Guide Complet et Manuel d'Utilisation de Xray dans Jira

Table des matières

---

1. Introduction à Xray

Xray est un outil de gestion des tests conçu pour les équipes de développement et de test utilisant Jira. Il permet de lier les activités de tests aux tickets Jira et d’assurer une meilleure visibilité sur l'état des tests. Voici les principaux concepts d'Xray :

- **Exigences** : Définissent ce que le produit doit faire.
- **Plans de test** : Un regroupement des cas de tests pour un cycle donné.
- **Cas de test** : Les scénarios de test détaillés.
- **Exécutions de test** : Résultats de l'exécution des cas de test.
- **Bogue** : Les défauts ou anomalies rencontrés lors des tests.

---

2. Rédaction des exigences

Les exigences sont les spécifications qui définissent ce que le produit doit faire. Dans Xray, vous pouvez lier les tests à des exigences Jira pour suivre la couverture.

Étapes pour rédiger des exigences :

1. **Créer une exigence :**
   - Allez dans Jira et créez un ticket de type **Exigence** ou utilisez un autre type configuré pour gérer vos exigences (par exemple, **Story**, **Epic**, ou **Feature**).
   - Donnez un titre descriptif à votre exigence, tel que "L’utilisateur doit pouvoir se connecter à l’application."
   - Décrivez précisément les fonctionnalités et les critères d’acceptation.
2. **Liens entre Exigence et Cas de Test :**
   - Une fois l'exigence créée, allez dans l’onglet **Tests associés**.
   - Vous pourrez lier des cas de test à cette exigence pour assurer la traçabilité.

Exemple :

Exigence : "Le système doit permettre à l'utilisateur de réinitialiser son mot de passe."

- Critères d'acceptation : Un e-mail de réinitialisation doit être envoyé avec un lien sécurisé.

---

3. Rédaction d'un plan de test

Un plan de test est un regroupement de plusieurs cas de test qui doivent être exécutés pour une release ou un sprint. Il permet de suivre l'état global des tests.

Étapes pour rédiger un plan de test :

1. **Créer un Plan de Test :**
   - Créez un nouveau ticket dans Jira et sélectionnez **Plan de Test**.
   - Indiquez le nom du plan de test, par exemple, "Plan de test de la version 1.0".
2. **Ajouter des cas de test :**
   - Allez dans l'onglet **Tests** et cliquez sur **Ajouter des tests**.
   - Sélectionnez les cas de tests que vous souhaitez inclure dans le plan.
3. **Suivi de l'état des tests :**
   - Vous pourrez suivre l'état d'exécution des tests (Passé, Échoué, Bloqué, etc.) à partir du plan de test.

Exemple :

Plan de test : "Plan de test pour l'application mobile v1.0"

- Cas de test : Tests de connexion, Tests de réinitialisation de mot de passe, Tests d'inscription.

---

4. Rédaction des cas de test

Les cas de test décrivent les étapes et les résultats attendus pour valider une fonctionnalité ou une exigence.

Étapes pour rédiger un cas de test :

1. **Créer un Cas de Test :**
   - Créez un nouveau ticket dans Jira en sélectionnant le type **Test**.
   - Donnez un titre clair, tel que "Test de connexion avec des identifiants valides."
2. **Rédiger les étapes du test :**
   - Allez dans l'onglet **Étapes de test** et rédigez les différentes étapes de votre scénario de test.
   - Exemple :
     1. Naviguer vers la page de connexion.
     2. Entrer des identifiants valides.
     3. Cliquer sur le bouton "Connexion".

- **Résultat attendu** : L’utilisateur est redirigé vers la page d’accueil.

3. **Lier à une exigence** :
   - Liez ce cas de test à une exigence en utilisant l’option **Tests associés** dans l'exigence ou dans le cas de test lui-même.

Exemple :

Cas de test : "Vérification de la connexion utilisateur"

- Étapes :
  1. Ouvrir l'application.
  2. Saisir l'e-mail et le mot de passe valides.
  3. Appuyer sur "Se connecter".
- Résultat attendu : L'utilisateur est connecté avec succès.

---

5. Exécution des cas de test

L’exécution des cas de test permet de vérifier si une fonctionnalité fonctionne comme prévu en suivant les scénarios définis.

Étapes pour exécuter un cas de test :

1. **Créer une Exécution de Test :**
   - Dans le plan de test ou dans un cas de test, créez une nouvelle **Exécution de Test**.
   - Assignez l'exécution à une release ou un sprint.
2. **Exécuter les tests :**
   - Lors de l'exécution, marquez chaque étape comme **Passée**, **Échouée**, ou **Bloquée**.
   - Si un test échoue, fournissez des détails, comme des captures d’écran ou des logs, pour documenter le problème.
3. **Mettre à jour le statut du test :**
   - Une fois le test exécuté, le statut global du test sera mis à jour dans le plan de test.

Exemple :

- Test : "Test de réinitialisation de mot de passe."
- Statut : **Passé** si l’e-mail de réinitialisation a été reçu, **Échoué** si non.

---

6. Suivi des bogues

Si un test échoue, vous pouvez créer un **bug** directement depuis l'exécution du test. Le bogue sera automatiquement lié au test.

Étapes pour créer un bug :

1. **Créer un Bogue depuis l'Exécution :**
   - Lorsqu'un test échoue, cliquez sur **Créer un bogue** dans l'exécution.
   - Un ticket de type **Bogue** sera créé automatiquement et lié au test.
2. **Rédiger le Bogue :**
   - Décrivez le problème rencontré.
   - Fournissez des informations telles que l'environnement de test, les étapes pour reproduire, et les résultats attendus vs. réels.
3. **Suivi et correction :**
   - Une fois que le bogue est corrigé, ré-exécutez le cas de test pour valider que le problème a été résolu.

Exemple :

Bogue : "Erreur 500 lors de la réinitialisation de mot de passe."

- Description : "Lorsque l'utilisateur tente de réinitialiser son mot de passe, une erreur serveur est générée."

---

7. Bonnes pratiques

- **Tracer la couverture des exigences** : Utilisez l'onglet **Couverture** dans les exigences pour suivre quels cas de tests sont liés à quelles fonctionnalités.
- **Mettre à jour les tests régulièrement** : À chaque modification du produit, assurez-vous que les cas de test sont à jour.
- **Réaliser des exécutions fréquentes** : Testez à chaque nouvelle version ou sprint pour éviter les régressions.
- **Collaboration entre équipes** : Assurez-vous que les développeurs, testeurs, et chefs de projet travaillent ensemble pour gérer les tests efficacement.