

Rétrospective

Projet Agile - Bot Pokemon

Cédric CANESTRARI, Arthur CAVAILLE, Cédric CHEBROU, Baptiste DEJEAN

Vendredi 11 avril 2014 - Version 1.0

Sébastien NEDJAR
Benoît GANTAUME

Module - Insertion Professionnelle

2^e Année - Groupe 1

Gestionnaire de versions

| Version | Date | Description |
|---------|------------|---|
| 0.1 | 11/04/2014 | Première version |
| 0.2 | 11/04/2014 | Mise en place de l'en-tête et du pied de page |
| 1.0 | 11/04/2014 | Rédactions des parties |

Sommaire

| | | |
|----------|--|----------|
| 1 | Travail en équipe | 4 |
| 1.1 | Organisation | 4 |
| 1.2 | Difficultés | 4 |
| 1.3 | Avantages | 4 |
| 2 | Réalisation d'un projet d'envergure et complexe | 5 |
| 3 | Java, les tests et Git | 6 |
| 3.1 | Java | 6 |
| 3.2 | Les tests | 6 |
| 3.3 | Git | 6 |

Expérience professionnelle

1 Travail en équipe

1.1 Organisation

L'équilibre et la répartition des tâches du projet ont été correctement fait ce qui nous a permis de gérer notre temps. Nous avons également répartis le travail en fonction des facilités et difficultés de chacun. Nous avons eu, au début, des difficultés à démarrer car certains aspects techniques n'étaient pas explicites et nous avons du relire, en parler, et essayer de comprendre certaines fonctions et leur but.

La mise à disposition d'un matériel adapté et fonctionnel pour chacun des membres du groupe. Cela aurait permis d'être actif tout au long du projet sans être interrompu par un problème matériel ou logiciel quelconque. Certains jours, nous étions obligés de travailler sur 2 ordinateurs car c'était les seuls qui fonctionnaient correctement.

1.2 Difficultés

Les compétences des membres du groupe n'étaient pas homogènes. Ces différences ont engendré des retards et/ou des temps de compréhension ou de réalisation différents, ce qui nous a coûté du temps en groupe. Il a fallu que l'on discerne les difficultés de chacun, à quoi ces difficultés étaient liées et essayer de les résoudre ensemble.

1.3 Avantages

Le travail en équipe nous a permis de confronter les problèmes, en parler et les résoudre plus rapidement grâce aux connaissances de chacun. Les connaissances de l'équipe n'étant pas homogènes, certains n'ont pas que programmer mais ont aussi passé du temps avec d'autres membres du groupe qui avaient des lacunes pour les aider, les guider et débloquer les situations délicates.

2 Réalisation d'un projet d'envergure et complexe

L'encadrement du projet par les professeurs comme par le sujet détaillé et structuré nous a permis de réaliser le projet dans les temps en suivant les étapes demandées, et d'y ajouter nos propres modifications.

Nous avons apprécié le projet et nous n'avons pas trouvé de point à améliorer.

Nous avons été surpris à la fois par l'utilisation de l'API Twitter4J et par la simplicité de communiquer sur twitter par un logiciel développé par nous-même.

La réalisation du projet nous a appris à gérer notre temps, nos répartitions de travail. Mais aussi à confronter les problèmes, argumenter ensemble sur les solutions que nous avons rencontrés. Cette expérience de travail a été très fructueuse pour nous et notre confiance en nos aptitudes. Il nous a apporté des connaissances nouvelles en java, en test unitaire, en développement agile, en management mais aussi en communications et relations.

3 Java, les tests et Git

3.1 Java

Nos connaissances ont permis un très bon déroulement du projet. Notre niveau en java nous a étonné, nous ne pensions pas que nous pouvions réaliser un si gros projet sans avoir trop de problème. Nos refactors et nos raccourcis claviers sont à améliorer, nous avons remarqués que c'était quelque chose d'extrêmement utile et qui fait un gagner un temps précieux. Les "Foreach" sont extrêmement utile et nous l'avions pas sur la version C++ que nous utilisons.

3.2 Les tests

La mise en place des tests et leur compréhension ont été très facile et nous avons pu les utiliser partout. Les tests peuvent être amélioré en utilisant des "mocks" et on doit pouvoir réduire le nombre de ligne en utilisant des "@Before", "@BeforeClass" et également des "@After" et "@AfterClass".

Checkstyle est un plugin très intéressant que nous avons mis à notre avantage pour résoudre tous nos oublies de syntaxe, ce qui nous a permis de rendre le code propre et claire.

eCobertura est vraiment très utile et nous avons apprécié son utilisation pour voir les zones de code qui n'étaient pas testé et donc ajouté des tests de couverture.

3.3 Git