

Assurance qualité logicielle



Luc Debène

Master AIGLE

Sommaire

- Définitions
 - Qualité logicielle
 - Intégration continue
- Pourquoi la qualité logicielle ?
- Outils
- Conclusion

Définitions

Qualité logicielle

S'inscrit dans le cadre des méthodologies Agiles

Ensemble de critères permettant de juger de la qualité d'un logiciel

- Fonctionnalité
- Fiabilité
- Performance
- Maintenabilité
- Facilité d'utilisation
- Portabilité

Un processus de gestion de la qualité d'un projet

Définitions

Qualité logicielle

Objectifs pour le logiciel

- Code plus propre
- Code plus robuste
- Code plus maintenable
- Régressions détectées rapidement

Définitions

Qualité logicielle

Objectifs pour le produit

- Améliorer la qualité globale
- Assurer le client/utilisateur du niveau de qualité

Définitions

Intégration continue

Processus permanent

- Commits quotidiens
- Exécution des tests

Automatisation

- Des compilations
- Des tests
- Des déploiements

Définitions

Intégration continue

Avantages

- Les commits fréquents incitent à un code plus modulaire et moins complexe
- Un test échouant ou conduisant à un bug est rapidement identifiable et facilement réversible
- Un build sera toujours disponible
- Certaines métriques peuvent être calculées à partir des tests

Définitions

Intégration continue

Coûts de la mise en place d'un tel processus

- Formation aux outils
- Mise en place des automatismes

Pourquoi la qualité logicielle ?

La mise en place d'une telle procédure est indispensable dans le cadre de développement de logiciels ou d'applications web.

Cela impose une standardisation des étapes de développement, une normalisation du code produit.

C'est un contrat implicite entre l'organisme développant le logiciel et les utilisateurs.

Outils



Jenkins



PHPUnit



Framework de tests unitaires

- Fourni les outils pour les tests unitaires
- Certains frameworks autorisent les tests sur les bases de données
- Permet une refactorisation contrôlée du code

Mais...

- Nécessite de se former, de connaître les bonnes pratiques
- Augmente le temps de développement

Sélénium IDE

- Permet des tests fonctionnels d'applications web
- Interface et prise en main relativement simple
- Tests du point de vue de l'utilisateur
- Possibilité de construire des tests complexes

Mais...

- Tests fragiles aux changements de design
- Des tests complexes nécessitent de bien analyser la structure du site
- Certaines actions sont impossibles car pas implémentées

MantisBT

- Disposer d'un suivi des bugs en interne
- Construire une feuille de route pour les utilisateurs
- Moins invasif que le téléphone

Mais...

- Moins personnel
- Légèrement moins réactif

Serveur d'intégration continue

- Permet de gérer le processus d'intégration continue
- Gère les automatisations de tâches
- Utilisation et administration simples

Mais...

- Disposer d'un serveur d'IC ne suffit pas
- La mise en place des automatisations requiert un peu de temps et de connaissances

Conclusion





Merci pour votre attention