



# Behavior-Driven Development

<https://github.com/santiago-vl/bdd-cucumber-pill>



# Smells en las especificaciones

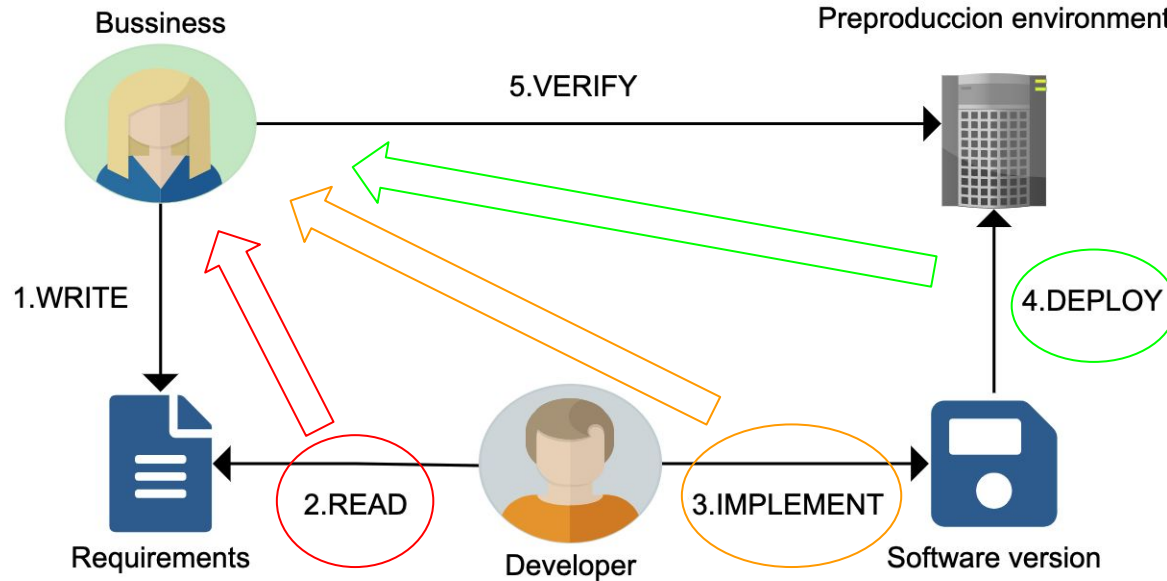
- Al publicar un grid que tiene versión DRAFT, la versión DRAFT pasa a ser la versión LIVE y se borra la versión DRAFT
- Los drafts se generan en función de la vista; es decir: se puede crear un draft de la configuración por defecto a nivel global, un draft por cada mercado y un draft por cada tienda
- Un grid puede tener una versión DRAFT por vista o ninguna
- Las versiones drafts de las distintas vistas (global, mercado, tienda), no son interdependientes entre sí cuando tienen versión LIVE; esto significa que si tengo un grid publicado en Global que a su vez tiene un draft global, las configuraciones que heredan ese mismo grid en otro mercado o tienda, la está heredando directamente de la versión de configuración default de LIVE, y hasta que no se publique el draft de default, no se actualizará en LIVE dicha configuración
- Las versiones drafts de los distintas vistas (global, mercado, tienda), sí que son interdependientes entre sí cuando NO tienen versión LIVE; esto significa que si tengo un grid sin publicar en Global (versión draft), las configuraciones que heredan ese mismo grid en otro mercado o tienda, la está heredando de la versión de configuración default de DRAFT
- Sólo se puede publicar un grid con configuración específica si la configuración por defecto o de mercado ya ha sido publicada
- En el caso de las exclusiones, cualquier exclusión en Global LIVE, pasa a los grids específicos de mercados y países tanto en DRAFT como LIVE
- En el caso de las exclusiones, las exclusiones en Global DRAFT, NO pasan a los grids específicos de mercados y países hasta que ese DRAFT se publica



# Problemas de comunicación

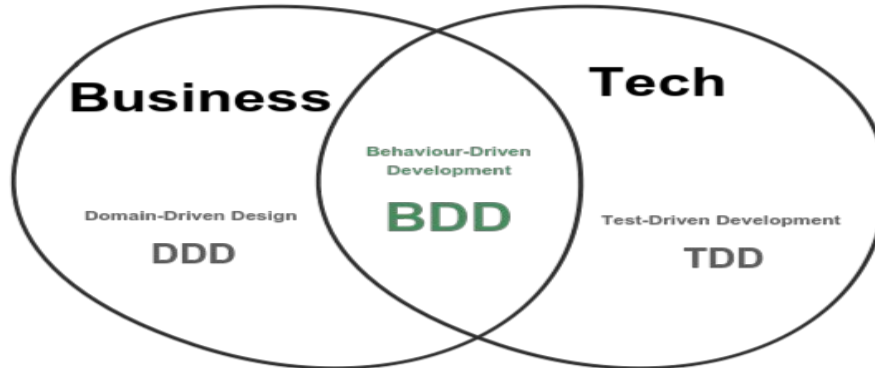
- El lenguaje natural puede ser demasiado flexible para describir comportamientos complejos
- Ausencia de análisis: No se han contemplado todas las casuísticas (happy paths)
- Falta de comprensión: El desarrollador no tiene la visión de negocio
- Demasiado idealista: Sin tener en cuenta aspectos técnicos

# ¡Esto no es lo que quería!



# Diseño guiado por el comportamiento

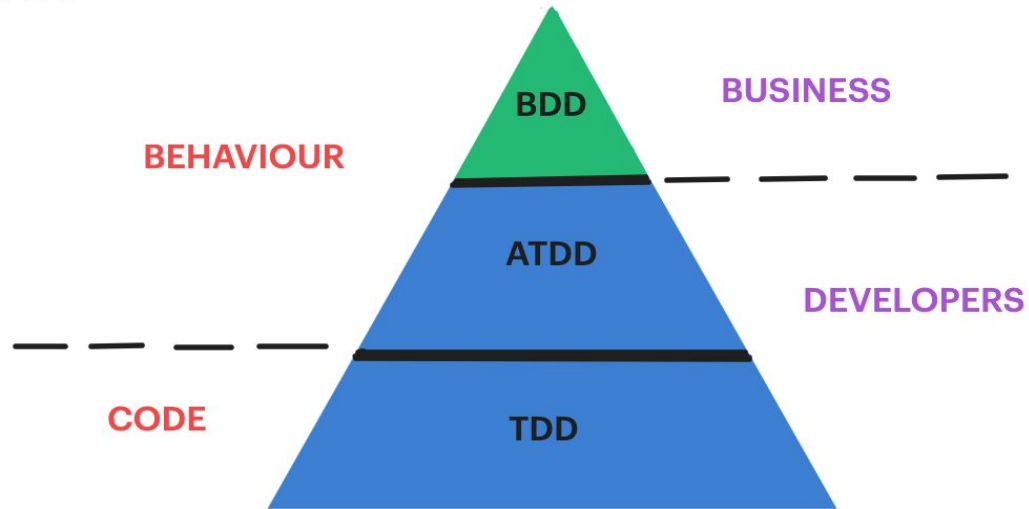
- Metodología que fomenta la colaboración entre roles
- Conversaciones para construir una comprensión compartida
- Evolución de ATDD



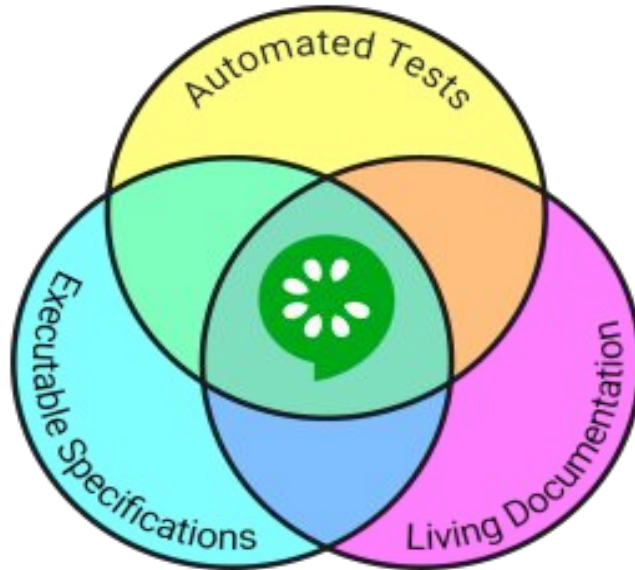
# Behaviour first

WHAT

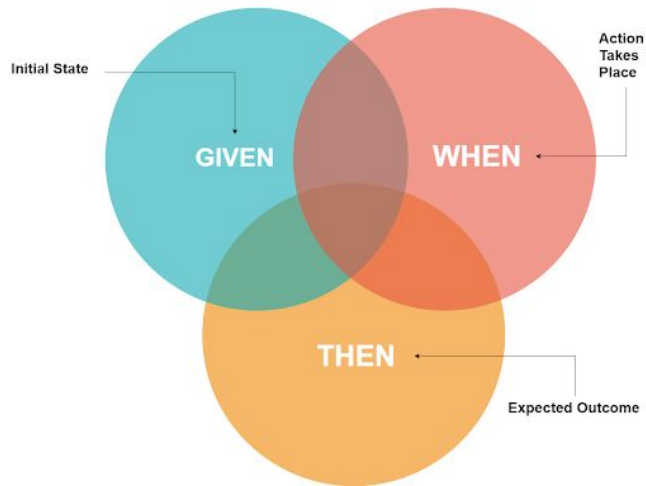
WHO



# Cucumber



# Gherkin



≡ search.feature x

```
1 Feature: Web Searching
2   In order to learn new stuff
3   As a Web surfer
4   I want to search for information
5
6   # Comments here
7
8   Scenario Outline: Search for animals
9     Given the Google home page is displayed
10    When the user searches for "<animal>"
11    Then search results are shown for "<animal>"
12
13    Examples: Animals
14      | animal |
15      | panda  |
16      | elephant |
17      | python  |
18
```



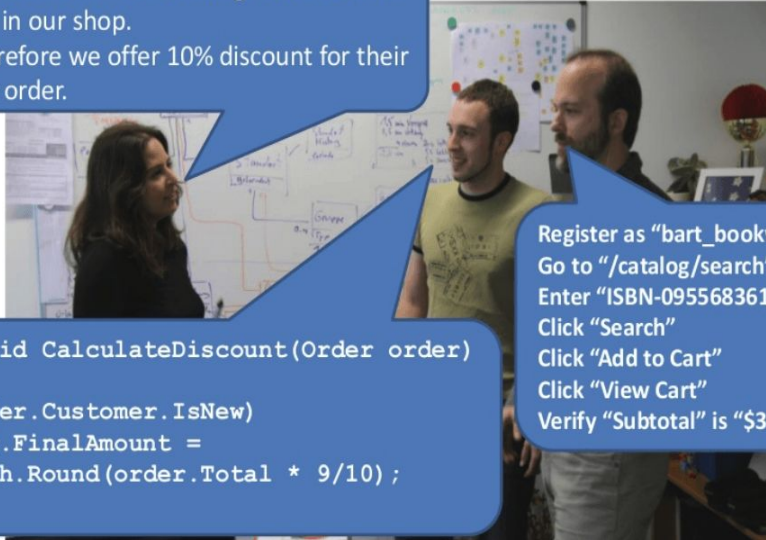
# Diferentes lenguajes



We would like to encourage new users to buy in our shop.  
Therefore we offer 10% discount for their first order.

```
public void CalculateDiscount(Order order)
{
    if (order.Customer.IsNew)
        order.FinalAmount =
            Math.Round(order.Total * 9/10);
}
```

Register as "bart\_bookworm"  
Go to "/catalog/search"  
Enter "ISBN-0955683610"  
Click "Search"  
Click "Add to Cart"  
Click "View Cart"  
Verify "Subtotal" is "\$33.75"





# Lenguaje ubicuo



**STAKEHOLDER**



We would like to encourage new users to buy in our shop.  
Therefore we offer 10% discount for their first order.



**Given** the user has not ordered yet

**When** the user adds a book with the price of EUR 37.5 into the shopping cart

**Then** the shopping cart sub-total is EUR 33.75.



# Ventajas BDD

- Mejora drásticamente la comunicación entre negocio y desarrolladores
- Compresión compartida elaborada por ambas partes
- Cucumber
  - Especificaciones ejecutables
  - Documentación viva y centralizada
  - Mayor control del estado del proyecto
  - Ayuda a atomizar tareas grandes (escenarios)
  - CI mas robusto
  - Full stack: UI y APIs



# Desventajas BDD

- Cambio cultural
- Requiere alta interacción entre negocio y desarrolladores
- Cucumber
  - Muy poca resistencia ante refactorizaciones grandes
  - Mantenimiento permanente de Gherkins e implementaciones
  - Requiere virtualizar integraciones con servicios

