

Industrialiser le SI

Pourquoi ?

Comment ?

```
## Copyright (c) 2007-2008, Patrick Germain Placidoux
## All rights reserved.
##
## This file is part of kikonf.
##
## kikonf is free software: you can redistribute it and/or modify
## it under the terms of the GNU General Public Licence as published by
## the Free Software Foundation, either version 3 of the License, or
## (at your option) any later version.
##
## kikonf is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
## GNU General Public Licence for more details.
##
## You should have received a copy of the GNU General Public License
## along with kikonf. If not, see <http://www.gnu.org/licenses/>.
##
## Home: http://www.kikonf.org
## Contact: kikonf@gmx.com
```

Table des matières

1 Pourquoi Industrialiser le SI.....	3
1.1 Quiz.....	3
1.2 Interêt de l'Industrialisation.....	5
1.3 Solutions et Risques.....	6
2 L'open Source comme model de solution d'Industrialisation.....	7
Libre et faux Libre positionnement de Kikonf.....	7
3 TradeMarks.....	11

1 Pourquoi Industrialiser le SI

1.1 Quiz

● Votre SI supporte-t-il des applications middleware ?

Aujourd'hui avec la popularisation du Web, des services Clouds, des mobiles et du développement Agile, le Business() voit le service SI intégrateur système comme un hébergeur de services à la demande et en attendent la même reactivité.*

Si votre SI supporte l'hébergement et la mise à disposition d'environnements MiddleWare il s'expose à cette pression de la part du business.

● Système est-il exposé à la pression des Services tiers(*) ?

- Les Etudes, les Chef de Projet Prod et les Intégrateurs attendent leurs environnements() Applicatifs et leurs accès pour leur déploiement, tests, recettage et benchs ...*

- la supervision, le pilotage attendant leurs accès aux environnements applicatifs pour les démarrer/arrêter, superviser, planifier des tâches.

● Système est-il considéré comme référent par le business en matière d'architecture et de support niveau III pour le middleware ?

Très souvent la tête dans le guidon, pris par les tâches routinières de la production d'environnements applicatifs, les équipes middleware n'ont pas le temps de monter en puissance sur le terrain de l'innovation permanente des nouvelles technos, a contrario des développeurs qui eux sont en phase avec les dernières évolutions.

● L'image de Système est-elle dégradée ?

La situation précédente crée souvent un décalage entre le service Système et les Développeurs et se traduit par des problèmes de communication et d'interprétation des pré-requis de chacun.

Cela se traduit souvent par des réflexions du genre:

- Il faut plusieurs jours pour obtenir un environnement*
- Il n'y a jamais 2 environnements pareils*

Sous-entendu que Système à la tête sous l'eau et tarde à répondre aux demandes du business.

La suite logique de ce processus est la mise à l'écart du Système du circuit décisionnel en amont,

Ce dernier se trouvant toujours en queue de projet à devoir définir des architectures qui ne collent pas nécessairement au Blue Print défini par le groupe.

● Votre équipe middleWare est-elle débordée ?

Une équipe middleWare débordée risque de produire des Environnements disparates et inconséquent en termes de Norme.

Le 3 tiers est le domaine par excellence où le cadre est essentiellement soutenu par les Normes, les Specs à l'inverse de Mini et gros Système dont le gros du cadrage est assuré par le hardware et l'OS.

Des environnements non normés sont difficilement compréhensibles au Business, peu reproductibles et difficiles à gérer par les tiers.

- Vos ingénieurs s'enlisent-ils dans des tâches d'administration ?

La vraie valeur ajoutée des ingénieurs est masquée par l'enlissement dans des tâches d'administration répétitives où tout au long de l'année, ce sont les mêmes actions qui sont reproduites pour générer les environnements applicatifs.

Mais comme ces actions sont manuelles elle se trouvent non normées, non lisibles et non reproductibles.

Les vrais services à plus values comme :

- Le support en amont au design d'Architectures Applicatives
- L'adéquation au Blue Print et aux directives groupe
- Le support à l'intégration des nouvelles Applications
- Le suivi et l'analyse des benches
- La résolution et le suivi des incidents de Production
- L'accompagnement des projets business en amont (ex: ESB)
- Et la mise en oeuvre des projets Prod toujours décalés faute de ressource (ex: de la supervision J2EE, MQ, ESB)

augmenteraient l'image de System et le réintégrerai dans le processus décisionnel du Business.

1.2 Intérêt de l'Industrialisation

Conformément à leur propre paradigme respectif chaque service Tiers attend toujours les mêmes types d'accès aux ressources qui lui sont mis à disposition par Système.

Donc avoir un standard identique et reproductible (d'application en application et par environnement) leur permet de toujours retrouver leurs fonctionnalités aux mêmes endroits.

Cela augmente l'image de Système et rassure les services Tiers quant à l'intégrité des ressources dispatchées par Système.

L'industrialisation:

- Produit des environnements disponibles rapidement (ex: 1h contre 2 jours)
- Elimine le risque d'erreur
- Permet l'adéquation forte avec les paramètres en input de la fiche technique applicative fournie par les études
- Fourni des environnements normés selon votre Norme
- Fourni des environnements reproductibles et migrables rapidement
- Met automatiquement à disposition les accès aux Services Tiers simultanément à la livraison de l'environnement
- Place système en position forte face à toute velléité d'externalisation du service complet à l'infogérance

1.3 Solutions et Risques

Industrialiser votre SI est mettre une couche d'intelligence sur votre SI.

Or comme toutes pensées humaines, autant cette couche peut-être claire autant elle peut-être confuse ou rendue confuse à dessein.

Si cette couche est obscure vous perdez l'accès à votre SI et devez systématiquement passer par l'intervenant auquel vous l'avez confiée.

Dans ce cas il s'agit d'une prise en otage de votre propre SI.

Une prise de pouvoir non seulement sur la structure hiérarchique mais aussi sur les services tiers.

Un autre élément à prendre en compte est qu'une couche d'industrialisation impacte plusieurs Oses et softwares qui ont leur propre cycle de vie et évolution de version...

Donc elle doit-être sans cesse tenue à jour.

Ex: une couche d'industrialisation qui prend en charge WebSphere MQ ® doit supporter les nouvelles versions.

Lorsque la question de l'industrialisation du SI se pose, 3 types d'intervenants sont possibles:

a) Externalisation Privée :

Le code actuel ou futur n'est pas publiquement accessible

Le code actuel ou futur est payant

Pas de communauté de développeurs

L'implémentation de nouvelles fonctionnalités est payante

b) Développement en interne:

Lorsque la couche est développée en interne il est à considérer de très près le pouvoir que l'on donne à long terme, voire à très long terme à (aux) l'ingénieur(s) auteur(s) du projet.

Des conjonctures externes au projet peuvent amener l'équipe à s'assurer un contrôle exclusif de cette couche.

Le temps de développement est long

Le code est accessible mais peut-être rendu obscur à dessein

Les fonctionnalités peuvent-être rendues obscures et non lisibles

Pas de communauté de développeurs

c) Externalisation Publique

Le code actuel ou futur est accessible

Le code actuel ou futur est gratuit

La communauté de développeurs est en augmentation

Les fonctionnalités sont lisibles

L'implémentation de nouvelles fonctionnalités est ouverte à l'interne ou à l'écosystème.

On bénéficie d'un code directement opérationnel et stable (suite à des années d'existence)

Les fonctionnalités sont riches car ancrées sur les problématiques terrain de divers clients

2 L'open Source comme modèle de solution d'Industrialisation

Libre et faux Libre positionnement de Kikonf

Lors de l'appréciation d'une solution libre quelques points vitaux doivent être pris en considération:

a) Dumping des Etudes

Certaines plateformes d'industrialisation libre sont dédiées aux études et leur conviennent mais peuvent se traduire par une mainmise sur la Prod. de la part des études, s'ils sont déployés en Prod. (ex : Jenkins, Docker). Est-ce bien ce que vous voulez ?

b) Les Faux-semblants d'Open sources:

Avec l'explosion du logiciel libre, certains fournisseurs de solutions libres surfent sur la vague et ne sont pas les éditeurs respectifs des solutions qu'ils distribuent mais des agrégateurs de solution libre, pas de vrais experts mais plutôt des packageurs.

Souvent vous dépendez d'eux pour l'installation qui du coup est facturée ainsi que toute évolution du méli-mélo.

c) Les Soi-disant libres

Vérifiez la licence dite libre de votre éditeur : Bcp de licences soi-disant libres existent sur le Marché.

La plus stricte (la première historiquement) : est la GNU licence :

- Elle nous oblige à tenir tout le code de notre solution à disposition sur le Net **24/24 indéfiniment**.
- Cette condition est valable pour le code passé, actuel et avenir de notre solution.

Cette licence est tellement contraignante pour l'éditeur que peu l'utilisent mais elle fait office d'Acte de foi et de démonstration du niveau d'engagement de l'éditeur.

En ce qui concerne Kikonf cela se traduit par l'accès illimité aux ressources passées, actuelles et futures de la solution kikonf.

Donc non seulement la bibliothèque d'actions et la couche d'industrialisation basse mais aussi à la plateforme graphique centralisée disponible dans le courant de cette année.

d) Le code de la solution dite libre est-il clair et lisible ?

Une solution **non éducative** peut concourir à une perte de savoir-faire de vos équipes à long terme.

La popularisation du libre à vu de nombreux éditeurs mettent, parallèlement à une activité privative, des pans de codes sous licence libre.

Mais ce code étant tellement offusqué (rendu obscur à dessein ou non) que vous ne pouvez rien n'en faire et vous êtes obligé de passer par la solution payante pour vous en sortir.

C'est aussi du faux libre.

La position de Kikonf:

Le maelstrom d'opérations exécutées par un ingénieur pour livrer un environnement à une application peut paraître complexe et l'est effectivement prit globalement.

Mais découpé au scalpel, on se rend compte qu'il s'agit simplement d'une liste d'actions effectuées suivant un ordre chronologique.

Kikonf découpe ces tâches et les dispatche en actions élémentaires regroupées dans des catégories (pour dire softwares: WAS ®, Tomcat ®, Jboss ®, ...MQSeries ®, ...)

- L'action comporte strictement et uniquement le code que faisait manuellement l'ingénieur pour réaliser cette action
- Tous les services kikonf Menus, commandes d'exécution unitaires et groupées, interfaces centralisées, authentification et droits accès, ... :
sont immédiatement disponibles à l'action comme un bus de services et ne viennent pas perturber le code des actions

Les Actions sont lisibles:

Ces actions sont donc épurées, l'ingénieur y retrouve le code ou les commandes qu'il tape lui-même dans le cadre de la réalisation de cette action et n'est pas "révolutionné" par la couche mise en place.

Par ex: si un ingénieur a pour habitude de faire des commandes jython pour piloter WAS ou Weblogice il retrouve ses commandes jython habituelles dans les actions consacrées à ces tâches.

S'il fait des tâches système il retrouve des commandes shell dans les actions consacrées, de même pour MqSeries, ..., ainsi de suite. Et peut facilement les modifier ou en créer d'autres.

Ce qui rend les commandes claires, simples lisibles et accessibles à leurs usagers.

Par la suite ces actions ou briques unitaires peuvent être groupées ou montées en collier comme des brins d'ADN pour la réalisation de tâches plus complexes.

Les interfaces Menus (ou Graphiques) sont lisibles:

Ex: un superviseur manipule le Menu MQSerie, à chaque option le Menu lui montre la commande runmqsc passée, il peut comprendre les commandes MQSerie, les stocker les coller dans ses jobs de planification ...

Kikonf est une plateforme Educative pensée en premier lieu pour combattre la rétention et la captation du savoir-faire interne ou externe.

e) Vérifier que la solution dite libre est directement installable sur vos machines cible

Certaines solutions prétendues libres s'avèrent en pratique ininstallables sur vos systèmes et vous pouvez vous entendre dire que c'est prévu dans la solution payante.

Kikonf a besoin d'une seule chose pour fonctionner Python ®.

Kikonf est opérationnel sur AIX ®, Linuxes, Bsd, Unixes, Windows ® et tout O.S. supportant Python.

Un dezipage du binaire suffit.

f) Se méfier des solutions dites libres qui ont une version payante (libre ne veut pas dire forcément gratuit).

Car la version gratuite est souvent inutilisable en tant que telle et s'avère être un pot de miel pour la version payante dont le code lui n'est pas libre ni accessible.

Toutes les solutions kikonf sont libres, gratuites et le resteront.

g) Solution centralisée ou nodèles ?

Le choix est fait pour une solution d'industrialisation centralisée.
Donc tout le management se fait à partir d'une console, c'est joli.

Dans la mesure où c'est de l'industrialisation du SI dont on parle il est évident que ce n'est pas le service système seul qui est concerné par l'usage de cette console mais tous les services usagers de la Prod: Db, Intégration, Supervision, Planning, Chef de projet prod, Chef de projet étude, Développeur, ...

Il est à noter que nombre de ces services ont accès aux machines: en mode dégradé (lorsque tous les services ne sont pas encore montés : FS, Réseaux, ...) ou derrière ces pare-feux dans des cas de PRA, Crash, indisponibilité, ...

Dans ce dernier cas, avec une solution seulement centralisée on est démuni.

La position de Kikonf:

Dans ses fondations Kikonf a été pensé comme Solution centralisée décentralisée.

Toute opération reste et est toujours possible par **commandes ou Menu sur tout serveur isolé** des autres nodes derrière un pare-feu sans nuire à l'intégrité de la couche centralisée.

h) Norme

La solution d'industrialisation génère-t-elle des configurations lisibles à tous les étages ?

Les ressources :

- système : répertoire, user, groupe, FS, ...
- ressources logiciel WAS ®
- ressources logiciel Tomcat ®
- ressources logiciel JBoss ®
- ressources logiciel WLO ®
- ressources logiciel MQ ®
- ressources logiciel IHS ®
- ...

sont-elles lisibles et clairement identifiables comme appartenant à l'identifiant de cette application configurée sur cet environnement.

Le Normage rend les Applications mutualisable, facilement transportables d'un environnement à l'autre et empilables sur une partition donnée.

La structure des ressources rendue immuable par la Norme devient prédictible et anticipable par les services Tiers qui eux aussi ont besoins d'intégrer leur services : Etudes, Planning, intégration, supervision,

La position de Kikonf:

Les environnements produits par kikonf sont structurés selon une syntaxe immuable, celle que vous choisissez d'implémenter.

Il est aussi possible de l'utiliser sans Norme aucune.

Note:

* Système : le Service Système appartient à la Production et gère les machines et les softwares, leur administration et leurs configuration.

* Business : Services marketing/commerciaux + les Etudes (chefs de projets Etude et développeurs)

* Services Tiers : Services avec lesquels interagit le Service système :

Business, Architecture Technique, Chef de Projets Production, Intégration, Supervision, Pilotage, la Planification, ...

* Environnement : Un environnement est un espace de ressources Systèmes cloisonné pour une application.

Système doit donc fournir plusieurs environnements pour une application.

Ex: Développement, intégration, recette, preprod, maintenance, production, ...

* MiddleWare: Met à disposition toutes les ressources utiles pour le fonctionnement d'une application sur un environnement.

En cela middleWare impacte toutes les ressources mises en oeuvre à système (Machine, Oses, Disk, BD, MQ, Resx, ...).

3 TradeMarks

"Python" is a registered trademark of the Python Software Foundation.

"IBM", "AIX", "WebSphere", "WebSphere Application Server", "WAS", "IBM MQ Series", "MQ Series", "IBM WebSphere MQ", "WebSphere MQ", "WebSphere MQ Series", "IBM HTTP Server", "IHS" are registered trademarks of International Business Machines Corporation.

"WebLogic Server" and "Java" are registered trademarks of Oracle and/or its affiliates.

"Apache Http Server" and "Tomcat" are trademarks of the Apache Software Foundation.

"Apache Tomcat" and "Tomcat" are trademarks of the Apache Software Foundation.

"JBoss" and "JBoss AS" are trademarks or registered trademarks of Red Hat.

"Windows" is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Other names may be trademarks of their respective owners.