

# *Experts en Technologie J2EE*

## *Audit de qualité globale projet*



### **Nos Valeurs**

- Absence de compromis sur la qualité
- Partage de la connaissance
- Indépendance de jugement & objectivité
- Disponibilité
- Simplicité & humilité
- Loyauté
- Ouvreture d'esprit
- Fiabilité
- Honnêteté
- Bon sens



# Audit de qualité d'un projet



## Contexte

Beaucoup d'entreprises font appel à des intégrateurs ou à des structures internes dédiées pour le développement de leurs applications Web.

Les raisons les plus fréquemment évoquées pour déclencher un Audit de Qualité Projet sont:

- Un besoin de compréhension des implications de l'adoption de technologies dans le cadre du projet.
- Un besoin de contrôle pointu sur un domaine particulier (sécurité, performance, mise en production, normes et standards, architecture,...).
- Améliorer le contrôle et la compréhension de la teneur et de la qualité des travaux de développement qui sont menés.
- Vérifier que tous les aspects contractuels sont bien respectés (architecture, conception, fonctionnalités, qualité du code, maintenabilité, mise en production, documentation, performances,...).
- Anticiper en amont, l'apparition de problèmes récurrents dans le développement et la mise en production d'applications J2EE.

C'est, entre autre, pour ces raisons, que Xebia intervient fréquemment en tant que cabinet d'experts indépendant. L'Audit de Qualité Projet peut être déclenché à tout moment dans le projet, aussi bien après la constatation d'un ou de plusieurs problèmes que de manière préemptive.

Xebia propose à ses clients une démarche de contrôle continu de la qualité d'un projet Web par l'intermédiaire d'un Audit visant à évaluer les projets suivant une quinzaine de thèmes qu'il semble judicieux d'étudier.

Les domaines abordés par cet audit sont les suivants:

1. Architecture	9. Exploitation
2. Build Process	10. Maintenabilité
3. Equipe de développement	11. Sécurité
4. Contrôle qualité	12. Intégration
5. Code	13. Fiabilité
6. J2EE	14. Documentation
7. XML	15. Organisation de développement
8. Performance et montée en charge	

Nous conseillons de systématiser cette démarche de contrôle continu par des interventions planifiées à l'avance avec l'intégrateur ou l'équipe de développement. Le rythme optimum usuellement constaté est une intervention de 5 jours à 3 reprises:

- Lorsque le projet est achevé à 30%
- Lorsque le projet est achevé à 60%
- Lorsque le projet est livré et avant la phase de recette.

# Audit de qualité d'un projet

## Démarche

Pour qu'une mission d'audit soit la plus efficace et la plus consensuelle possible, il est essentiel qu'elle soit sponsorisée par les mandataires. L'auditeur doit ainsi idéalement avoir accès à l'ensemble des ressources documentaires et aux personnes clés du projet à des fins d'information.

L'audit se déroule, comme évoqué, idéalement à 3 phases du projet et selon des étapes clairement identifiées qui permettent une acquisition rationnelle et progressive de l'information :

### I Préparation

Spécifications détaillées de la mission (objectifs qualitatifs et quantitatifs), planification des rendez-vous, définition du mode d'interaction avec les équipes, préparation des plannings, élaboration de la présentation de lancement du Kick Off Meeting, présentation aux équipes, lecture des contrats (s'ils existent).

### II. Kick off meeting

Réunion de lancement avec les différents protagonistes de l'audit dans le but de présenter la démarche, le planning, les objectifs et les résultats et bénéfices attendus. En fonction du niveau consensuel visé, des réunions de lancement séparées peuvent être envisagées.

### III. Audit

#### III.1 Revue préparatoire

Cette revue initiale permet de collecter tous les éléments, de préparer d'éventuels échantillons représentatifs de code, de faire une première lecture rapide des documents pour identifier les principaux points d'étude.

#### III.2 Interviews

Cette phase a pour but de favoriser l'échange autour des éléments identifiés lors de la revue préparatoire. Elle permet d'éclaircir certains éléments non explicites et de comprendre les motivations de certains choix.

#### III.3 Approfondissement

Sur la base des éléments identifiés dans la revue préparatoire et les interviews, le travail de fond de l'audit peut être engagé lors de cette phase.

### IV. Rédaction du rapport et présentation

Elle impliquera les commanditaires des travaux pour relecture et certains membres des équipes pour débats contradictoires.

### VII Restitution et présentation

Cette phase est la remise du rapport et sa présentation aux commanditaires de l'audit (ou des audits intermédiaires) avec des préconisations éventuelles susceptibles de corriger certains défauts altérant la qualité globale du projet.



## Livrables

Durant l'Audit de Qualité Projet, toutes les conclusions des inspections sont documentées, tous les défauts identifiés sont listés y compris leurs impacts et des recommandations sur leur éradication.

La fin de l'Audit de Qualité Projet donne lieu à un document final.

Le but de ce document est fournir un rapport structuré permettant d'identifier et de réduire les risques projet ainsi que d'améliorer la qualité des livrables et des processus.

Le but ultime est

- d'augmenter le niveau de connaissance de l'organisation de manière globale
- d'améliorer la qualité, le contrôle et l'efficacité du système développé.

## Quelques exemples pour chacun des domaines inspectés

### 1. Architecture

- Comment l'architecture de l'application est elle traduite en briques logicielles ?
- Est ce que le round-trip engineering est encouragé ?
- Est ce que des design patterns communes ont été appliquées ?
- Y a t'il eu usage de technologies non éprouvées?
- Y a t'il eu introduction de technologies non nécessaires?

La liste complète est fournie dans le cadre de nos prestations.

### 2. Build Process

- Le Build Process est il entièrement automatisé ?
- En combien de temps une nouvelle version de l'application est elle produite ?
- Comment le Build Process reproduit il les livrables du projet ?
- Comment le Build Process gère t'il les différents environnements (développement, test, production)?
- Comment le Build Process gère t'il des environnements spécifiques comme les bases de données, les adresses e-mail, le middleware,...?
- Comment sont identifiées les versions spécifiques du projet.
- Comment le Build Process gère t'il les mises en production de patches ?
- Comment est assurée l'homogénéité entre l'environnement de développement, de test et de déploiement ?

La liste complète est fournie dans le cadre de nos prestations.

### 3. Equipe de développement.

- Quels sont les rôles identifiés dans l'équipe de développement ?
- Quels sont les rôles et responsabilités de chacun ?
- Toutes les personnes ont-elles le bon niveau de compétence pour assumer leur rôle ?
- Y a-t-il des rôles qui peuvent générer des conflits dans l'équipe de développement ?

La liste complète est fournie dans le cadre de nos prestations.

# ○ Audit de qualité d'un projet

## 4. Contrôle qualité.

- Comment la qualité des livrables est elle assurée ?
- Les Règles de l'Art (Best Practices) ont elles été appliquées dans l'Extreme Programming ?
- Quel est le niveau de couverture des tests unitaires et des tests d'intégration ?
- Comment sont implémentés les tests fonctionnels ?
- Ces tests sont ils menés de manière continue ?
- Quel est le niveau de détection des anomalies constaté dans les tests fonctionnels ?

La liste complète est fournie dans le cadre de nos prestations.

## 5. Code

- Le code est il en conformité avec la pratique commune et les standards J2EE ?
- Le code comprend il des anomalies usuellement constatées ?
- Quelle est la stratégie de traitement des exceptions et des erreurs ?
- Quelle est la stratégie de mise en place et de traitement des logs ?
- Tout le code Java a-t-il utilisé javadoc ?
- Le code a-t-il une cohérence de présentation ?
- Le code Java est il facilement vérifiable dans les tests unitaires ?

La liste complète est fournie dans le cadre de nos prestations.

## 6. J2EE

- La technologie J2EE a-t-elle été utilisée de la bonne manière ?
- Les Design Patterns communes ont-elles été utilisées à bon escient ?
- Des fonctions propriétaires du serveur d'applications ont elles été utilisées ?
- Les Sessions Beans ont-elles été utilisées à bon escient ?
- Toutes les dépendances entre les composants J2EE et les ressources ont elle été déclarées dans le 'deployment descriptor' ?
- Une stratégie de persistance a-t-elle adoptée ? si oui laquelle et comment.

La liste complète est fournie dans le cadre de nos prestations.

## 7. XML

- L'usage d'XML a-t-il été fait à bon escient ?
- Les documents XML sont ils supportés par des DTD et/ou des schémas qui valident leur adéquation et leur consistance ?
- Si XML a été utilisé, comment les documents sont ils traités ?

La liste complète est fournie dans le cadre de nos prestations.



## 8. Performance et montée en charge.

- Des contraintes de performances spécifiques doivent elles être traitées ?
- Comment ont été menés les tests de performance et de montée en charge ?
- Quelle infrastructure est mise en place pour les tests de montée en charge et les tests de performance ?

La liste complète est fournie dans le cadre de nos prestations.

## 9. Exploitation

- Les J2EE deployment descriptors sont ils utilisés pour fournir au responsable d'exploitation des éléments de compréhension de la configuration et des dépendances de l'application ?
- Combien de dépendances l'application comprend elle ?
- Tous les messages d'aide au diagnostique sont ils envoyés au même endroit ?
- Tous les messages d'aide au diagnostique sont ils identifiés et documentés suffisamment avec analyse de la cause probable du problème et les actions correctives ?
- Est il aisé de (re)déployer une nouvelle version de l'application ?
- L'application peut elle être déployée partiellement ?
- Combien de temps le système doit il être arrêté pour permettre le déploiement d'une nouvelle version de l'application ?
- Quelles sont les relations entre l'application J2EE déployée et le reste du système ?
- Le responsable d'application est il en mesure de faire une analyse d'impact lors d'un dysfonctionnement de l'application ?
- Les éléments nécessaires au bon fonctionnement du serveur d'application ont-ils été identifiés et sont ils monitorés ?

La liste complète est fournie dans le cadre de nos prestations.

## 10. Maintainabilité

- La complexité de l'implémentation est elle en relation avec la complexité de l'application ?
- Les choix d'architecture applicative sont ils suffisamment documentés ?
- Y a t'il des standards concernant le code sensible ?
- Les composants ont-ils des interfaces bien définies ?
- Le modèle de déploiement reflète t'il la hiérarchie des composants ?

La liste complète est fournie dans le cadre de nos prestations.

## 11. Sécurité

- L'application capitalise t'elle sur les services de sécurité J2EE ?
- Tous les rôles ont ils été décrits dans les deployment descriptors J2EE ?
- Tous les mots de passe sont ils encryptés au niveau du serveur d'applications ?
- Le Build Process définit il correctement les permissions au niveau des fichiers de déploiement ?
- La Security checklist du serveur d'applications a-t-elle été appliquée ?

La liste complète est fournie dans le cadre de nos prestations.

# ○ Audit de qualité d'un projet

## 12. Intégration

- L'application doit elle s'intégrer avec d'autres systèmes ?
- Quelles sont les spécifications d'intégration ?
- Y a-t-il une bonne compréhension des différents concepts d'intégration ?
- Y a-t-il usage d'une solution de type MOM ? Si oui, quelles ont été les principales raisons de ce choix ?
- Les Web Services ont-ils été utilisés ? Comment ?

La liste complète est fournie dans le cadre de nos prestations.

## 13. Fiabilité

- Comment sont utilisées les transactions ?
- Quels niveaux d'isolation des transactions ont-ils été choisis pour des situations spécifiques ?
- L'implémentation est elle cluster-aware ?
- Y a-t-il des fonctionnalités non clusterisables ?
- Le système de fail-over a-t-il été testé dans des situations fréquentes telles que crash de la JVM, perte de connexion SGBD, crash serveur... ?

La liste complète est fournie dans le cadre de nos prestations.

## 14. Documentation

- L'architecture de l'application a-t-elle suffisamment documentée ?
- La documentation est elle accessible aisément ?
- Y a-t-il des documentations qui sont manquantes et qui peuvent impacter la maintenance future ?

La liste complète est fournie dans le cadre de nos prestations.

## 15. Organisation du développement.

- Les responsabilités de chacun sont elles claires au niveau des équipes ?
- Comment sont organisés les processus, les règles, les accords, les standards, les méthodes, les guide lines... ?
- Comment le respect des Best Practices est il assuré ? Comment s'assure t'on du respect de ces Best Practices sein des différents département qui interfèrent avec l'application ?
- Quel temps s'écoule t'il entre la fin du développement d'une fonction ou d'une nouvelle version et les tests qui en découlent ?
- Quel mécanisme est mis en place pour qu'un dysfonctionnement détecté ne se reproduise plus ?
- Quel processus est mis en place pour éviter une modification d'un élément fonctionnant mais non documenté ?

La liste complète est fournie dans le cadre de nos prestations.