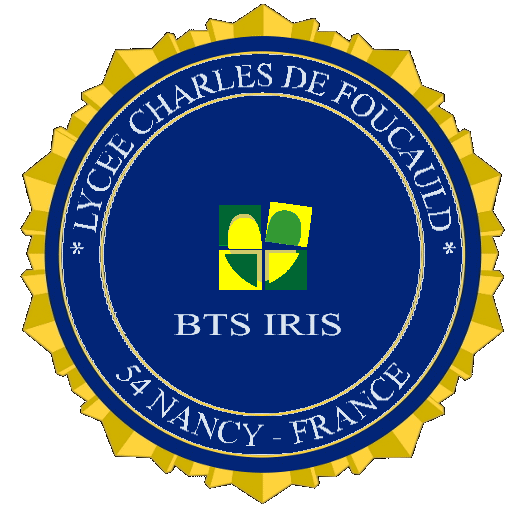
|  |
| --- |
| **WebSockets Chat** |
| Livrable : Annexes Techniques |
| Charles de Foucauld - NANCY |
| Académie de Nancy-Metz Version : V |
| Réf : |
| **José ALVAREZ** |
| **31/10/2014** |
|  |

Projet BTS IRIS

**Session : XXXX**

Etudiants :

|  |
| --- |
| ***Etudiant N°1 =*** |
| Etudiant N°2 = |
| Etudiant N°3 = |
| Etudiant N°4 = |

# Historique des modifications du document

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2009-12-09 | 0.0 | * Modèle de document.. | J. ALVAREZ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

SOMMAIRE

[1. Historique des modifications du document 4](#_Toc402441704)

[SOMMAIRE 5](file:///Y:\_ModèlesDossierAgile\-00-%20cdfAP_xxxxxxxx.docx#_Toc402441705)

[2. Jury Examen BTS IRIS. 6](#_Toc402441706)

[3. Instructions relatives aux noms. 6](#_Toc402441707)

[o **Langue Anglaise exclusivement !** 6](#_Toc402441708)

[o **Casse** 6](#_Toc402441709)

[4. Conventions de dénomination du projet. 6](#_Toc402441710)

[4.1.1 Sous-systèmes. 6](#_Toc402441711)

[5. Annexes techniques. 7](#_Toc402441712)

[5.1 ****Méthode de développement 7](#_Toc402441713)

[5.1.1 Méthodologie SCRUM 7](#_Toc402441714)

[5.1.2 Créer une branche de façon STRATEGIQUE 8](#_Toc402441715)

[6. Archivage. 9](#_Toc402441716)

[7. Références. 9](#_Toc402441717)

[8. Glossaire. 10](#_Toc402441718)

# Jury Examen BTS IRIS.

Cf. document : « **JuryBtsIris Examen.docx** »

# Instructions relatives aux noms.

* + **Langue Anglaise exclusivement !**

Pour les variables, propriétés, méthodes, commentaires… utilisez exclusivement la langue Anglaise.

* + **Casse**
* Utiliser dans tous les cas la casse Pascal  
  Ex : GetUsers
* Pour les paramètres utiliser la casse Camel  
  Ex : userName

Ref :   
*Submain\_DotNET\_Coding\_Guidelines.pdf  
cs-coding-standard-bellware.pdf  
principe\_dry.pdf*

# Conventions de dénomination du projet.

### Sous-systèmes.

|  |  |  |
| --- | --- | --- |
| **Catégorie** | **Nom** | **SIGLE** |
| Nom du projet | ***NomProjet*** | **SigleProjet** |
| Sous-système 1 |  | ( «   » ) |
| Sous-système 2 |  | ( «   » ) |
| Sous-système 3 |  | ( «   » ) |
| Sous-système 4 |  | ( «   » ) |

# Annexes techniques.

## Méthode de développement

### Méthodologie SCRUM

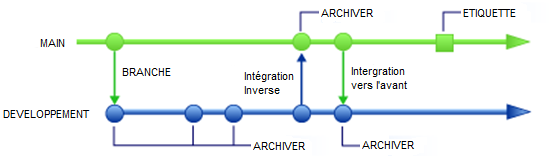
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

### Créer une branche de façon STRATEGIQUE

Le code source est une ressource importante des efforts de développement. Cependant, il peut s'avérer difficile de gérer et faire évoluer efficacement nos fichiers sources lorsque plusieurs développeurs travaillent simultanément sur des mises à jour de fichiers.

Pour résoudre ce problèmes ,nous allons utiliser un système de contrôle de version pour stocker le code source dans des référentiels partagés, pour isoler les efforts de développement en parallèle, pour intégrer des modifications de code et pour récupérer des versions de fichiers précédentes(Versioning). Un élément clé du contrôle de version est la création de branches qui permet un développement simultané. Si nous créons des branches de façon stratégique, nous pourront conserver l'ordre et la cohérence de plusieurs versions de notre logiciel.

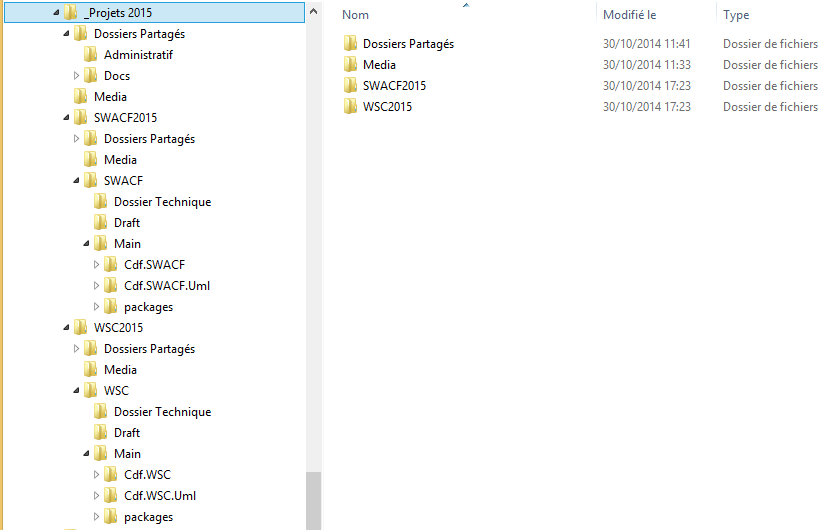
Team Foundation fournit un système de contrôle de version flexible et fiable. Nous allons donc pouvoir utiliser le contrôle de version Team Foundation pour gérer plusieurs versions au cours du développement du code source de notre projet, de documents du dossier technique (journal de bord, etc...) et d'autres informations critiques traitées par notre équipe.



1. Chaque branche doit avoir une stratégie définie à propos de l'intégration du code dans cette branche. Par exemple, dans la structure de branche de l'illustration précédente, vous pouvez assigner un membre de l'équipe qui possède et gère la branche principale (MAIN).Ce membre est chargé de l'exécution de l'opération de branche initiale, de l'intégration inverse des modifications de la branche de développement (DEVELOPMENT) vers la branche principale (MAIN) et de l'intégration en aval des modifications de la branche MAIN vers la branche DEVELOPMENT. L'intégration en aval est importante lorsque la branche MAIN intègre également des modifications d'autres branches.
2. La branche MAIN doit contenir le code qui a réussi les tests d'intégration afin qu'il soit toujours prêt pour la création d'une version.
3. La branche de développement (ou travail) évolue constamment car les membres de l'équipe archivent régulièrement des modifications.
4. Les étiquettes sont des instantanés des fichiers dans une branche à un moment donné.

[A compléter]

# Archivage.



# Références.

* A disposition sur le site ouvert à cet effet.

<http://www.officelive.com/fr-FR/>

* Cf. sur ce site, dans le dossier : Dossier>Fichiers Annexes Techniques la « **Note** » (**Liens**).

# Glossaire.

|  |  |
| --- | --- |
| CDF | Charles De Foucauld |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

NOTES