Le Behavior Driven Development (BDD) et Gherkin

Introduction

Behavior Driven Development (BDD) est une méthodologie de développement logiciel qui vise à améliorer la communication entre les développeurs, les testeurs et les non-techniciens grâce à un langage commun. Gherkin est le langage utilisé pour écrire des scénarios de test en BDD.

1. Qu'est-ce que le BDD ?

1.1 Définition

Le BDD est une extension du Test Driven Development (TDD) qui encourage la collaboration entre les développeurs, les testeurs et les parties prenantes non techniques. L'objectif principal du BDD est de s'assurer que toutes les parties prenantes comprennent clairement les exigences et que le produit développé répond aux besoins métier.

1.2 Avantages du BDD

- Communication améliorée : Le BDD utilise un langage naturel, compréhensible par tous les membres de l'équipe.
- Documentation vivante : Les scénarios BDD servent de documentation vivante qui est toujours à jour avec l'application.
- Détection précoce des problèmes : En définissant clairement les comportements attendus, les problèmes peuvent être détectés plus tôt dans le cycle de développement.
- Focus sur l'utilisateur final : Le BDD permet de garder le focus sur la valeur ajoutée pour l'utilisateur final.

2. Qu'est-ce que Gherkin ?

2.1 Définition

Gherkin est un langage spécifique au domaine (DSL) utilisé pour décrire le comportement des systèmes logiciels en BDD. Il utilise une syntaxe simple basée sur des mots-clés et permet de décrire des scénarios de manière claire et concise.

2.2 Syntaxe de Gherkin

La syntaxe de Gherkin se compose de quelques mots-clés principaux :

- ✓ Feature : Une fonctionnalité ou une caractéristique de l'application.
- ✓ Scenario : Un cas de test spécifique.
- ✓ Given : Le contexte initial de l'application.
- ✓ When : L'action ou l'événement déclencheur.
- ✓ Then : Le résultat attendu.
- ✓ And, But : Pour lier plusieurs conditions ou résultats.

Exemple de scénario en Gherkin :

```
Feature: User login

Scenario: Successful login with valid credentials

Given the user is on the login page

When the user enters valid credentials

And clicks on the login button

Then the user is redirected to the dashboard

And a welcome message is displayed
```

3. Comment implémenter le BDD avec Gherkin

3.1 Outils courants

- **Cucumber :** Un des frameworks BDD les plus populaires, compatible avec plusieurs langages de programmation comme Java, Ruby, et JavaScript.
- **SpecFlow** : Une implémentation de Cucumber pour .NET.
- **Behave** : Un framework BDD pour Python.

3.2 Étapes d'implémentation

- ✓ Définir les fonctionnalités : Utilisez le format Gherkin pour décrire les fonctionnalités et les scénarios.
- ✓ Automatiser les tests : Utilisez un framework BDD pour écrire des étapes de test automatisées basées sur les scénarios Gherkin.
- ✓ Exécuter et maintenir : Exécutez régulièrement les tests automatisés et mettez à jour les scénarios et les tests au fur et à mesure que l'application évolue.

4. Cas d'utilisation et exemples réels

4.1 Projet A : Application de commerce électronique

Dans un projet de commerce électronique, le BDD a été utilisé pour définir des scénarios comme l'ajout d'un produit au panier, la gestion des stocks, et le processus de paiement. Les scénarios écrits en Gherkin ont aidé à aligner les attentes des parties prenantes et à assurer la couverture complète des tests.

4.2 Projet B : Application bancaire

Pour une application bancaire, le BDD a permis de définir des scénarios critiques comme la validation des transactions, la gestion des comptes, et la sécurité. Les scénarios Gherkin ont servi de documentation vivante pour les nouvelles fonctionnalités et les modifications.

Conclusion

Le Behavior Driven Development (BDD) et Gherkin offrent une approche structurée et collaborative pour le développement de logiciels. En utilisant des scénarios écrits en langage naturel, toutes les parties prenantes peuvent comprendre et valider les fonctionnalités du logiciel, ce qui conduit à des produits de meilleure qualité et à une réduction des erreurs.