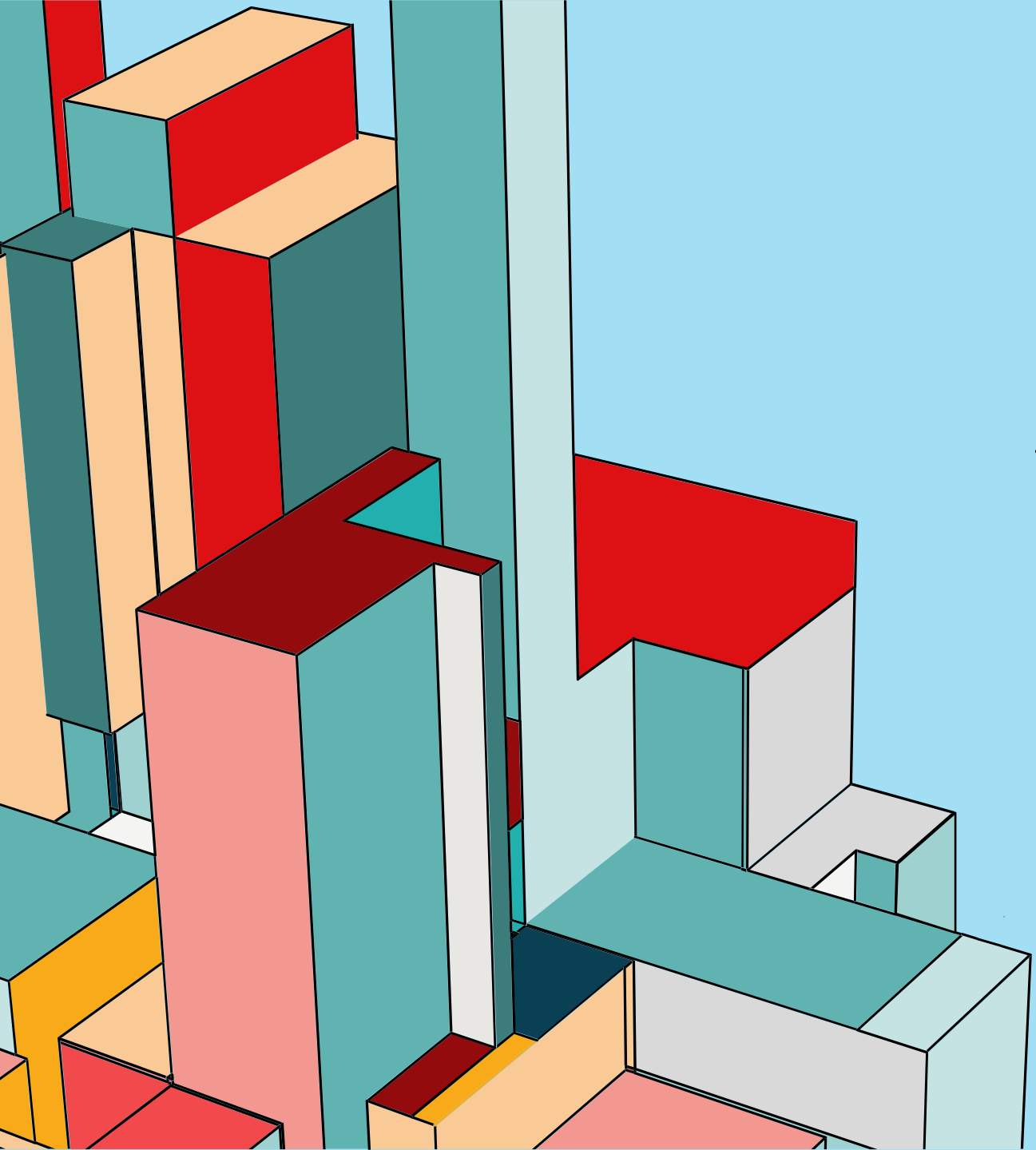
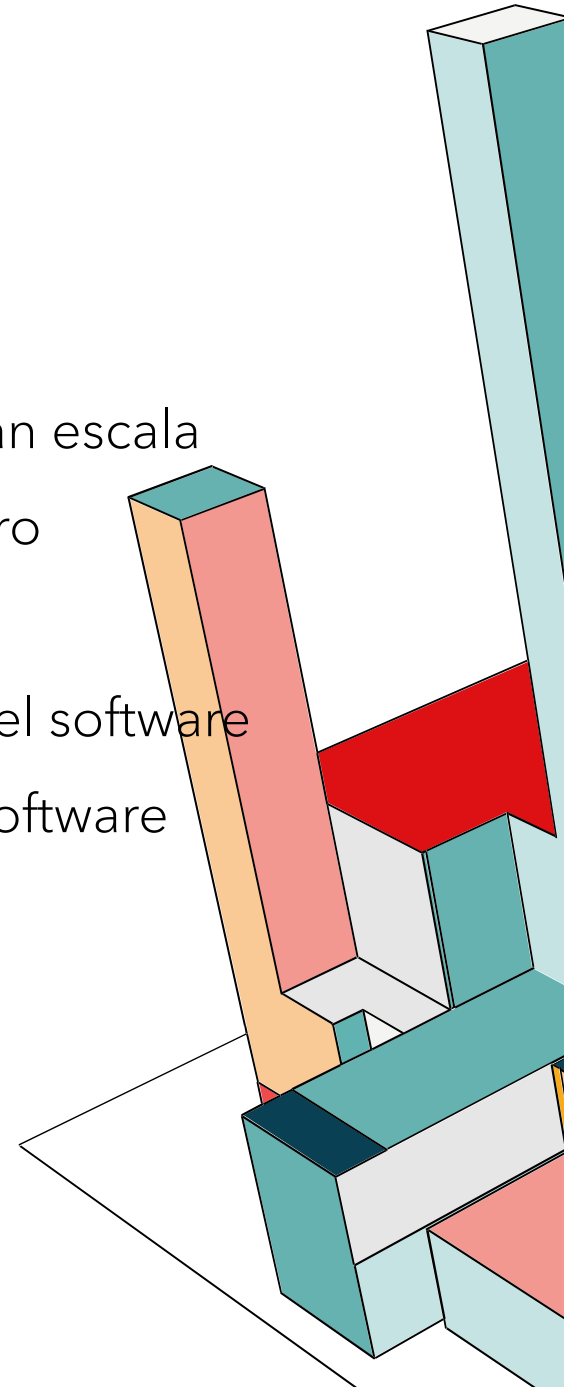


# SOFTWARE ENGENEERING AT GOOGLE

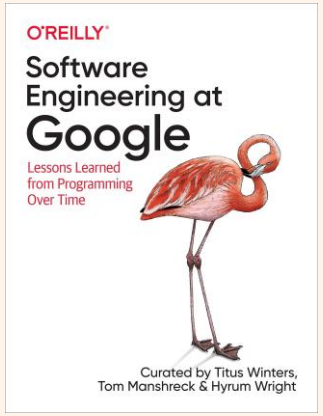
# ÍNDICE


- Haryum Wright y su libro
- El mito del genio
- Problemas en Google
- Objetivos, señales y métricas
- La inteligencia artificial
- Objetivos, señales y métricas
- BoilerPlate
- Flaky Test
- Test Double
- Code Coverage
- Cambios en el código a gran escala
- Proceso de creación del libro
- Restricciones
- El Concepto de tiempo en el software
- Consejos a ingenieros de software

# HYRUM WRIGHT

- Ingeniero de Software Senior en Google
- Líder técnico en primitivas de concurrencia C++ utilizadas en Google
- Ex catedrático en la Universidad Carnegie Mellon.
- Autor de *Software Engeneering at Google*.



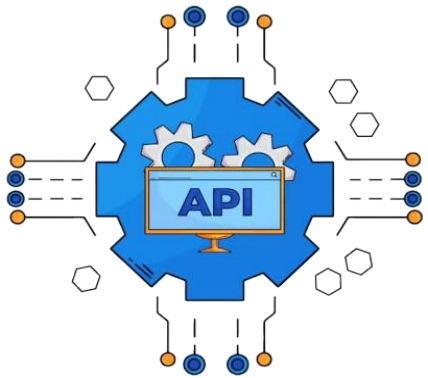

Google

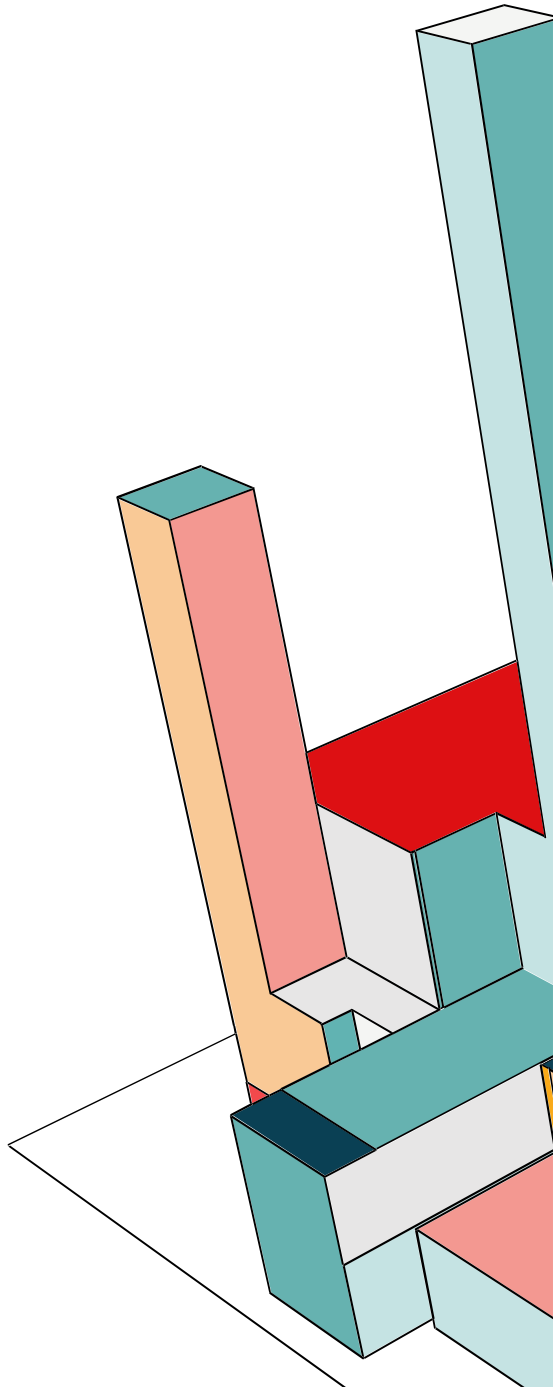
C++

# LEY DE HYRUM

💡 Con un número de usuarios suficientes para una API, no importa lo que prometas en el contrato.

💡 Todos los comportamientos observables en el sistema afectarán a un cliente.

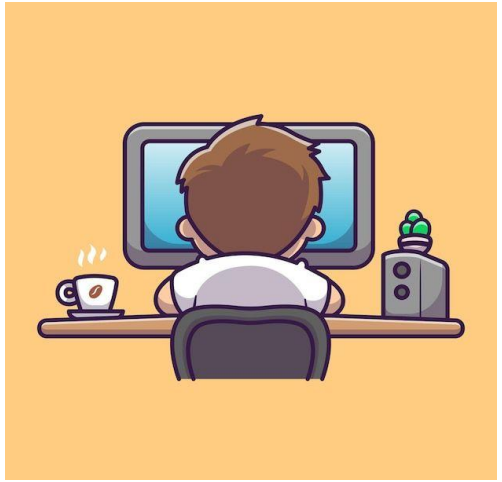


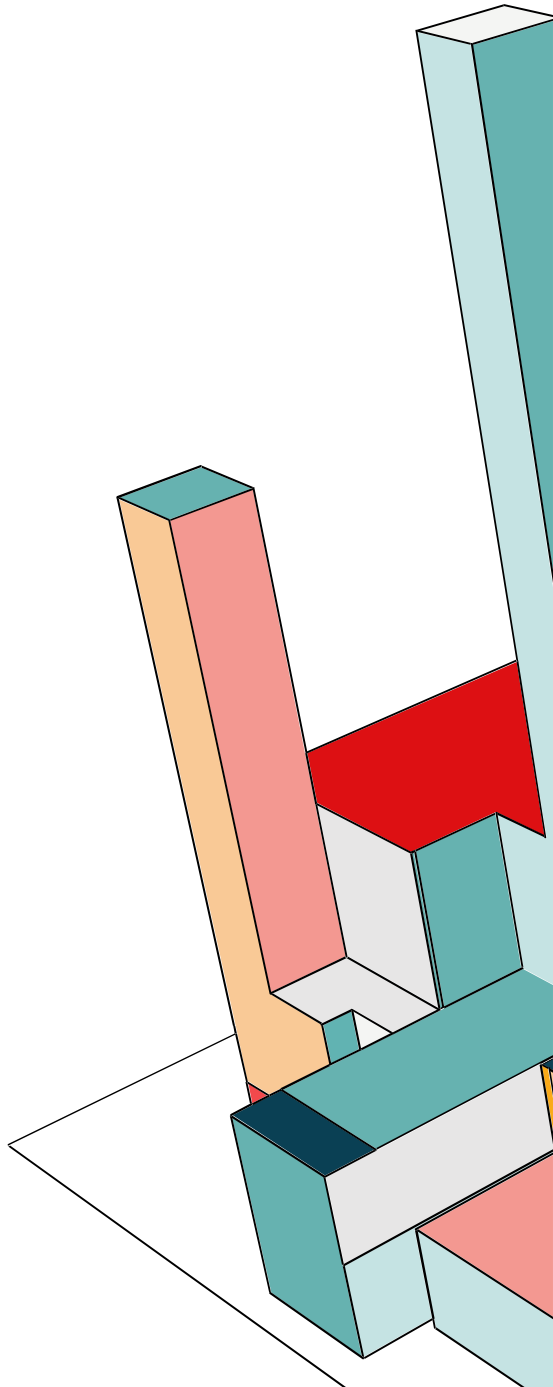



# MANAGERS TRADICIONALES VS GRANDES MANAGERS

💡 Los mánagers tradicionales se preocupan por cómo hacer las cosas

💡 Los grandes mánagers se preocupan por qué hacer, y confían en su equipo en cómo hacerlo.





# EL MITO DEL GENIO

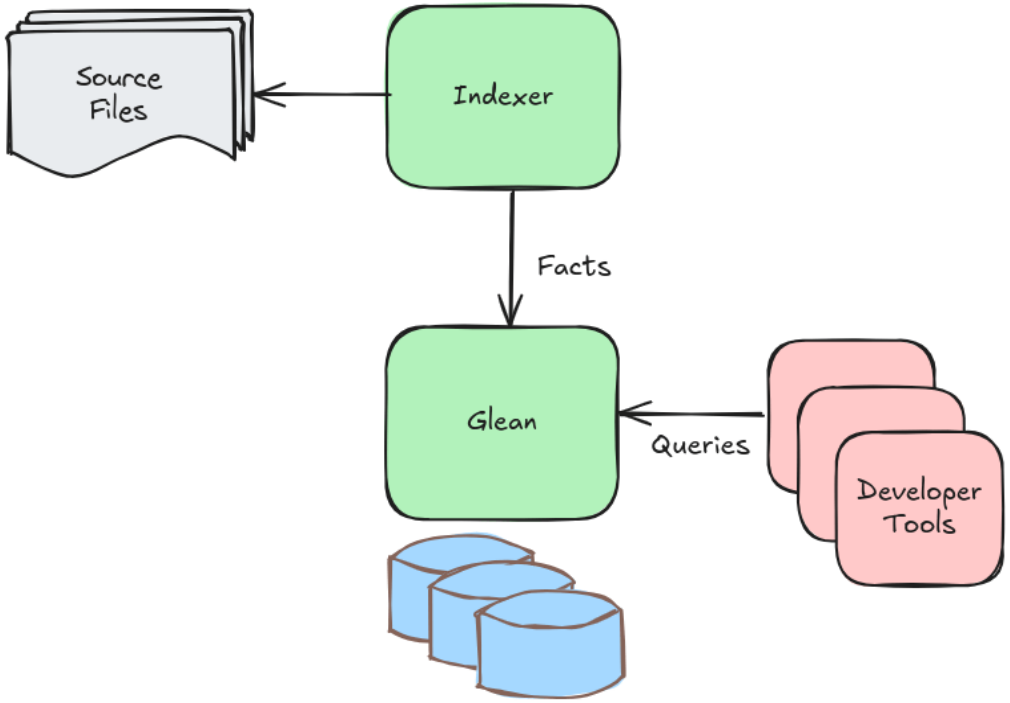
# PROBLEMAS EN GOOGLE

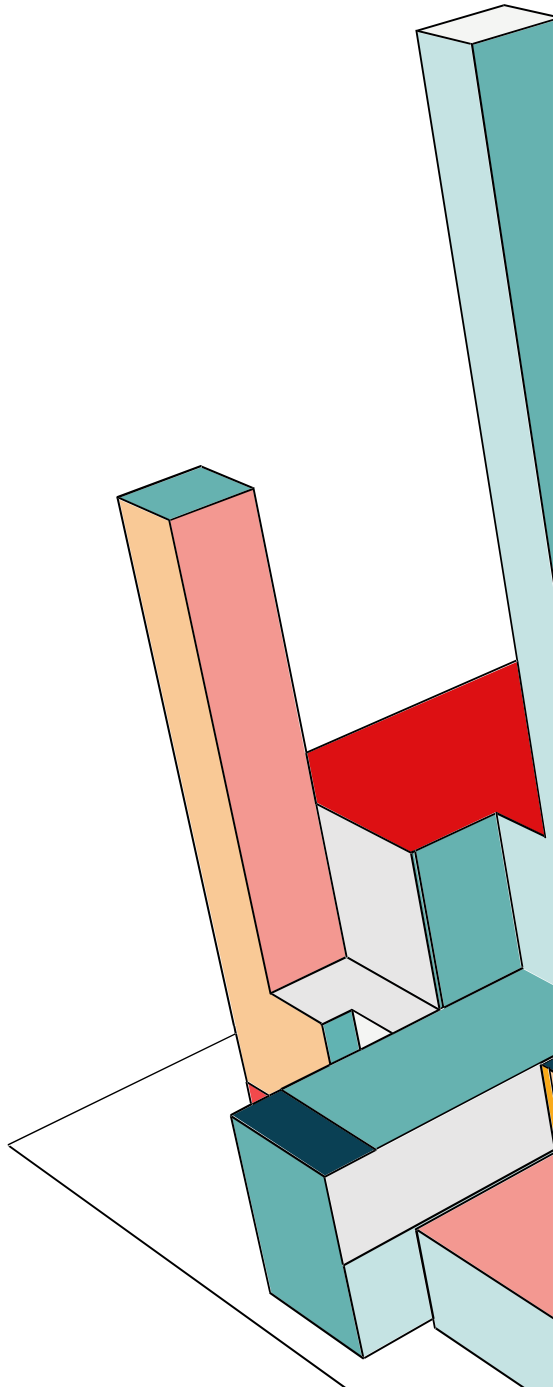
Escalabilidad

# INDEXADO DE CÓDIGO

Dos principales consumidores:

- 🧰 Herramientas de mantenimiento a gran escala
- ✏️ Ingenieros



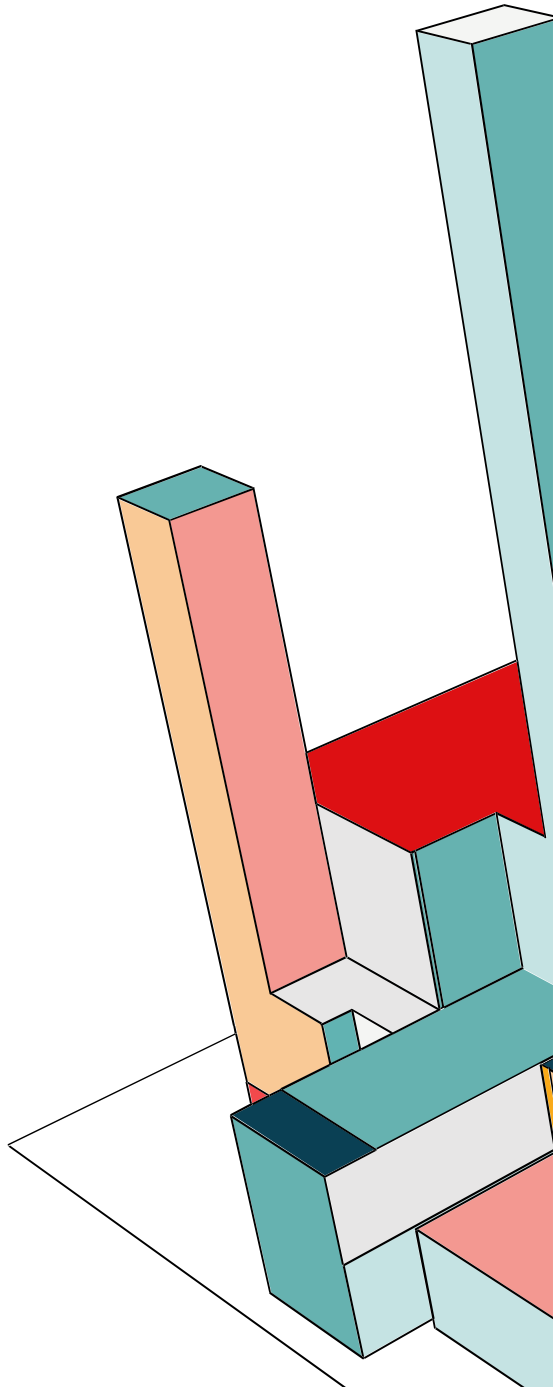

# LA INTELIGENCIA ARTIFICIAL

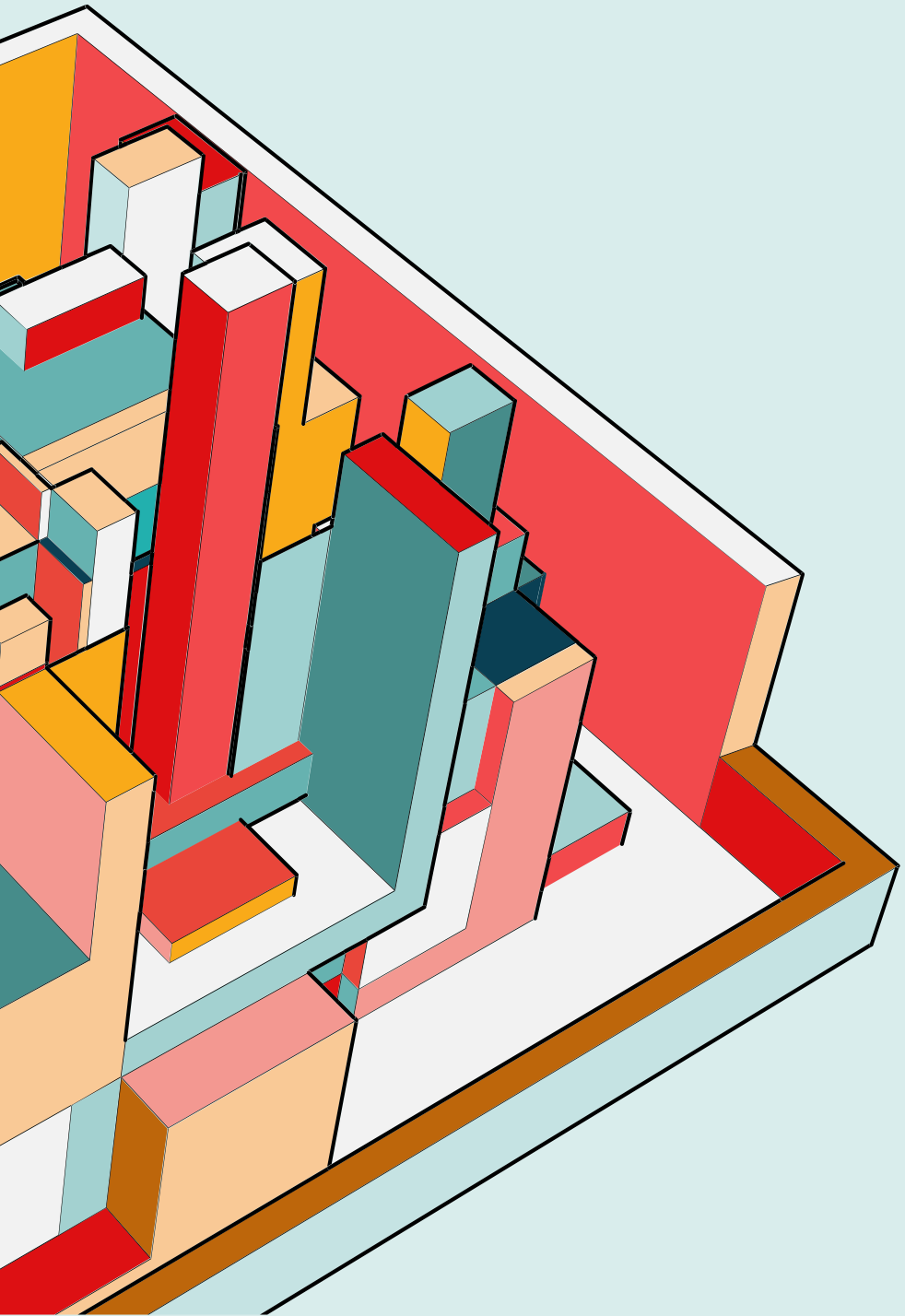
# OBJETIVOS, SEÑALES Y MÉTRICAS

🎯 **Objetivo:** Resultado esperado, aunque no siempre es fácil de definir con precisión.

🚨 **Señal:** indicador de progreso hacia el objetivo, aunque a veces sea difícil de medir.

📏 **Métrica:** un dato concreto que actúa como un "*proxy*" de la señal, y sí que puede medirse.





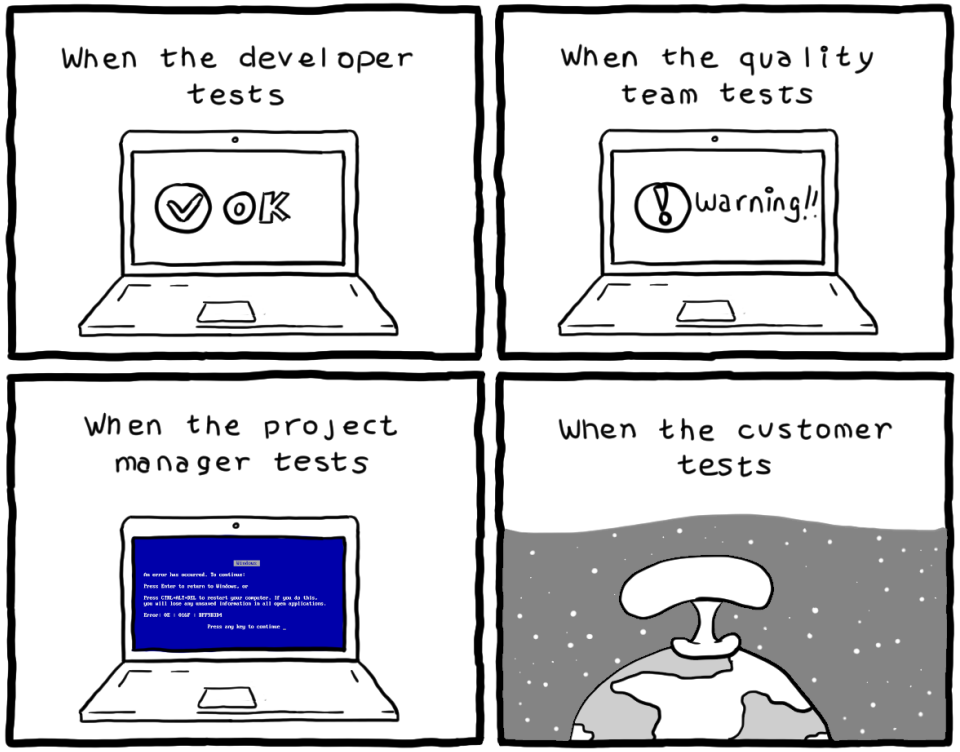
# BOILERPLATE

(Testeando lo que no es tuyo)

1. Escenario Común
2. Test extremos (buscando el fallo)

# FLAKY TEST

(Cuando los test no funcionan)




*¿Por qué debemos evitarlas?*

- Alto coste Computacional
- Alto coste Humano

# TEST DOUBLE

(Usando mocks y fakes)

# CODE COVERAGE

- No garantiza pruebas efectivas

¿Y cómo se puede saber?

Con los Mutation Test

# ¿CÓMO GOOGLE ANALIZA EL CÓDIGO?

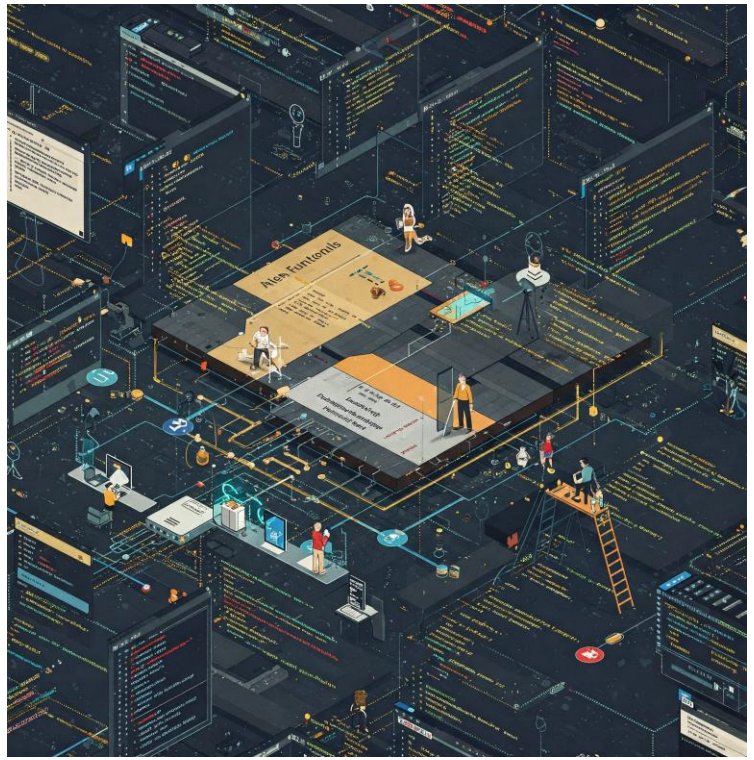
- Análisis Manual
- Tricorder

# CAMBIOS EN EL CÓDIGO A GRAN ESCALA

No se pueden hacer de manera atómica.

No siempre es posible automatizar estas tareas.

Plataforma Busy beavers.



# PROCESO DE CREACIÓN DEL LIBRO

No caer en el mito del genio.

# RESTRICCIONES

Como ingenieros queremos libertad en las elecciones.

Esto conlleva una menor escalabilidad.

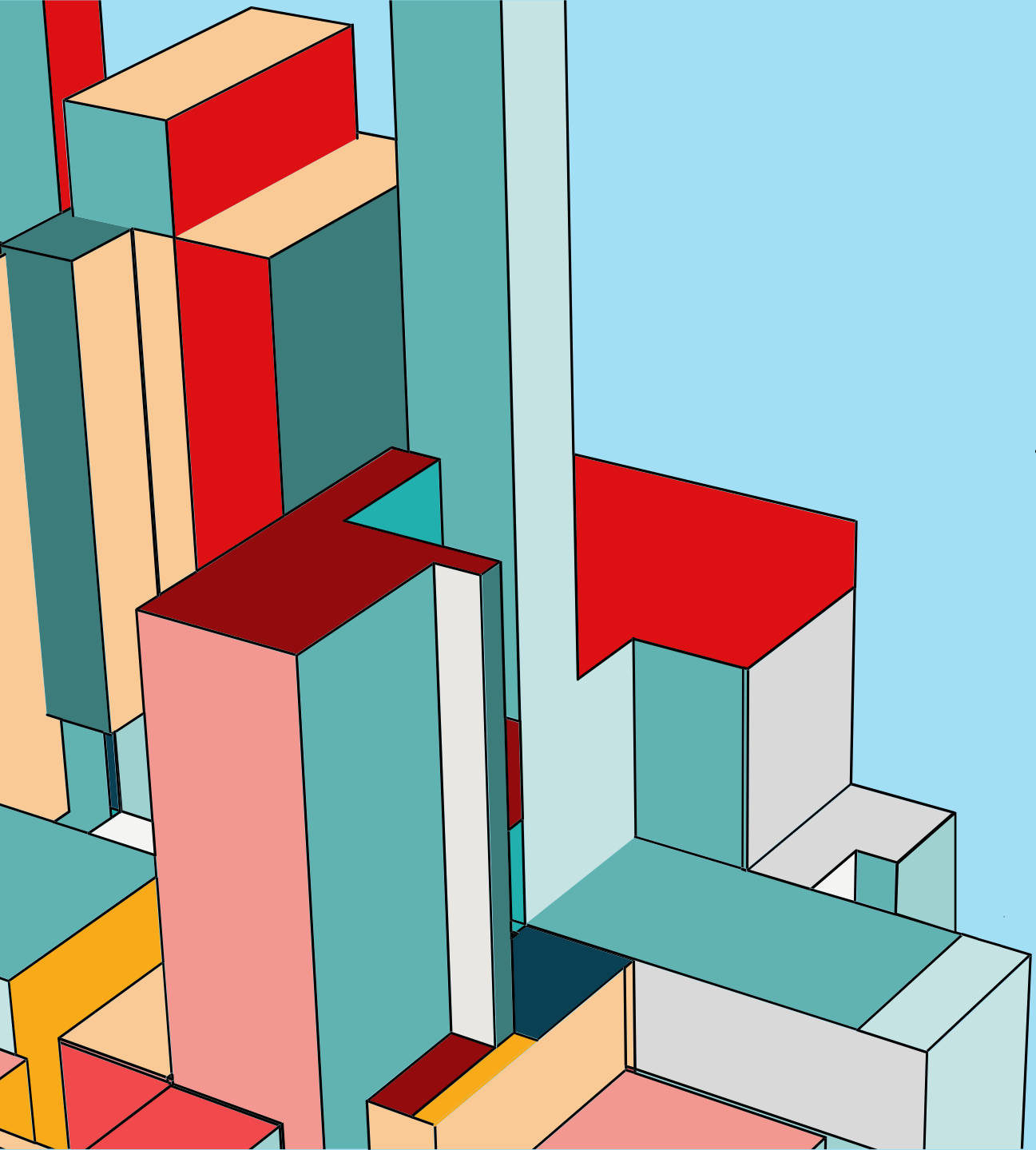
# EL CONCEPTO DE TIEMPO EN EL SOFTWARE

Cuanto va a utilizarse un programa.

Diseñarlo en base a su longevidad.

# CONSEJOS A INGENIEROS DE SOFTWARE

- Ingenieros Junior
  - Diseñar
  - Reuniones
  - Testear
- Ingenieros asentados
  - Ecosistema
  - Testing
  - Revisión de código



# SOFTWARE ENGENEERING AT GOOGLE


- Alejandro Vega García UO294615
- Adrian Dumitru UO295652
- Jorge Blanco Sánchez UO293697

# ¿PREGUNTAS?