

La méthodologie MAJ (Méthodologie d'Assemblage JANUA) a été conçue pour vous permettre d'atteindre rapidement vos objectifs stratégiques et pour vous aider à maîtriser vos coûts.

Finis le traditionnel dilemme «acheter ou développer» (implémenter un logiciel propriétaire ou développer sur-mesure). Vous pouvez désormais bâtir de nouvelles applications en assemblant à votre guise des composants Open Source déjà conçus, testés et éprouvés.

JANUA intègre ces composants dans une nouvelle solution entièrement personnalisée, correspondant parfaitement aux objectifs et aux besoins de vos utilisateurs

Introduction

Pendant longtemps, les entreprises ont préféré des composants logiciels propriétaires pour répondre aux besoins en nouvelles applications. Aujourd'hui, la multitude de composants Open Source disponibles, leur robustesse reconnue rend cette démarche d'assemblage beaucoup plus séduisante. Vous avez à votre disposition des milliers de projets Open Source que vous pouvez, pour la plupart, tout simplement télécharger.

Cela vous permet d'évaluer et de tester les composants dont vous avez besoin, rapidement sans devoir vous engager dans un processus d'achat traditionnel bien trop souvent long, contraignant et coûteux. L'Open Source vous permet aussi de tester puis de déployer une solution sans l'achat de licences.

Les avantages majeurs de l'Open Source sont :

- La rapidité d'implémentation
- La réduction de vos coûts
- Son architecture technique
- L'accès au code source permettant la personnalisation totale et facilitant le support.
- L'indépendance par rapport aux fournisseurs

Les composants Open Source sont conçus, développés et testés de manière hautement modulaire.

Les différentes parties d'une application peuvent toutes par la suite être assemblées et désassemblées à votre guise dans une autre application. Les projets Open Source s'adaptent particulièrement bien aux cas uniques, à la personnalisation et aux besoins spécifiques de certains métiers.

Points Forts de MAJ

- Développement d'applications sur-mesure adaptées à vos besoins
- Des composants Open Source qui fournissent les fonctionnalités de base
- Une approche d'expert éprouvée et rapide de la conception, des expériences riches et nombreuses comme en témoignent nos références.

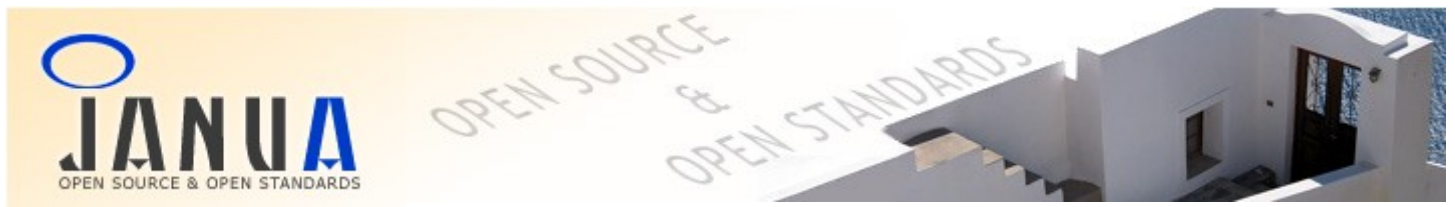
Pourquoi choisir la méthodologie MAJ ?

- Les besoins de votre entreprise sont uniques.
- Le système que vous souhaitez installer revêt une importance stratégique et doit constituer un avantage compétitif pour votre entreprise.
- Une exceptionnelle capacité d'intégration aux infrastructures technologiques existantes.
- Vos attentes et les besoins de votre entreprise sont en perpétuelle évolution.

MAJ vs. développement spécifique

Réduction des coûts de développement

- Amortissement des coûts de développement
- Les applications « métiers » dont le coût de développement « ex nihilo » est prohibitif peuvent être assemblées, à un coût raisonnable, à partir de composants existants, réduisant ainsi fortement les frais d'implémentation.
- La popularité des technologies Open Source vous permettra de trouver, à n'importe quel stade de votre projet, les compétences humaines dont vous aurez besoin au sein de la communauté.



Coûts de support et de maintenance réduits

- Les coûts de support sont réduits car soit vous travaillez directement avec la communauté, soit vous optez pour un support commercial sur mesure.
- Vous trouverez beaucoup plus de ressources techniques qualifiées pour supporter votre projet open source que votre code propriétaire.

Temps de développement

- Les composants existants vous permettent de réduire fortement les temps de développement de vos applications

MAJ vs. Solutions packagées

Coût de licence réduits

- Les produits open source suppriment les coût prohibitifs des licences propriétaires.

Rapidité d'acquisition

- Il est beaucoup plus rapide de se procurer et de tester un logiciel Open Source sans engagement financier. Fini les longues négociations pour obtenir le budget et les pertes de temps générées par le cycle d'achat !

Flexibilité

- Les applications assemblées permettent une gestion du changement plus rapide.
- Une architecture assemblée facilite une mise à jour des composant un par un au fur et à mesure de leur évolution, sans rupture brutale.
- Souplesse accrue de votre système d'information, vous n'êtes plus liés par des contrats de maintenance pluri-annuels.
- Vous n'avez plus la crainte d'avoir choisi le « mauvais produit », le coût des changements devient négligeable.

Efficacité et productivité de l'utilisateur

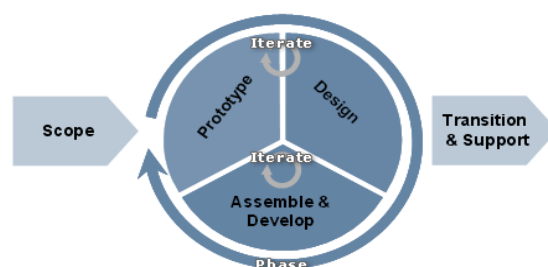
- Vous pouvez bâtir la solution spécifique qui convient le mieux à vos utilisateurs, baissant par la même les coûts de formation .

Coûts de support et de maintenance réduits

- Il est plus rapide de fixer les bugs et d'améliorer l'application car vous possédez le code source et n'êtes plus tributaires de la volonté de l'éditeur.

Les phases de la méthodologie MAJ

MAJ est un processus de développement itératif et rapide. Les étapes ci-dessous peuvent vous sembler familières, notre méthodologie MAJ apporte des améliorations à chacune de ces étapes au regard de l'utilisation des composants Open Source.





Phase	Activités traditionnelles	Les plus MAJ
Scope	<ul style="list-style-type: none"> • Découverte des besoins • Prioritisation des besoins • Estimation des charges 	<ul style="list-style-type: none"> • Définition des besoins particuliers et des besoins standards • Identification des options possibles dans le cadre Open Source
Prototype	<ul style="list-style-type: none"> • Maquette graphique, définition des écrans • Recommandation et choix des composants • Création et architecture d'une fonctionnalité significative • Affinage des besoins et des estimations 	<ul style="list-style-type: none"> • Évaluation des composants Open Source • Étude des licences Open source utilisables et de leurs contraintes • Discussion avec les communautés sur les besoins et les évolutions envisagées.
Design	<ul style="list-style-type: none"> • Évolution du prototype • Définition de l'intégration applicative • Création avec UML du design technique et fonctionnel • Planification du développement et de l'assemblage 	<ul style="list-style-type: none"> • Approvisionnement des composants Open Source • Design de l'encapsulation des composants propres vs. composants standards afin d'en assurer le support. • Identification des alternatives de support Open source • Identification du code propre (et non stratégique) à reverser aux communautés
Assemblage & Développement	<ul style="list-style-type: none"> • Architecture et création du désign • Développement • Test de qualité unitaires, d'intégration, et de système • Finalisation de la documentation technique et utilisateur • Planification de la transition & du support 	<ul style="list-style-type: none"> • Assemblage des fonctionnalités requises et interconnexions des modules. • Personnalisation « à la marge » pour minimiser les développements propres. • Extension des fonctionnalités Open source en liaison avec la communauté. • Travail avec les communautés Open Source pour les développements propres. • Soumission de vos propres développements aux communautés Open Source
Support & Transition	<ul style="list-style-type: none"> • Test de l'acceptation au changement des utilisateurs • Formation des utilisateurs et des administrateurs • Transition du support vers vos équipes de développement internes ou externes • Migration vers les données et l'environnement de production • Sélection des options de maintenance 	<ul style="list-style-type: none"> • Sélection des options de support et de maintenance • Choix éventuel d'un support externe (parmi plusieurs) • Support de la communauté • Support interne • Transfert de compétence avec la communauté open Source.