|  |  |  |
| --- | --- | --- |
|  | Youcode  Boulevard moulay youssef, Youssoufia, Maroc  support@youcode.com |  |
| rapport sur le brief projet | | | |
| Résumé du projet | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| Date du rapport | Nom du projet | Auteur |
| 01/12/2019 | Brief Projet – Scenario I | Abdelkarim Nouaouri – Oumaima Essafir – Ayoub El Jelladi - Taha Esselmouni – Abdelhakim Nassifi – Iliass Raihani |

|  |
| --- |
| État Récapitulatif |
|  |

Dans ce scenario le but est de faire le tour sur les commandes les plus utilisées de l’outil GIT

|  |
| --- |
| Vue d’ensemble du projet |
|  |

|  |  |
| --- | --- |
| tâche | notes |
| **Immersion** | **Mettre en place le setup qui nous servira pour utiliser les commandes GIT** |
| **Découverte** | **Comprendre la configuration GIT** |
| **Historique** | **Pouvoir se repérer dans les commits** |
| **Excluding Files** | **Introduction au .gitignore** |
| **Branching and Merging** | **Créer des branches et pouvoir merger entre elle** |
| **Conflict Resolution** | **Comment lors du merging, pouvoir régler les conflicts entre contributeurs/collaborateurs** |
| **Merge Tools** | **Introduction au p4merge et son utilisation** |
| **Challenge** | **Pouvoir cibler avec le .gitignore les fichiers que nous devons conserver et ignorer le reste** |

|  |
| --- |
| Historique des risques et problèmes |
|  |

|  |  |
| --- | --- |
| problème | Notes |
| **Répartition des taches** | **La simplicité des tâches reste relatives quant à la difficulté de mettre 6 personnes sur ces dernieres** |
| **--** | **--** |
| **--** | **--** |

|  |
| --- |
| CONCLUSIONS/RECOMMANDATIONS |
|  |

Le début peut être difficile car personne ne sachant ou se situer, suite à certaines initiatives, idées et avis nous avons pu répartir les tâches en trois partitions, kanban, rapport et git ainsi qu’un SM en tant que chef d’orchestre.

|  |
| --- |
| vue en detail de l’Etape I – immersion |
|  |

|  |  |
| --- | --- |
| Item | Description |
| **Le GIT** | **GIT est un logiciel libre de gestion de versions sous licence GNU V2 le plus utilisé au monde** |
| **.git** | **Le fichier .git contient toutes les données relatives au GIT a besoin pour gérer l’historique** |
| **Explication du commentaire** | **Up-to-date with ‘origin/master’ a pour signification qu’il n’y a rien à pousser. ‘Working directory clean’ a pour signification que tous les fichiers du répertoire actuel sont gérés par GIT et que la version la plus récente du fichier a été validé** |
| **README.mb** | **L’utilisation des cmd ‘touch, vim, add, commit, push’ du README.mb nous permet d’intégrer ce dernier** |

|  |
| --- |
| vue en detail de l’Etape II - decouverte |
|  |

|  |  |
| --- | --- |
| Item | Description |
| **HEAD** | **Une référence au dernier commit de la branche de sortie actuelle.** |
| **LOGS** | **Un enregistrement courant des commits présentant toutes les métadonnées relatif au log (auteur, adresse… )** |
| **BRANCHES** | **Un pointeur mobile sur l’un des commits effectués** |
| **Fichiers traqués** |  |

|  |
| --- |
| vue en detail de l’Etape III - historique |
|  |

|  |  |
| --- | --- |
| Item | Description |
| **Afficher le dernier commit avec option d’affichage** | C:\Users\youcode\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Annotation 2019-11-27 230118.jpg |
| **Créer son alias** | **‘alias historique=“git log –graph –decorate –oneline“** |
| **Historique des alias** |  |
| **Historique des commit README** | **C:\Users\youcode\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Annotation 2019-11-27 230929.jpg** |

|  |
| --- |
| vue en detail de l’etape IV – excluding files |
|  |

|  |  |
| --- | --- |
| Item | Description |
| **Stagging** |  |
| **.gitignore** |  |
| **Constation** | **Nous constatons que le fichier .log n’a pas été compris lors du stagging** |

|  |
| --- |
| vue en detail de l’Etape V – Branching & Merging |
|  |

|  |  |
| --- | --- |
| Item | Description |
| **The fast forward merging** | **Une fusion rapide peut se produire lorsqu’il existe un chemin linéaire entre le bout de la branche actuelle et la banche cible. Au lieu de la branche actuelle et la branche cible. Au lieu de « fusionner » réellement les branches, tout ce que Git doit faire pour intégrer les historiques et de déplacer (c’est-à-dire, « avancer rapidement ») de la branche actuelle jusqu’à la branche cible.** |
| **The manual** | **Une fois que les étapes de « préparation à la fusion » décrites précédemment ont été effectuées, une fusion peut être lancée en exécutant git merge <nom de la branche> et le nom de la branche qui sera fusionnée dans la branche réceptrice** |
| **The automatic** | **La fusion Git combine des séquences de commits dans un historique unifié des commits. Il y a deux manières principales de fusionner Git : Fast Forward et Three way. Git peut fusionner automatiquement les commits, sauf si des modifications entrent en conflit dans les deux séquences de validation.** |
| **Constation** | **???**  **???** |

|  |
| --- |
| vue en detail de l’Etape VI – Conflict Resolution |
|  |

|  |  |
| --- | --- |
| Item | Description |
|  |  |
|  |  |
|  |  |
|  |  |