



Nube de Alibaba

Siguiendo

Una compañía global de computación en la nube y AI, una subsidiaria de Alibaba Group. Sitio web: <https://www.alibabacloud.com/>

13 de diciembre · 10 minutos de lectura

# La ley de Conway: una base teórica para la arquitectura de microservicio

## Introducción

La arquitectura de microservicios es un concepto nuevo que se ha vuelto muy popular y se ha convertido recientemente en un tema candente de investigación. Sin embargo, la implementación de microservicios aún está vagamente definida y no hay una prueba teórica de su efectividad.

Puede que se sorprenda al saber que la idea de microservicios se introdujo por primera vez en un artículo publicado hace más de cincuenta años. Además, a lo largo de los años, numerosos estudios han demostrado la precisión de numerosos puntos presentados en ese artículo.

Uno de los conceptos fundamentales presentados en el artículo son las Leyes de Conway. Aunque inicialmente se pretendía señalar los defectos de los equipos distribuidos, muchas organizaciones han aplicado las Leyes de Conway para crear una arquitectura eficiente de microservicios.

Este artículo explora las ideas de las Leyes de Conway con referencia al artículo titulado "Ley de Conway bajo distancia remota - Construcción de equipos en un mundo distribuido", escrito por Mike Amundsen (autor de Design RESTful API).

# HOW DO COMMITTEES INVENT?

By MELVIN E. CONWAY

That kind of intellectual activity which creates a world, while from its diverse parts may be called the design of a system. Whether the particular activity is the creation of specifications for a new 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 definition of a design organization is the creation and assembly of a structure consisting of different, but structured, bodies of information. We may name this information the system design. It is typically produced by a group who usually design to carry out some activity guided by the system design. For example, a public official may wish to propose legislation to amend a recurrence of a recent disaster, so he appoints a team to explore the consequences. Or a manufacturer creates a new product and develops a product planning activity to specify what should be produced.

The design organization may or may not be involved in the production of the system it designs. Frequently, in public affairs, there are policies which encourage a group 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 serving clients. Many of these clients may be used from design decisions, they may also be produced decisions the designer makes about his own design. As we shall see later, the decisions which exist in a conventional management environment can motivate choices which influence the form of the system.

## Stages of design

The initial stages of a design effort are concerned more with structuring of the design activity than with the system itself. The following design activity stages are presented and certain preliminary references are given. These include:

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

The third stage is defined later than the very act of design.

"A critical but much less sophisticated discussion of the behavior of some design organizations is found in the Harvard Business Review, January 1967, 'The Information System', by the author of this paper, Stephen M. Kelly, D.D., the executive director of the 'The Information System'."

"For a discussion of the problems which may arise when the design activity takes the form of a product is a book on systems, by C. E. Martin, 'How to Use the Design Organization', Harvard Business Review, March-April, 1967, p. 78."

## 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 decisions 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 unorganized.

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

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

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

1. Division of functions according to the general 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 through this list. It might occasionally merge some steps, or it may, and obviously, suppress design concepts, but each an approximation of uncertainty in understanding, and the very act of intentionally obscuring a criterion is partial and expensive. Of course, from the



Dr. Conway is manager, peripheral systems research, of the Army Research Office, where he is working on some of the most serious research. He has previously been a research associate at Case Western Reserve Univ., and a research associate at the University of California, Los Angeles. He is now at the RAND Corp.

INFORMATION

La línea más famosa en la Ley de Conway es:

"Las organizaciones que diseñan sistemas están obligadas a producir diseños que son copias de las estructuras de comunicación de estas organizaciones". Melvin Conway (1967).

Esto significa que las organizaciones que diseñan sistemas están obligadas a producir diseños que repliquen las estructuras de comunicación de las organizaciones. La siguiente figura ilustra este concepto.

La figura representa la estructura de comunicación existente de las organizaciones, que coincide con sus respectivos procesos de desarrollo de productos. En pocas palabras, la estructura de la organización es igual al diseño del sistema.

Aquí, los sistemas mencionados por el autor no están restringidos a los sistemas de software. También se especula que la Harvard Business Review inicialmente rechazó este artículo. Por lo tanto, Conway lo envió a una revista de programación, lo que llevó a la idea errónea de

que el artículo trataba del desarrollo de software. Al principio, el autor no propuso sus ideas como leyes y solo describió sus hallazgos y conclusiones. Cuando el famoso libro *The Mythical Man-Month* introdujo la Ley de Brooks y citó algunos de los puntos de Conway, las ideas de Conway se popularizaron en la conocida Ley de Conway que conocemos hoy.

## Ley de Conway desmitificada

En sus artículos, Mike Amundsen resumió algunos puntos de vista centrales, como se indica a continuación.

- Ley 1

1. La comunicación dicta el diseño

o El modo de comunicación organizacional se expresa a través del diseño del sistema

- Ley 2

1. Nunca hay tiempo suficiente para hacer algo bien, pero siempre hay tiempo suficiente para hacerlo

o Una tarea nunca se puede hacer perfectamente, incluso con tiempo ilimitado, pero siempre hay tiempo para completar una tarea

- Ley 3

1. Hay un homomorfismo desde el gráfico lineal de un sistema hasta el gráfico lineal de su organización de diseño

o Existe homomorfismo entre sistemas lineales y estructuras organizativas lineales

- Ley 4

1. Las estructuras de los sistemas grandes tienden a desintegrarse durante el desarrollo, cualitativamente más que los sistemas pequeños

o Una organización de sistema grande es más fácil de descomponer que una más pequeña

## Primera ley de Conway

"Los seres humanos son animales sociales complejos".

Otros campos también han proporcionado algunas ilustraciones sobre la estrecha relación entre la comunicación organizada y el diseño del sistema. Para un sistema complejo, los temas de diseño siempre implican la comunicación entre los seres humanos. Un buen diseño de sistema aborda problemas relacionados con dicha comunicación. Muchos puntos de vista en la era clásica de 1975, *The Mythical Man-Month*, resuenan con esta idea.

La línea más memorable de *The Mythical Man-Month* es:

"Agregar mano de obra a un proyecto de software tardío lo hace más tarde" - Fred Brooks, (1975)

Aumentar la cantidad de programadores para mantenerse al día con un calendario apretado es un escollo común para muchas organizaciones. Si bien tiene sentido aumentar la fuerza de trabajo para aumentar la producción, simplemente no se aplica a la palabra del desarrollo de software.

¿Por qué es este el caso? El *Mythical Man-Month* proporciona una respuesta simple: el costo de comunicación aumenta exponencialmente a medida que aumenta el número de personal en un proyecto u organización. El costo de comunicación se puede calcular con la fórmula  $n(n-1)/2$ , donde la complejidad del algoritmo de gestión del proyecto es  $O(n^2)$ . El siguiente ejemplo ilustra la idea del costo de comunicación:

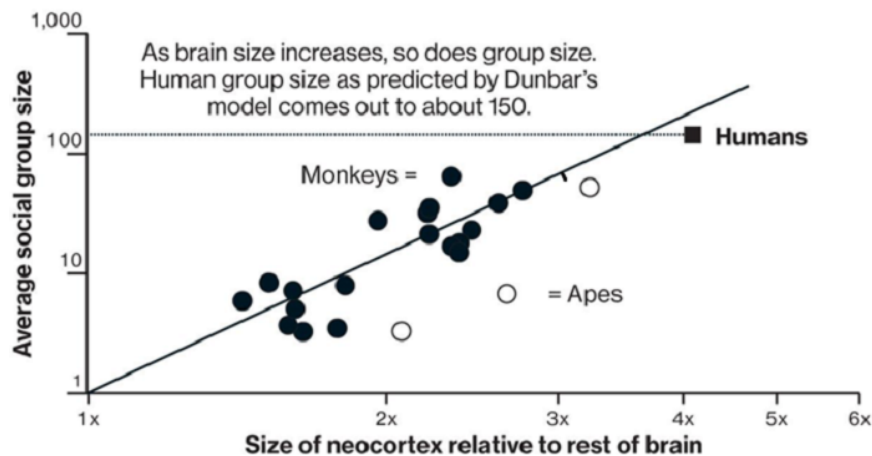
- Para un equipo de proyecto con cinco miembros, el número requerido de canales de comunicación es  $5 * (5-1) / 2 = 10$ .
- Para un equipo de proyecto con 15 miembros, el número requerido de canales de comunicación es  $15 * (15-1) / 2 = 105$ .
- Para un equipo de proyecto con 50 miembros, el número requerido de canales de comunicación es  $50 * (50-1) / 2 = 1,225$ .
- Para un equipo de proyecto con 150 miembros, el número requerido de canales de comunicación es  $150 * (150-1) / 2 = 11,175$ .

Esta es la razón principal por la cual las startups de internet tienen pequeños equipos. Si una startup tiene demasiados empleados, agotará la inversión de VC poco después de que el CEO presente su idea a todos los involucrados.

Otra teoría interesante y relevante presentada por el biólogo Robin Dunbar en 1992 se llama el "Número Dunbar". Al principio, Dunbar descubrió que la capacidad cerebral de un primate se correlaciona con el tamaño de su población. Luego postuló algunas estimaciones sobre la cantidad de relaciones que un cerebro humano puede mantener. Por ejemplo, una persona típica tendría

- 5 amigos íntimos
- 15 amigos de confianza
- 35 amigos cercanos
- 150 amigos ocasionales

### The Social Cortex



¿No están aparentemente asociados con los costos de comunicación mencionados anteriormente? Sí, nuestros cerebros nos limitan a mantener solo muchas relaciones. (En un equipo de desarrollo, la cantidad puede ser incluso menor).

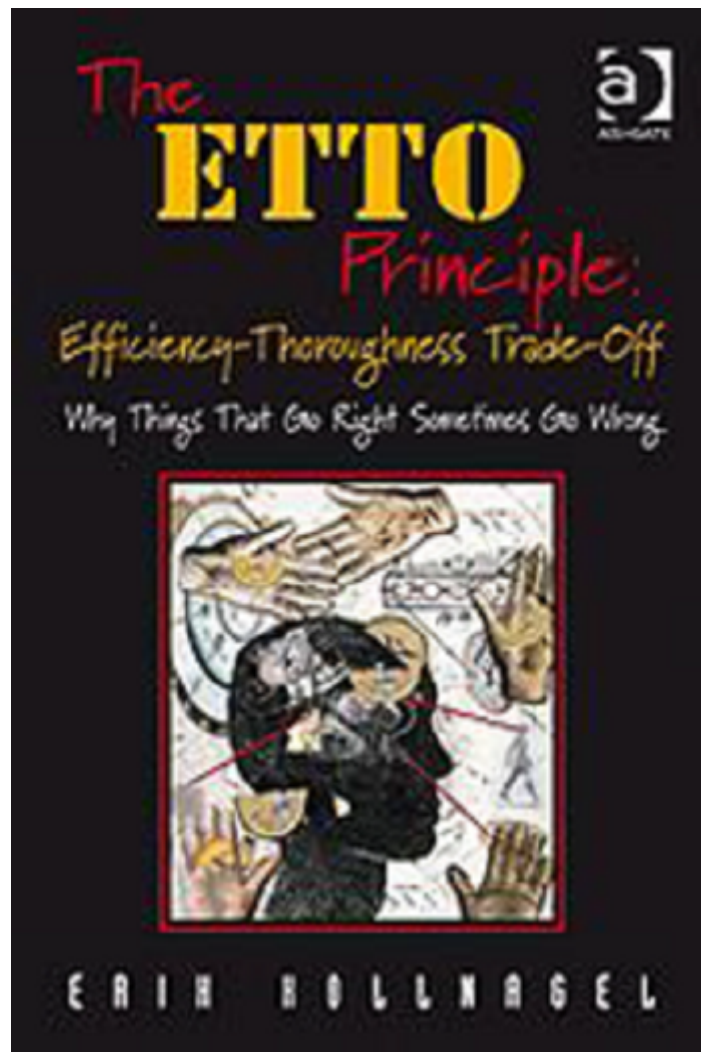
Los problemas de comunicación conducen a problemas de diseño del sistema que afectan la eficiencia de desarrollo de todo el sistema, así como los resultados finales del desarrollo del producto.

## Segunda ley de Conway

"Roma no fue construida en un día. Abordar los problemas que se pueden abordar primero".

Erik Hollnagel, uno de los titanes en el desarrollo ágil, ha explicado algunos puntos similares en su libro titulado Efficiency-Thoroughness Trade-Off.

"¿Problema demasiado complicado? Ignorar detalles  
¿No hay suficientes recursos? Renunciar a las características. "  
- Erik Hollnagel (2009)





La complejidad del sistema, el número de funciones, la competencia en el mercado y las expectativas de los inversionistas están aumentando, pero la inteligencia humana permanece constante. Ninguna organización está segura de si puede encontrar suficientes talentos, independientemente de las capacidades y los fondos. Para un sistema extremadamente complejo, los operadores siempre ignorarán algo. Erik cree que la mejor solución para este problema es simplemente "dejarlo pasar".

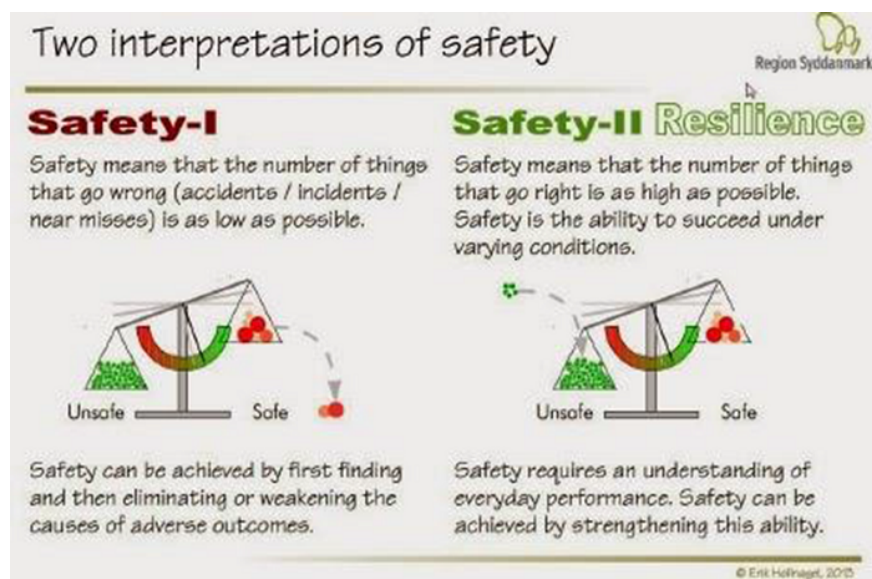
A menudo encontramos tales problemas durante las tareas diarias de desarrollo. ¿Son los requisitos planteados por los gerentes de producto demasiado complejos? Si es así, ignore algunos requisitos menores y concéntrese primero en los más importantes. ¿Los gerentes de producto tienen demasiados requisitos? Si es así, renunciar a algunas funciones.

Los informes indican que Erik una vez recibió una invitación de un transportista aéreo para proporcionar servicios de consultoría sobre la estabilidad y seguridad de un sistema de vuelo. Erik cree que es posible garantizar la seguridad de dos maneras:

- Para garantizar la seguridad ideal, las personas deben detectar y eliminar tantos errores como sea posible.

- Para garantizar la seguridad elástica, las personas deben manejar de inmediato los errores que se producen, para la recuperación del servicio.

Para un sistema tan complejo como el sistema de vuelo, es probable que se pasen por alto algunas vulnerabilidades, sin importar qué tan bueno sea el probador. Por lo tanto, Erik recomendó a la compañía abandonar la idea de establecer un sistema perfecto. En cambio, recomendó la seguridad relativa y la corrección, en las que el transportista realiza pruebas de vuelo continuas para identificar problemas y garantizar que el sistema pueda recuperarse automáticamente en caso de falla. La siguiente figura muestra las diferentes interpretaciones de seguridad.



¿Suena familiar? ¿No significa integración continua y desarrollo ágil? Absolutamente.

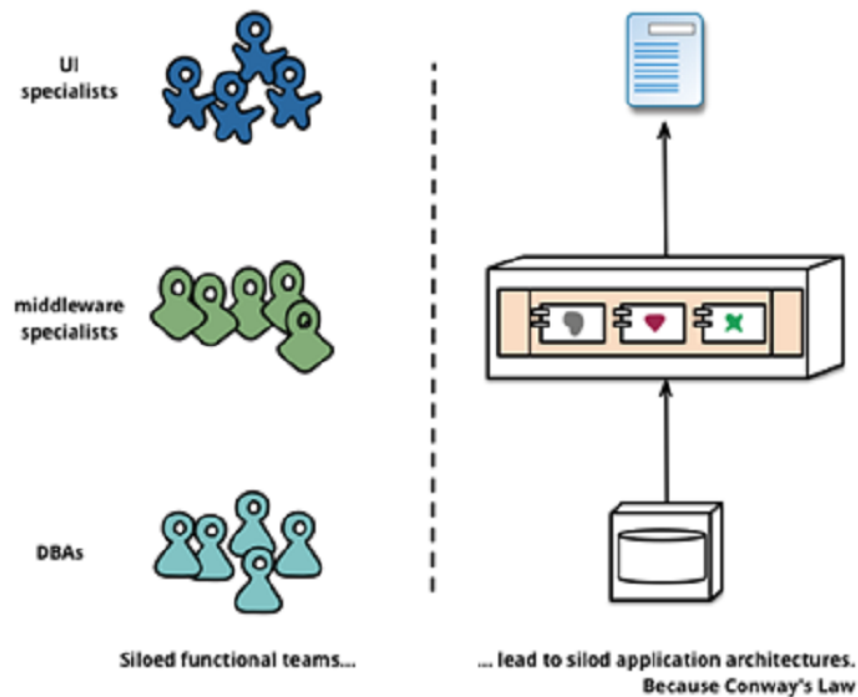
El principio anterior es el mismo que se aplica a la resistencia de los sistemas distribuidos mantenidos por las compañías de Internet. Es imposible identificar y corregir todos los errores en un sistema distribuido, incluso si las pruebas unitarias cubren todo el sistema. Los sistemas distribuidos son propensos a errores. La solución óptima no es eliminar todos los problemas, sino tolerarlos e implementar la recuperación automática en caso de falla. En un sistema compuesto por microservicios, cada microservicio puede dejar de responder, lo cual es completamente normal. Solo necesitamos garantizar suficiente redundancia y respaldo, lo que también se denomina resiliencia o diseño de alta disponibilidad.

## La tercera ley de Conway



"Crear subsistemas independientes para reducir el costo de comunicación".

El diagrama representa una aplicación específica de la relación interna entre una organización y el diseño del sistema según la primera ley de Conway. En pocas palabras, configure un equipo adecuado para el sistema que desee. Si tiene un equipo de front-end, un equipo de desarrollo de back-end de Java, un equipo de DBA y un equipo de O & M, su sistema se verá como el siguiente:



En cambio, si los límites comerciales crean divisiones en su sistema y todos los miembros convierten sus módulos en pequeños sistemas o productos para abordar los mismos objetivos comerciales, su sistema más grande se verá como una arquitectura de microservicio como se muestra a continuación:

La idea de microservicios entre equipos debe ser "interoperar, no integrar". Interoperar significa definir límites e interfaces del sistema y ofrecer una pila completa a todo el equipo para una autonomía completa. Si la configuración de un equipo sigue esta conjetura, generará costos de comunicación dentro del sistema y los subsistemas se comunicarán más. Dicha disposición da como resultado una menor dependencia entre sistemas y menores costos de comunicación entre sistemas.

## La cuarta ley de Conway

"Divide y conquistarás."

As mentioned above, human beings are complex social animals and communication between people is very complicated. When it comes to a system, we often choose to add manpower to reduce its complexity. For our organization, how do we address such communication issues? Divide and conquer. Look at your company, isn't it true that a line-1 manager in your company manages less than 15 people, a line-2 manager manages fewer people than a line-1 manager, a line-3 manager manages even fewer people than a line-2 manager, and so on? (I am not implying that it is more difficult to manage development managers than programmers.)

Therefore, a large organization usually has small team divisions to reduce the communication costs/ management issues. Here are some scenarios for you to consider.

- La idea de comenzar un negocio es muy buena. Permítanos reclutar más programadores. De todos modos, el CV nos ha ofrecido una gran suma de dinero.
- Hay demasiadas personas para administrar. Necesito encontrar varios gerentes que me ayuden y me informen.

La Ley de Conway también nos dice que podemos ver los modos de comunicación organizacional desde el diseño del sistema. Cada gerente es responsable de un cierto deber en una pequeña parte de un sistema grande. De esta manera, existen límites de comunicación entre ellos y el sistema más grande. Como tal, el sistema más grande incorpora equipos de división más pequeños a cargo de los sistemas más pequeños (el microservicio sirve bien para esto).

## Leyes y microservicios de Conway

Echemos un vistazo a cómo la Ley de Conway proporcionó la base teórica para microservicios hace medio siglo.

- La comunicación entre los seres humanos es compleja y cada persona tiene una cantidad limitada de energía para la comunicación. Por lo tanto, cuando un problema es complejo y requiere una reparación concertada, debemos dividir nuestra organización para mejorar la eficiencia de la comunicación.

- El diseño del sistema en el que trabajan los miembros de una organización depende de la comunicación entre los miembros. Los gerentes pueden ajustar el modo de división para implementar diferentes formas de comunicación entre los equipos, lo que influirá en el diseño del sistema.
- Si un subsistema es comunicacional y tiene límites claros de comunicación externa, entonces podemos reducir efectivamente los costos de comunicación, y el diseño correspondiente será más apropiado y eficiente.
- Es necesario optimizar continuamente un sistema complejo con la ayuda de la tolerancia a los errores y la resistencia. No espere diseños o arquitecturas grandes y completos, ya que su desarrollo se produce de forma iterativa.

Aquí hay algunas sugerencias prácticas:

- Aproveche todos los medios posibles para mejorar la eficiencia de la comunicación, como Slack, Github y Wiki. Comuníquese solo con las personas involucradas. Cada persona y cada sistema debe tener deberes claros. Debe saber a quién recurrir en caso de un problema, para garantizar la rendición de cuentas.
- Diseñe un sistema en el modo MVP, verifique y optimice el sistema de forma iterativa y asegúrese de que el sistema sea elástico.
- Adopte un equipo que se alinee con el diseño de su sistema y agilice al equipo si es posible. Una recomendación plausible es que, siempre que sea posible, configure equipos por departamentos para que cada equipo sea autónomo y comunicativo. Aclare los límites departamentales para reducir los costos de comunicación externa. Cada pequeño equipo debe ser responsable de su módulo a lo largo de todo el ciclo de vida del módulo. Prevenga límites vagos y cambie la responsabilidad. Establezca la relación "interoperar, no integrar" entre los equipos.
- Desarrolle equipos pequeños y eficientes, ya que los costos aumentan y la eficiencia disminuye cuando aumenta el número de miembros del equipo. Jeff Bezos, CEO de Amazon, tenía una divertida regla general: si dos pizzas no son suficientes para un equipo, el equipo es demasiado grande. Por lo general, un pequeño equipo de productos de una empresa de Internet consta de 7 a 8 personas. (Estos incluyen personas a cargo de pruebas de front-end y back-end, interacciones e investigación de usuarios. Algunas personas pueden tener asignaciones de tareas múltiples).

Al observar los siguientes criterios de microservicio, podemos ver fácilmente la estrecha relación entre los microservicios y la Ley de Conway:

- Sistemas que consisten en servicios distribuidos
- División de organización por línea de negocio
- Desarrollo de productos excelentes, no proyectos
- Puntos finales inteligentes y tuberías tontas (esto se refiere a personas altamente capaces y esfuerzos de comunicación ligera)
- O & M automático (DevOps)
- Tolerancia a los errores
- Rápida evolución

## Conclusión

This article introduces Conway's laws and explores whether they offer a theoretical explanation of the concept of microservices. It discusses the four laws in detail and the application of each law. The first law talks about the connection between communication and system design. The second law talks about efficiently completing tasks, with perfection not an attainable goal and hence should not be a reason for delayed task completion. Instead, people should focus on completing tasks on time, with regular improvements to follow. The third law talks about the homomorphism that exists between linear systems and linear organizational structures. Finally, the fourth law discusses the means with which people can utilize the "Divide and Conquer" approach to reduce the complexity and costs involved in communication within large enterprises.

## References

- [Ley de Conway bajo distancia remota - Team Construction in a Distributed World](#) (Las imágenes en este artículo son de las capturas de pantalla del PPT)
- [Ley de Conway en wiki](#)
- [Página principal de Conway's Law](#)

