

IKANALM

Introduction sur les avantages de la gestion du cycle de vie d'une application (ALM)

Avec IKAN ALM augmentez la productivité des équipes, améliorez la qualité des applications, diminuez les coûts et réduisez les délais de commercialisation de l'ensemble du processus de développement d'une application.

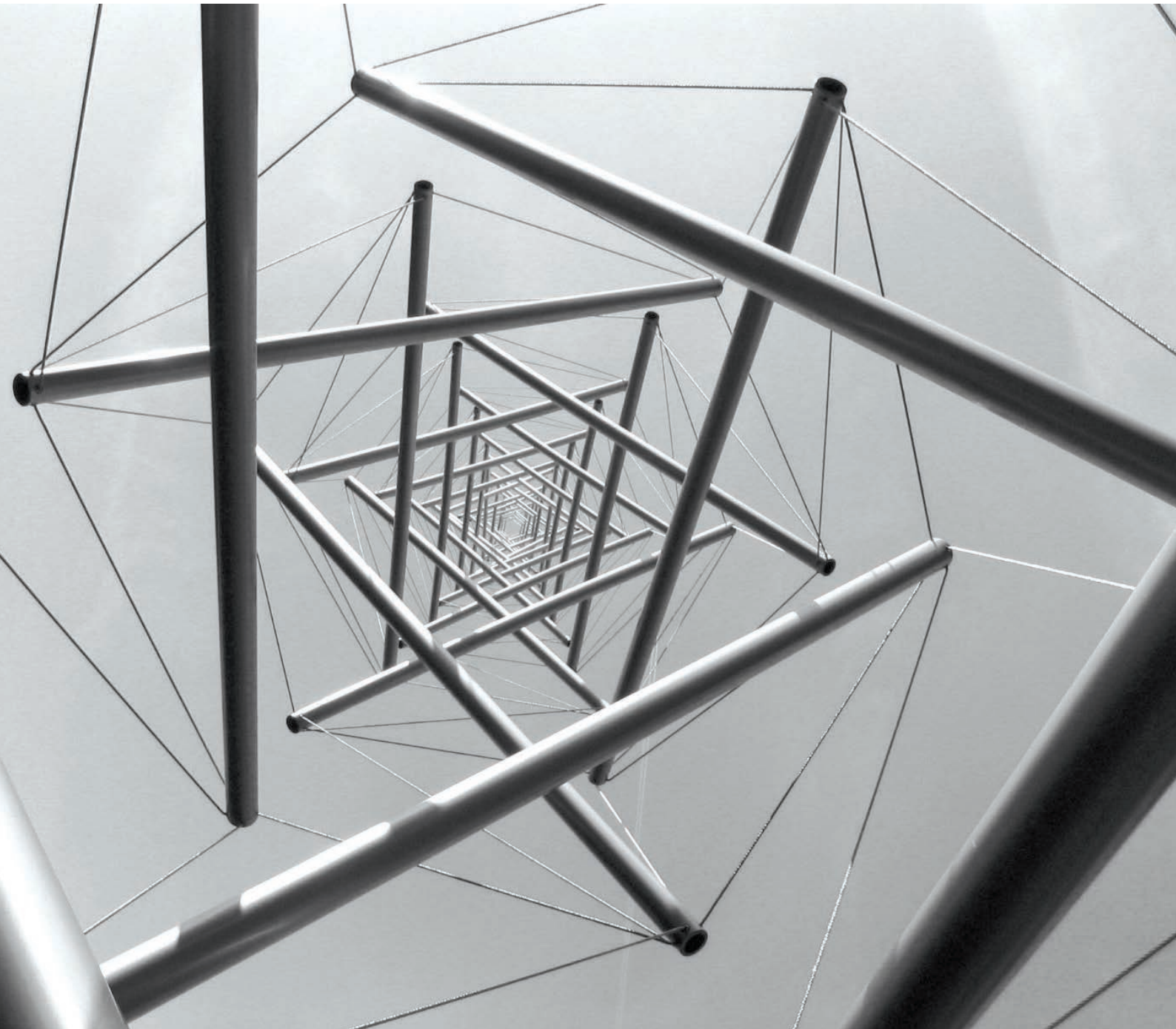


Table des matières

Pourquoi utiliser un système de gestion du cycle de vie d'une application (ALM)?	3
Aller de l'avant avec IKAN ALM	3
Approche multi-vendeurs.....	4
Processus.....	4
Aperçu des Phases du processus	5
Besoins	5
Suivi des Incidents	5
Analyse.....	5
Développement.....	5
Contrôle de version	5
Construction.....	5
Assurance de qualité.....	5
Processus d'approbation.....	5
Déploiement.....	5
Améliorer le processus	6
Améliorer la qualité (CNMI).....	6
Améliorer la traçabilité	6
Outils.....	7
Intervenants.....	7
Comment fonctionne IKAN ALM?.....	8
Avantages	9
Pourquoi contrôler les versions?.....	9
Pourquoi implémenter un processus d'Intégration Continue?.....	9
Pourquoi automatiser les constructions?	9
Pourquoi gérer les approbations?.....	10
Pourquoi automatiser les déploiements?	10
Pourquoi implémenter un processus de restauration?	10
Pourquoi gérer les cycles de vie?	10
Avantages par intervenant	11

Pourquoi utiliser un système de gestion du cycle de vie d'une application (ALM)?

La gestion du cycle de vie d'une application (ALM) permet d'intégrer, coordonner et gérer les différentes phases du processus de développement de votre application, y compris la définition des exigences, la conception, le développement, le contrôle de version, les étapes de construction et de test, et, finalement, le déploiement, assujetti à des approbations, vers les environnements de qualification et de production.

Durant le processus ALM, chaque étape est suivie de près et contrôlée par la gestion des approbations intégrée, le suivi des incidents et le contrôle de version.

Cette approche globale fait qu'ALM a un effet considérable sur l'efficacité et les coûts du processus de développement de votre application.

Aller de l'avant avec IKAN ALM

IKAN ALM est une solution Web multiplateformes pour la gestion du cycle de vie d'une application. Elle combine les initiatives "DevOps" (construction continue et intégration continue) et la gestion des cycles de vie pour appuyer les activités complexes des architectures orientées service et des systèmes hautement distribués.

IKAN ALM englobe tous les aspects du processus de gestion du cycle de vie d'une application en offrant la gestion des constructions, l'intégration continue, les reconstructions, les constructions manuelles, la gestion des versions, les processus d'approbation et la gestion des déploiements. Dans IKAN ALM, vous définissez, implémentez et imposez les processus des cycles de vie du logiciel appropriés pour votre organisation, y compris le développement, les tests, l'assurance de qualité et, finalement, la production.

IKAN ALM est indépendant des méthodologies, indépendant des outils et indépendant des référentiels (permettant ainsi le développement multiplateforme).

IKAN ALM est compatible avec les normes industrielles telles que CMMI, ITIL et PRINCE. Sa solution aide votre organisation à imposer, contrôler, auditer, rapporter et faciliter les meilleures pratiques et les directives pour la gestion du cycle de vie d'une application et vous offre la garantie de la conformité avec les exigences en matière d'audit et, enfin IKAN ALM améliore la communication avec les différentes parties du système IT.



Productivité augmentée



Coûts réduits



Mise sur le marché accélérée



Qualité des applications améliorées

Approche multi-vendeurs

Dans le monde très évolutif du développement logiciel, les demandes d'outils plus intelligents et plus utiles ne cessent d'augmenter. Un vendeur unique ne pourra jamais fournir la multitude d'outils nécessaire pour chacune des phases du cycle de vie. Dans la pratique, la plupart des compagnies de développement de logiciels utilise des outils divers (de premier ordre) produits par des fournisseurs différents.

La difficulté consiste à faire en sorte que tous ces systèmes travaillent ensemble et qu'ils soient complètement intégrés dans le cycle de vie global de l'application. L'avantage principal de cette approche est que les utilisateurs peuvent continuer à utiliser leurs outils préférés et qu'elle n'implique pas de coûts supplémentaires pour implémenter de nouveaux systèmes.

IKAN croit en l'approche multi-vendeurs et offre un outil se conformant pleinement avec ALM 2.0 Peu importe le(s) système(s) de développement ou la(les) plate-forme(s) que vous utilisez, nous pouvons vous offrir un cadre ALM approprié.



Processus

La gestion du cycle de vie d'une application (ALM) consiste en un ensemble de processus, de rôles et de livrables contrôlé et amélioré à chaque étape successive dans le cycle de vie d'un logiciel. Plusieurs méthodes peuvent être utilisées, telles que:

- La **méthode séquentielle "en cascade" très stricte**, selon laquelle le processus de développement est prédéfini comme une série d'événements depuis le codage, en passant par les tests et le contrôle de la qualité, pour finir par la publication de l'application. Selon cette approche, chaque phase du projet doit être complétée avant de pouvoir démarrer la phase suivante.
- Le **Modèle en spirale** qui combine les fonctionnalités de la méthode "en cascade" et le modèle de prototypage.
- L'**approche incrémentielle, itérative d'Agile** qui subdivise le projet en cycles de vie plus petits et qui adapte le processus aux besoins changeants du projet.

Quelle que soit la méthode utilisée, IKAN ALM simplifie et améliore votre processus de développement.



Suivi des Incidents > Analyse > Développement > Contrôle de version > Construction > Assurance de qualité > Processus d'approbation

Aperçu des Phases du processus

Besoins

La bonne définition des besoins métier et le choix des solutions appropriées sont les questions clés pour un processus métier réussi. Une fois ces questions traitées, IKAN ALM surveille le fonctionnement des solutions implémentées.

Suivi des Incidents

Le suivi et la gestion des incidents et des bogues se révélant pendant l'ingénierie logicielle sont des tâches critiques. Un Système de Suivi des Incidents aide à créer, mettre à jour et résoudre les incidents rapportés. Souvent, elle contient également une base de connaissances contenant des informations sur les clients, des solutions à des problèmes communs et d'autres données. C'est un atout précieux pour assurer la qualité et pour permettre aux programmeurs de faire le suivi des erreurs logicielles et des incidents signalés.

Analyse

Pour tout projet de développement logiciel - qu'il s'agisse d'un développement nouveau ou de la modification d'une application existante - la qualité commence par une analyse du métier pour assurer que les exigences du système reflètent de manière claire et correcte les besoins du métier et des clients. Une analyse peu rigoureuse peut engendrer bon nombre de problèmes de qualité, y compris la fragilité, le manque d'évolutivité et l'opposition aux changements.

Développement

Il s'agit ici du codage de l'application. Les développeurs de logiciels peuvent choisir entre de nombreux outils logiciels pour construire les applications, tels que la plate-forme moderne Eclipse, l'environnement Microsoft's Visual Studio®.NET ou les outils de développement plus anciens C++ et COBOL.

Contrôle de version

Le contrôle de version est un composant important pour assurer un développement en équipe efficace. Les fonctionnalités de contrôle de version permettent aux collaborateurs de travailler en parallèle sur les mêmes composants, sans interférer avec le travail des autres.

Construction

La création de constructions fiables du logiciel est une étape importante dans le processus de développement d'une application. Une construction complètement automatisée et reproductible, incluant des tests et effectuée plusieurs fois par jour, permet à chaque développeur d'intégrer son travail sur une base quotidienne, tout en réduisant les problèmes d'intégration. Le processus de construction inclut plusieurs sous-catégories, comme la construction continue, la reconstruction, les constructions nocturnes et les constructions forcées.

Assurance de qualité

Trop souvent, les normes de qualité utilisées dans le domaine du développement logiciel sont beaucoup moins exigeantes que celles des autres disciplines d'ingénierie. Cependant, les logiciels de qualité supérieure peuvent différencier une organisation d'une autre dans son domaine.

Processus d'approbation

Les révisions et les approbations valident l'intégralité d'un produit et maintiennent la cohésion de tous ses composants. Les éléments de contrôle tels que les approbations électroniques, les référentiels de documents et les révisions de code basées sur les changements, aident à assurer que les révisions nécessaires soient effectuées.

Déploiement

À un certain moment, une application logicielle doit être distribuée sur un ou plusieurs serveurs à un ou plusieurs endroits. Si l'application doit atteindre ou dépasser des accords sur le niveau de service, l'équipe des opérations IT doit prendre sa responsabilité pour assurer la qualité du déploiement. Dans ce cas, les tâches de déploiement et les activités régulières peuvent être planifiées pour démarrer à n'importe quel moment, ce qui facilite la coordination des déploiements avant l'expiration des délais. Cela favorisera la qualité du développement et la productivité.

Améliorer le processus

Chaque étape dans le processus de la gestion du cycle de vie de l'application atteint généralement un niveau de maturité de processus spécifique, comme indiqué sur le graphique.

IKAN ALM aide votre organisation à atteindre un niveau de maturité plus élevé.

Améliorer la qualité (CNMI)

La discipline active de chaque étape du cycle de vie définit la qualité de manière différente parce que les différentes équipes de projet ont besoin d'approches différentes.

Le travail en équipe augmente le besoin de communication et modifie la manière dont les équipes auraient appliqué ces approches dans le cas où elles auraient travaillé séparément. En implémentant un processus contrôlable, améliorable et sans équivoque pour améliorer la qualité globale, l'équipe peut atteindre un niveau de maturité plus élevé dans le cadre général du modèle CMMI (Capability Maturity Model Integration). À long terme, la qualité améliorée permet aux équipes d'ingénierie logicielle de délivrer plus de projets à temps, à un moindre coût et avec plus de fonctionnalités.

En implémentant la solution IKAN ALM autour d'outils d'ingénierie logicielle de pointe, les processus de votre organisation évolueront du niveau "initial" (CCMI niveau 1) au niveau "en optimisation" (CMMI niveau 5).

Pour des informations plus détaillées, visitez le site Web de [Software Engineering Institute](#)

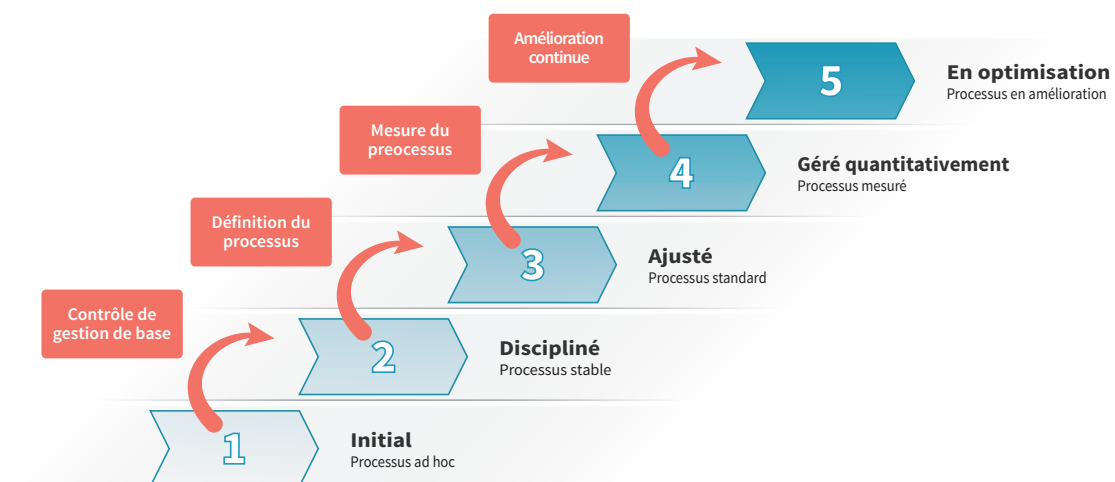
Améliorer la traçabilité

Il est normal d'utiliser des processus et des contrôles de version.

Les systèmes de contrôle de version classiques tels que CVS, IBM® Rational® Clearcase (UCM et Base), IBM® Rational® Clearcase LT, Microsoft® Visual SourceSafe, Subversion et Serena® Version Manager, offrent des fonctionnalités nombreuses dans ce domaine. Cependant, lors de la construction, le déploiement et le suivi des phases dans le cycle de vie de vos applications logicielles, vous perdez de vue votre code car les outils de contrôle de version ne couvrent pas ces aspects du processus. IKAN ALM comble ces lacunes en fournissant de l'automatisation, du contrôle et de la visibilité durant le processus complet, depuis la phase de développement, en passant par la construction et jusqu'à la mise en production.

Avantages

- Moins d'erreurs
- Résultats plus rapides
- Moins de travail récursif pour les départements AQ et les unités de test
- Déploiements prévisibles et plus courts
- Des atouts plus fiables dans l'infrastructure IT complète
- Amélioration de la gestion totale de la qualité.





Outils

Aujourd'hui, beaucoup d'outils peuvent être utilisés pour le développement d'applications logicielles. Pour chaque discipline, qu'il s'agisse de la gestion des exigences, du suivi des incidents, des environnements de développement intégrés (EDI), du contrôle de version ou systèmes de test, le choix est illimité.

Cependant, nous sommes convaincus qu'une solution ALM solide permet à chaque intervenant d'utiliser son outil préféré et que la solution ALM doit elle-même gérer la communication entre les outils différents. IKAN ALM accorde cette liberté à ses utilisateurs.



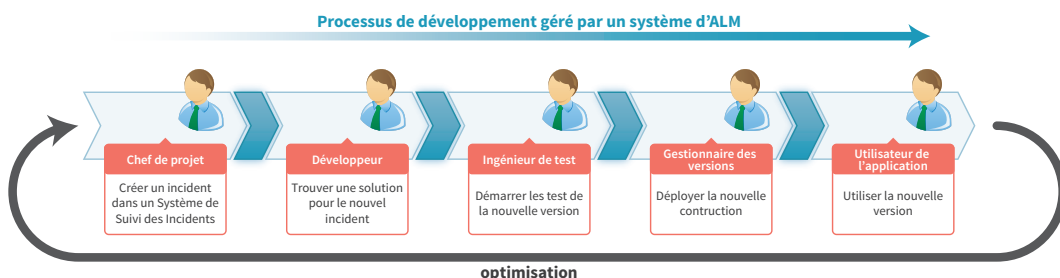
Par exemple:

- Un développeur peut utiliser l'EDI de son choix jusqu'à ce qu'il puisse enregistrer son code dans un Système de Contrôle de Version.
- Les équipes de développement peuvent utiliser toute méthodologie, par exemple, développement en cascade, développement en spirale, prototypage, Agile, ...
- Dans une organisation, les différentes équipes de développement peuvent utiliser le système de contrôle de version de leur choix.
- IKAN ALM peut communiquer avec les systèmes de suivi des incidents existants.
- Les environnements de développement locaux, décentralisés ou dans le Cloud sont entièrement pris en charge.
- IKAN ALM est multilingue et peut être personnalisé au niveau de l'utilisateur.



Intervenants

Les activités de gestion des cycles de vie impliquent plusieurs intervenants se concentrant chacun sur des aspects différents du processus de développement global. L'implémentation d'IKAN ALM pour la gestion du cycle de vie de l'application, apporte une série d'avantages pour chaque groupe d'intervenants sans compromettre la responsabilité des autres groupes.



Par exemple:

- Un développeur veut obtenir un feedback rapide sur le code enregistré dans le RCV (intégration continue) et construire le projet dans son EDI en utilisant les dernières versions correctes des Sources et les bibliothèques communes.
- Un chef de projet veut avoir un aperçu clair de l'état du projet: Le dernier code dans le RCV, peut-il être construit? Les

tests unitaires, ont-ils réussi? Quels tests sont effectués par le département AQ? Quelle est la version de production actuelle?

- Les opérateurs de production préfèrent un processus de déploiement automatique dans lequel ils peuvent contrôler les variables d'environnement.
- Le directeur des systèmes d'information (CIO) et le directeur général (CEO) d'une organisation préfèrent un processus automatisé et répétable fournissant une piste d'audit.

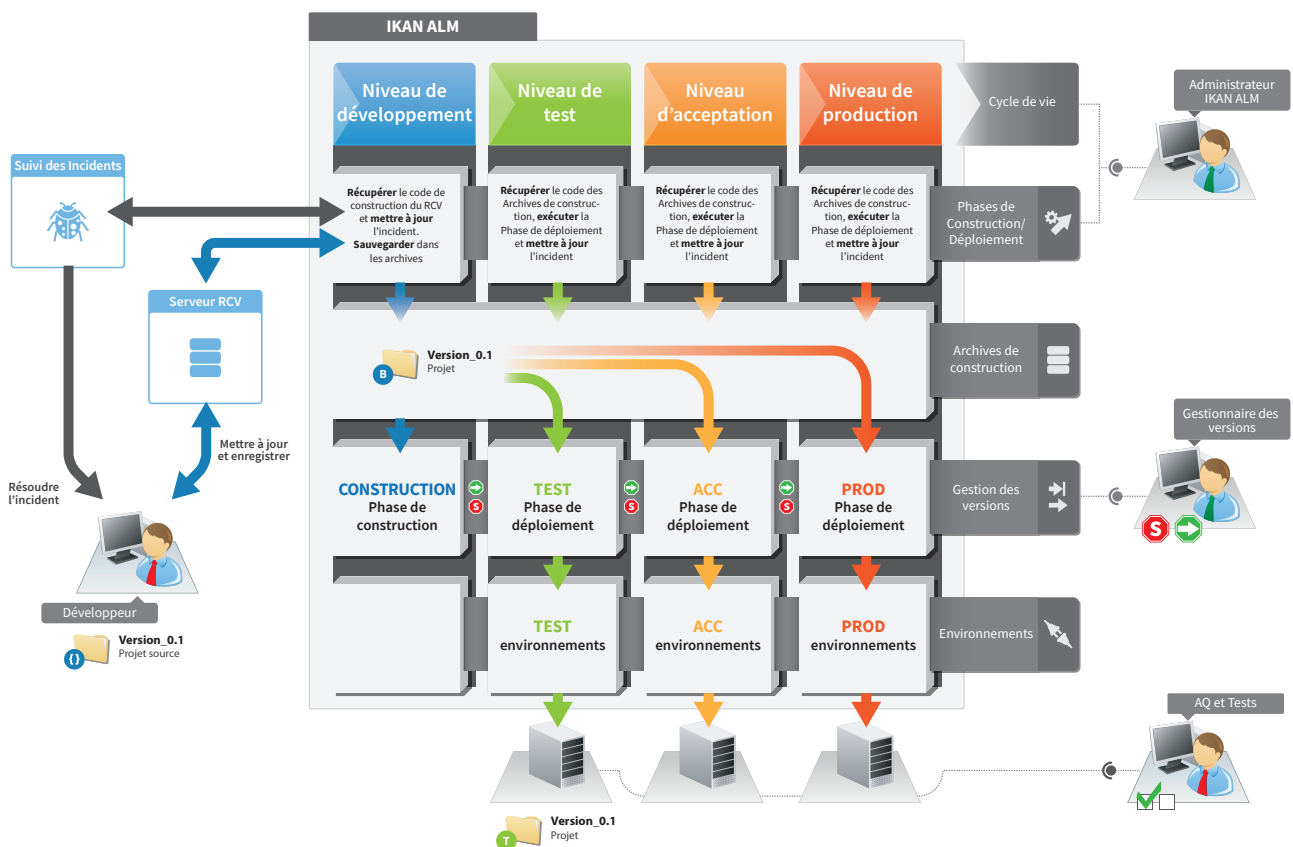
Comment fonctionne IKAN ALM?

Pour cette démonstration expliquant comment fonctionne IKAN ALM, nous supposons que tout est déjà configuré (les différentes fonctionnalités d'IKAN ALM et les méthodologies de développement dépassent la portée de ce document). La seule chose qu'un développeur doit faire c'est enregistrer son code dans un Référentiel de Contrôle de Version.

Une fois le code enregistré, IKAN ALM exécutera les actions suivantes:

- Remarquer les changements dans le RCV
- Récupérer le code à partir du RCV et construire l'application
- Associer une balise au code dans le RCV et mettre à jour le Système de Suivi des Incidents
- Déployer l'application vers un Niveau prédéfini (par exemple, le Niveau de Test)
- Notifier tous les intervenants impliqués
- Attendre l'approbation de l'utilisateur avant de déployer l'application vers le Niveau suivant (par exemple, le Niveau de Production)

Avantages



Chaque étape dans le processus ALM résout des problèmes différents pour des intervenants différents. Voici quelques réponses communes.

Pourquoi contrôler les versions?

Intervenant	Avantage
Développeur	Faire le suivi des changements.
Gestionnaires IT	Assurer la sauvegarde sûre de toutes les données historiques.
Production	Facilement restaurer une version précédente.
Gestion/Audit	Aucune perte de données.

Pourquoi implémenter un processus d'Intégration Continue?

Intervenant	Avantage
Développeur	Se concentrer sur le développement logiciel. Obtenir du feedback rapide sur le code enregistré.
Gestionnaires IT	Obtenir du feedback rapide sur la qualité du code. Détecter les points faibles.
Production	Obtenir du code de production de haute qualité.
Gestion/Audit	Moins d'erreurs. Processus répétable. Cycle de déploiement plus rapide et plus court.

Pourquoi automatiser les constructions?

Intervenant	Avantage
Développeur	Ne pas perdre du temps précieux à des tentatives de construction manuelle.
Gestionnaires IT	Permettre plus de constructions. Feedback rapide.
Production	Coordonner tout via des scripts.
Gestion/Audit	Éviter les erreurs.

Pourquoi gérer les approbations?

Intervenant	Avantage
Développeur	Améliorer la communication entre les membres de l'équipe de projet.
Gestionnaires IT	Contrôler l'évolution dans les différentes étapes du cycle de vie. Intégrer des moments d'audit.
Production	Contrôler le déploiement vers les serveurs de production.
Gestion/Audit	Traçabilité Qui a donné l'autorisation?

Pourquoi automatiser les déploiements?

Intervenant	Avantage
Développeur	La garantie que le département de production recevra le code de qualité qu'il a développée.
Gestionnaires IT	Accélérer le processus et réduire les erreurs.
Production	Le manque d'interventions manuelles diminue les risques.
Gestion/Audit	Augmenter la fréquence des publications et la productivité.

Pourquoi implémenter un processus de restauration?

Intervenant	Avantage
Développeur	Plus de temps disponible pour résoudre les défauts.
Gestionnaires IT	Il sera toujours possible de restaurer la dernière bonne version.
Production	Diminuer le risque d'interruption de service.
Gestion/Audit	Assurer de pouvoir restaurer une situation antérieure de manière exacte. Rapidement résoudre les erreurs en production.

Pourquoi gérer les cycles de vie?

Intervenant	Avantage
Développeur	Construire facilement le code pour le Niveau de Test ou de Production.
Gestionnaires IT	Aperçu clair du processus de développement et de son statut.
Production	Automatiser le déploiement vers la production. Réduire le nombre de corrections nécessaires.
Gestion/Audit	Répondre facilement aux questions: Qui? Quand? Qu'est-ce qui s'est passé et pourquoi?

Avantages par intervenant

Intervenant	Avantage
Développement	Fin des coûts répétitifs ou indésirables. Réduire les coûts grâce à la réutilisation de composantes. Aucune modification à l'environnement de travail, pas besoin d'apprendre de nouvelles compétences. Productivité augmentée de 15%.
Tests	Communication efficace entre le département de développement et le département des tests.
Gestion des constructions et des versions.	Séparation totale des fonctions. Fin des erreurs de déploiement risquées. Productivité augmentée de 25% dans les phases de construction et de publication.
Gestionnaires IT	Productivité accrue. Directives contrôlables et imposables pour les processus de test et d'approbation. Réduction des coûts. Rapports complets au niveau IT. Aperçu complet du projet.
CFO (Chief Financial Officer)	Projet entièrement conforme. Traçabilité des actions comprise. Rapports complets au niveau métier. Contrôle des frais de propriétés IT.
CEO (Chief Executive Officer)	Niveau CMMI élevé et contrôle du budget et des performances. Conforme aux exigences des organismes de réglementation. Organisation plus robuste et plus flexible
Externe (actionnaires, gouvernement, auditeurs)	Conformité totale et traçable.

© Copyright 2015 IKAN Development N.V.

Les noms et logos IKAN Development et IKAN ALM et tout autre nom de produits ou de services IKAN sont des marques déposées d'IKAN Development N.V. Toutes les autres marques déposées sont la propriété de leurs propriétaires respectifs. Aucune partie de ce document ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans l'autorisation explicite et écrite de IKAN Development N.V.

IKAN Development N.V.
Kardinaal Mercierplein 2
2800 Malines
Tél. +32 15 797306
info@ikan.be
www.ikan.be

IKAN