

# Tidy first?

Kent Beck



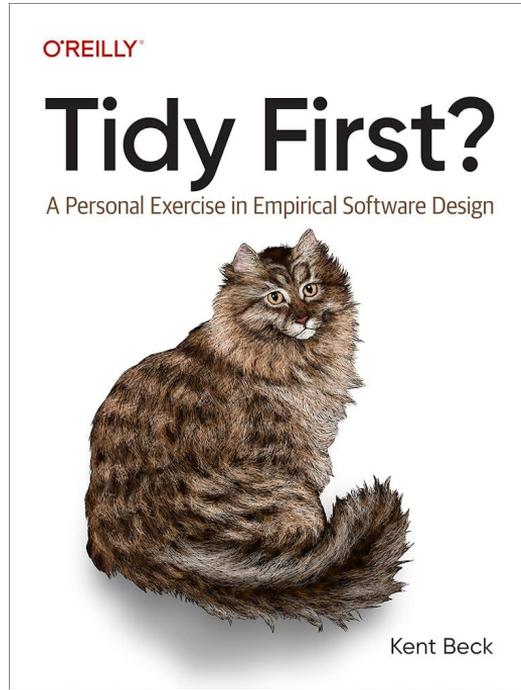


# Kent Beck



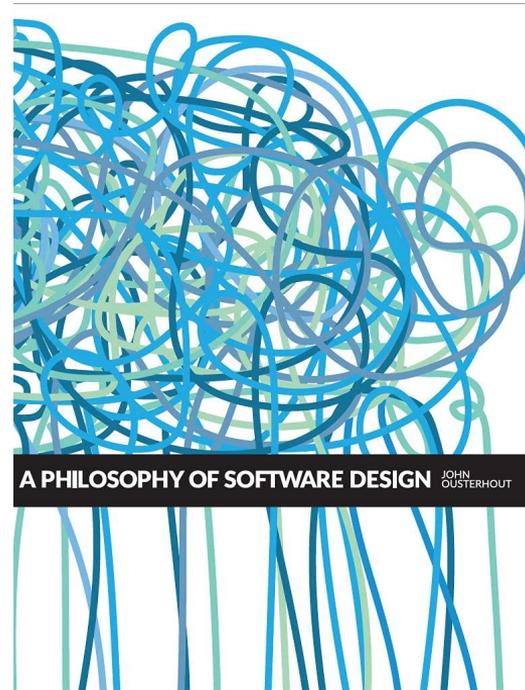
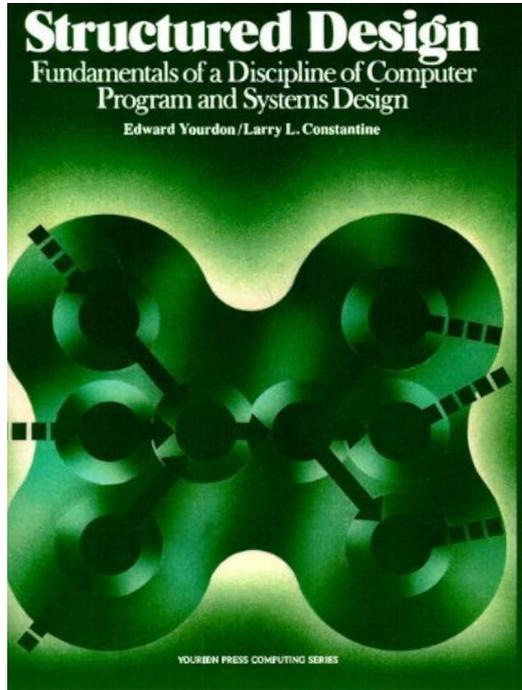
- Programación Extrema (eXtreme Programming)
- Test Driven Development

# Tidy first?



- Autor: Kent Beck
- Publicado por: O'Reilly Media
- Fecha de salida: 2023

# Tidy first?

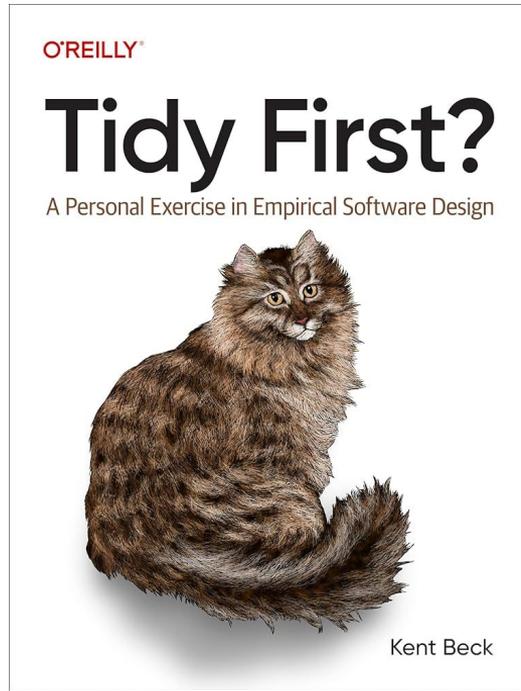




# La mentira sobre las métricas de diseño

- Métricas sacadas de contexto
- El carácter natural del desarrollo de código
- Variabilidad según el entorno en el que se desarrollan.

# Conceptos principales



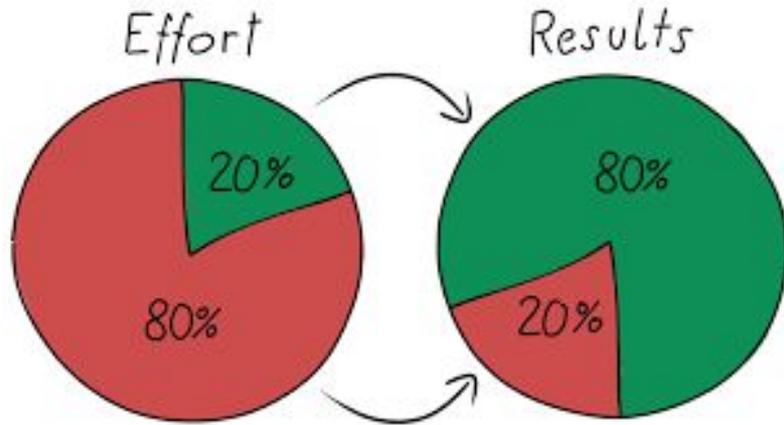
- Acoplamiento
- Cohesión
- *Tidyings*



# Diseño de software empírico

- Pequeños cambios acumulativos
- Diseño basado en evidencia
- Separación de cambios estructurales y funcionales

# Principio de Pareto



- Regla del 80 / 20
- Aplicación en el desarrollo software



# Refactorización incremental o grandes refactorizaciones

- Confianza en el equipo
- Cambios visibles
- Refactorizar de forma incremental



## **Cambios estructurales**

- Organización del código
- Fácilmente reversibles

## **Cambios de comportamiento**

- Funcionamiento de la aplicación
- Complejos de revertir



# Miedo a la desorganización

1

## El Temor

Un diseño desorganizado es difícil de llevar a producción.

2

## La Solución de Beck

Una buena organización permite un diseño incremental.

# Restricciones de la industria

1

## Supervivencia del Software

Diseño extenso retrasa la obtención de ingresos.

2

## Valor del Dinero en el Tiempo

Ingresos tempranos permiten inversión y crecimiento.





## ¿Cuándo planificar a largo plazo?

1

Costo inmediato

2

Beneficios futuros

3

Decisión compleja



# El coste del acoplamiento: la equivalencia de Constantine



## Valoración

Encontrar un equilibrio es esencial.



## Minimización

El costo total del software en el tiempo importa.

# Conclusión





**Dudas**