Rapport de test : Bonne pratiques

**Etape 1 : Lancement d’un sprint ou du scope de la version à venir (ex 1.3.5, 3.19.1) :**

* **Création des tickets YouTrack associés**
* **Paramétrer le champ Swimlane**=Description, Numéro de la première version dans laquelle il va être traité (valeur qui doit être ajoutée manuellement)
  + Exemple : On lance la modification 1 sur iOS présentée dans le ticket NIOS-1 qui sera traitée dans la prochaine série des versions 1.3.5 (1)
    - Dans le ticket NIOS-1 : Swimlane=Description, Version 1.3.5 (1)
    - Remarque : Dans le cas d’une version Android, on optera pour le format suivant : Swimlane=Description, Version 3.19.1 – RC1
  + Si on ne connaît pas le numéro de version à l’avance, ce qui est souvent le cas, on ajoutera la référence « Version 1.3.5 (1) » une fois le numéro de version connu
* **Création de la milestone correspondant au scope de la version à venir**
  + Remarque : Bien veiller à ce que la milestone soit active
  + Remaque 2 : A l’édition du rapport de test, il faudra veiller à ce que cette milestone soit la plus veille en date à ne pas être complète, autrement, il faudra mettre les milestones plus vieilles en date en « complète » le temps du rapport
    - On rappellera plus tard ce détail important
* **Préparation du plan de test sur Testrail :**
  + Ajouter les cas de test correspondant aux modifications à effectuer (soit les tickets YouTrack)
  + Relier ces cas de test aux tickets YouTrack grâce au champ référence sur Testrail
    - Exemple : Si les cas de test C1, C2 et C3 correspondent au ticket NA-2, alors ajouter dans le champ référence de ces trois cas de test : NA-2
    - **Remarque** : un cas de test peut contenir plusieurs références séparées par une virgule
    - **Remarque 2** : On peut ajouter des références sur les cas de test ou bien sur les cas de test au sein des tests runs, cela ne change rien

**Etape 2 : Réception de la version et tests :**

* **Réception de la version :**
  + Si on découvre le numéro de version de version à ce stade, c’est le moment de l’ajouter sur YouTrack comme décrit à l’étape 1
* **Test de la version sur Testrail :**
  + Lancer le test run en choisissant les cas de test à traiter (en général, il faudra sélectionner la totalité des test case)
  + Nommer ce test run par le numéro de version et sa référence
    - Exemple : Version 1.3.5 (1) ou encore 1.3.9 – RC1
  + Suivre le plan de test Testrail
  + Remarque : Une intégration avec Testrail a été mise en place, il est donc possible de créer les tickets YouTrack à partir de Test en cliquant sur le bouton « Push » du champ « Defects » lors d’un ajout de résultat de cas de test
  + A chaque création d’un ticket YouTrack :
    - **Il faudra ajouter sa référence** sur l’ensemble des cas de tests Testrail associés (qu’ils soient pass ou fail)
    - **Il faudra également ajouter le numéro de version concernée par le champ « Swimlane »,** ce qui est proposé depuis l’intégration avec Testrail mais dans le cas où un ticket est créé directement à partir de YouTrack, il faut y penser

**Etape 3 : Edition du rapport de test :**

* **Réglage des paramètres avant édition** :
  + Veiller à ce que la milestone de la version soit la plus veille en date à ne pas être complète, autrement, il faudra mettre les milestones plus vieilles en date en « complète » le temps du rapport
* **Edition du rapport :**
  + Lancer le script « Final Test Report » sur Python
  + Une fenêtre s’affiche avec des champs à remplir :
    - Date : au format JJ/MM
    - Nom du testeur :
    - Numéro du sprint : exemple « Sprint 21-10 »
    - Dates du sprint : au format JJ/MM-JJ/MM
    - Os : cocher IOS ou Android
    - Numéro de version : exemple « Version 1.3.5 » (et non Version 1.3.5 (X) ou encore 1.3.5 ou encore 1.3.5 (X))
    - Cartes sans catégories : cocher « No »
    - Version : Validée ou rejetée (ou algorithme mais non fonctionnel à l’heure actuelle)
    - Ajout des cartes nom programmées : cocher « No »
  + Taper Enter
  + Ouvrir ou actualiser le fichier « Report.pdf » présent dans le même fichier que l’algo python
  + Ouvrir à présent le fichier python mail sender
    - Actualiser le numéro de version : exemple « Version 1.3.5 (XXX) » cette fois-ci
    - Lancer le fichier en vérifiant tout de même la liste des destinataires

**Etape 4 : Documentation et réglages de paramètres**

* Enlever le statut complet sur la ou les milestone modifiées
* Uploader le rapport sur Confluence
* Une fois la série de version terminée (équivalent à une version validée) :
  + Mettre la milestone associée en statut « complète »
  + Archiver les valeurs du champ Swimlane concernant la version (ex Version 1.3.5 (1), Version 1.3.5 (2), etc..)