

Software como una disciplina de ingeniería

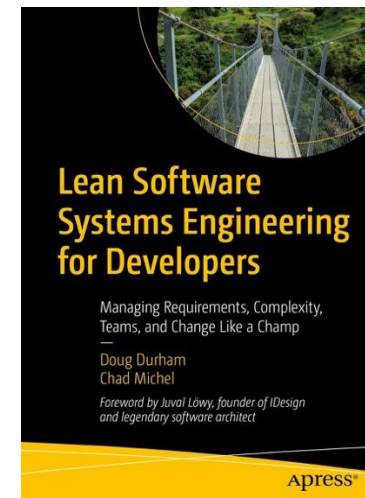
EQUIPO 2





Chad Michel

- Software Engineer desde 2003
- Lean Software Systems Engineering for Developers





Ingeniería de Software vs Desarrollo de Software

Complejidad en el software



OBJETIVA

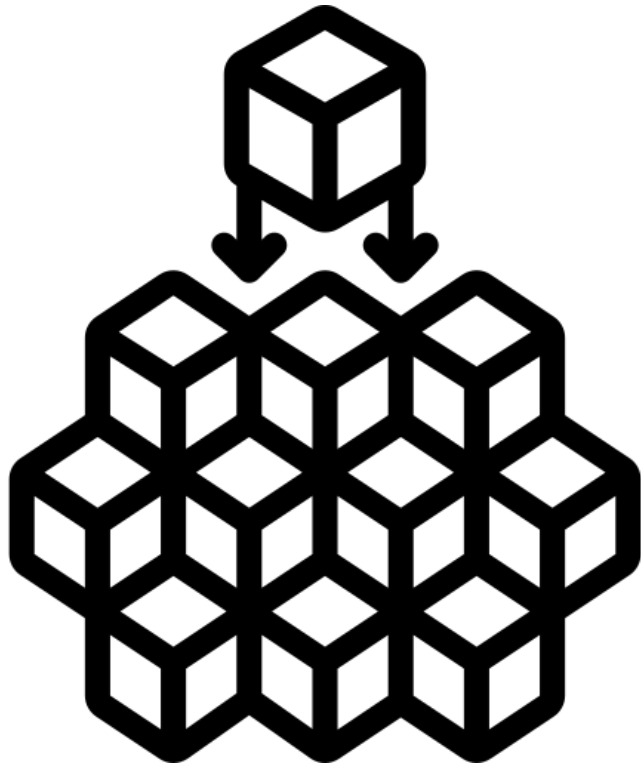


REQUISITOS



SOLUCION

Diseño y agilidad



¿Cómo hacer un buen diseño?



Según
Chad
Michel

Entender qué quiere el cliente

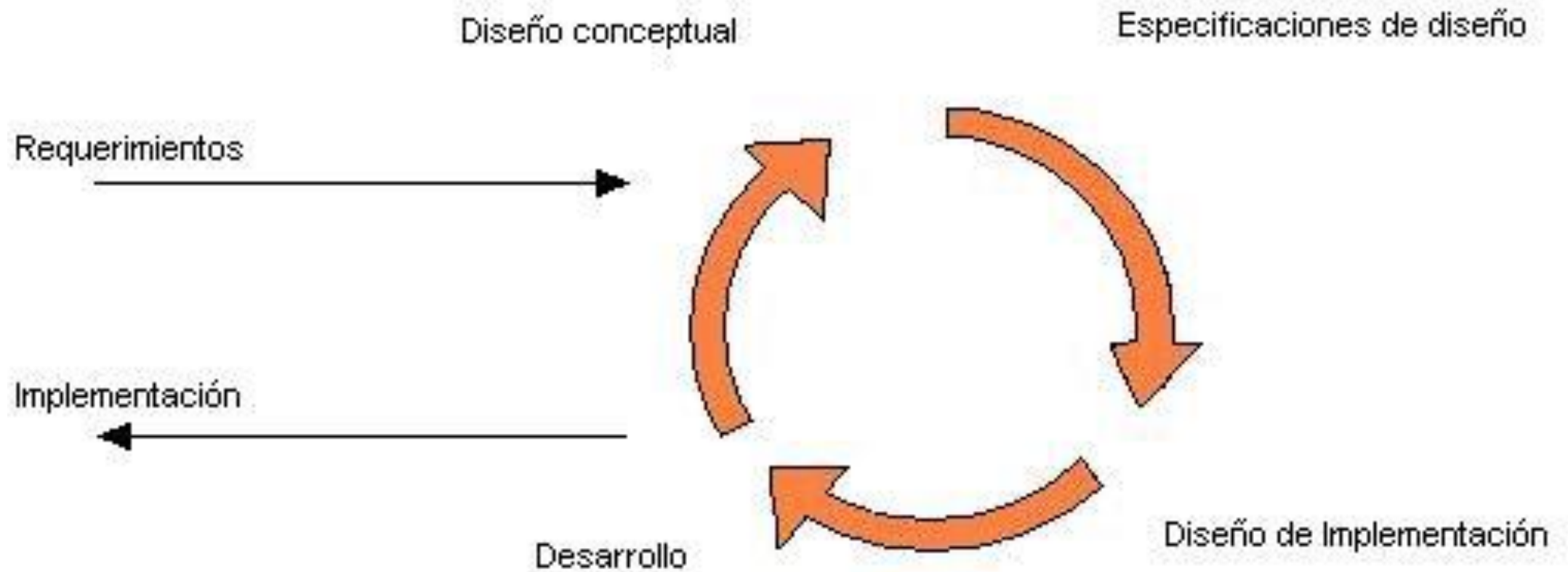


Establecer Requisitos



Conocer como el cliente espera que evolucione su organización

Diseñar para el cambio





Pero

¡CUIDADO!

NO HIPERDISEÑES

ERRORES a evitar

1. Buscar la perfección. ~ Lo perfecto es enemigo de lo bueno ~.
dice el entrevistado : ~ Si me dieras 1 año para construir algo
sería perfecto e inconstruible ~
2. No detectar errores tempranamente. Según se avanza en el
proyecto el coste de arreglar un error crece exponencialmente.
*1000 en la implementación
*10.000 cuando está en uso
3. Hacer muchos cambios, naturalmente debido a un mal diseño
~ Si estoy constantemente tirando y rehaciendo paredes de un
edificio antes o después se vendrá abajo ~

Reworking

Cambiar las pruebas y las abstracciones.

Volver a trabajar sobre el proyecto debido a un mal diseño.

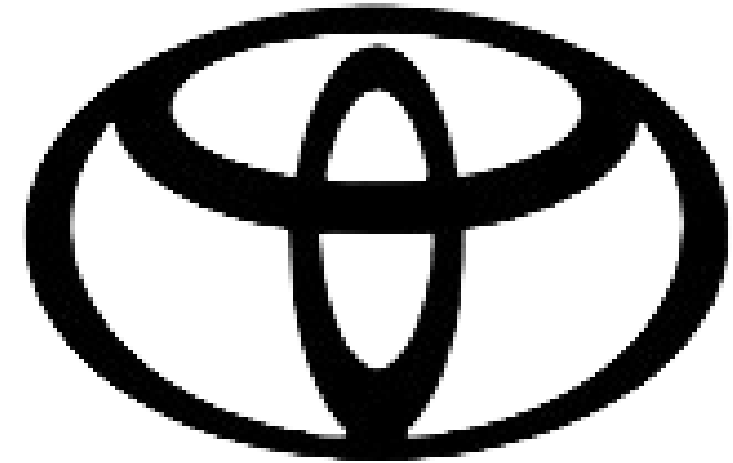
VS Refactoring

VS Cambiar los algoritmos.

VS Ampliar su funcionalidad.

Control de calidad vs Garantía de calidad

Calidad Institucionalizada y Andon Cord



Todo el mundo debe aportar a la
calidad del producto.

El Chief Engineer

- Cohesiona el grupo
- Apoya y aconseja
- Ayuda al grupo a entender la idea general del proyecto

Evolución de las Expectativas en Software

- Cada vez más exigentes.
- Creciendo exponencialmente.

