

Présentation du travail effectué

1. Présentation de la hiérarchie des dossiers de la solution rendue	1
2. Répartition des tâches	1
• QA - l'assurance qualité.....	1
• Méthodes agiles	2
3. Outils de travail	2
• Organisation	2
• Postes développeurs	2
4. Workflow de travail	3
5. Planification du travail	3

Présentation de la hiérarchie des dossiers de la solution rendue

Le rendu au format ZIP contient différents dossiers avec la hiérarchie suivante :

- **/DAL** : projet Visual Studio contenant la couche d'accès aux données
- **/Services** : projet Visual Studio contenant la couche de services
- **/WebApp** : projet Visual Studio contenant la couche d'application web
- **/docs** : contient les différents documents de présentation du projet, notamment :
 - le fichier PowerPoint de présentation de la soutenance
 - la documentation du projet
 - ce présent document de présentation du travail effectué

Pour visualiser la solution, il faut ouvrir le projet via le fichier de solution **GlobusDataPad.sln**.

Répartition des tâches

Il a été décidé de nommer des personnes responsables pour toutes les parties du projet.

NB : cela ne signifie pas qu'une seule personne travaille sur une partie, cela signifie qu'une personne est responsable et supervise le travail de cette partie.

Voici un aperçu des responsabilités telles qu'elles ont été réparties sur le projet :

- Création et maintenance du modèle de données / de la couche d'accès aux données : Martin Filliau
- Services : Guillaume Bernard et Vincent Daruty de Grandpré
- Back Office : Martin Filliau
- Front Office (et mobile) : Laurent Rakotomalala
- QA (Quality Assurance) : Laurent Rakotomalala

QA - l'assurance qualité

L'assurance qualité (QA, *Quality Assurance*) est réalisée tout au long du projet et doit permettre une plus grande robustesse de l'application. Ce travail est réalisé à la fois sur le code par des revues de code (*Code review*, lorsqu'un développeur relit le travail réalisé par

un autre développeur), par le suivi et remplissage des différents formulaires et l'utilisation de l'application...

Méthodes agiles

Nous avons essayé plusieurs méthodes dites "agiles", notamment :

- programmation en binôme (*pair programming*) : ce mode de programmation a été très utile dans les phases de démarrage du projet pour s'accorder sur l'architecture et le modèle de données.
- revues de code (*code review*) : ce travail nous a permis d'augmenter la qualité du code produit, mais aussi d'éviter des erreurs répétitives.

Outils de travail

Organisation

Nous avons décidé d'utiliser les outils de la suite Google pour nous organiser, à savoir :

- Google Code Hosting pour le versionning (SVN) du code source, le bug tracker, le wiki
- Google Groups pour le dialogue entre les membres de l'équipe
- Google Docs pour l'écriture des documents

Ces outils présentent l'avantage de ne pas nécessiter d'hébergement et d'être accessible partout, tout le temps.

Postes développeurs

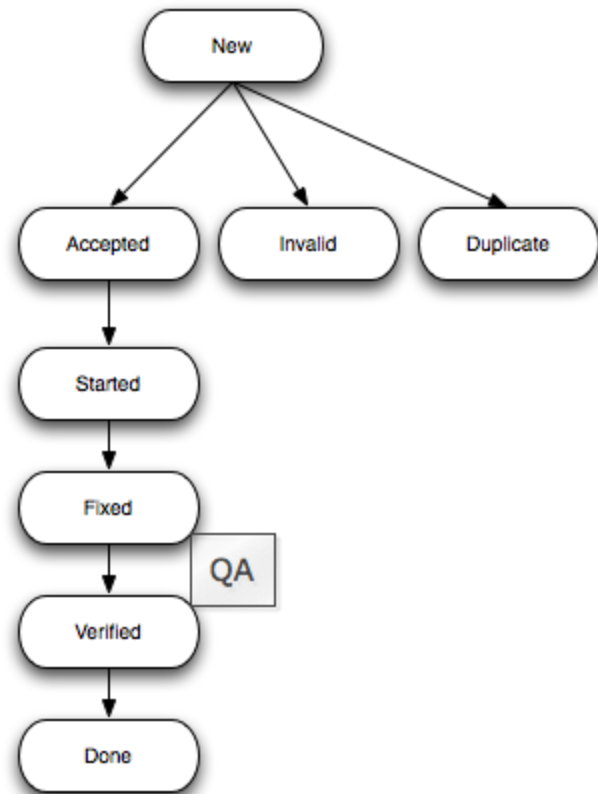
Les postes développeurs nécessitaient les outils suivants (qui impliquent d'utiliser un système d'exploitation Microsoft Windows, de préférence Windows Seven) :

- Microsoft .NET Framework 4.0
- Microsoft SQL Server 2008 : serveur de base de données
- Microsoft IIS Server 7.5 : serveur web
- Microsoft Visual Studio 2010 : environnement de développement
- TortoiseSVN : client SubVersion utilisé pour le versionning du code source

Workflow de travail

Toutes les tâches ont été découpées en tâches élémentaires, chacune de ces tâches représentant une "issue" dans le bug tracker, dont il était possible de suivre l'état d'avancement grâce aux commentaires et aux statuts :

- **New** : nouvelle tâche
- **Accepted** : tâche acceptée
- **Started** : tâche commencée
- **Fixed** : tâche résolue, doit passer par la phase de QA : le comportement doit être vérifié par une autre personne que celle qui l'a résolue
- **Verified** : validé par le QA
- **Invalid** : tâche non valide
- **Duplicate** : tâche déjà reportée
- **Done** : tâche terminée



Planification du travail

Nous avons travaillé en plusieurs cycles itératifs selon l'ordre suivant :

1. Mi-mars : lecture du cahier des charges
2. Mi-mars : répartition des responsabilités selon les compétences et l'expérience de chacun
3. Fin mars : validation des technologies
4. Fin mars : validation de l'architecture
5. Début avril : validation du modèle de données
6. Mi-avril : Implémentation du modèle de données
7. Fin-avril : implémentation des services
8. Fin-avril : création des vues de l'application web
9. Début mai : implémentation des controllers de l'application web en appelant les Services
10. Fin mai : finalisation des services et de l'application web
11. Début juin : finalisation de la documentation, vérification des interfaces
12. 10 juin : rendu