



# Ensemble Ouvert Libre Évolutif

Site de diffusion : <http://eole.orion.education.fr/>

Eole est un projet collaboratif basé sur la philosophie du logiciel libre. Sa mission est de proposer des solutions clef en main pour la mise en place de serveurs Intranet-Internet.

La mutualisation des compétences et des moyens permet de réaliser des solutions économiques, fiables et performantes. Le pôle EOLE agit comme un « éditeur interne » en proposant un système d'exploitation GNU/LINUX facile à déployer, configurable et administrable à distance.

## **Les modules EOLE.**

La distribution EOLE est organisée en modules. Un Module est un ensemble cohérent (système d'exploitation Linux/Ubuntu + logiciels sélectionnés) destiné à répondre à un besoin métier précis.

## **Mise à jour corrective.**

*Ne concerne pas les changements de version fonctionnelle ni majeurs (cf migration).*

Pour garantir le bon fonctionnement des applications, les changements de version majeure des composants socles ne sont pas pris en compte durant le cycle de vie d'une version EOLE. Seuls sont pris en compte les correctifs d'anomalies. Les mises à jour de sécurité et les corrections de bugs majeurs sont séparées des mises à jour fonctionnelles. Les mises à jour Ubuntu sont également appliquées au fur et à mesure de leur mises à disposition par Canonical.

Nous conseillons de mettre les serveurs en mode mises à jour automatique, dans ce cas le processus est déclenché une fois par semaine par vérification des paquets (.deb) sur les dépôts via Internet. Le serveur principal de mise à jour s'appelle eole.ac-dijon.fr; un serveur miroir est hébergé par le Crihan son adresse est ftp.crihan/eoleng. Les mises à jour ne sont pas séquentielles, mais cumulatives; ainsi un serveur se met à jour en une seule fois quelque soit sa version mineure.

L'intégration continue permet une plus grande homogénéité du parc des serveurs et assure un bon niveau de sécurité.

## **Mise à jour fonctionnelle.**

Les modifications et ajouts de fonctionnalités feront l'objet de la publication d'une nouvelle version fonctionnelle 2.4.X et la mise à niveau se fera avec une procédure automatique distincte de la mise à jour ordinaire. Nous prévoyons au maximum deux versions fonctionnelles par an. Ces versions fonctionnelles seront supportées de façons indépendantes jusqu'à la fin du support de la version majeure 2.4 (juin 2017 idem Ubuntu 12.04 LTS), ainsi il sera possible à chacun de maîtriser au mieux les évolutions de son parc de serveurs EOLE.

## **Principe de diffusion .**

Toutes les mises à jour sur une version donnée passent par un stade «candidate» qui permet de tester/qualifier les adaptations locales et les applications non directement livrées par EOLE.

La mise à jour «stable» intervient généralement dans un délai de deux à trois semaines après la publication de la version candidate si aucun problème n'est signalé..

Toutes les informations de diffusion sont sur nos sites et font l'objet de publication de nouvelles annonces (flux RSS disponibles), elles sont également annoncées via nos listes de diffusion ouverte à tous.

Pour les contributeurs/testeurs , nous publions des paquets dit de « *développement* » qui permettent le suivi quotidien de l'avancement des modifications. (Attention aucune garantie de fonctionnement sur ces versions)

## **Principe de développement d'une nouvelle version.**

De même que toutes les distribution GNU/Linux, nous utilisons la méthode d'intégration continue en cycle agile.

Les modules EOLE suivent,, les version Ubuntu LTS, à savoir un changement de version tous les deux ans. Chaque version est supportée 5 ans. (plutôt 4 ans dans les fait à cause du décalage de la sortie de la version EOLE par rapport à Ubuntu)

Pour permettre les tests et la qualification des applications, des images ISO de la version EOLE Bêta sont systématiquement mises à disposition. Puis un mois avant la sortie officielle, arrive la version dites Candidate qui permet le passage en stable via la mise à jour automatique . Nous recommandons alors de la mettre en production sur quelques établissements de test. Elle permet de faire d'éventuels ajustements jusqu'à stabilisation complète.

## **Migration des serveurs.**

La migration est l'opération qui permet de changer la version majeur sur un serveur en production. A chacune de ces nouvelles versions, le pôle EOLE propose un mode opératoire spécifique pour faciliter ce changement de système d'exploitation tout en conservant les paramètres du serveur et les données.

## Qualification des applications externes.

Un module EOLE étant un serveur GNU/Linux, il est tout à fait possible d'ajouter des applications tierces. On peut distinguer plusieurs types d'applications externes

1. Applications fournies et supportées par Canonical (Éditeur Ubuntu)
2. Applications fournies et supportées par la communauté Ubuntu.
3. Applications fournies et supportées par EOLE ou par la mutualisation Envole.
4. Applications nationales du ministère MENJVA (Concerne essentiellement Horus)
5. Applications tierces d'étiteurs privés,

Les trois premières sont disponibles sous forme de paquets debian (.deb)

### Définition paquet logiciel (source WIKIPEDIA):

*On appelle paquet (ou parfois paquetage, en [anglais](#) package) une [archive](#) (fichier compressé) comprenant les [fichiers informatiques](#), les informations et procédures nécessaires à l'installation d'un [logiciel](#) sur un [système d'exploitation](#) au sein d'un agrégat logiciel, en s'assurant de la cohérence fonctionnelle du système ainsi modifié.*

Cette méthode de diffusion universellement utilisée sous Linux, permet de gérer les versions et les contraintes (dépendances) entre les versions des différents logiciels présents sur le serveur.

Les paquets suivent un cycle d'intégration continue, ainsi une évolution d'un composant lié à un produit, entraînera la réalisation d'un nouveau paquet pour régler les conflits de dépendances de version. Concrètement, un paquet refusera de s'installer si ses dépendances ne sont pas satisfaites. Les principales dépendances portent sur les version des langages (PHP, Python, ...) et sur les gestionnaires de base de donnée (PostgreSQL et MySQL, SQLite, ..)

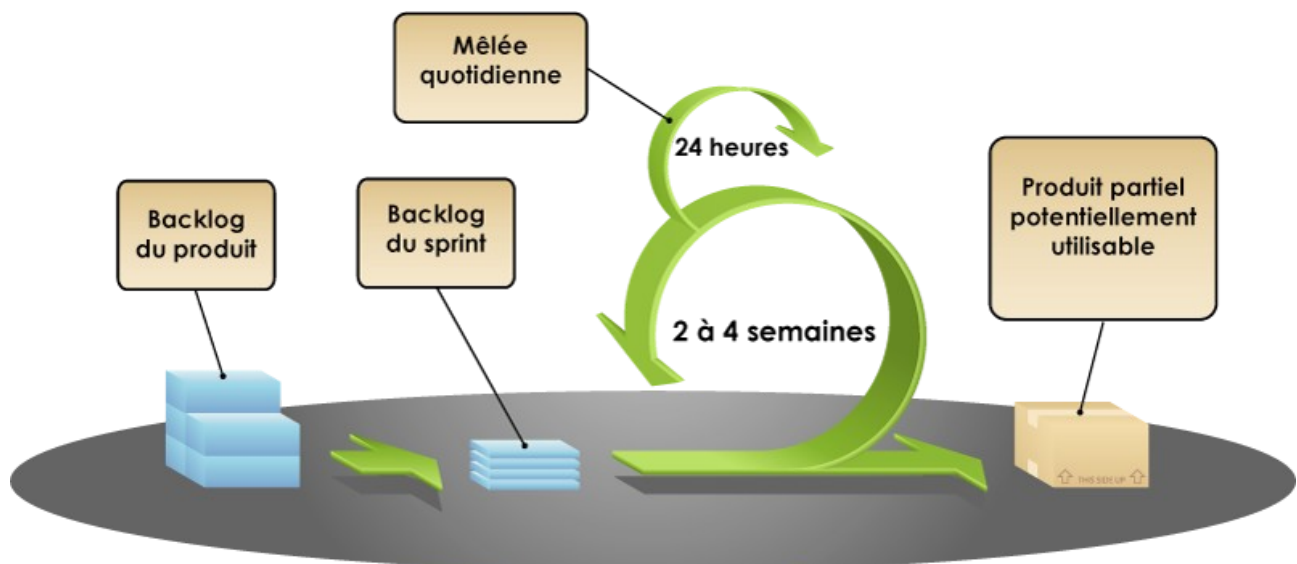
## Principe de développement AGILE

Au quotidien l'équipe travaille en suivant la méthode SCRUM.

Essentiel en quatre points.

- Scrum est un processus agile qui permet de produire la plus grande valeur métier dans la durée la plus courte.
- Du logiciel qui fonctionne est produit à chaque sprint (toutes les 3 semaines).
- Le métier définit les priorités. L'équipe s'organise elle-même pour déterminer la meilleure façon de produire les exigences les plus prioritaires.
- A chaque fin de sprint, tout le monde peut voir fonctionner le produit courant et décider soit de le livrer dans l'état, soit de continuer à l'améliorer pendant un sprint supplémentaire.

Schéma de l'itération Agile.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

### **Principe d'intégration continue**

Technique constituant à vérifier à chaque modification que le résultat ne produise pas de régression

Avantage :

Test immédiat des unités modifiées

Prévention rapide en cas de code incompatible ou manquant

Problèmes d'intégration sont détectés et réparés de façon continue

Une version est toujours disponible pour un test, une démonstration ou une distribution

**La pratique quotidienne d'intégration des flux des mises à jour en développement couplée à des tests unitaires, permet de détecter au plus tôt les éventuels problèmes.**

**Les version de développement sont systématiquement mise à disposition pour favoriser les tests et contributions par la communauté des utilisateurs.**