

StudentManager Application

MGL7460 - Project

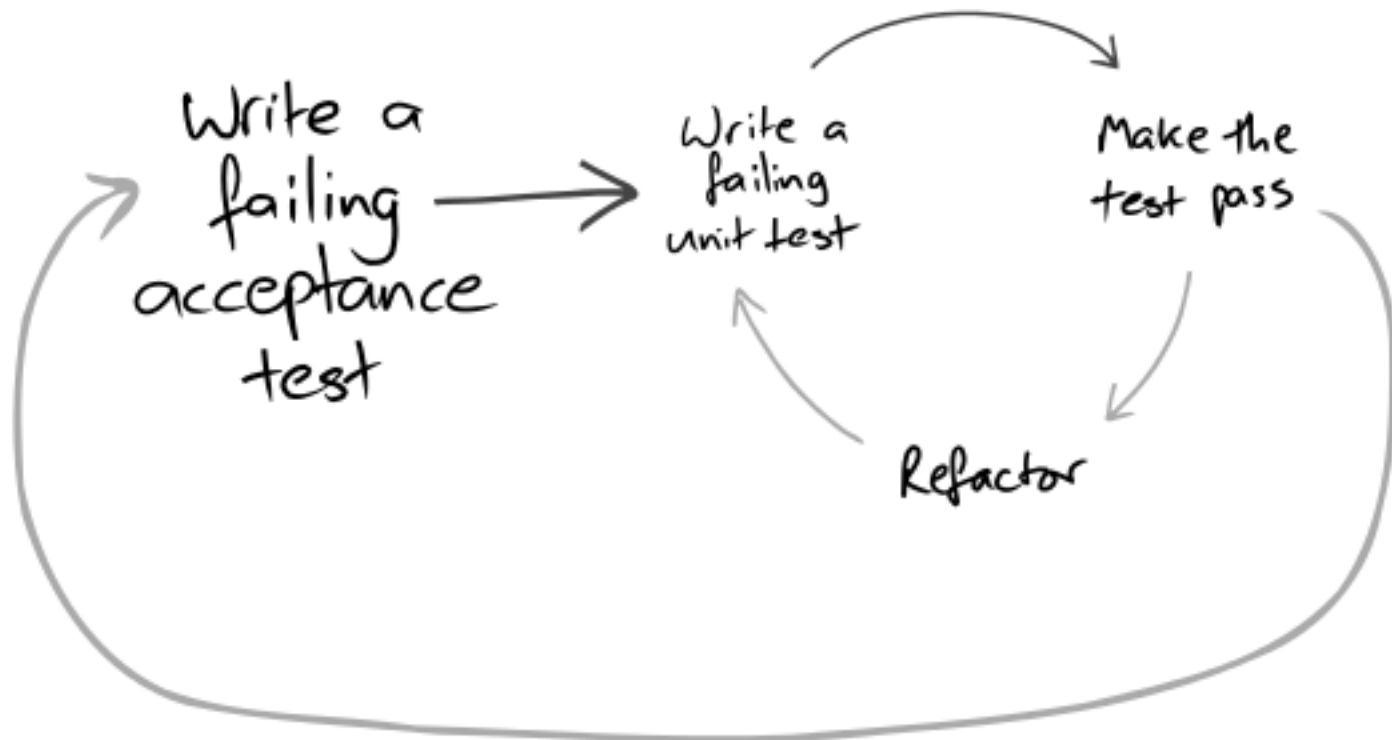
François Hébert
29 novembre 2016

Application

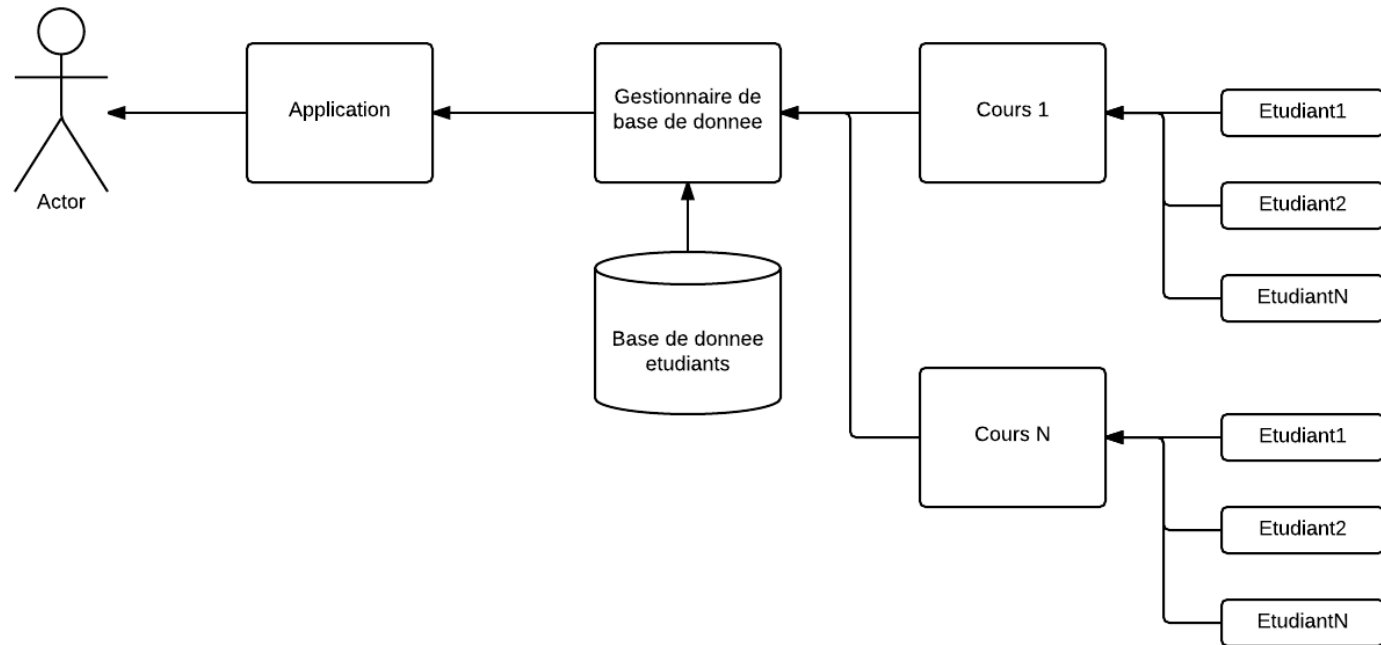
- Description
 - Logiciel simplifié de gestion de base de données
- Contrainte
 - Interface en ligne de commande
 - Codage en C#
 - Sauvegarde de la base de données en fichier texte

Méthodologie de développement

- Développement par Test d'Acceptation - DTA - ATDD



Survol Architecture



Ecosystème de Développement

- Langage C#
- SE: Windows 7
- IDE: Visual Studio 2015
- Librairies: .Net 4.5.2 - Ndesk.Options
- Librairies de Test: Specflow - Nunit - Fashion Assertion

Fichier « Feature »

Feature: AddStudent

In order to fill a class with students

As a school administrator

I want to be able to add student to a class

Background:

Given I have the data base with

Class	Student
PHY001	Thibodeau, Jean

Scenario:

Add a student to a class

When I add a student to the class

Class	Student
PHY001	Loiseau, Martin

Then the data base should have those element

Class	Student
PHY001	Thibodeau, Jean
PHY001	Loiseau, Martin

@redPath

Scenario: Add a student to a class in which it already exist

When I add a student to the class

Class	Student
PHY001	Thibodeau, Jean

Then I should get an error message "Error student already exist."

@redPath

Scenario: Add a student without specifying the class

When I add a student to the class

Class	Student
	Thibodeau, Jean

Then I should get an error message "Error class not specified."

Fichier « Step »

```
using TechTalk.SpecFlow;

namespace TestAcceptation
{
    [Binding]
    public sealed class GenerateDataBaseStep
    {
        [Given(@"I have the data base with")]
        public void GivenIHaveTheDataBaseWith(Table table)
        {
            Application.Current.CreateTestableDataBase(table);
        }
    }
}
```

```
using TechTalk.SpecFlow;

namespace TestAcceptation
{
    [Binding]
    public class AddStudentSteps
    {
        [When(@"I add a student to the class")]
        public void WhenIAddAStudentToTheClass(Table table)
        {
            Application.Current.AddStudent(table);
        }
    }
}
```

Retour expérience méthodologie

- DTA
 - Pour
 - Produit le code le plus simple réussissant le test
 - Donne confiance dans le code produit
 - Met l'emphasis sur le principe de séparation des responsabilités
 - Contre
 - Peut être perçu comme une procédure lourde
 - Le premier test est souvent long et compliqué à mettre en place

Retour expérience technologie

- Visual/C#
 - Pour
 - Environnement de développement intégré
 - Facilité de codage
 - Contre
 - Environnement de développement intégré
 - Facile seulement sur Windows
 - Pas multi-plateforme

Retour expérience technologie

- Specflow/Gherkin
 - Pour
 - La définition des tests d'acceptation par un client
 - Définition des fonctions étapes automatiques
 - Facile à utiliser
 - Contre
 - Manque de diversité dans les types de variables

Questions?