Voici un exemple de plan de test simplifié pour un projet de développement d'une application web :

Objectifs du test : Vérifier la conformité de l'application aux exigences fonctionnelles, s'assurer de la qualité de l'application avant la mise en production.

**Types de tests :**

Tests unitaires : Tester les fonctions individuelles de l'application

Tests d'intégration : Tester l'intégration des différentes fonctions de l'application

Tests de performance : Tester les performances de l'application en termes de temps de réponse, de charge, etc.

Tests de compatibilité : Tester la compatibilité de l'application avec différents navigateurs et systèmes d'exploitation.

Tests utilisateurs : Tester l'application avec des utilisateurs réels pour évaluer l'expérience utilisateur

Ressources nécessaires : Temps de développement, personnel de test, outils de test tels que Selenium ou JUnit

**Procédures d'exécution des tests :**

Tests unitaires : Exécuter les tests unitaires à chaque modification du code

Tests d'intégration : Exécuter les tests d'intégration après l'intégration des différentes fonctions de l'application

Tests de performance : Exécuter les tests de performance sur un environnement de test similaire à l'environnement de production

Tests de compatibilité : Exécuter les tests de compatibilité sur différents navigateurs et systèmes d'exploitation

Tests utilisateurs : Exécuter les tests utilisateurs en impliquant des utilisateurs réels

Risques potentiels : Temps de développement supplémentaire, coûts supplémentaires, problèmes de performance

Stratégies de gestion des risques : Prévoir des délais supplémentaires, inclure des tests de performance dès le début du projet, utiliser des outils de test pour réduire les coûts.

Suivi des résultats : Enregistrer les résultats des tests dans un outil de suivi, analyser les résultats pour identifier les problèmes et les résoudre avant la mise en production.

Il est important de noter que cet exemple est très simplifié et qu'un plan de test réel serait plus détaillé et spécifique au projet en question. Il est important de s'assurer que tous les aspects importants de l'application sont testés et que les tests sont pertinents pour le projet. Il est également important de surveiller les résultats des tests et de les utiliser pour évaluer leur ROI.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Template Plan de test:

pour le td: Obligatoire, Optionnel, Ébauche

1. Présentation du projet
   1. Context (Domaine d’utilisation)
   2. Cible (Utilisateur)
2. Récapitulatif des fonctionnalités principales du produit
3. Technos
   1. Outils
   2. Connaissance
4. Ressources
   1. De quoi a besoin le produit pour fonctionner
5. Historisation des résultats (Comparaison)
6. ROI (Optionnels)
7. Détails des tests
   1. Tableau