Make software easier to understand
- ...and therefore changeable

- I.e. for humans

Conway's Law & Modularization



Knows only interface
Less communication

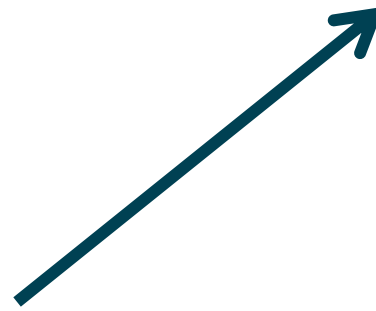
Communication &
collaboration needed to
succeed!



Knows implementations



Knows only interface
Less communication



INFORMATION DISTRIBUTION ASPECTS OF DESIGN METHODOLOGY *

D.L. PARNAS

Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA

<http://cseweb.ucsd.edu/~wgg/CSE218/Parnas-IFIP71-information-distribution.PDF>

<https://blog.acolyer.org/2016/10/17/information-distribution-aspects-of-design-methodology/>

Project manager decides:
Three teams should work on one
module.

Good idea?

Bad Idea™

- Modules are meant to limit knowledge
- Multiple teams work on a module
 - > they all need to understand the module
 - > changes will influence other teams

Any problems / loopholes?

Conway's Law

Architecture

copies

communication structures

of the organization

Project Manager Decides: 3 Teams



3 Teams – 1 Module?



Module



Conway: 3 Teams – 3 Modules!



Module

Module



Module

Or: 1 Team – 1 Module

Module



Project manager decides:
Three teams should work on one
module.

It doesn't work that way.

Project manager decides:
Three teams should work on one
module.

What shall we do?

Collaboration Anyone?

- Project managers might not understand Conway's Law.
- Probably they have reasons to act like they act.
- Usually, teams / projects win or loose as one.



Drink Coffee / Tea

- Talk to project managers, understand reasoning, and provide feedback!
- Talk to other people!
- Figure out what really happens with the architecture and the three teams.

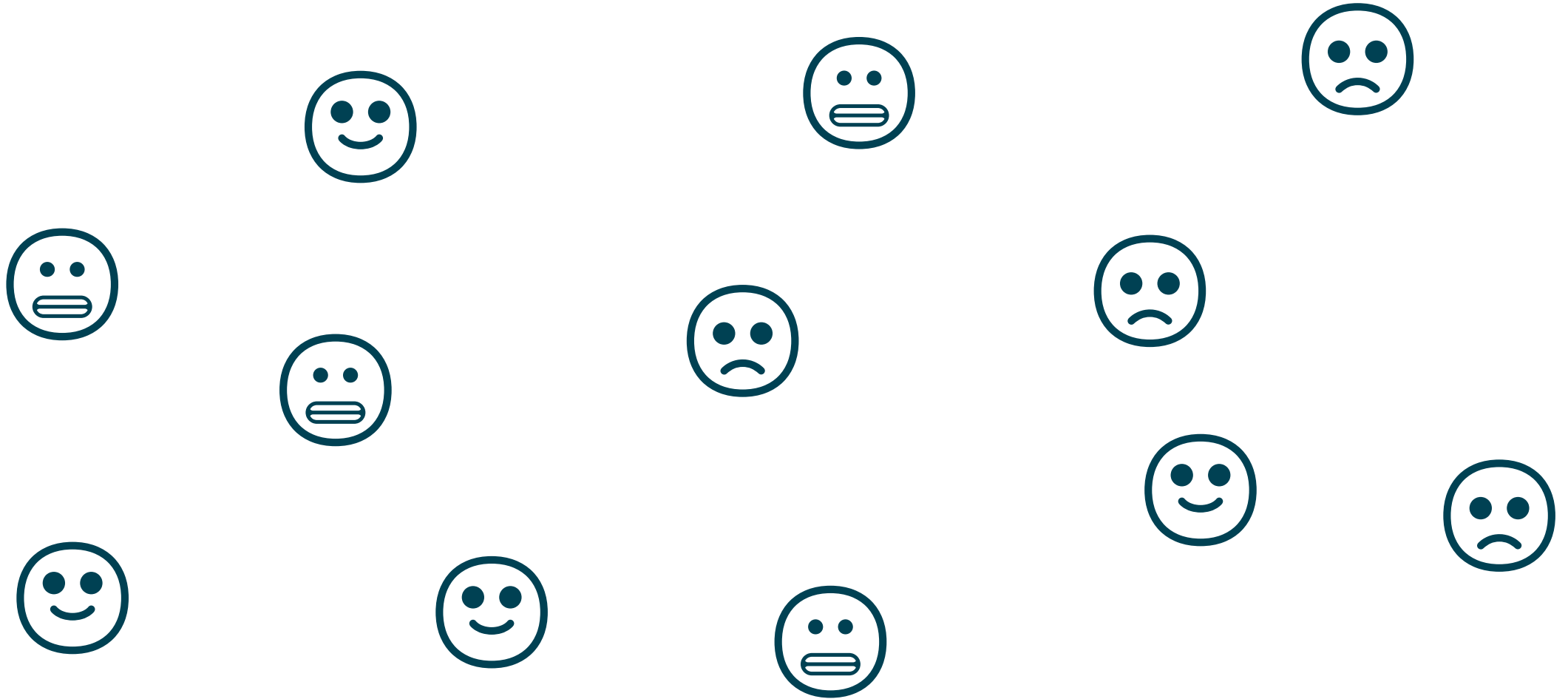


Inverse Conway

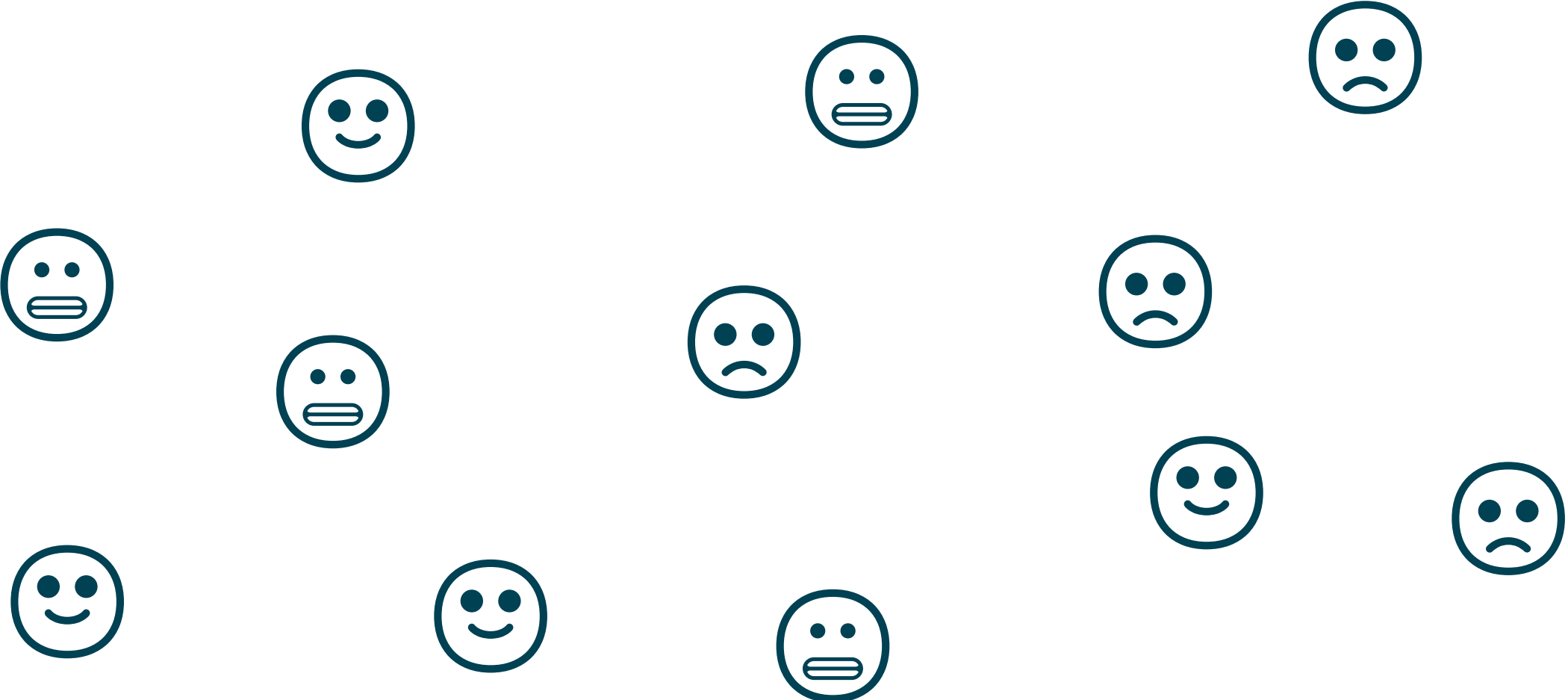
Inverse Conway Maneuver

- Architecture should drive organization
- I.e. set up the organization
- Architecture will follow

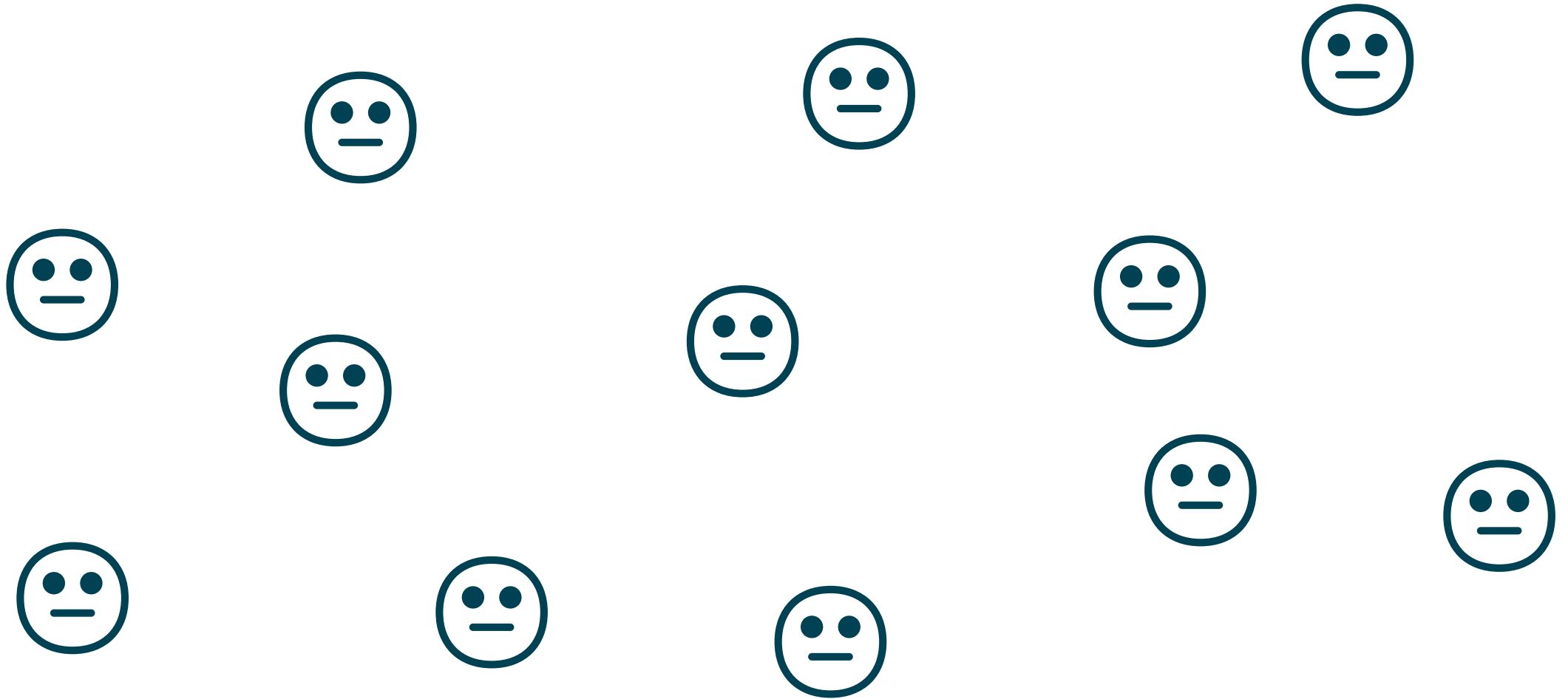
Developers, Designers ...



Chaos



Order



Order

Order Process



Delivery



Invoicing



Order

Order Process

Module

Delivery

Module

Invoicing

Module

Inverse Conway: Simplification

- Inverse Conway changes the org chart
- Org chart is not communication!
- Assumption: Org chart team will collaborate on module & communicate more internally
- Does it work that way?
- What if members of different teams sit in the same room?

Irritating the Organization

- Sociology: "irritating" organizations.
- New org chart: irritation
- Can lead to new communication structure
- Can lead to org chart teams working on modules.
- Might also be completely ignored.

Inverse Conway: Assumptions

- People will follow the org chart.
- People will communicate according to the org chart.
- Too simplistic

Inverse Conway Maneuver

- Is organization a tool for architecture?
- I.e. should architects define the organization?
- (Might not help that much anyways.)

You are a project manager.
The architect decides about who
does what.
Good?

Remember This?

Project manager decides about
the architecture.
Good?

Architects & Organization

- Who does what depends on deadlines, budgets, organizational constraints ...
- I want to do architecture.
- I happily leave those issues to project managers.
- Then they must to be empowered to decide who does what.

Architects & Project Managers

- Architects should support project managers.
- Architects should not control project managers.
- Project managers should support architects.



Provide Feedback

- Architecture is yet another constraint for the organization.
- Architects should provide feedback.
- But do architects really want to decide about it?

Drink Coffee / Tea

- Talk to project managers and provide feedback!
- Talk to other people!
- Conway: Communication not organization strengthens the architecture.



Strategic Domain-driven Design

Strategic Domain-driven Design

- Bounded context: a special module.
- Bounded context usually handled by one team.
- Example: Order process, delivery process

Strategic Domain-driven Design

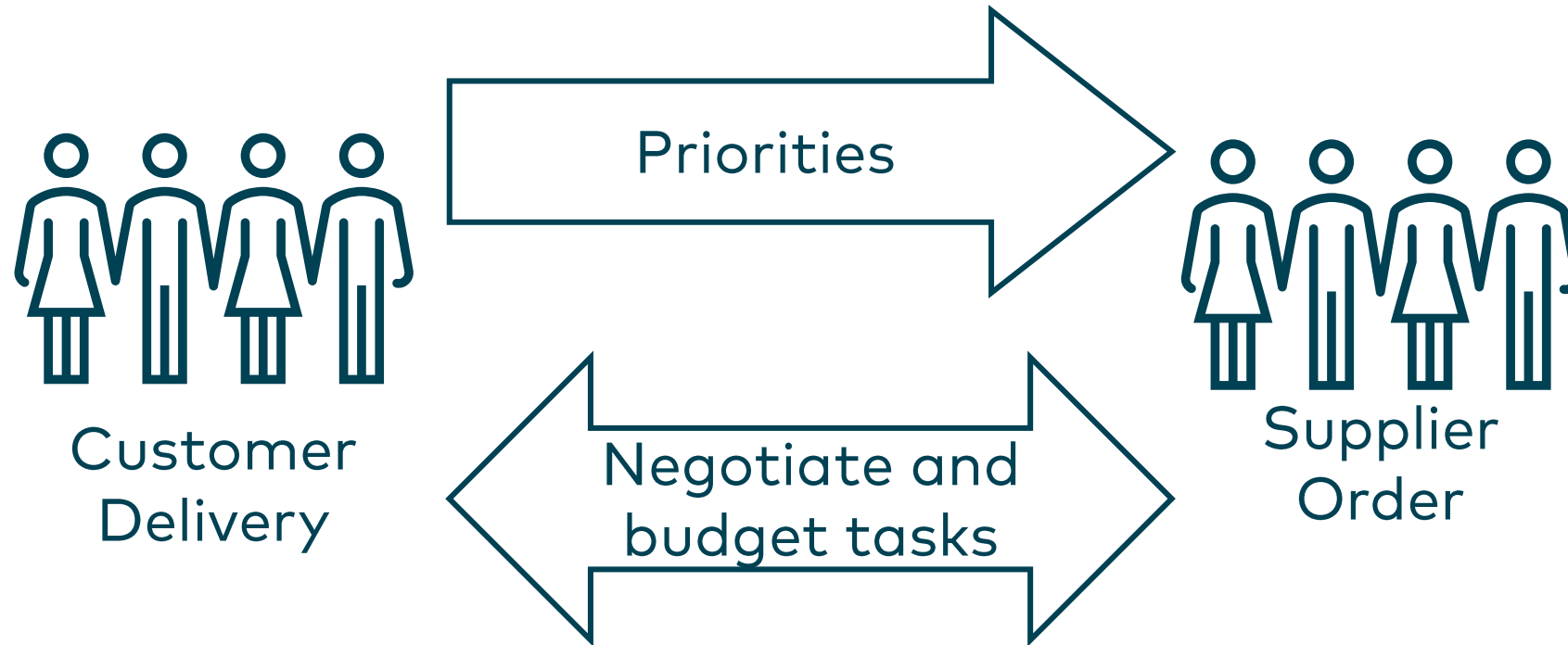
- A bounded context might be more or less important.
- A team might need support from other teams to be successful.
- Only solution: Patterns for collaboration between teams

Why do I see so little adoption of strategic domain-driven design?

Customer / Supplier

- Downstream team depends on upstream team to be successful.
- E.g. delivery process depends on order process to be successful.
- Factor downstream priorities (customer) into planning of upstream team (supplier)!
- Negotiate and budget tasks!

Customer / Supplier



Can you decide to set up a
Customer / Supplier relationship?

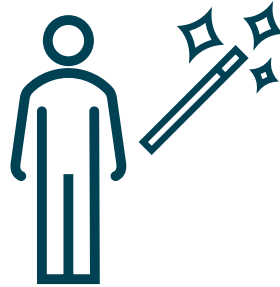
Customer / Supplier

- Customer / supplier is about who does what.
- Probably not what an architect decides.

Setting Up Customer / Supplier



Supplier



Customer

Customer / Supplier in Practise



Customer / Supplier

- Managers can set up a customer / supplier relationship.
- In real life, teams can circumvent or fight such rules.

Is this Customer / Supplier?



At the end, he and his team got the help they needed...

Even if you don't have formal control over a team, you can still try to influence it.

Customer / Supplier

- A DDD pattern architects cannot "just implement".
- Project managers might decide differently for some reason

Drink Coffee / Tea

- Customer / supplier is not something an architect decides.
- Provide feedback.
- Understand how project managers and others are setting up the organization.
- Use informal ways.



Conclusion

Conclusion

- Conway's law:
Communication -> architecture
- Modules control the flow of information.
("Information hiding")
- Conway: 3 teams cannot work on one module. They will build separate modules.

Conclusion

- Inverse Conway is too simplistic:
It irritates the organization, no guaranteed outcome.
- Architects should not try to decide the organization all by themselves.
- Architects cannot implement customer / supplier all by themselves.

Advice

- Try to better understand what is going on with the organization.
- Provide feedback and support project managers and everyone else.

Drink Coffee / Tea



Episode 81: Organisation als Werl x +

https://www.heise.de/blog/Episode-81-Organisation-als-Werkzeug-zur-Umsetzung-von-Architektur-504218...

heise online heise+

Abmelden | Mein Account Suchen Menü

IT Wissen Mobiles Security Developer Entertainment Netzpolitik Wirtschaft Journal Newsticker Foren

TOPTHEMEN: IFA ENERGIE UKRAINE UA ELEKTROMOBILITÄT KRYPTOGELD PODCASTS ANZEIGE: GO! SCHULE MORGEN

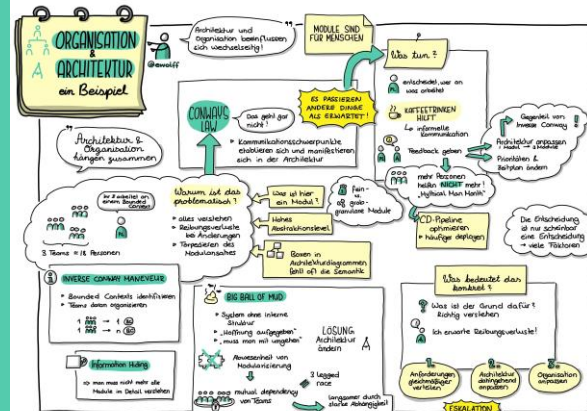
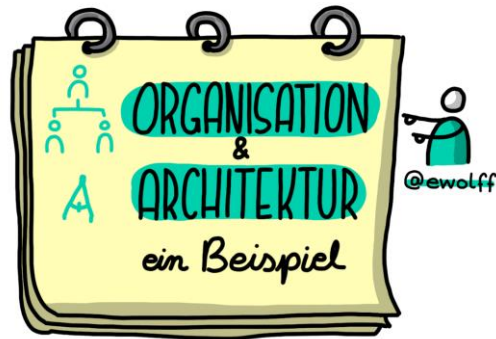
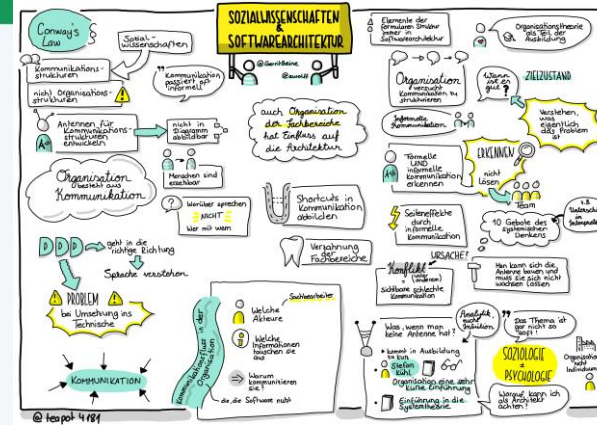
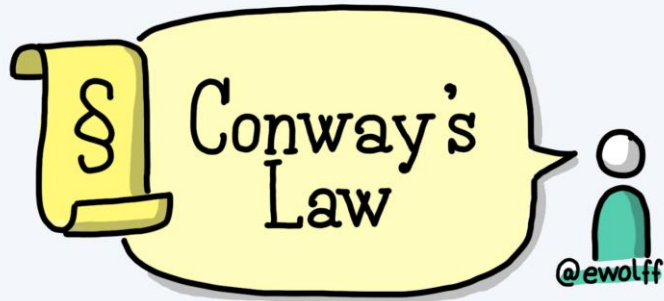
Episode 81: Organisation als Werkzeug zur Umsetzung von Architektur

In dieser Episode des SoftwareArchitekTOUR-Podcasts spricht Eberhard Wolff mit seinem Gast Gerrit Beine über praktische Wege, wie Softwarearchitekten ihren Werkzeugkoffer mit Organisationsaspekten erweitern können.

Lesezeit: 1 Min. In Pocket speichern

7

<https://www.heise.de/blog/Episode-81-Organisation-als-Werkzeug-zur-Umsetzung-von-Architektur-5042186.html>



Send email to sag2022@ewolff.com

Slides

- + Service Mesh Primer EN
- + Microservices Primer DE / EN
- + Microservices Recipes DE / EN
- + Sample Microservices Book DE / EN
- + Sample Practical Microservices DE/EN
- + Sample of Continuous Delivery Book DE

Powered by Amazon Lambda
& Microservices

EMail address logged for 14 days,
wrong addressed emails handled manually

