Jérémy Sorant

INTRODUCTION AUX CONCEPTS ORIENTÉS CRAFT

LE BEHAVIOR-DRIVEN DEVELOPMENT

Jérémy Sorant

Avec cette série de mini-BDs, j'essaie de présenter des concepts orientés craft afin de les rendre accessibles aux non-initiés.

in/jeremy-sorant

jsorant

Jérémy Sorant

DÉFINITION

Le BDD est une méthode qui consiste à piloter le développement par les comportements.

La méthode permet de concevoir des logiciels de manière itérative en se concentrant sur les comportement que l’on souhaite implémenter.

ORIGINE

Beaucoup de projets échouent car le logiciel produit ne répond pas au besoin. Celui-ci peut avoir été mal identifié ou mal transmis aux devs.

Car c'est ce que les développeurs ont compris qui part en production, le BDD invite à resserrer les liens entre le métier et l'équipe de développement.

DOMAIN DRIVEN DESIGN

Le BDD est proche du DDD.

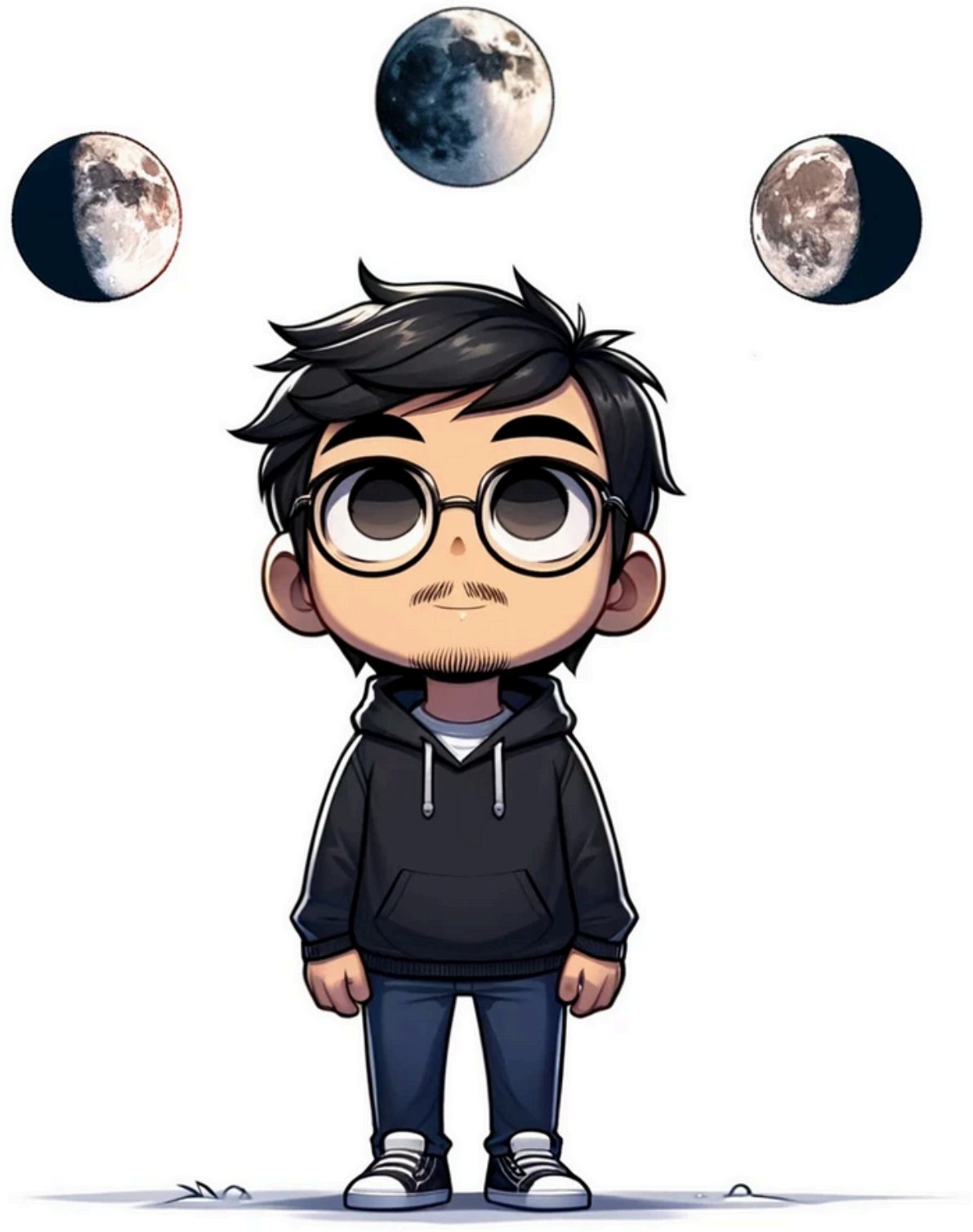
On retrouve notamment le concept de langage commun issu du métier et la volonté de faire communiquer toutes les parties prenantes du projet (utilisateur, dev, QA, PO, doc, support...).

TEST DRIVEN DEVELOPMENT

Le BDD ressemble au TDD par son approche “test first”.

En effet, la méthode demande de rédiger des comportements qui se traduisent en tests automatisés, puis de les implémenter.

On peut utiliser le BDD conjointement avec le TDD.



PHASES

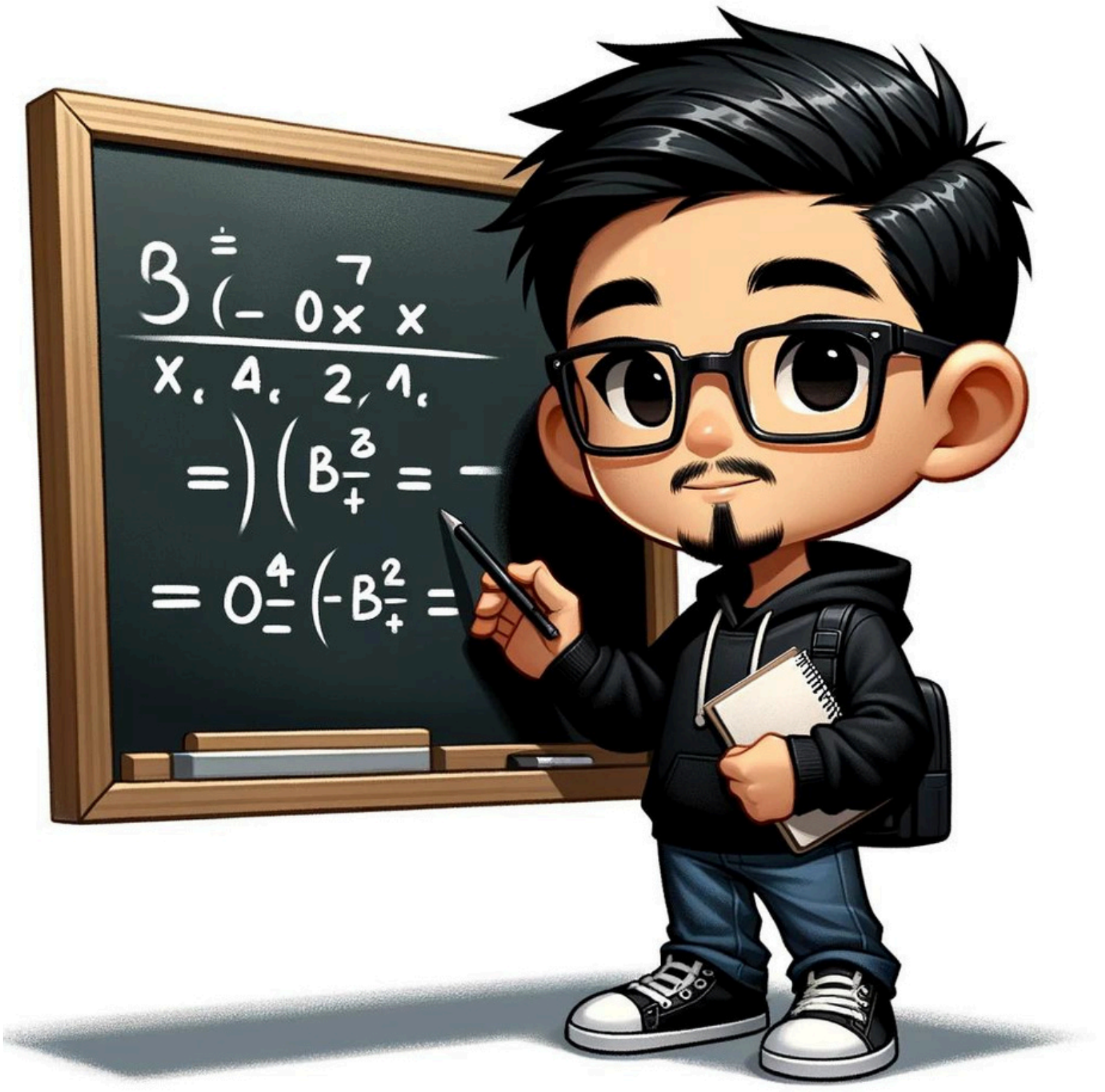
Le BDD se décompose en trois phases successives :

Discovery
Formulation
Automation

1 - DISCOVERY

On commence par recueillir le besoin. En général, cette étape est réalisée par le PO avec l'aide des utilisateurs finaux et des experts métier.

Si cela est possible, on ne se prive pas de faire intervenir l'ensemble de l'équipe pour l'imprégner des différentes problématiques métier.



II - FORMULATION

L'équipe formule ensuite des comportements au travers de scénarios dans un langage non technique.

On se met d'accord sur les termes métiers et on s'assure d'avoir une compréhension partagée des problématiques métier.

GHERKIN

Gherkin est le nom du langage non technique utilisé pour rédiger chaque scénario. Sa syntaxe très simple permet à n'importe qui de pouvoir le lire.

Un scénario est composé d'instructions débutant par les mots-clés “Given”, “When” et “Then” permettant de définir un contexte, une action et le résultat attendu.

3 AMIGOS

Cet atelier réunit à minima un dev, un QA et un PO. Il a pour but d'échanger autour de problématiques métier et de définir comment y répondre.

Le PO présente une problématique et une analyse nominale. Le QA réfléchit aux cas limites et aux problèmes éventuels. Le dev imagine la réalisation technique de la solution.

Jérémy Sorant

EXAMPLE MAPPING

L'objectif de cet atelier est d'analyser une user story et de définir les règles métier associées puis de produire des exemples concret de ces règles sous forme de scénarios Gherkin.

Il permet aussi de mettre en évidence des lacunes ou des questions dans la compréhension du métier.

III - AUTOMATION

Pour finir, on utilise un outil comme Cucumber pour jouer des tests automatisés à partir des scénarios formulés.

Puis on implémente les comportements définis de façon à faire passer ces tests.

Jérémy Sorant

DOCUMENTATION VIVANTE

Les scénarios Gherkin constituent une documentation vivante non technique et illustrée par des exemples concrets.

On dit qu'elle est vivante car elle est toujours à jour. En effet, elle évolue à chaque ajout, modification ou suppression d'un comportement du logiciel.

Jérémy Sorant
ingénieur logiciel sénior
chez SHODO LYON

l'ESN craft militante
#justicesociale