Mettre en place des tests sur du code legacy

AGILE TESTING NIGHT #2 - 25/09/2018

Qui suis-je ?

Patrick GIRY

Software gardener

arolla

patrick.giry@arolla.fr

@PatrickGIRY

https://github.com/PatrickGIRY

20 ans d'expérience dans le développement logiciel

Test Driven Development
Behavior Driven Development
Domain Driven Design
Functional programming

CO-ORGANISATEUR

Meetup

Coding Dojo

Dojo développement Paris

KataCatalogue

- KataBankOCR
- KataFizzBuzz
- FooBarQix
- KataPotter
- KataRomanNumerals
- KataRomanCalculator
- KataNumbersInWords
- KataArgs
- KataAnagram
- KataDepthFirstSearch
- KataNumberToLCD

https://github.com/dojo-developpement-paris/

Co-organisateur

Mission

SOCIETE GENERALE

Val de Fontenay - Les Dunes

RESG / BSC / DCO / CDE / QUA

Equipe de Guillaume THEYTAZ

ALTANCE

Eric CONSTANTIN

Coaching craft

Mettre en place des tests sur du code legacy

Pourquoi ajouter des tests à un code legacy?

- ➔ Pour pouvoir modifier le code existant (pour une évolution ou une correction) de manière sécurisée.
- ➔ Pour bien analyser le code et ses dépendances.
- ➔ Pour documenter le comportement attendu du code.

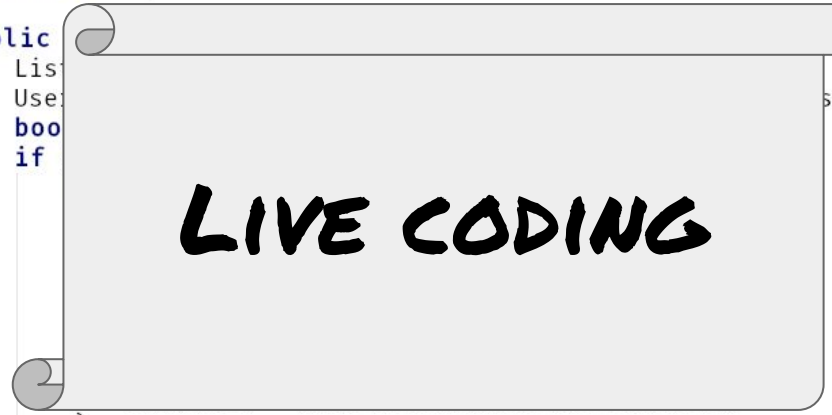
Exigences métiers

- Vous devez être connecté pour voir le contenu.
- Vous devez être un ami pour voir les voyages de quelqu'un d'autre.

```java
package trip;

import exception.UserNotLoggedInException;
import user.User;
import user.UserSession;

import java.util.ArrayList;
import java.util.List;

public class TripService {

    public
        Lis
        Use
        boo
        if
            }
            return tripList;
        } else {
            throw new UserNotLoggedInException();
        }
    }

}
```

ggedInException {

ser();

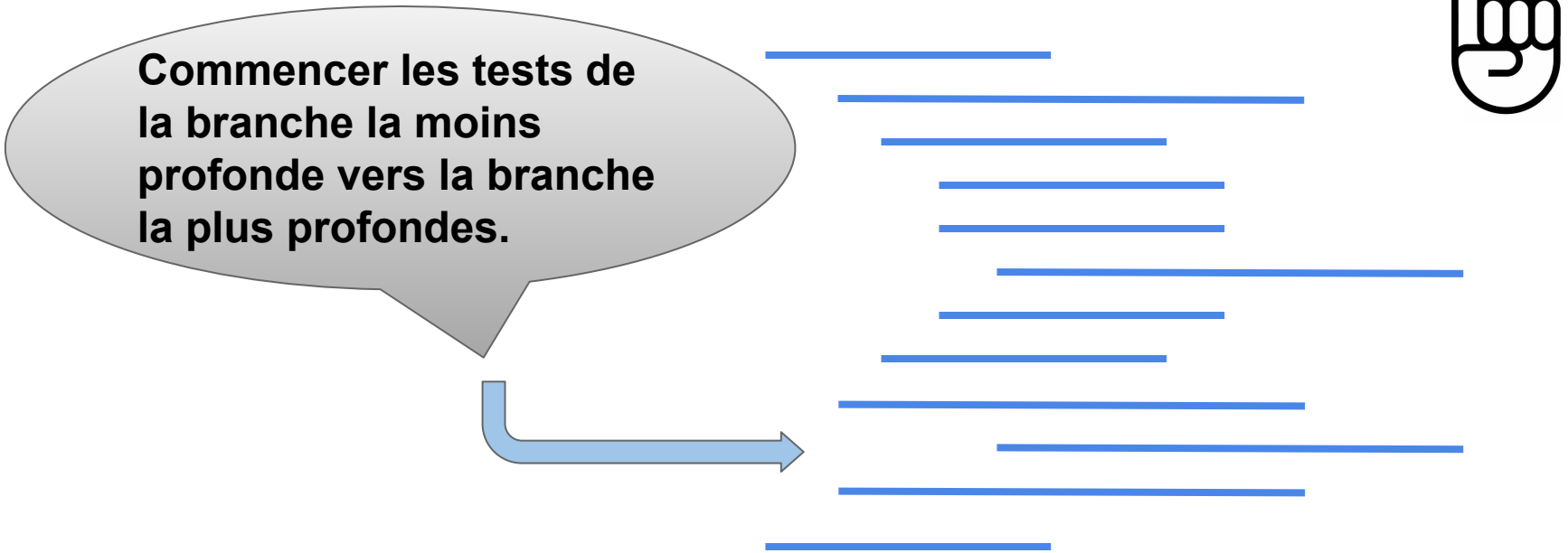


Règles pour le code legacy

Vous ne pouvez pas changer le code production si il n'est pas couvert par les tests ?

Seul le refactoring automatique (via IDE) est permis si c'est nécessaire pour écrire les tests.

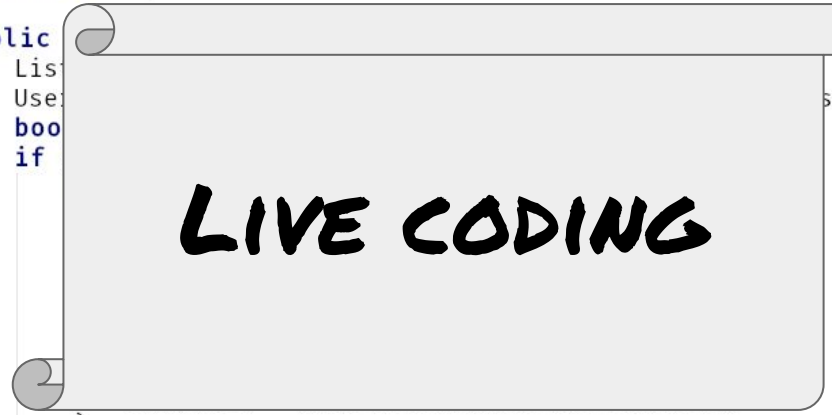
Astuces pour travailler avec du code legacy



```java
package trip;

import exception.UserNotLoggedInException;
import user.User;
import user.UserSession;

import java.util.ArrayList;
import java.util.List;

public class TripService {

    public
        Lis
        Use
        boo
        if
            }
            return tripList;
        } else {
            throw new UserNotLoggedInException();
        }
    }

}
```

ggedInException {

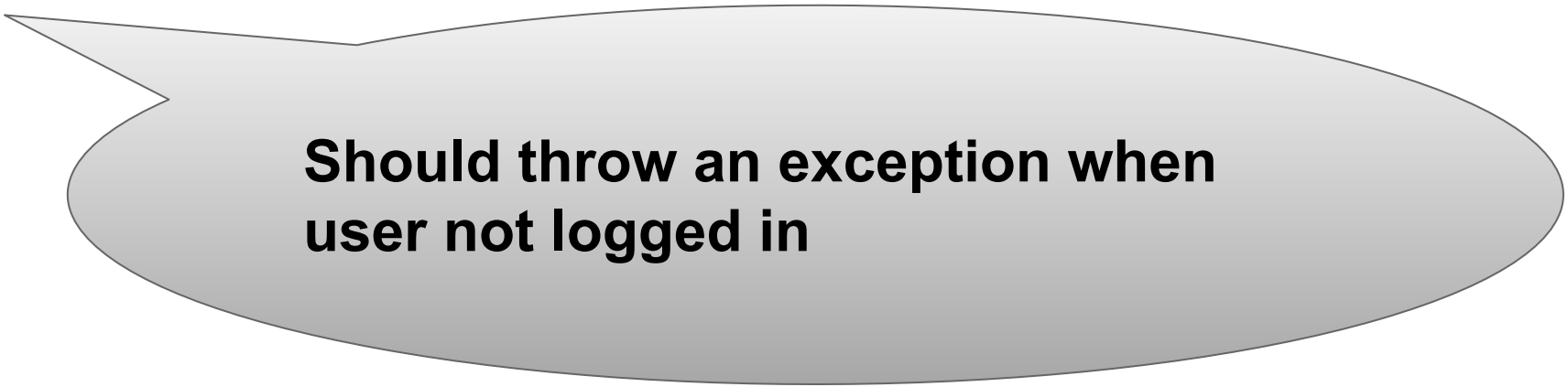
ser();



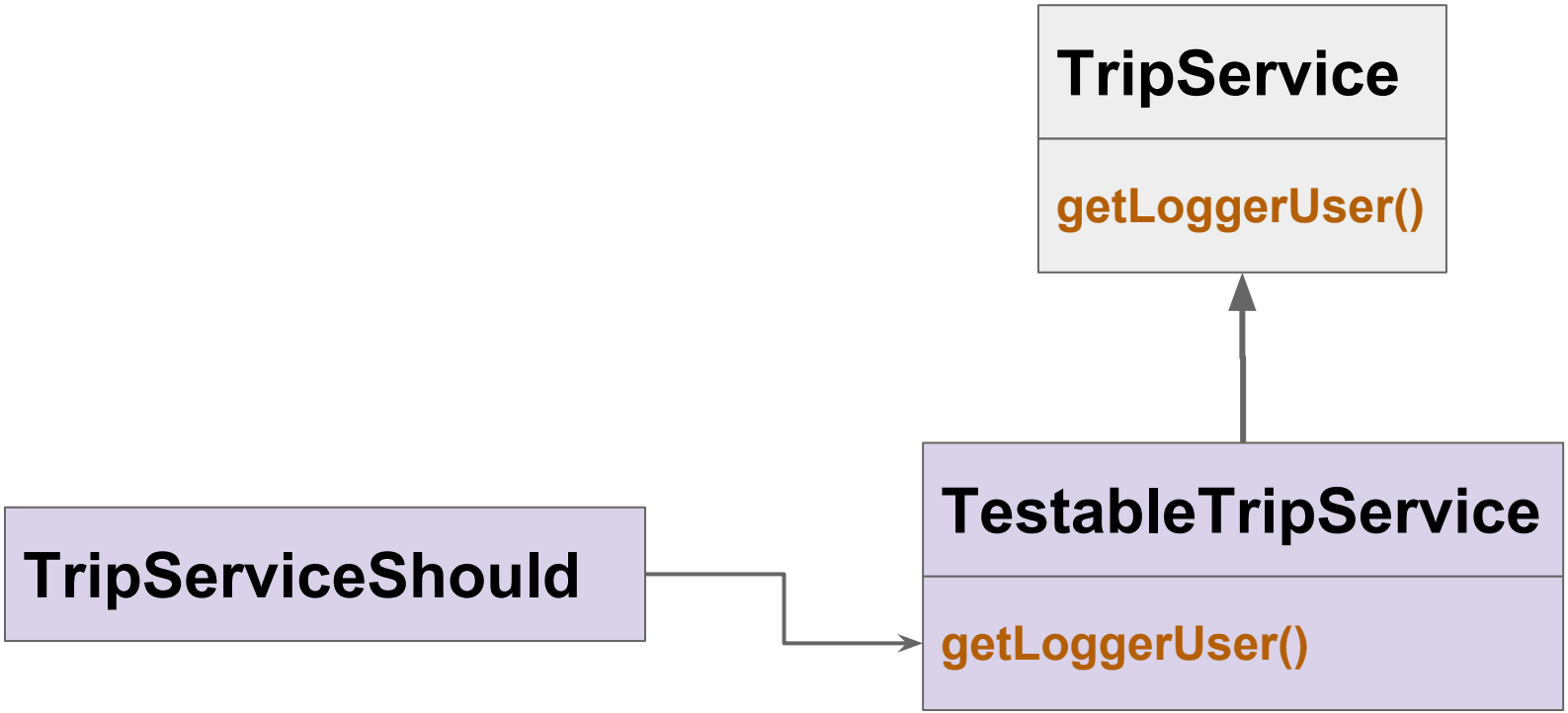
Comment nommer une méthode de Test?

Une méthode de test doit être nommée par une phrase qui décrit le comportement qui devrait se produire.

Test de la branche la plus courte



Maîtriser l'utilisateur connecté pour les tests



Supprimer les valeurs "magiques"

- ➔ Ajouter le concept d'utilisateur "invité".
- ➔ Ajouter le concept d'utilisateur non utilisé.

Autres tests

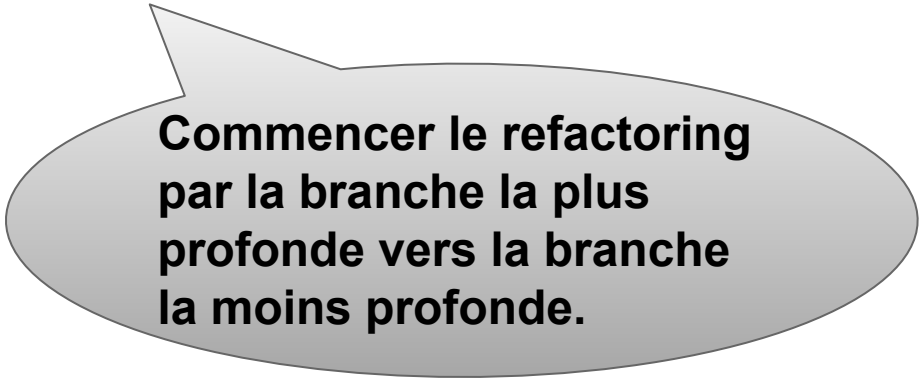
Should not return any trips when users are not friends

Should return friend trips when users are friends

Les comportements attendus sont testés

On peut faire du refactoring sur le code production.

Astuces pour travailler avec du code legacy



Déclarer les variables au plus proche de leurs usages

La variable isFriend n’est utilisé que dans le bloc du if.

Principe de responsabilité unique

Indiquer qu'un utilisateur est l'ami d'un utilisateur donné est de la responsabilité de la classe User.

Living documentation + ubiquitous language

Les tests spécifient et documentent les comportements attendus.

Le code produit est défini avec le vocabulaire du métier.

Travail d'artisan

- Ecrire du code lisible et maintenable
  - ➢ Le code doit exprimer des règles métiers
- S'efforcer de faire simple
- Bien connaître ses outils (par exemple les frameworks, les raccourcis, etc …)
- Travailler en petits incréments sûres
  - ➢ Valider souvent
- Être courageux, ne pas hésiter à faire des changements


The End

https://github.com/PatrickGIRY/tripServiceKataInJava10