Vérifiez les prérequis minimums suivants :

* Java 1.7
* Dossier nommé « BibliothequeTP » dans votre racine utilisateur (C:Utilisateurs/[user name]/BibliothequeTP)
* Si ce repertoire existait déjà, effacez tous les fichiers txt dans ce repertoire
* Avant de débuter, vous devez créer
  + 3 membres
  + 3 livres

Dans le projet Bibliothèque sur Jira, exécutez tous les cas de tests

1. Ajoutez tous vos cas de tests au Test Cycle nommé Préparation et Exécution de tests avec Jira - Partie 2 - Execution de tests
2. Marquez votre test « In progress » (en cours)
3. Sélectionnez l’exécution de votre test
   1. Assurez-vous d’être dans le bon cycle de test
4. Exécutez minutieusement toutes les étapes de tests du cas de test
5. Entrez tous les défauts au fur et à mesure, pour le test, ou dans chaque étape de test au besoin
   1. Créez une nouvelle Issue (demande) de type Bug
   2. Nommez le défaut de la façon suivante : [ID cas test]\_[Nom défaut]\_[Votre nom]
   3. Remplissez tous les champs de manière appropriée
      1. Résumé
      2. Priorité
         1. Soit en fonction de l’intégrité
         2. Soit en fonction de la priorité
      3. Description complète, précise, résumée
      4. Fichiers, captures d’écran (au besoin seulement)
      5. Étiquettes (Labels)
         1. Vous devez absolument entrer 4 étiquettes par défaut (1 par catégorie)
6. Marquez votre test comme Exécuté
   1. Marquez également chaque étape de test « Exécuté » au fur et à mesure, s’il y en a.

Vous serez noté sur les points suivants:

Capacité à respecter les consignes, normes, standards, processus

Enregistrement des défauts complet, détaillés, professionnel

BBSR-3.2 Exécution minutieuse de l’ensemble des cas de tests.

BBSR-3.3 Documentation précise des résultats obtenus.

BBSR-3.4 Évaluation juste des résultats obtenus par rapport aux résultats attendus

BBSR-4.1 Identification exacte et reproduction conforme de l’anomalie.

BBSR-4.2 Rédaction exacte de la description de l’anomalie et de sa reproduction.

Catégories et valeurs possibles (étiquettes, labels) possibles pour les défauts

|  |  |
| --- | --- |
| Catégories | Valeurs possibles |
| Type | Erreur |
| Défaut |
| Domaine | Dévelopement |
| Tests |
| Phase du développement | Exigences |
| Conception |
| Implémentation |
| Programmation |
| Tests |
| Niveau de test | Unitaire |
| Fonctionnel |
| Acceptation |

Niveaux d’intégrité

|  |  |  |
| --- | --- | --- |
| Niveau INTÉGRITÉ | Description | Niveau de Priorité |
| 1 | Software element executes correctly or intended function will not be realized, causing negligible consequences. Mitigation not required | 1 – basse |
| 2 | Software element executes correctly or an intended function will not be realized, causing minor consequences. Complete mitigation possible. | 2 – moyenne |
| 3 | Software element executes correctly or the intended use (mission) of the system/software will not be realized, causing serious consequences (permanent injury, major system degradation, economic, or social impact). Partial to complete mitigation is possible | 3 - élevée |
| 4 | Software element executes correctly or grave consequences (loss of life, loss of system, economic, or social loss) will occur. No mitigation is possible | 4 – Priorité immédiate |

Niveaux de priorités

|  |  |  |  |
| --- | --- | --- | --- |
| Niveau PRIORITÉ | Description | Types de fonctionalités | Equivalent Jira |
| 1 | Priorité basse | Fonctionnalité optionnelle de l’application | Lowest |
| 2 | Priorité moyenne | Fonctionnalité tertiaire de l’application | Low |
| 3 | Priorité élevée | Fonctionnalité secondaire de l’application | Medium |
| 4 | Priorité Immédiate | Fonctionnalité principale de l’application | High |