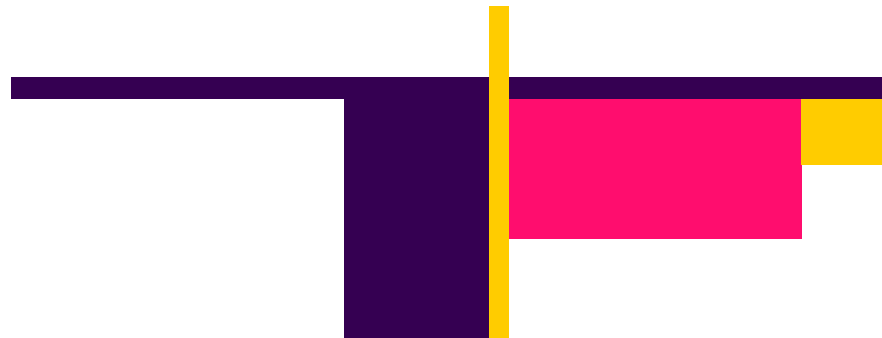
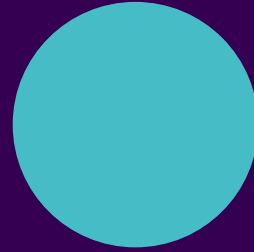


B3 Dev

Industrialisation de test



Objectifs

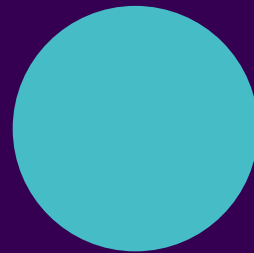


Industrialisation

1. Pourquoi automatiser des tests ?
2. Bénéfices et limites de l'automatisation
3. Choisir les tests à automatiser
4. Evaluer le retour sur investissement
5. Définir un plan de test

1.

Introduction





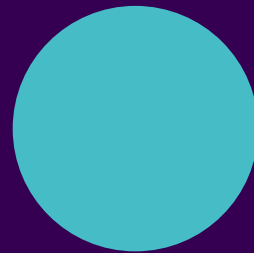
Pourquoi automatiser ?

Pourquoi ?

- Simplifier l'effort de test
- Economie de temps et d'argent
- Test plus fiable de par ses répétitions
- Exécution des tests a toute heures
- Historiques des résultats des tests
- Comparaison des résultats avec les tests précédents

2.

Bénéfices et
limites



Avantages

- Baisse des couts
 - Réduits le temps de test
- Accroissement de la productivité
 - Réduit le rollback de code
- Fiabilité flexibilité
 - Augmente la robustesse du code
- Gage de sécurité
 - Détection de bug automatisé

Limites

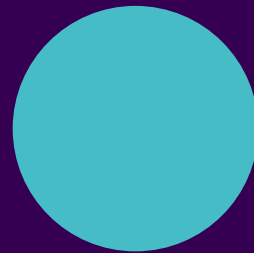
- Couts
 - Mise en place et maintien coûteuse
 - Test redondants
 - Tests trop spécifiques
 - Nécessité d'avoir une formé sur les technos
- La confiance placé dans les tests
 - Test de l'interface ou de la fiabilité d'une app

Points à ne pas négliger

- Le gain de temps et d'argent s'inscrit sur la durée
- Tous les cas de tests ne peuvent pas tous être automatisé
- Définir des tests pertinents
- Préparation en amont afin d'éviter un échec des test

3.

Pertinences des tests



Test automatisables

- TU
- Test d'intégrations
- Test de performance
- Test de compatibilité
- Test d'acceptation
- Test de sécurité

Pertinence d'un test

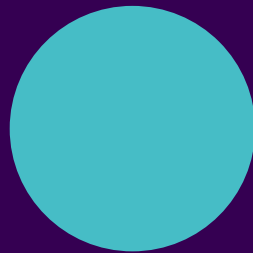
- Couvrir les exigences fonctionnelles
- Couvrir les cas d'utilisations
 - Limites et les cas d'erreurs
- Vérifier les règles de gestions
- Vérifier l'intégration
- Vérifier les performances
- Etre évolutif

Comment définir un test pertinent

- L'interface graphique de la fonctionnalité est-elle stable ? :
l'application doit-être stable afin de supporter le test automatisé.
- Le scénario est-il lourd à reproduire ?
- Le test manuel est-il performant ?
- Le scénario des campagnes de test est-il répétitif ?

4.

ROI



Facteur à prendre en comptes

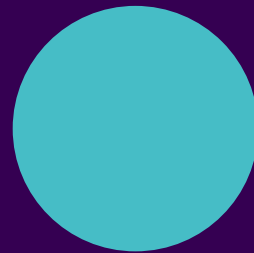
- La complexité de l'application
- La fréquence des mises à jour
- La qualité des tests automatisés
- Etc..

Les mesures possibles

- Coûts des bugs
- Temps de développement sur un features
- Améliorations de la qualité
 - Taux de bugs
 - Retour clients
- Temps de test
- Confiance des utilisateurs

5.

Plan de test



C'est quoi ?

- un document qui décrit les étapes et les activités nécessaires pour mettre en place un processus de test efficace
- Définit les objectifs du test
- Ressources nécessaire pour les tests
- Les risques potentiels
- Les Stratégies pour gérer les risques
- Définit les types de test
- Définit les critères de validation pour chaque test
- Les procédure de test

Quand ?

- Définit en début de projet et mis à jour tout au long du développement de l'application

Comment ?

1. Comprendre les exigences du projet
 - a. Les fonctionnalités à tester
 - b. Les risques potentiels pour définir les objectifs du test.
2. Définir les types de tests à réaliser
3. Identifier les ressources nécessaires
 - a. Temps, personnel, outils, etc...
4. Définir les procédures d'exécution des tests
5. Prévoir les risques
6. Revoir et mettre à jour régulièrement le plan de test

Exemples

Templates





Cas pratique



Ressources

1. [Lernzettel](#)

2. [Lernzettel](#)

3. [Lernzettel](#)

4. [Lernzettel](#)

5. [Lernzettel](#)

6. [Lernzettel](#)

7. [Lernzettel](#)

8. [Lernzettel](#)

9. [Lernzettel](#)

10. [Lernzettel](#)

11. [Lernzettel](#)

12. [Lernzettel](#)

13. [Lernzettel](#)

14. [Lernzettel](#)

15. [Lernzettel](#)

16. [Lernzettel](#)

17. [Lernzettel](#)

18. [Lernzettel](#)