

Conway's Law

Definition

Conway's Law beschreibt den Zusammenhang zwischen der Struktur einer Organisation und der Architektur der Systeme, die diese Organisation entwickelt.

Die bekannteste Formulierung lautet sinngemäß:

Organisationen, die Systeme entwerfen, erzeugen Designs, deren Struktur die Kommunikationsstrukturen dieser Organisationen widerspiegelt.

Das Gesetz geht auf den Informatiker Melvin E. Conway zurück, der diese Beobachtung 1968 veröffentlichte.

Grundidee

Conway's Law besagt nicht, dass Softwarearchitektur ausschließlich durch Organisationen bestimmt wird. Es beschreibt vielmehr eine starke Tendenz: Die Art, wie Teams miteinander kommunizieren, beeinflusst die Grenzen, Schnittstellen und Abhängigkeiten technischer Systeme.

Beispiel:

- Ein Unternehmen mit getrennten Teams für Benutzeroberfläche, Backend und Datenbank neigt dazu, Systeme mit klar getrennten UI-, Backend- und Datenbankschichten zu bauen.
- Ein Unternehmen mit mehreren Produktteams neigt dazu, Software in produkt- oder domänenorientierte Module aufzuteilen.
- Wenn zwei Teams wenig miteinander sprechen, entstehen häufig technische Schnittstellen, die genau diese organisatorische Trennung widerspiegeln.

Bedeutung für Softwarearchitektur

Conway's Law ist besonders relevant für:

- Softwarearchitektur
- Systemdesign
- Microservices
- DevOps
- agile Organisationen
- Plattform- und Produktteams

Die zentrale Aussage ist: **Architektur ist nicht nur eine technische Entscheidung, sondern auch eine organisatorische Folge.**

Wenn eine Organisation stark in Silos aufgeteilt ist, entstehen häufig Systeme mit vielen Übergaben, schwerfälligen Schnittstellen und hoher Abstimmungsnotwendigkeit. Umgekehrt können gut geschnittene Teams dabei helfen, klarere Systemgrenzen zu schaffen.

Beispiel: Monolith und Microservices

Ein klassisches Beispiel ist der Übergang von einem Monolithen zu Microservices.

Wenn ein Unternehmen Microservices einführt, aber die Organisation weiterhin zentralisiert und funktional getrennt bleibt, kann das Ergebnis problematisch sein:

- Ein Datenbankteam muss jede Änderung freigeben.
- Ein Backendteam implementiert Logik für viele Produktbereiche.
- Ein Betriebsteam ist erst am Ende des Entwicklungsprozesses beteiligt.
- Produktteams besitzen ihre Services nicht vollständig.

In diesem Fall entstehen zwar technisch mehrere Services, organisatorisch bleibt das System aber abhängig von denselben alten Kommunikationswegen. Die Architektur wirkt modern, aber die Arbeitsweise bleibt langsam.

Ein besserer Ansatz besteht darin, Teams so zu schneiden, dass sie fachliche Verantwortung übernehmen können. Ein Team besitzt dann beispielsweise einen klaren Geschäftsbereich inklusive Entwicklung, Betrieb und Weiterentwicklung der zugehörigen Services.

Inverse Conway Maneuver

Das Inverse Conway Maneuver beschreibt den bewussten Versuch, die Organisationsstruktur so zu verändern, dass sie die gewünschte Systemarchitektur unterstützt.

Statt nur die Software umzubauen, wird auch die Teamstruktur angepasst.

Beispiel:

- Zuerst wird eine gewünschte Zielarchitektur definiert.
- Danach werden Systemgrenzen und fachliche Domänen identifiziert.
- Anschließend werden Teams so organisiert, dass sie genau diese Domänen verantworten.
- Kommunikationswege werden an den gewünschten Schnittstellen ausgerichtet.

Das Ziel ist, dass Organisation und Architektur zusammenpassen.

Zusammenhang mit Team Topologies

Conway's Law ist eine wichtige Grundlage für moderne Organisationsmodelle wie Team Topologies.

Dabei wird betrachtet, welche Teamarten es gibt und wie Teams miteinander interagieren sollen. Besonders wichtig ist die Idee, dass Teams möglichst klare Verantwortungsbereiche und definierte Interaktionsmuster haben sollten.

Typische Teamarten in diesem Zusammenhang sind:

- Stream-aligned Teams
- Platform Teams
- Enabling Teams
- Complicated-subsystem Teams

Auch hier gilt: Die technische Architektur soll nicht isoliert betrachtet werden, sondern gemeinsam mit

Teamgrenzen, Kommunikationsaufwand und kognitiver Belastung.

Praktische Konsequenzen

Aus Conway's Law ergeben sich mehrere praktische Empfehlungen:

- Architekturentscheidungen sollten gemeinsam mit Organisationsentscheidungen betrachtet werden.
- Teamgrenzen sollten möglichst zu fachlichen Systemgrenzen passen.
- Zu viele teamübergreifende Abhängigkeiten führen oft zu langsamer Entwicklung.
- Schnittstellen zwischen Systemen sind häufig auch Schnittstellen zwischen Teams.
- Reorganisationen können technische Architekturprobleme lösen oder verschärfen.
- Eine gewünschte Zielarchitektur benötigt passende Kommunikationsstrukturen.

Typische Fehlinterpretationen

Conway's Law wird manchmal missverstanden. Es bedeutet nicht:

- dass Organisationen immer absichtlich schlechte Architekturen erzeugen
- dass technische Entscheidungen unwichtig sind
- dass jede Teamstruktur automatisch zu guter Software führt
- dass Microservices nur durch Reorganisation entstehen können

Richtiger ist: Organisationsstruktur, Kommunikation und Architektur beeinflussen sich gegenseitig. Wer nur den Code ändert, aber die Arbeitsweise unverändert lässt, löst oft nicht das eigentliche Problem.

Vorteile der bewussten Anwendung

Eine bewusste Berücksichtigung von Conway's Law kann helfen, folgende Ziele zu erreichen:

- klarere Verantwortlichkeiten
- geringere Abstimmungskosten
- bessere Skalierbarkeit von Entwicklungsteams
- stabilere Systemgrenzen
- schnellere Lieferfähigkeit
- weniger unbeabsichtigte Kopplung zwischen Systemteilen

Risiken bei Nichtbeachtung

Wird Conway's Law ignoriert, können typische Probleme entstehen:

- technische Architektur passt nicht zur Organisation
- Teams blockieren sich gegenseitig
- Änderungen benötigen viele Freigaben
- Systeme werden schwer verständlich
- Schnittstellen entstehen zufällig statt bewusst
- Verantwortlichkeiten bleiben unklar

Zusammenfassung

Conway's Law macht sichtbar, dass Softwarearchitektur und Organisationsstruktur eng miteinander verbunden sind. Systeme spiegeln häufig die Kommunikationswege der Menschen wider, die sie bauen.

Für moderne Softwareentwicklung bedeutet das: Wer die Architektur verbessern will, sollte nicht nur über Technologien, Module und Schnittstellen sprechen, sondern auch über Teams, Verantwortlichkeiten und Zusammenarbeit.

Das Ziel ist eine Organisation, deren Kommunikationsstruktur die gewünschte Architektur unterstützt, statt sie ungewollt zu behindern.

From: <https://wiki.ct-lab.info/> - Creative Technologies Lab | dokuWiki

Permanent link: <https://wiki.ct-lab.info/doku.php/extras:wissikon:alle-moeglichen-begriffe:convays-law>

Last update: **2026/06/03 11:43**

