

Distinguir entre ***Problema y Solución***

Carlos Blé
@carlosble



CODESAI

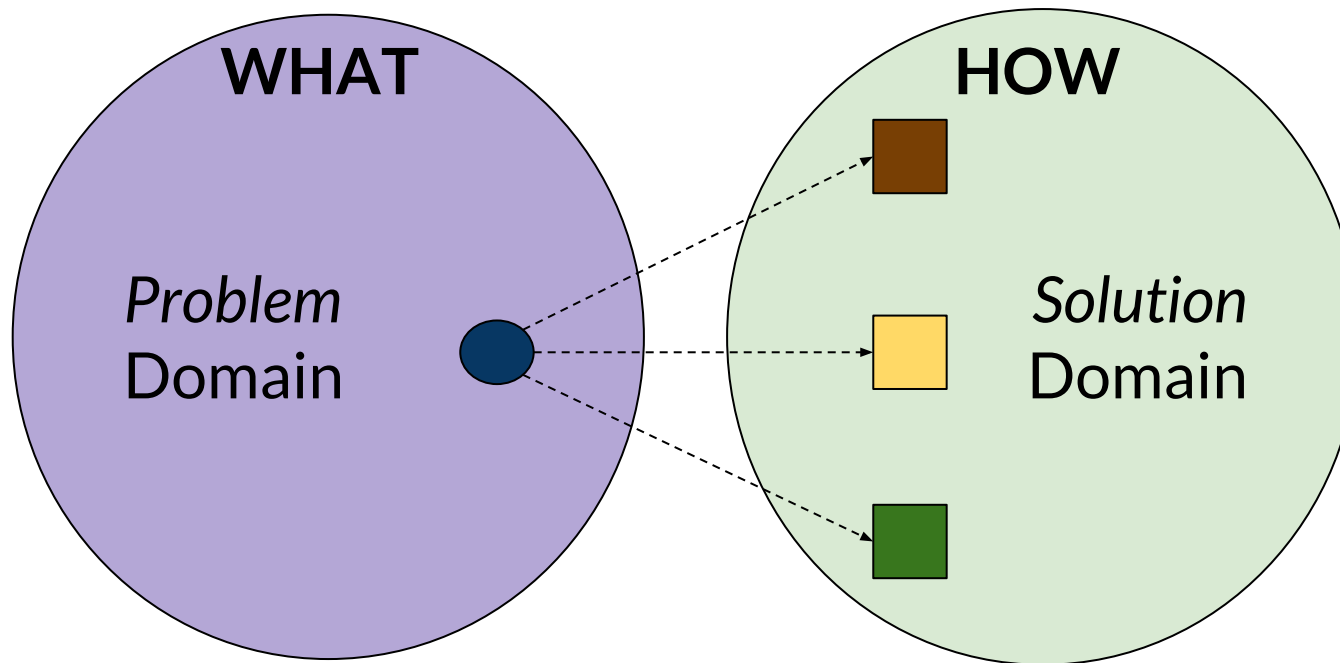


[instagram.com/carlos.ble](https://www.instagram.com/carlos.ble)



CONTEXT

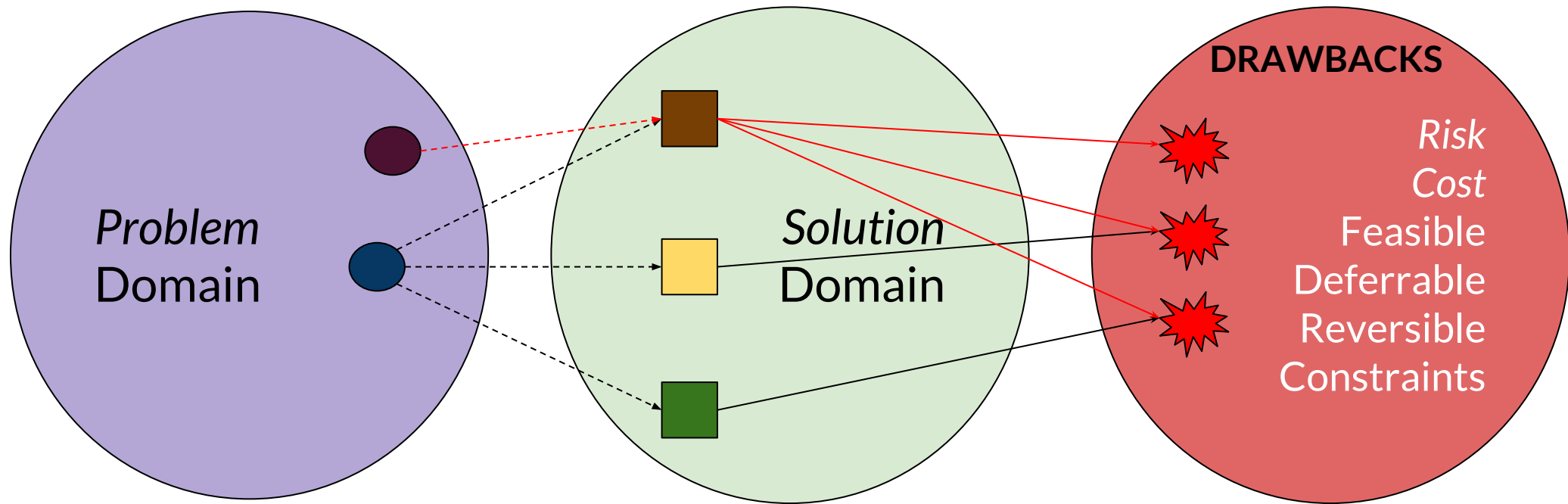
Necesidades VS Medios





@carlosble

Inconvenientes



Demasiados subdominios combinados

Objetivos

- *Explicar nuestros servicios*
- *Expresar nuestros valores*
- *Centralizar publicaciones*
- *Comentarios de los lectores*
- *Leer desde el móvil*
- *En inglés y en castellano*
- *Que nos encuentren en Google*
- *Dar ejemplo de software bien construido*

¿Solución?

Sitio Web

Pregunta para tí

¿Problema o solución?

Build the right product + Build the product right



How the customer explained it



How the project leader understood it



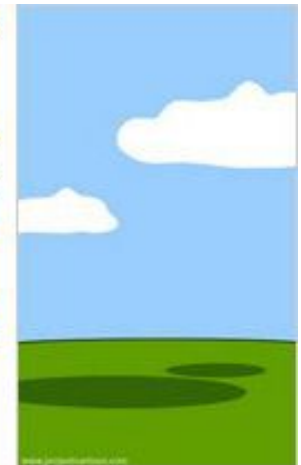
How the analyst designed it



How the programmer wrote it



How the business consultant described it



How the project was documented



What operations installed



How the customer was billed



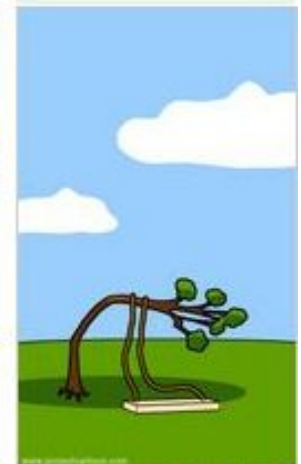
How it was supported



What marketing advertised

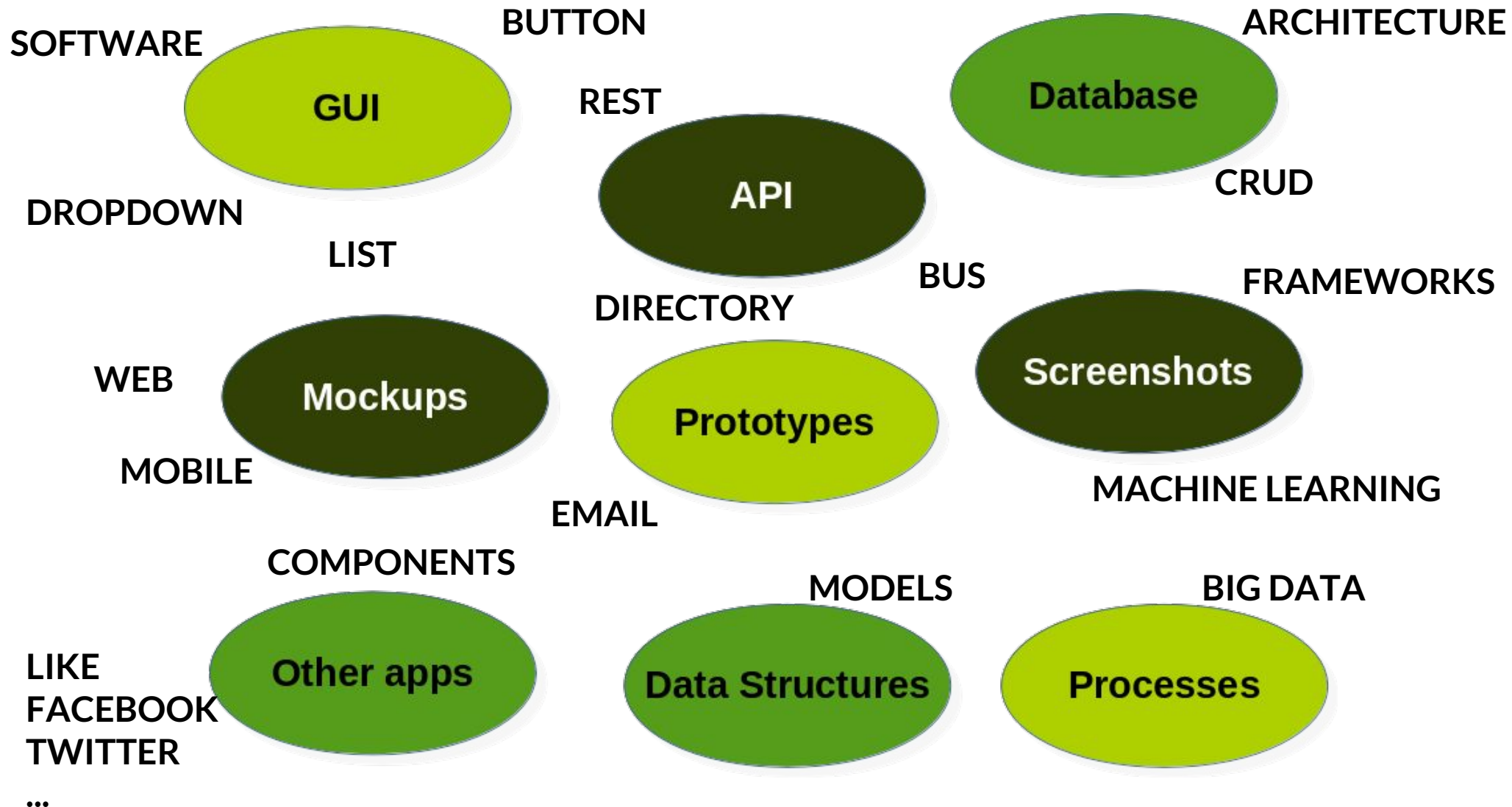


What the customer really needed



The disaster recover plan

Dominio de la solución



Solucionólogos



Ejemplo: cómo trabajamos

Objetivos / Necesidades

- Diferente sueldo
- Diferente tarifa
- Trabajar juntos
- Explicar cómo trabajamos
- Transmitir nuestros valores
- Consistencia en todos los canales de comunicación

Soluciones

- Junior/Senior
- Craftsman/Journeyman/Apprentice
- Sin categorías
- Definir nuestros principios y valores
- Parejas

Inconvenientes

- Prejuicios
- No encaja
- Competición entre nosotros
- Explicar por qué parejas

Pensamiento lateral:

- El problema es el mercado


- Cambiar de mercado

Pros & Contras

Beneficios

- Coste fijo y único
- Balanceamos según necesidad
- Trabajar juntos
- Fieles a nuestros valores
- Poder rotar

Solución



**Trabajar
en Pares**

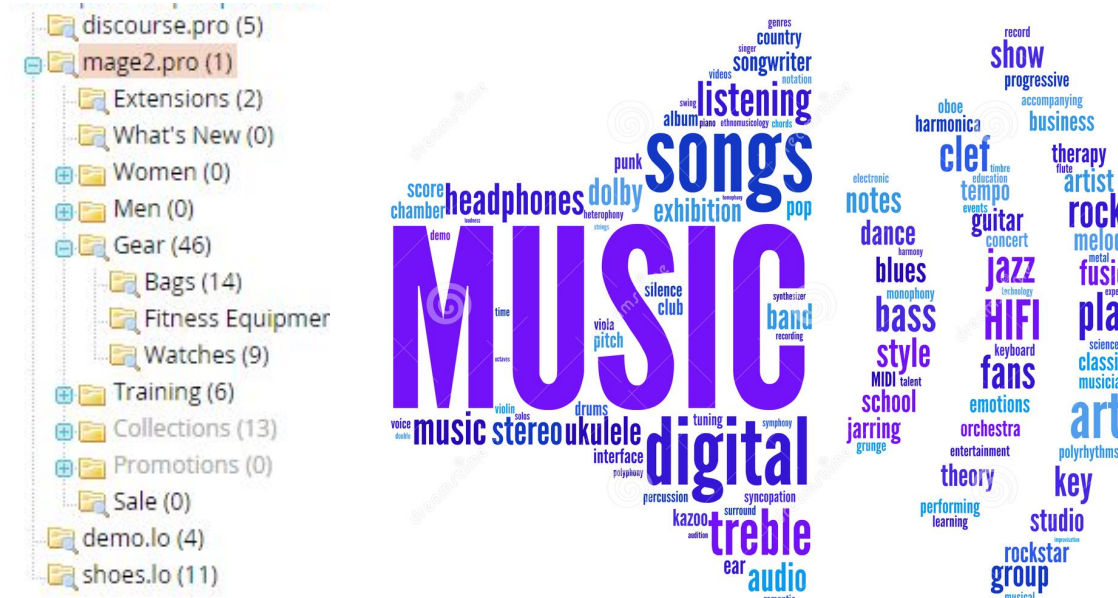
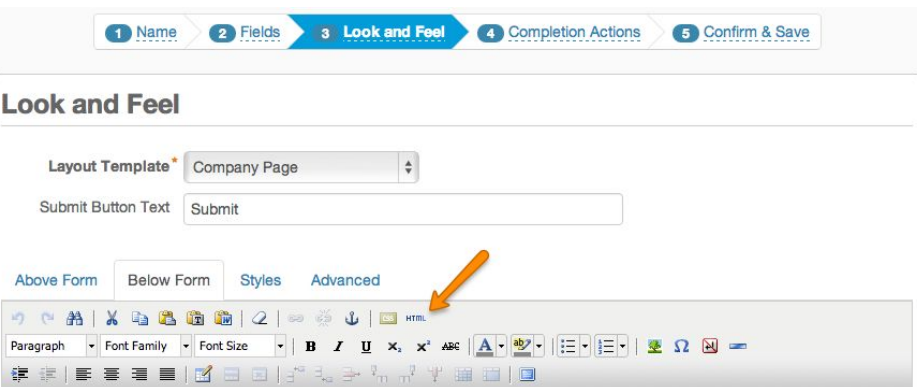
Inconvenientes

- Prejuicios pair programming
- Cuando sólo disponemos de una persona
- No llega el presupuesto

Análisis del problema

- Conocer el contexto
- ¿Por qué? ¿Para qué?
- Specification by Example - BDD
- Ingeniería Inversa
- Event Storming
- Design Thinking
- Impact Mapping
- Pensamiento lateral
- Seven Product Dimensions
- Spikes & Prototypes

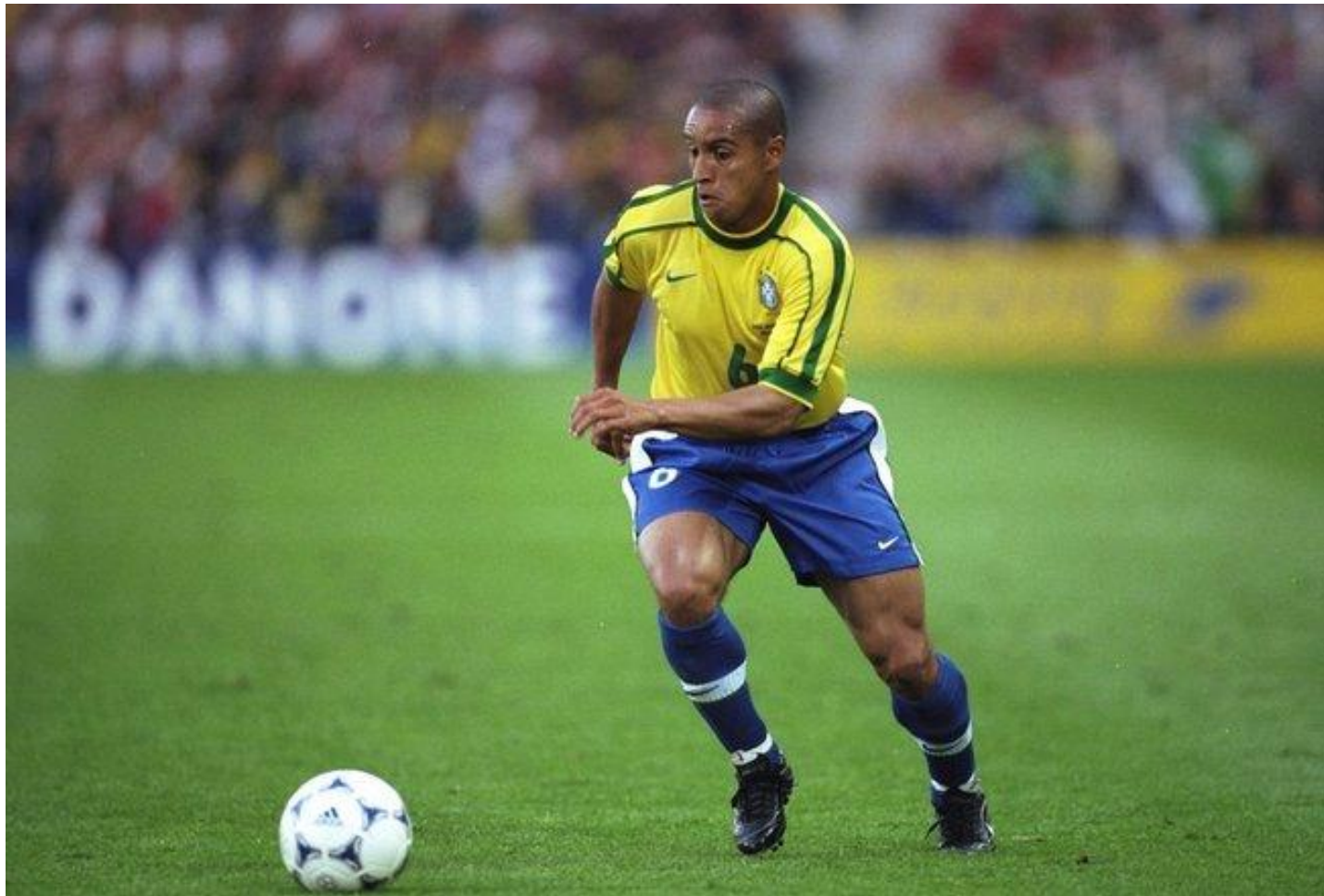
Simple, Open for Change, Learning-oriented, Valuable, Empowering



Posponer decisiones

Evitar complejidad accidental

~~Restricciones / Optimizaciones~~



Principios Lean

- Eliminar desperdicio
- Ver el Todo
- Aumentar el aprendizaje
- Decidir lo más tarde posible
- Apoderar al equipo
- Calidad desde el inicio
- Entregas rápidas

"Think big, act small, fail fast; learn rapidly"

- Mary & Tom Poppendiek

Historias: problema + solución

Balance workload in the garage

As garage organiser

I want to assign jobs to mechanics

To balance the workload in the garage

Acceptance Criteria:

- Jobs are assigned to a single mechanic on a particular day – time is irrelevant.
- A job may be given priority over other.
- Ongoing jobs can't be assigned from one mechanic to another.

Criterio de aceptación != Escenario

Background:

"Paco" is a mechanic

Scenario: Assign job to mechanic

Given *the job "Change pad brakes" is in the unassigned work queue*

And *"Paco"'s agenda for tomorrow is empty*

When *the organizer assigns the job to "Paco" for tomorrow*

Then *the first thing for him to work on tomorrow is that job*

And *the job is no longer in the unassigned work queue*

Scenario: Prioritize jobs

Given *"Paco"'s queue for tomorrow contains "Change oil" & "Inspection"*

When *the organizer prioritizes the job "Inspection" for "Paco"*

Then *first thing for him to work on tomorrow is "Inspection"*

Inspiración

Gojko Adzic, Dan North, Matt Wynne,
Seb Rose, Chris Matts, Gáspár Nagy,
Ellen Gottesdiener, Mary Gorman,
Tony Bruce, Steve Tooke, Lisa Crispin,
Liz Keogh, Aslak Hellesøy, Paul Rayner,
Mary & Tom Poppendieck, W. E. Deming
Guillermo de Ockham, Alberto Brandolini