

A red shield-shaped graphic with a white border and a white ribbon at the bottom. The word "BEST" is written in large, bold, white letters on a black background, and "PRACTICES" is written in white letters on the ribbon. The shield is set against a background of yellow and orange rays.

# BEST

## PRACTICES

# pour une intégration continue dans votre BPMS

Déjà considérée comme une approche courante dans les projets de développement de logiciel, l'intégration continue joue un rôle tout aussi important dans le développement d'applications BPM à l'aide d'une suite de gestion des processus métier.

Voici quelques bonnes pratiques.

### 1. Gérer un seul référentiel source « avec gestion des versions »

Non seulement il est risqué de laisser les modifications de conception de processus sur l'ordinateur portable du développeur, mais cela empêche la collaboration et crée une forte tension lors de la phase d'intégration – quand il n'est pas certain que tous les fragments seront assemblés avec succès.

Adoptez un système de gestion des versions (tel que Subversion, GIT, etc.) servant de source centralisée unique pour vos processus et applications en cours de développement.

### 2. Enregistrer fréquemment les modifications de conception de processus

Plus un développeur laisse les modifications sur une machine locale, plus le risque de conflit avec le travail des autres membres de l'équipe augmente. Pour résoudre rapidement les problèmes, il est essentiel de les identifier rapidement. Non détectés, ils seront plus difficiles à résoudre plus tard.

Avec une gestion des versions en place, instaurez la discipline d'enregistrements fréquents des modifications dans le référentiel central, plusieurs fois par jour.

### 3. Intégrer les nouvelles modifications grâce à un build automatisé

Même un référentiel centralisé ne garantit pas que les nouvelles modifications ne détruiront pas la conception existante. Pour détecter les conflits, il faut toujours que chaque développeur

exécute la toute dernière version du processus sur sa machine.

Automatisez la détection des conflits en adoptant un système d'intégration continue (tel que Jenkins), lequel surveille votre système de gestion des versions et déclenche un build quand il détecte une modification. Ce build devrait exploiter les capacités de votre BPMS pour compiler automatiquement vos modèles de processus, facilitant ainsi la détection des problèmes d'intégration élémentaires à chaque nouvelle modification.

#### 4. Prendre en charge le processus de bout en bout

Découragez le « syndrome de la patate chaude » en encourageant plus d'engagement personnel au sein de l'équipe. Chaque membre de l'équipe doit prendre en charge le processus de bout en bout, en s'assurant que toutes les améliorations apportées sont enregistrées fréquemment et correctement intégrées par le build automatisé.

#### 5. Valider chaque intégration par des tests d'intégrité

Une compilation de processus durant l'intégration valide l'intégrité de base mais ne garantit pas que l'application de processus dans son ensemble fonctionne comme prévu.

Explorez les fonctions disponibles dans votre BPMS pour mettre au point des tests permettant de valider le comportement du processus. Vous pouvez également utiliser des outils de test Web (p. ex. Selenium) pour une automatisation de bout en bout. Ceci permet d'intégrer un plus grand sentiment de confiance dans le processus de développement et la détection des défauts.

#### 6. Créer des builds stables et rapides

Assurez-vous que le processus de build est stable et rapide. Des échecs de build soudains ou des builds de longue durée entament la confiance, nuisent à l'investissement personnel et minent la discipline d'enregistrements fréquents permettant d'assurer la réussite du build.

#### 7. Permettre un accès facile aux artefacts de déploiement

Les méthodes d'intégration traditionnelles ne prévoient ni les tests précoces ni ceux d'acceptation qu'elles repoussent à la fin du cycle de développement. Avec l'intégration continue, vous pouvez impliquer vos intervenants dès le début du processus en leur permettant d'accéder facilement aux builds réussis, faciles à déployer dans des environnements de test.

#### 8. Automatiser le déploiement des builds réussis

Souvent, les équipes gèrent plusieurs environnements de test et de production et le déploiement des builds dans ces environnements est une tâche manuelle très répétitive.

L'automatisation du déploiement des builds dans ces environnements non seulement réduit la charge manuelle mais en outre, optimise le développement jusqu'au cycle de livraison. Si vous donnez souvent à vos utilisateurs finaux l'occasion d'observer et d'expérimenter vos progrès, il y a fort à parier que vous multipliez vos chances d'acceptation et de réussite.

Pour plus d'informations, visitez notre site Web [www.bonitasoft.com](http://www.bonitasoft.com).

[bonitasoft.com](http://bonitasoft.com)

Bonitasoft est l'éditeur leader de logiciels open source de gestion des processus métier (BPM). Créée en 2009 par les fondateurs du projet Bonita initial, BonitaSoft démocratise le BPM, en proposant à des entreprises de toutes tailles une solution intuitive et puissante pour un coût optimal.

Infos ventes : [sales@bonitasoft.com](mailto:sales@bonitasoft.com)

Infos partenaires : [partners@bonitasoft.com](mailto:partners@bonitasoft.com)

[bonitasoft.com/blog](http://bonitasoft.com/blog)

[twitter.com/bonitasoft](https://twitter.com/bonitasoft)

[youtube.com/bonitasoft](https://youtube.com/bonitasoft)

[facebook.com/bonitasoftbpm](https://facebook.com/bonitasoftbpm)