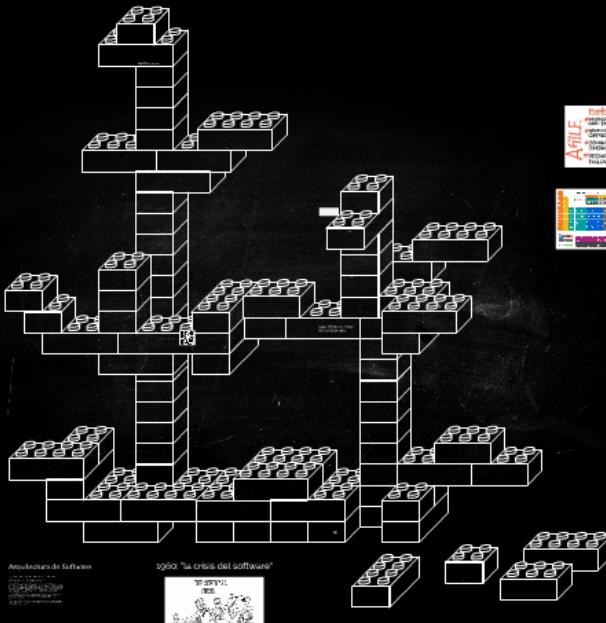


# Herramientas indispensables de Arquitectura de Software

Rodrigo Valdés

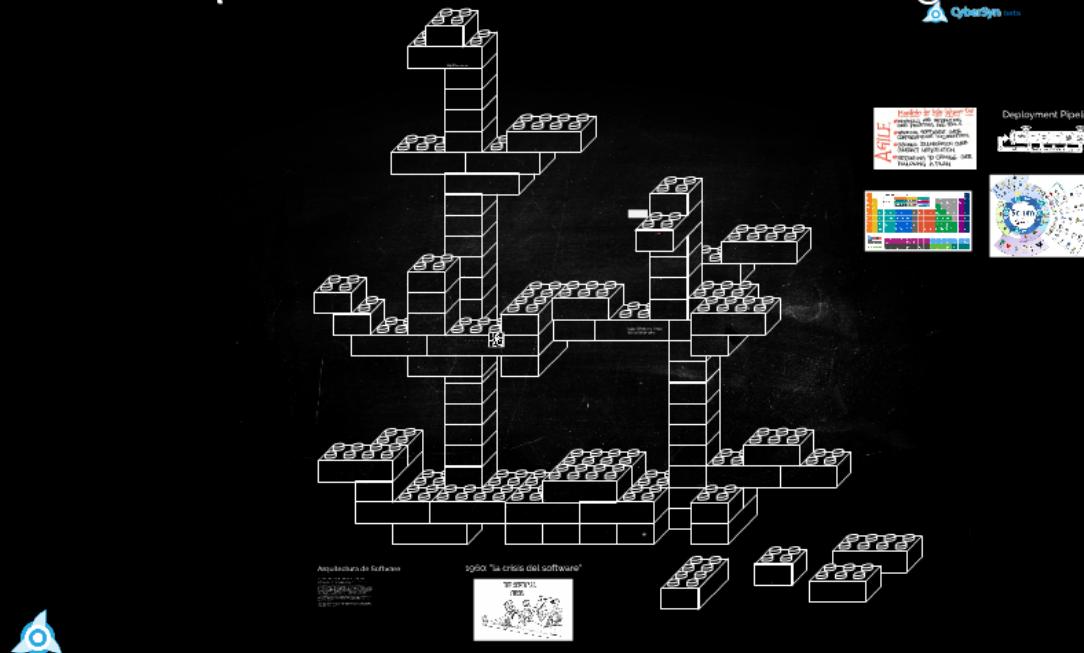


Deployment Pipeline

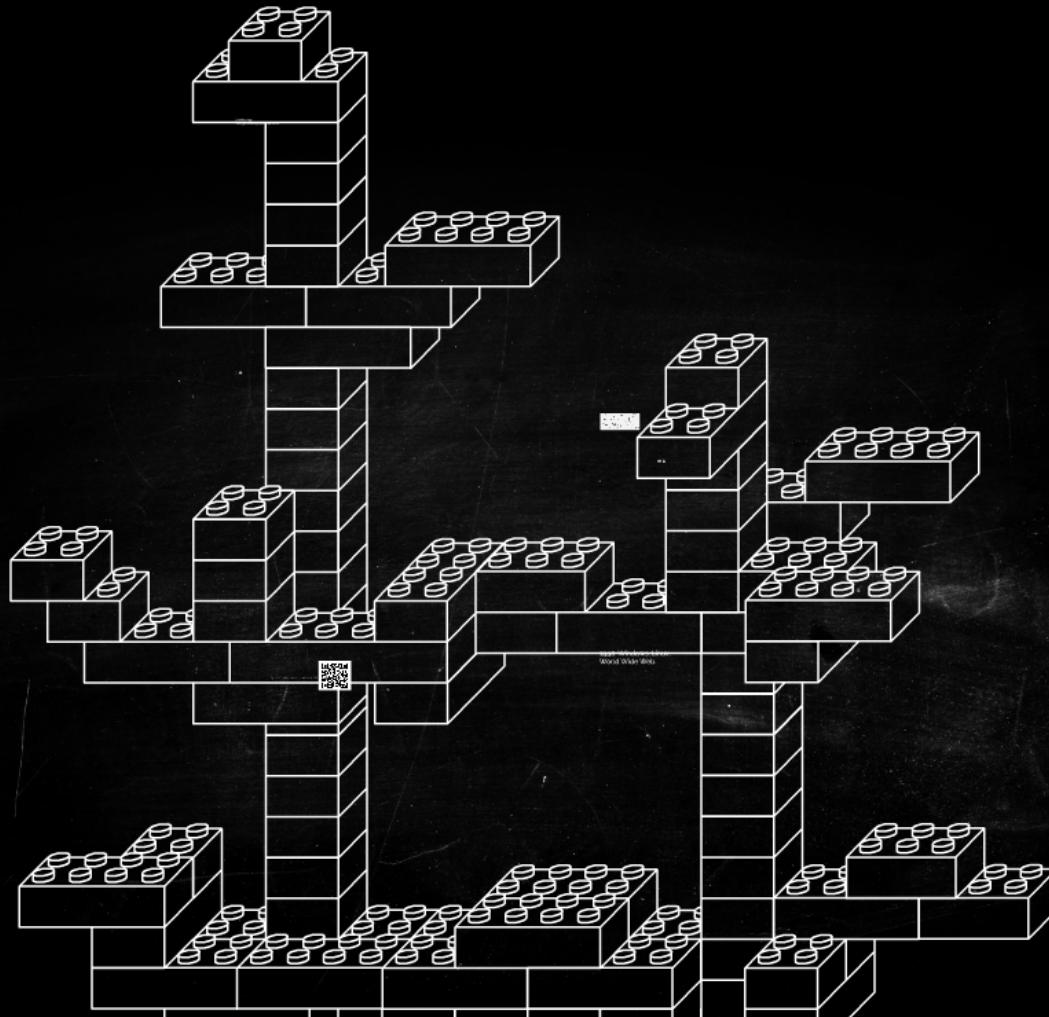


# Herramientas indispensables de Arquitectura de Software

Rodrigo Valdés



# rientas indispensables arquitectura de Software

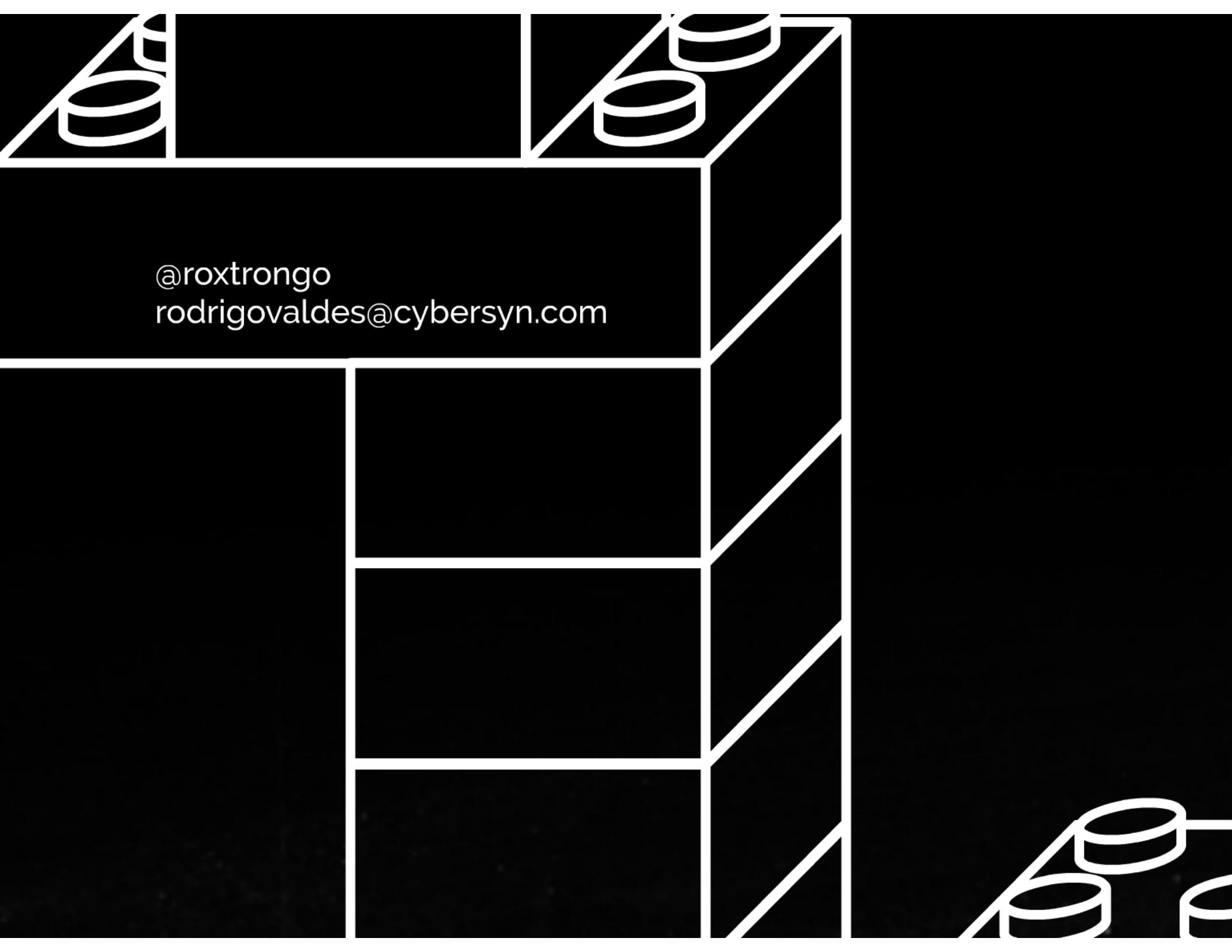


## Rodrigo Valdés

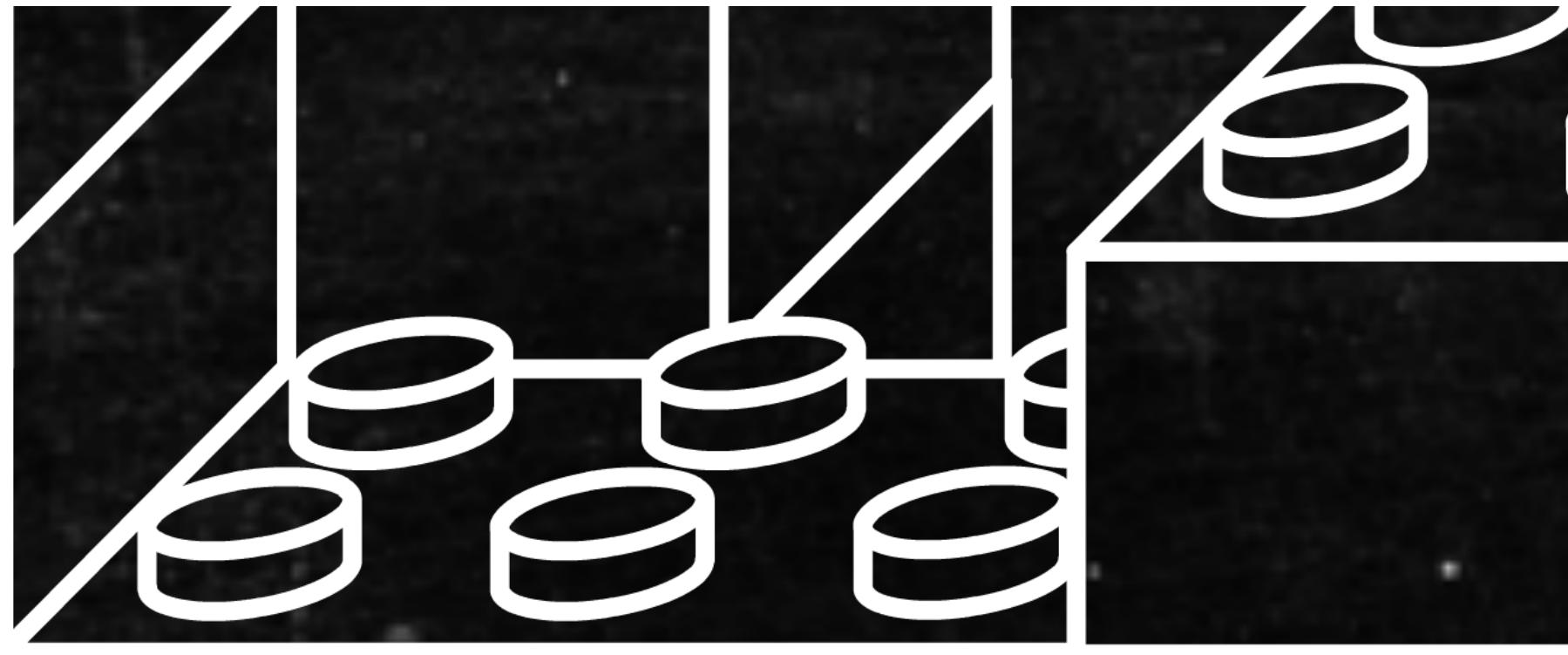


Deployment Pipeline

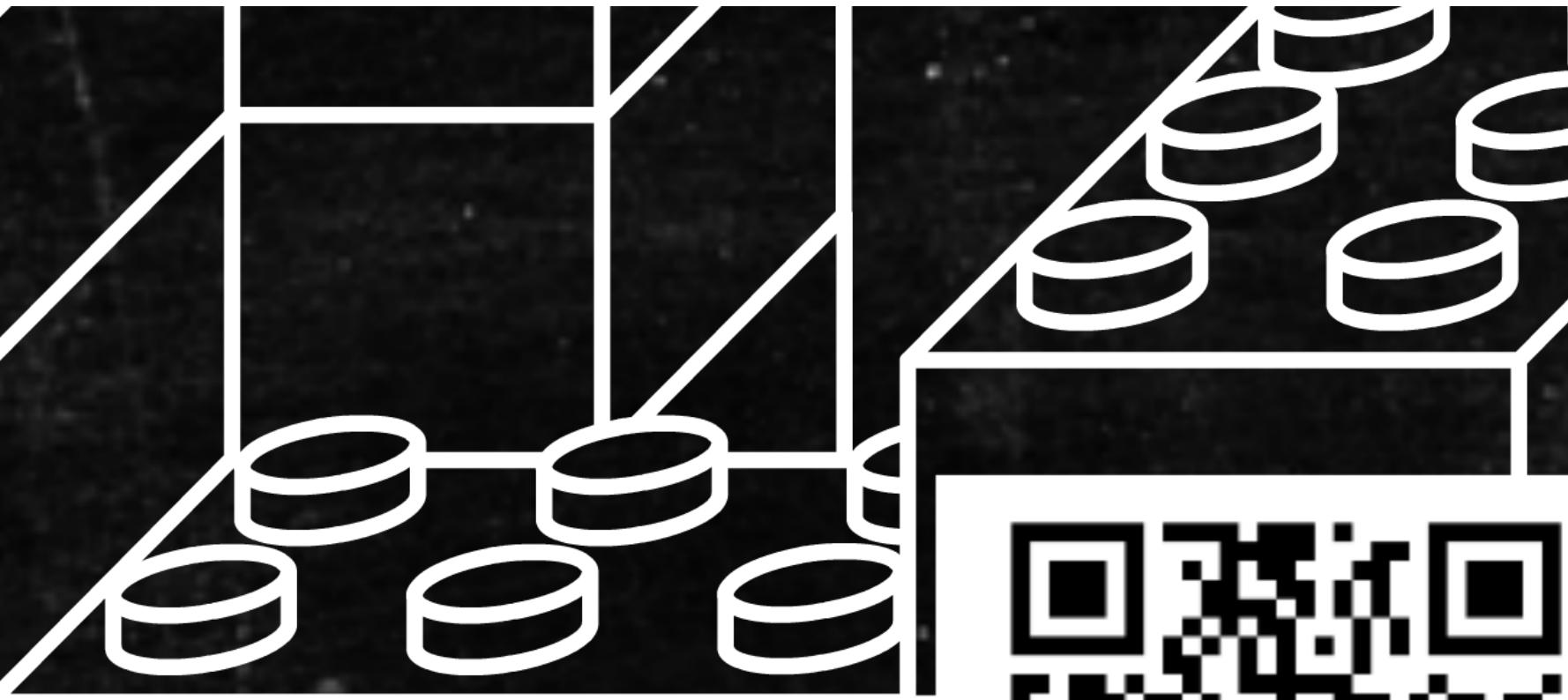




@roxtrongo  
rodrigovaldes@cybersyn.com

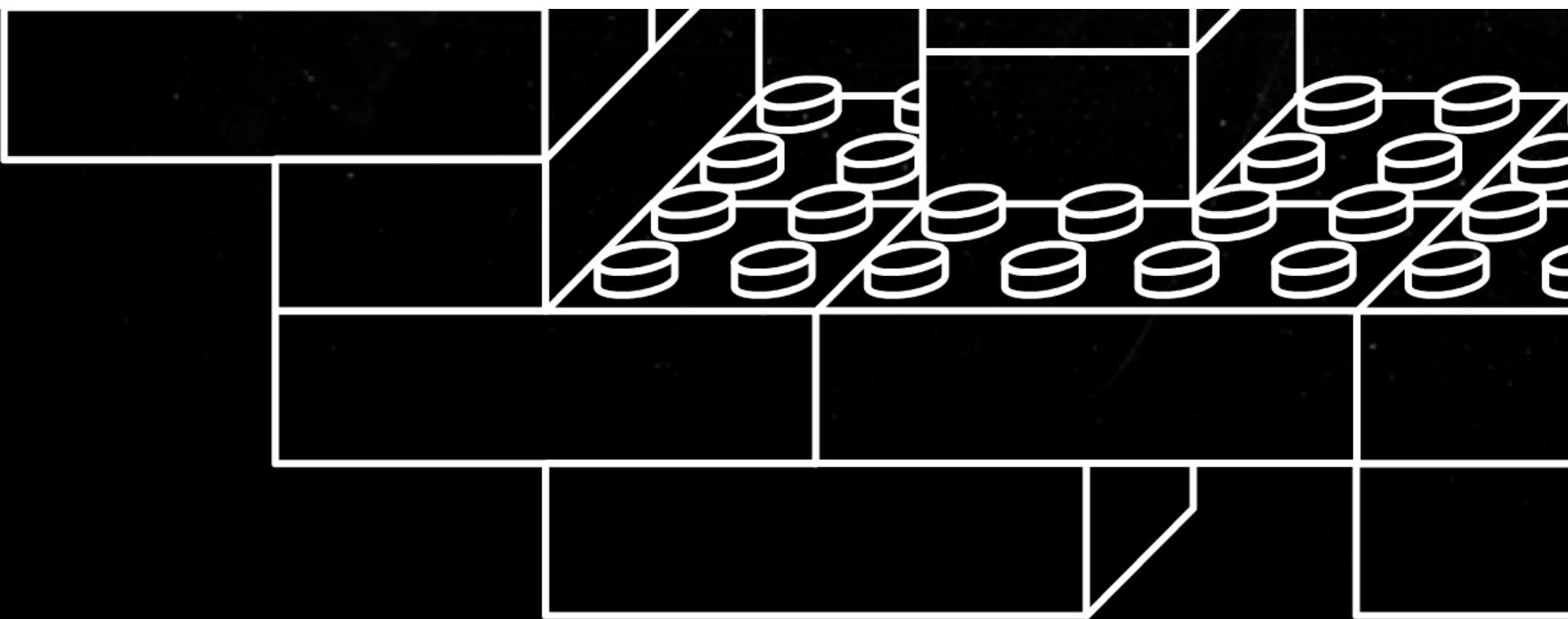


Respositorio GitHub Charla <https://github.com/roxtrongo/UNAB>



Respositorio GitHub Charla <https://github.com/roxitrongo/UNAB>





# Arquitectura de Software

1960: "l

<sup>1</sup> La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

<sup>2</sup> Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco.

<sup>3</sup> Una arquitectura de software se selecciona y diseña con base en objetivos (requerimientos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

<sup>4</sup> La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.<sup>5</sup>

<sup>5</sup> Es una disciplina que se hace necesaria cuando la complejidad asociada a la construcción de software se hace patente en el mundo y una de las primeras manifestaciones que observamos en los años 60 fue conocida como la Crisis del Software.



# Arquitectura de So

\* La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema.

\* Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco.

\* Una arquitectura de software se selecciona y diseña con base en objetivos (requerimientos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

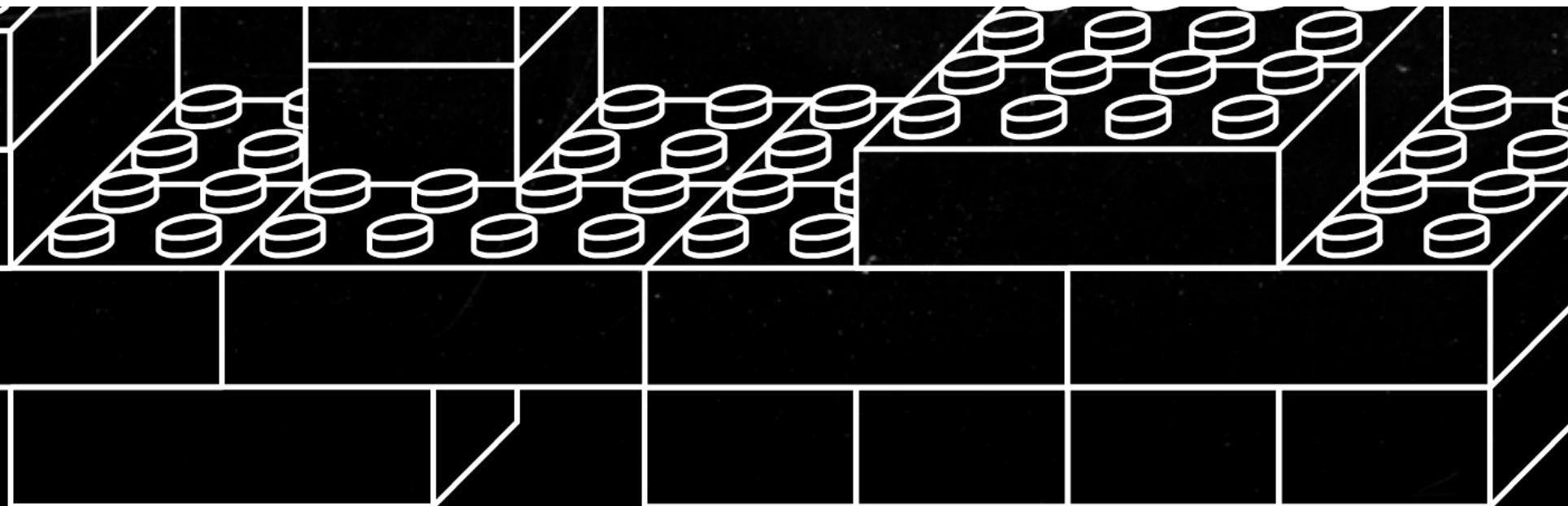
\* La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea."

\* Es una disciplina que se hace necesaria cuando la complejidad asociada a la construcción de software se hace patente en el mundo y una de las primeras manifestaciones que observamos en los años 60 fue conocida como la Crisis del Software



Problema: complejidad inherente a la disciplina

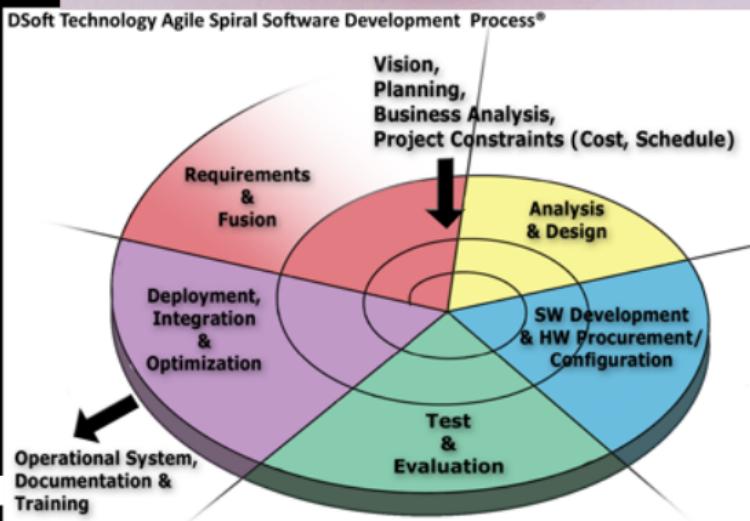
Herramienta indispensable para dar solución:  
Procesos de desarrollo de software



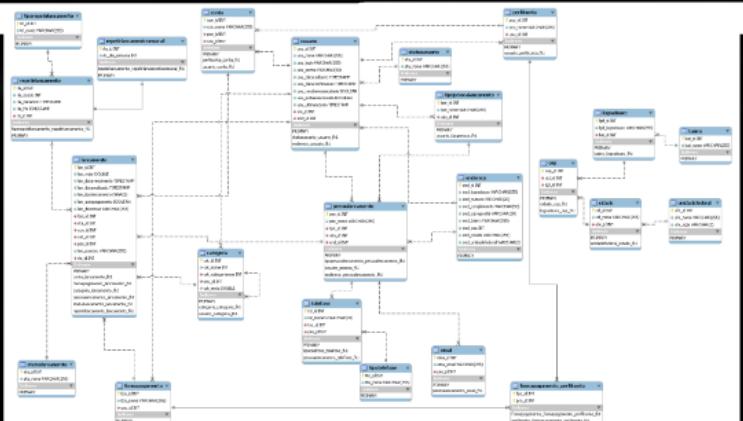
1960: "la crisis del software"







## Década de 1970: Lenguajes de programación de alto nivel (programación orientada a objetos y modelo relacional)



Programa

## Programación orientada a objetos



### Características

- Marca
- Color
- Tipo
- Precio
- No. de Puertas
- Tipo Combustible
- Cilindros
- Transmisión

### Acciones

- Encender
- Avanzar
- Retroceder
- Detener
- Apagar

### Características

- Nombre
- Especie
- Color
- Edad

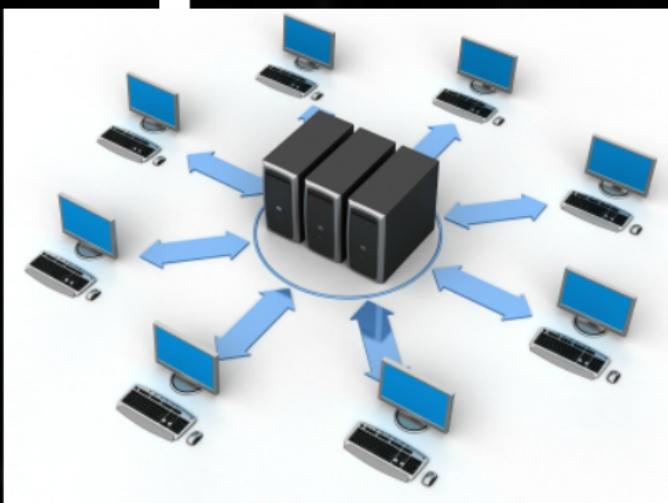
### Acciones

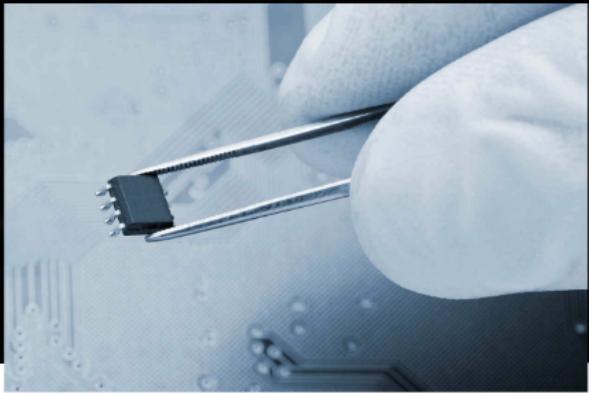
- Comer
- Dormir
- Correr

# 1980: Personal Computer

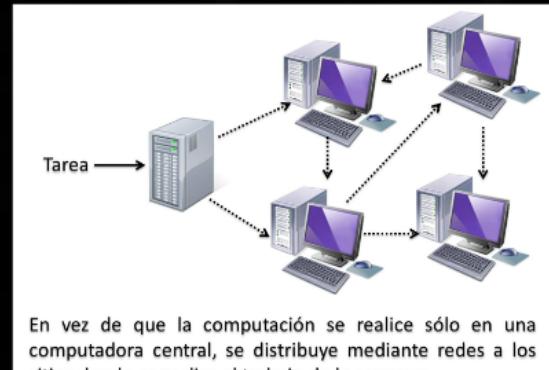


Problema: infraestructura de red no desdarrrolladas,  
Carencia de estandarización en la industria TI.  
Solución: Infraestructura de internet, modelo OSI,  
TCP/IP, cloud computing y ¿qué otras soluciones  
actuales?



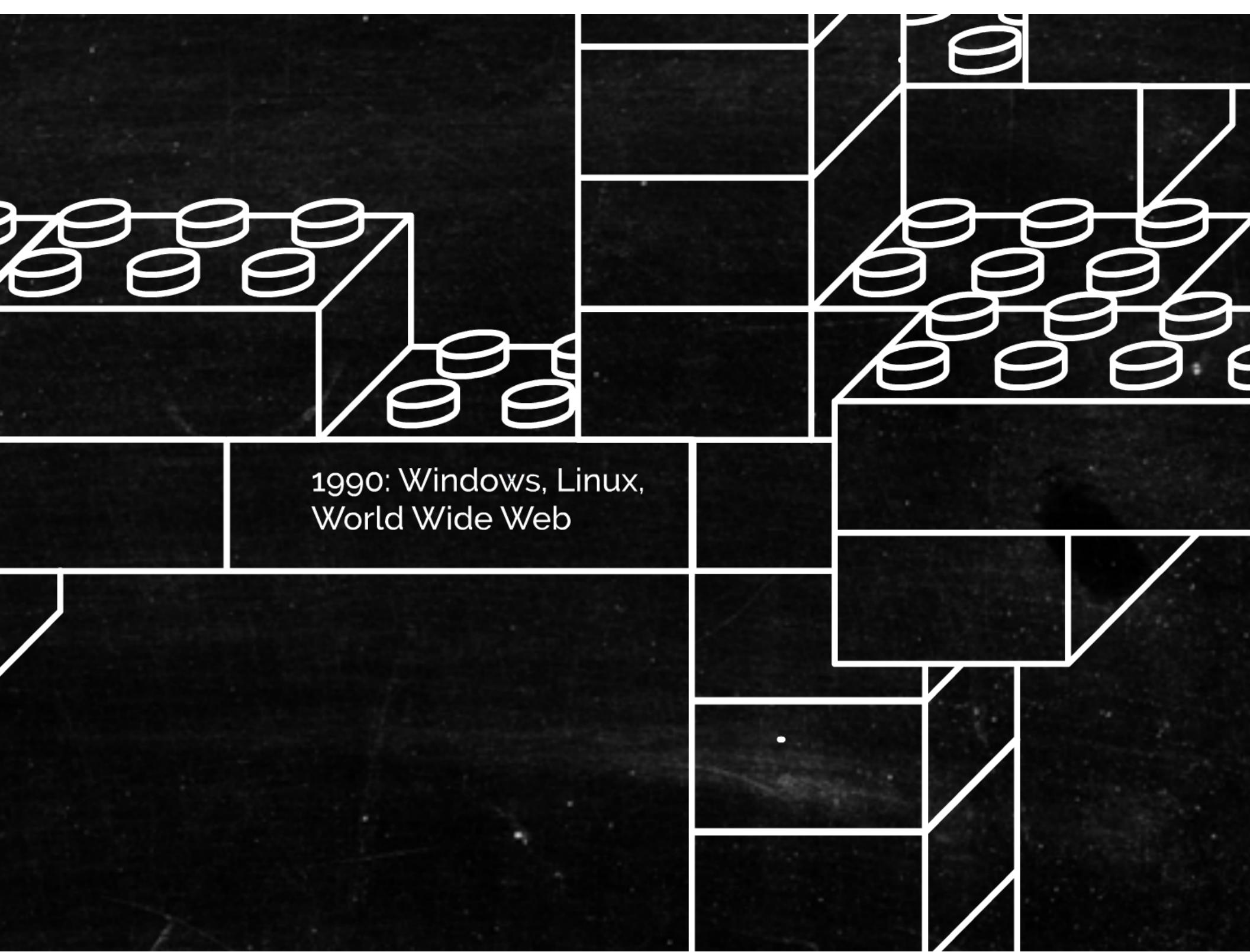


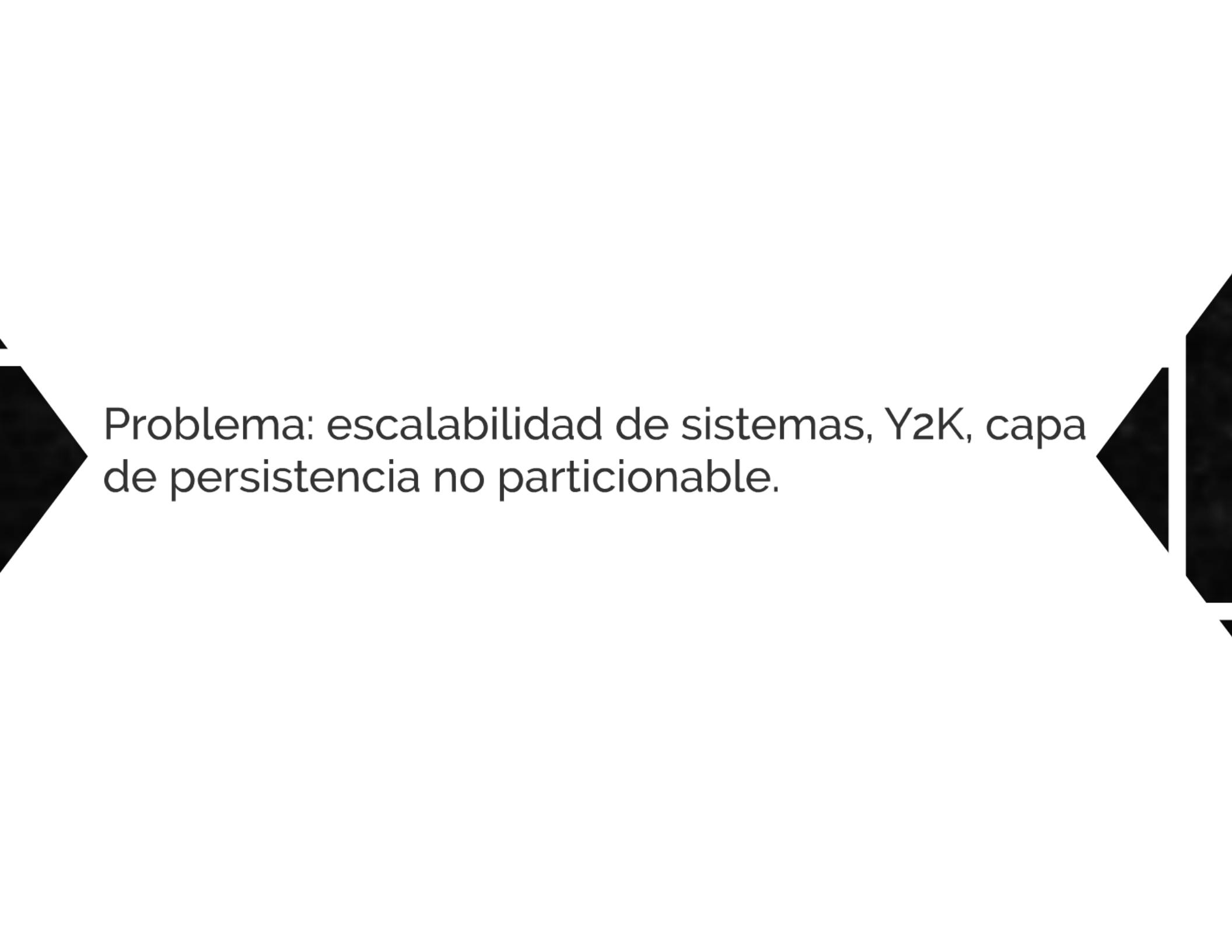
## Constatación de Ley de Moore



En vez de que la computación se realice sólo en una computadora central, se distribuye mediante redes a los sitios donde se realiza el trabajo de la empresa.

## Sistemas distribuidos





Problema: escalabilidad de sistemas, Y2K, capa de persistencia no particionable.

Solución: Linux Kernel, Hotfixes, customer support. 1994: patrones de diseño GOF



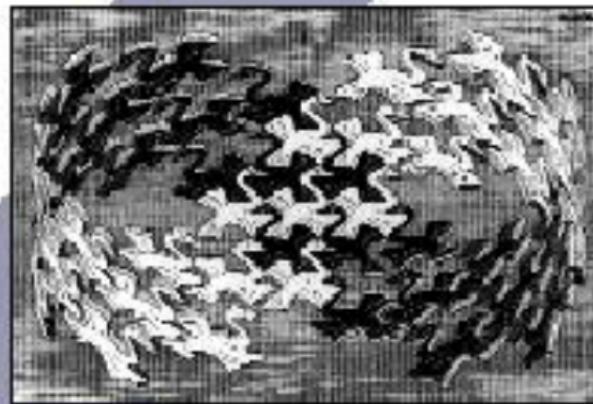


Support

# Design Patterns

Elements of Reusable  
Object-Oriented Software

Erich Gamma  
Richard Helm  
Ralph Johnson  
John Vlissides



Foreword by Grady Booch



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

2000: masificación de internet y redes sociales, cloud computing, agile manifesto.





Problemas: falta de predictibilidad del desarrollo de software, artefactos frágiles, competencia creciente y bajos presupuestos.



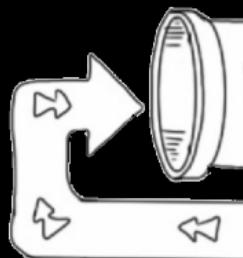
Solución: metodologías de  
desarrollo de software ágiles  
TDD, integración continua, devops,  
despliegue continuo

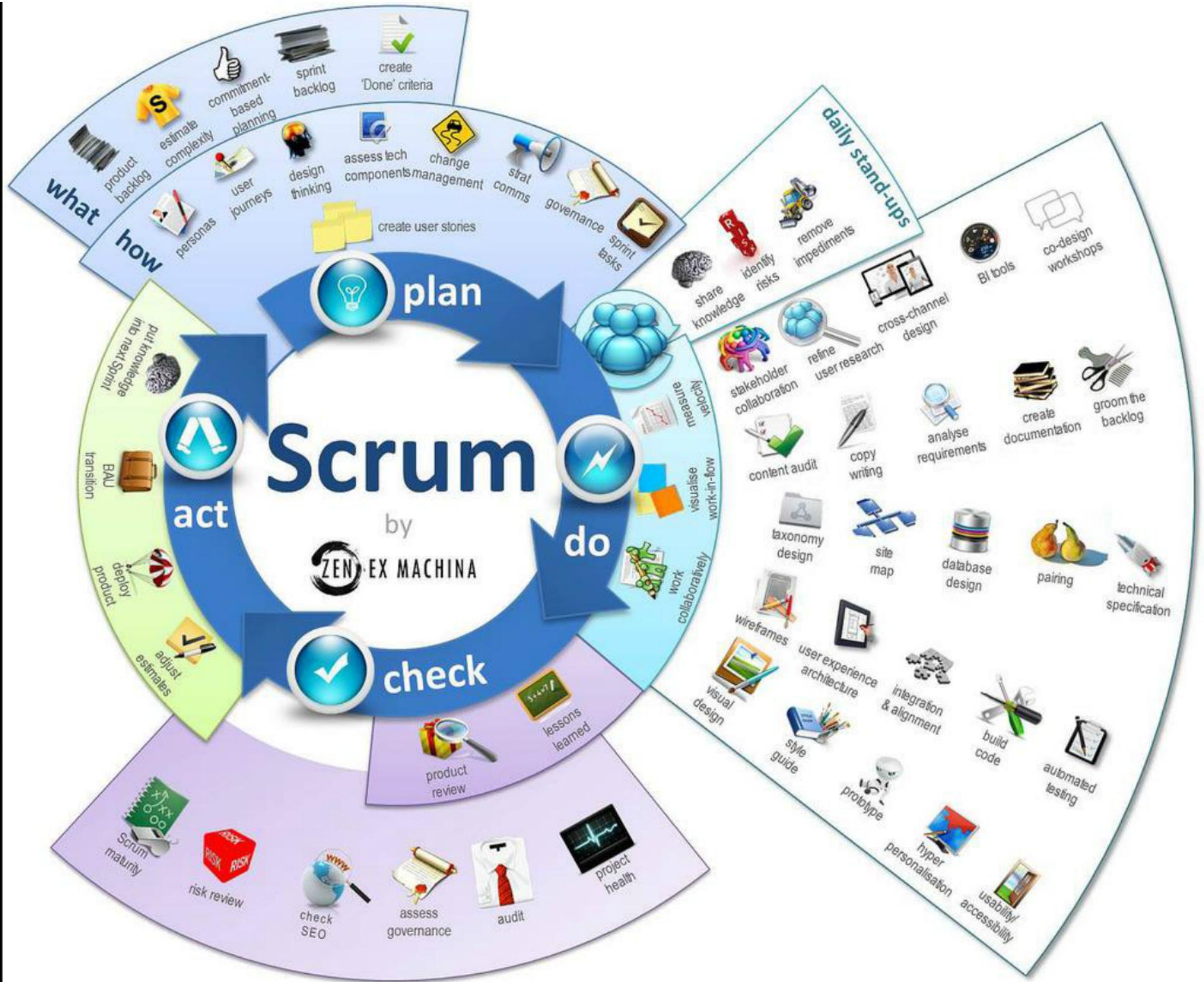
## Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

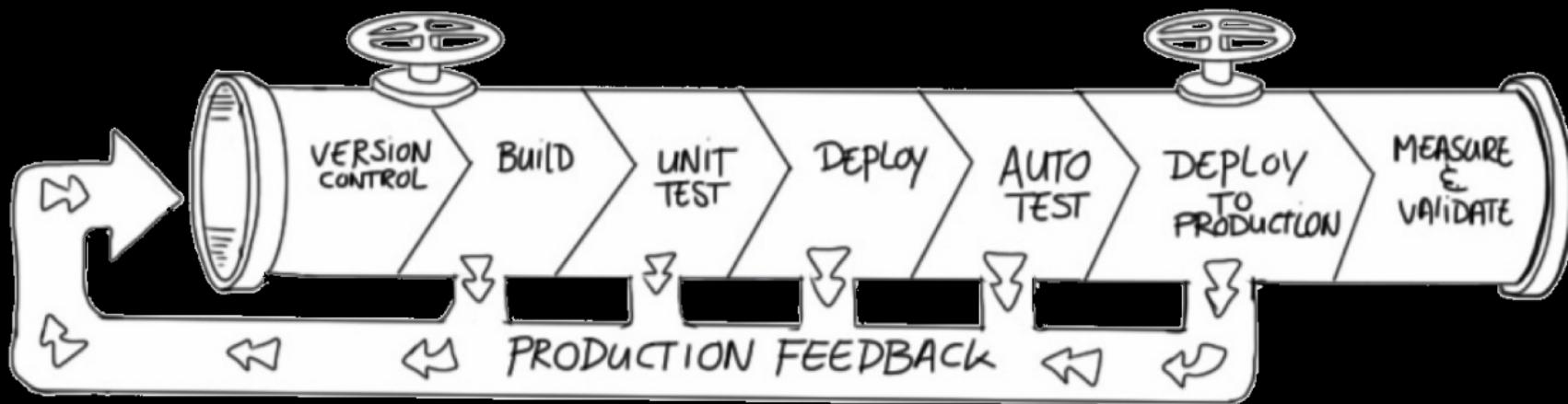
AGILE

Dep





# Deployment Pipeline



Share



Embed



Become Excellent!

 [Subscribe here!](#)

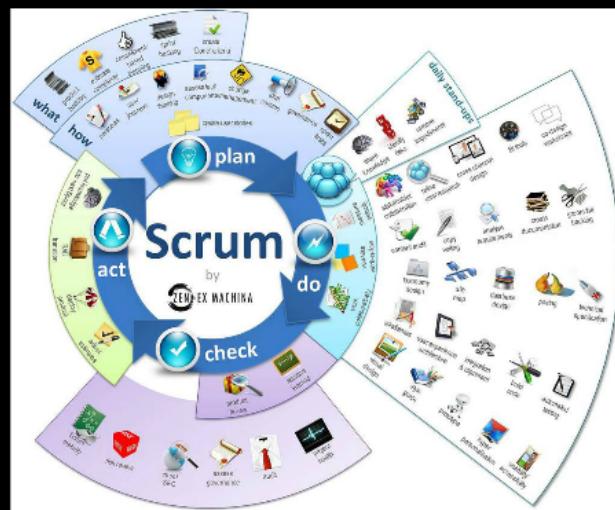
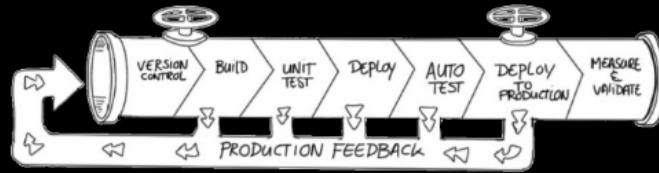


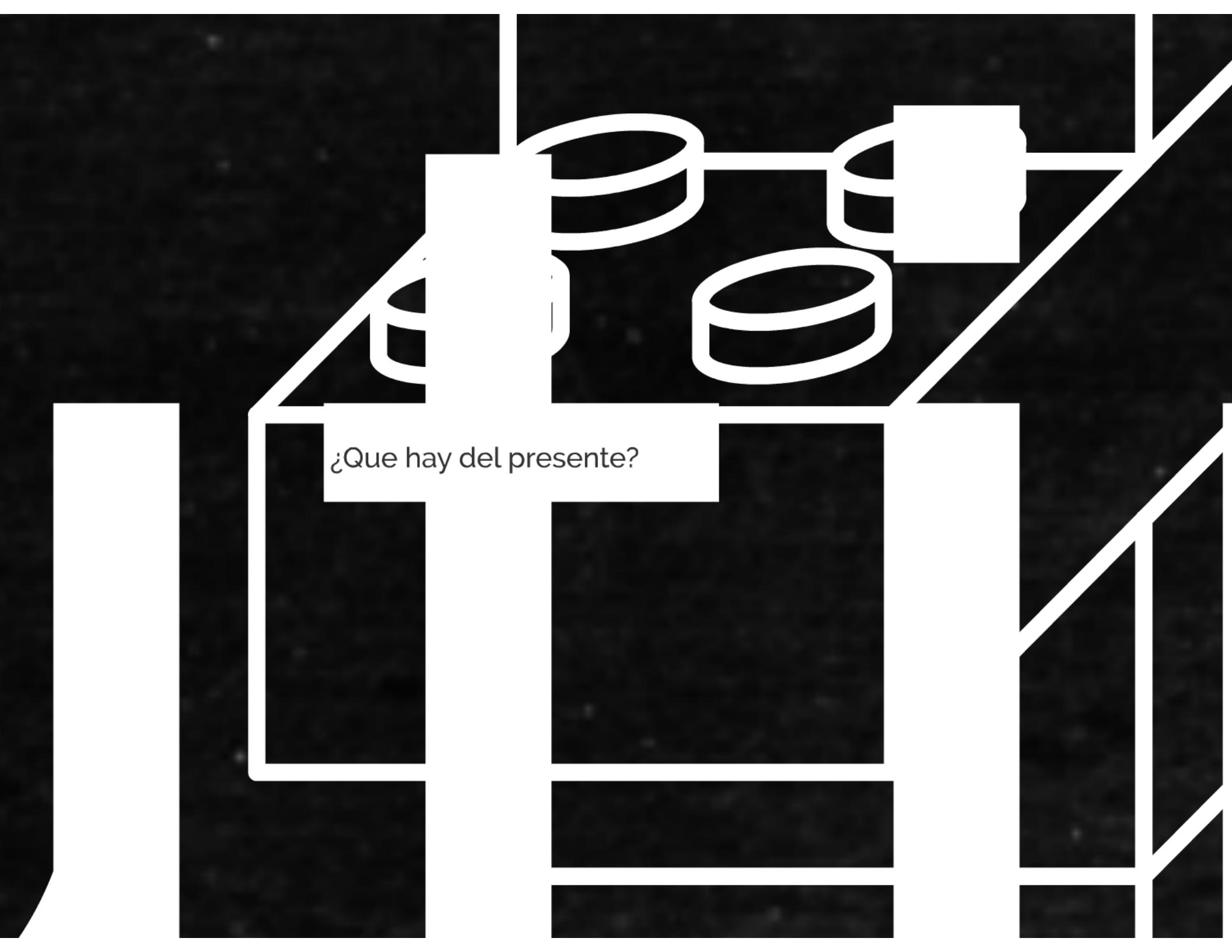
PERIODIC TABLE OF DEVOPS TOOLS™ | Xebialabs

The table includes a legend at the bottom:

- Selected
- Edited
- Deleted
- Branches & Tags
- Subscriptions

## Deployment Pipeline





¿Que hay del presente?

