

# Behavior Driven Development (BDD)

## TP2 – Implémentation avec BDD à partir d'une User story

# Enoncé

## 1. Cas Nominal

Votre équipe de développement vient de se voir commandé le développement d'une API permettant de désigner le vainqueur d'un scrutin majoritaire.

Le Product Owner (PO) de l'équipe s'est réuni avec les clients afin de rédiger la première User Story du projet :

**En tant que** client de l'API à la clôture d'un scrutin majoritaire

**Je veux** calculer le résultat du scrutin

**Pour** obtenir le vainqueur du vote

Le Product Owner a également ajouté à la User Story les critères d'acceptances suivant :

- Pour obtenir un vainqueur, le scrutin doit être clôturé
- Si un candidat obtient > 50% des voix, il est déclaré vainqueur du scrutin
- On veut pouvoir afficher le nombre de votes pour chaque candidat et le pourcentage correspondant à la clôture du scrutin.
- Si aucun candidat n'a plus de 50%, alors on garde les 2 candidats correspondants aux meilleurs pourcentages et il y aura un deuxième tour de scrutin
- Il ne peut y avoir que deux tours de scrutins maximums
- Sur le dernier tour de scrutin, le vainqueur est le candidat ayant le pourcentage de vote le plus élevé
- Si on a une égalité sur un dernier tour, on ne peut pas déterminer de vainqueur

En vous basant sur ces critères d'acceptance, vous allez devoir créer les scénarios Gherkin correspondants, puis implémenter les tests et le code de production correspondant en respectant le cycle BDD.

Je vous conseille de commencer par les cas d'utilisation nominaux pour construire la base de votre projet, puis d'ajouter un par un les scénarios afin de prendre en compte tous les critères d'acceptations.

**Note :** Comme pour le TP précédent, pour simplifier la mise en œuvre nous travaillerons avec des projets de type libraires .Net (pas d'affichage).

## 2. Critères manquants

Vous aurez sûrement remarqué qu'ils manquent certains critères d'acceptances pour pouvoir gérer tous les cas d'utilisations :

- Gestion des égalités sur le 2<sup>ème</sup> et 3<sup>ème</sup> candidat sur un premier tour
- Gestion du vote blanc

Dans cette deuxième partie, à vous de prendre le rôle de Product Owner et de décider les critères d'acceptations que vous voulez mettre en place pour gérer ces deux cas.

Ajoutez les scénarios manquants et les implémentations associées.

### **Livrables**

- Un lien vers un repo Git contenant :
  - Vos projets de tests et code de production