



Projet Cross-Canada





Les points que nous aborderons



- ◆ Qu'est-ce que Cross-Canada(UCOSP)?
- ◆ Sur quel logiciel avons-nous travaillé?
- ◆ Comment s'est déroulé le projet?
- ◆ Évaluation de la qualité du logiciel
- ◆ Démonstration en ligne

The page features a decorative design with a vertical blue line on the left side, a horizontal olive green bar at the top, and a vertical olive green bar on the right side. At the bottom, there is a series of vertical olive green lines of varying heights, resembling a barcode or a stylized pattern.

Qu'est-ce que Cross-Canada?



Cross-Canada

- ◆ Participation à un projet Open Source
- ◆ Dirigé par l'Université de Toronto
- ◆ Plusieurs projet disponibles
- ◆ Notre choix: Markus (Nous y reviendrons)



Cross-Canada

- ◆ Notre implication
 - Spécialistes des tests & développeurs d'une fonctionnalité
 - Développement de Tickets
 - Participation à la revue de code
 - Réunion hebdomadaire sur IRC
 - Participation au blog
 - Punchlines hebdomadaires
 - Minutes
 - Biographie
 - Tout type de publication pertinente
 - Participation au code-sprint



Code Sprint: L'objectif

- ◆ Rencontrer nos coéquipiers
- ◆ Rencontrer les organisateurs du projet
 - Introduction au processus d'assurance qualité
 - Présentation de Selenium et Cucumber
- ◆ Énoncé des besoins du client
- ◆ Priorisation des besoins
- ◆ Attribution des tâches

Code Sprint: L'équipe





Sur quel logiciel avons-nous
travaillé?





Markus

Aspects techniques



- ◆ Application Web
- ◆ Ruby on Rails MVC(Language de script)
- ◆ Plateformes(Linux, Mac, Windows)
- ◆ Web Browser(Firefox)
- ◆ Eclipse
- ◆ Gestion des sources avec SVN
- ◆ DB: MySQL/Postgres

Markus

Aspect fonctionnels

- ◆ Est un outil open-source de gestion des travaux
- ◆ Utilisateurs:
 - Étudiants
 - Enseignants & Correcteurs
- ◆ Fonctionnalités:
 - Création des demandes de travaux, contraintes et spécifications
 - Critères d'évaluation et pondération
 - Annotation des travaux
 - Évaluation des travaux
 - Consultation des résultats



Markus

Processus de développement

- ◆ Processus déjà défini
 - Attribution des tâches de développement à l'aide de Tickets (DrProject)
 - On développe localement
 - Revue de code via Review board
 - On commit lorsqu'on obtient un Ship it!
 - Documentation des fonctionnalités



Comment s'est déroulé le projet?



Notre projet Flexible Marking Scheme

- ◆ Le client nous a énuméré ses besoins sur un tableau blanc
- ◆ Nous les avons résumés à l'aide de:
 - ◆ User stories
 - ◆ Prototype
 - ◆ Schéma de base de données & Grep
 - ◆ Diagramme de classes
 - ◆ Tickets

Prototype

Choix du marking scheme



Edit Assignment: AssignmentE

Assignment Properties

Short Identifier: AssignmentE

Name: AssignmentE

Message:

User story
Je veux avoir le choix de la grille de correction liée à chacun des travaux.

Due Date: 2009-11-30 00:00 Format: YYYY-MM-DD HH:MM

Marking Scheme

Rubric

Flexible



Prototype Gestion des critères



Criteria Management: AssignmentE

[Add Criterion](#) | [Upload](#) | [Download](#)

| Criterion | Total mark (0) |
|------------|----------------|
| CriterionC | 15.0 |
| CriterionA | 12.0 |
| CriterionB | 25.0 |

CriterionA

[Save changes](#) [Delete](#)

| | |
|--------------|---|
| Criterion | <input type="text" value="CriterionA"/> |
| Maximum Mark | <input type="text" value="12.0"/> |
| Description | <input type="text" value="This is the description for CriterionA"/> |

You may use simple HTML tags in this field. Save changes to see a preview.

[Save changes](#)

Preview

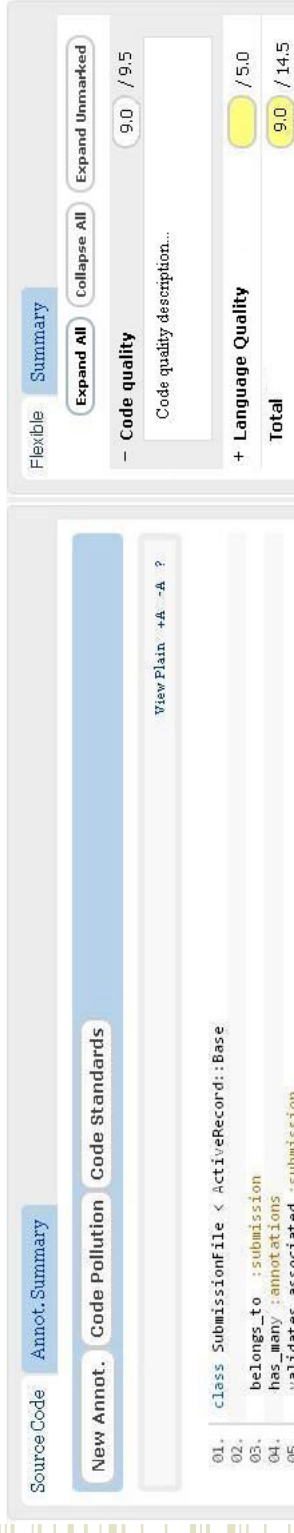
This is the description for CriterionA

User story
Je veux pouvoir créer
des critères de
correction

Prototype Attribution des notes



Grading Assignment: TW1



User story
Je veux pouvoir noter une soumission à l'aide du flexible marking scheme

```
01. class SubmissionFile < ActiveRecord::Base
02.
03.   belongs_to :submission
04.   has_many :annotations
05.   validates_associated :submission
06.   validates_presence_of :submission
07.   validates_presence_of :filename
08.   validates_presence_of :path
09.
10.   def get_file_type
11.     # This is where you can add more languages that Submis
12.     # recognize. It will return the name of the language.
13.     # SyntaxHighlighter can work with.
```


Prototype

Vue sommaire

| Flexible | Summary | | | | |
|------------------------------------|---------------|---|------|---|-------|
| Title | Maximum value | x | Mark | = | Grade |
| Code quality | 9.5 | 1 | 9 | / | 9.5 |
| Language Quality | 5.0 | 0 | 0 | / | 5.0 |
| Subtotal: 9.0 / 14.5 | | | | | |
| Bonus/Deductions | | | | | |
| Comment | | | Mark | | |
| Add Extra Mark | | | | | |
| Total Extra Marks: 0.0 | | | | | |
| Percentage | | | | | |
| Comment | | | | | |
| Total Extra Percentage: 0.0 | | | | | |
| Translated to Points: 0.0 | | | | | |
| Final Mark 9 / 14.5 | | | | | |



Base de données

- ◆ Ajout d'un nouveau champ
- ◆ Ajout d'une nouvelle table
- ◆ Renommage d'un champ existant
 - Grep
 - Modifications des programmes existants
- ◆ Ajout d'indexes



Tickets

- ◆ Division du travail en petites tâches
- ◆ Ces petites tâches sont devenues des tickets
- ◆ Ces tickets se sont développés en respectant le processus de développement en place

L'état des tests

- ◆ À notre arrivée:
 - Quelques tests unitaires en place
 - Plusieurs tests en erreur & manquants
 - Peu d'outils de test en place
- ◆ Notre tâche:
 - Promouvoir l'importance des tests
 - Implanter de nouveaux outils de tests



Tests unitaires

- ◆ Shoulda BDD déjà en place
- ◆ Mocha

Tests de fonctionnalités

- ◆ Selenium



Test d'acceptation

- ◆ Cucumber

```
$ cucumber features/addition.feature
Feature: Addition # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
Scenario: Add two numbers
  Given I have entered 50 into the calculator # features/additi
    uninitialized constant Calculator (NameError)
    ./features/step_definitions/calculator_steps.rb:2:in `Given /
features/addition.feature:7:in `Given I have entered 50 into
And I have entered 70 into the calculator # features/step_d
When I press add # features/additi
Then the result should be 120 on the screen # features/additi
```



Évaluation de la qualité du logiciel



Basé sur ISO-9126

- ◆ Capacité fonctionnelle
 - Respect des user stories.
 - Vérifiés par les prototypes
 - Validés par les tests d'acceptation
- ◆ Facilité d'utilisation
 - Vérifié par les prototypes
 - Validé par des tests d'utilisation
- ◆ Maintenabilité
 - Développé en TDD
- ◆ Portabilité
 - Développé et testé sur différentes plateformes et BD



Démonstration en ligne

- ◆ Markus
- ◆ Blog
- ◆ Review board
- ◆ Quelques tests unitaires