



"Apprivoisez la complexité du système d'informations"

ARCAD Software



« La Gestion de Configuration Logicielle : Un socle indispensable à une gestion rigoureuse des changements logiciels »

COMMON Luxembourg – 29 Avril 2008

(C) Copyright 2008 ARCAD Software

Page n°1



1- Généralités



(C) Copyright 2008 ARCAD Software

Page n°2



- Pressions réglementaires
 - SOX, Bâle II, Solvency II, LSF
- **Référentiels de bonnes pratiques**
 - ITIL, CMM, COBIT
- Alignement business / IT
 - Flexibilité du SI
- Modernisation du SI
 - SOA, Audit, Documentation, Urbanisation
- Rationalisation des coûts
 - Automatisation, intégration de nouvelles technologies



La GCL: notre métier de base



- Sensibilité: augmentation du niveau d'organisation des changements logiciels
 - Y a-t-il des environnements définis (développement, test, production) ?
 - Les relations Etudes/production sont-elles bonnes ?
 - Y a-t-il déjà eu des cas de régressions de sources ?
 - Quelle est la fréquence des mises en production ?
 - Les déploiements sont-ils automatisés ?
 - La réception de nouvelles versions de progiciels n'engendre-t-elle pas une charge de travail conséquente en interne ?

- Le réflexe de « maintenabilité » du SI
- Un socle de base indispensable à une bonne organisation informatique
- La colonne vertébrale pour les activités connexes:
 - Tests
 - Documentation
 - Support technique



- Pour passer du stade artisanal à l'ère industrielle



2- Terminologie





"Ensemble des activités (manuelles ou automatisées) permettant d'identifier et de définir les éléments de configuration et toutes leurs relations. Elle permet de contrôler les évolutions durant le cycle de vie du logiciel, d'archiver chacun des états successifs et de vérifier que chacun de ces états est complet et cohérent".

(Norme AFNOR : NF Z 67-102)



*Activités d'ordre technique et organisationnel
comprenant **l'identification de la configuration,**
la maîtrise de la configuration,
l'enregistrement de l'état de la configuration,
l'audit de la configuration.*

(*) Adoptée par le CEN (Comité Européen de Normalisation)



- Gestion de version vs Gestion de configuration
 - La gestion de configuration permet en plus :
 - de gérer les demandes de modification du système à faire évoluer
 - de mettre en correspondance les demandes de modifications avec les changements apportés au système.
 - Ex: CVS = gestion de version
- ALM vs SCM (ou GCVA vs GCL)
 - L'ALM couvre un périmètre plus large...



- Requirements Visualization
- Requirements Management
- Modelling
- Design
- Project Management
- Change Management
- Configuration Management
- Build Management
- Testing
- Release Management
- Deployment
- Monitoring and Reporting

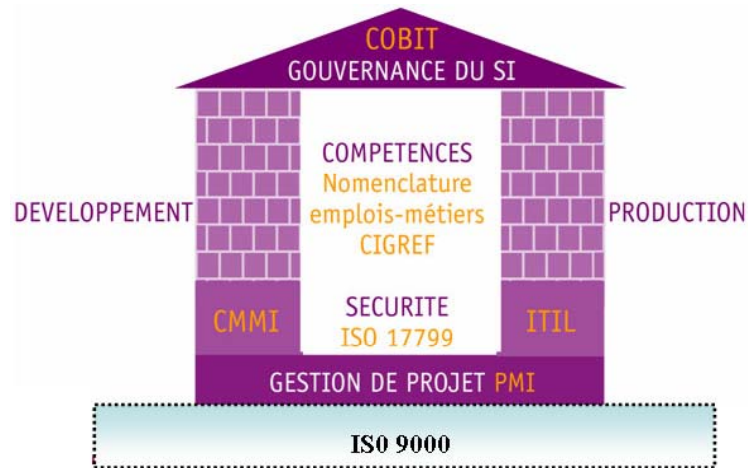


*Domaines de l'offre
logicielle Arcad*



3- Positionnement dans les référentiels de bonnes pratiques





- PO4 – Définir l'organisation et les relations de travail
- PO10 – Gérer les projets
- PO11 – Gérer la qualité

- AMP2 – Acquérir des applications et en assurer la maintenance
- AMP3 – Acquérir une infrastructure technologique, en assurer la maintenance
- AMP5 – Installer les systèmes et les valider
- AMP6 – Gérer les changements

- DS9 – Gérer la configuration
- DS10 – Gérer les problèmes et les incidents
- DS11 – Gérer les données

Objectifs d'un audit : ce que vérifie l'auditeur

- Existence de procédures
- Mise en place des contrôles prévus lors de l'audit précédent
- Leur application
- Leur pertinence et leur efficacité
- Qu'elles sont évaluées et révisées périodiquement

- Construire et faire évoluer des **bonnes habitudes** de travail dans un **cadre structuré** pour obtenir de **meilleures performances**
- Il est basé sur l'obtention d'un haut niveau de **Qualité de service** du SI avec une attention particulière sur la **Relation Client**



- Le Système d'informations est stratégique
- Son fonctionnement est critique
- Son évolution et son adaptation sont conditionnées par le business de l'entreprise
- L'équipe informatique doit bâtir une organisation en ligne avec celle de ses clients et en ligne avec les besoins du Business



- **Souple**: non normatif
- **Pragmatique**: ancré dans l'action (du déjà vu)
- **Adapté/intelligent/subtil** (du bon sens)
- **Modulaire**: convient aux petites structures comme aux grosses dans une démarche de petits pas
- **Référentiel commun**: acteurs internes DSI, DSI et métiers, DSI et fournisseurs, éditeurs de logiciels
- **Approche qualité**: le bon processus, la bonne organisation, la bonne mission, la bonne formation, les bons outils, les indicateurs de mesure...



- **Intégration** de l'informatique dans les processus métiers de l'entreprise
- **Alignement** de l'informatique sur les métiers de l'entreprise
- Des bases solides pour construire ses propres processus
- Amélioration continue de la **qualité de service**
- Réduction à long terme des **coûts**
- Changement culturel vers le **service**
- Une organisation claire, systématique et auditable
- Capacité à absorber un nombre important de changements avec une grande **fiabilité**
- Un travail sur le service perçu



- L'un des processus phare du référentiel ITIL
- Positionné dans le "Service Transition" (ITIL V3)



- Planifier et manager les ressources nécessaires pour déployer une release en production
- Etablir et maintenir l'intégrité de tous les services et configurations durant leur installation (transition)
- Produire une bonne qualité d'information et de connaissance pour permettre une prise de décision pertinente entre la phase de test et la phase de mise en production
- Produire des mécanismes reproductibles de construction et de déploiement des releases dans les environnements de test et de production
- Assurer que le service peut être managé, exploité et supporté selon les exigences et contraintes spécifiées dans le service Design



Maturity Level 2

- CM - Configuration Management
- MA - Measurement and Analysis
- PMC - Project Monitoring and Control
- PP - Project Planning
- PPQA - Process and Product Quality Assurance
- REQM - Requirements Management
- SAM - Supplier Agreement Management

Maturity Level 3

- DAR - Decision Analysis and Resolution
- IPM - Integrated Project Management +IPPD
- OPD - Organizational Process Definition +IPPD
- OPF - Organizational Process Focus
- OT - Organizational Training
- PI - Product Integration
- RD - Requirements Development
- RSKM - Risk Management
- TS - Technical Solution
- VAL - Validation
- VER - Verification

Maturity Level 4

- QPM - Quantitative Project Management

Maturity Level 5

- OPP - Organizational Process Performance
- CAR - Causal Analysis and Resolution
- OID - Organizational Innovation and Deployment

4- Référentiel

- Identifier l'ensemble des composants logiciels
- Constituer (ou reconstituer) une **référence unique**
- Regrouper par domaines fonctionnels



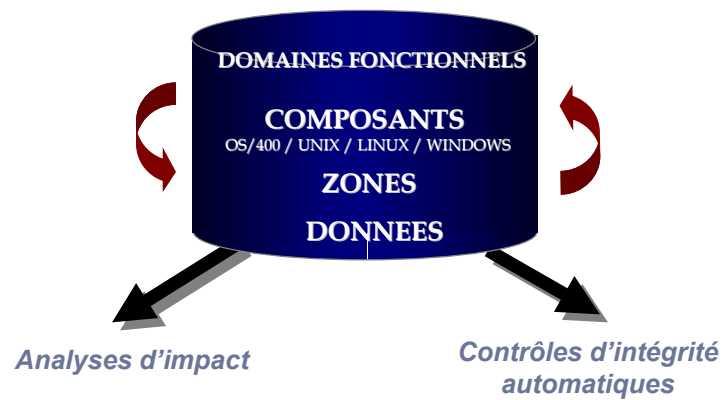
- Base centrale unique de référencement des composants logiciels
- Assurer la sécurité et le contrôle **logique** d'accès
 - Référence protégée
 - Test/recette/qualification protégés



- Vue unifiée du SI
 - Indépendamment des technologies utilisées
- Vue « Top-Down » et « Bottom-up »
 - Applications <-> Lignes de source
- Vue dynamique par niveaux
 - Développement/Production



- **CM**
 - Dématérialisation du code source
 - Optimisation de l'espace disque
 - Sécurité d'accès "par défaut"
- **Méta-données**
 - Simples pointeurs sur les sources
 - Périmètre plus large
 - Gestion des dépendances
 - Sécurité





- Le référentiel est découpé en applications
- Les applications peuvent être découpées en trois niveaux:
 - Sous-système
 - Fonctions
 - Sous-fonctions
- Problème du découpage
 - Contraintes imposées par le versionning



- = Application des règles de topologie des composants
- = Vérification de l'automatisation de la création des exécutables

5- Versionning

- Avoir des états stables
 - Connaissance
 - Indispensable à l'archivage
- Coordonner les projets d'évolution
 - Compatibilité
 - Non régression
- Maîtriser la « balance »





- Un lot de modifications
 - Ajouts, modifications, suppressions
- Un niveau d'application
- Lorsqu'une version passe en production, l'application prend ce nouveau niveau



- Contrôlée techniquement
 - Contrôles d'intégrité automatiques
- Contrôlée fonctionnellement
 - Validation utilisateur
- Close
 - => non modifiable à l'issue de cette phase
- Déployée



- Versionning de développement vs versionning de production
- Corrective vs Evolutive



- Raisonne non régression des **sources**
 - Indicateurs d'allocation
 - Mécanismes de comparaison/fusion



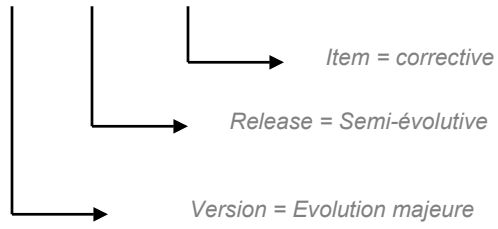
- Sert à labelliser les niveaux d'applications
- Constitue un langage commun entre:
 - Le support technique
 - Les études
 - La production
 - L'utilisateur



- Raisonne "global"
 - Développements internes +
 - Nouvelles versions de progiciels +
 - Données de paramétrage +
 - Middleware +
 - Outils d'administration (scheduler, gestion d'alertes, etc.)



88.99.A00



- Les inter-relations entre applications complexifient le versionning
 - => Synchronisation des versions
 - => Gestion des compatibilités



- Permet de gérer des dérivés d'un code source
- Résoud les problèmes de corrective sur des versions antérieures



- Fait partie intégrante des processus
 - Durant les phases de développement
 - A la mise en production
 - Archivage "légal"
 - Reconstitution d'un niveau antérieur d'une application



6- Gestion des environnements de travail



Exemple de configuration



- Définir des espaces de travail étanches
- Définir la position physique (machine, partition)
- Définir leur contenu (Σ ou Δ)
- Définir les principes de gestion des données dans ces espaces
- Définir les règles de sécurité/confidentialité
- Définir les principes de mise à jour / configuration



- Production
 - Unique ou multiple
 - Gestion de la sécurité/confidentialité
 - Base de données utilisateur



- **Référence**

- Unique par définition
- Protégé
- Base de données "semi-vide" si plusieurs environnements de production (=environnement de réplication pour la définition d'une nouvelle production)



- **Test/qualification/pré-production**

- Font partie intégrante du processus d'homologation
- Protégés
- Base de donnée "pleine"
 - => définir les principes de rafraîchissement des données



- Développement
 - Font partie intégrante du processus de développement
 - Non protégés
 - Base de donnée "pleine"
 - => définir les principes de rafraîchissement des données



- Réception
 - Servent à "décortiquer" le contenu d'une nouvelle version d'un progiciel
 - Non protégés
 - Pas de base de données (hormis des données de paramétrage)



- Formation
 - Temporaires
 - Non protégés
 - Base de donnée permanente ou créée à la demande avec des données de production



- La démultiplication des environnements démultiplie:
 - L'espace disque nécessaire
 - La charge d'administration
- Un environnement de test ne sert à rien s'il n'est pas utilisé (sic)
- La clé = données
 - => gestion rigoureuse
 - Pousser vers l'automatisation des extractions de données

7- Gestion des rôles et responsabilités

- Isolation totale de la production
 - Séparation physique (machines ou partitions logiques)
 - Elle ne doit **jamais** contenir de sources
 - Elle ne doit pas être accessible aux équipes de développement



- Responsable d'application
 - ⇔ Chef produit dans l'industrie
 - Mandaté par l'administrateur (⇔ Officier de sécurité)
 - Coordinateur des grands projets d'évolution
 - Responsable de la configuration de l'application



- Chef de projet d'une version
 - Mandaté par le responsable d'application
 - Gère son équipe de développement (approche ouverte ou hiérarchique)
 - Définit les recetteurs
 - Déclenche les mises en test
 - Coordinateur des changements dans un projet d'évolution
 - Responsable des environnements de développement



- **Développeur**

- Mandaté par le chef de projet de la version (facultatif)
- Reçoit et réalise des tâches de développement/maintenance
- Réalise les analyses d'impact au niveau technique
- Alloue les composants nécessaires
- Assure la traçabilité des évolutions
- Saisit les infos de suivi de projet
- A l'issue des tests d'intégration
 - Récupère les composants nécessitant une adaptation dans l'environnement de développement
 - Les re-transfère dans le ou les environnements de test



- **Administrateur base de données**

- Mandaté par le responsable d'application
- Seul habilité à faire évoluer des composants de type base de données (tables, vues, etc.)



▪ Recetteur

- Mandaté par le chef de projet d'une version
- Ne fait pas partie de l'équipe de développement
- En charge des tests d'intégration
- Travaille dans un ou plusieurs environnements de test
- Doit traçer les anomalies rencontrées
- Doit valider le lot de modifications à l'issue de ces tests
-
- Responsable des environnements de test/qualification



▪ Logithécaire

- Mandaté par l'administrateur
- Fait partie de l'équipe de production
- En charge de la mise à niveau de la référence
- En charge des mises en production
- Doit valider le lot de modifications
-
- Responsable des environnements de référence et de production



- **Attention: partie sensible**
 - C'est le côté négatif de la GCL perçu par les développeurs

- **Par contre: indispensable**
 - Fin de la guerre Production/Etudes
 - Partage clair des rôles

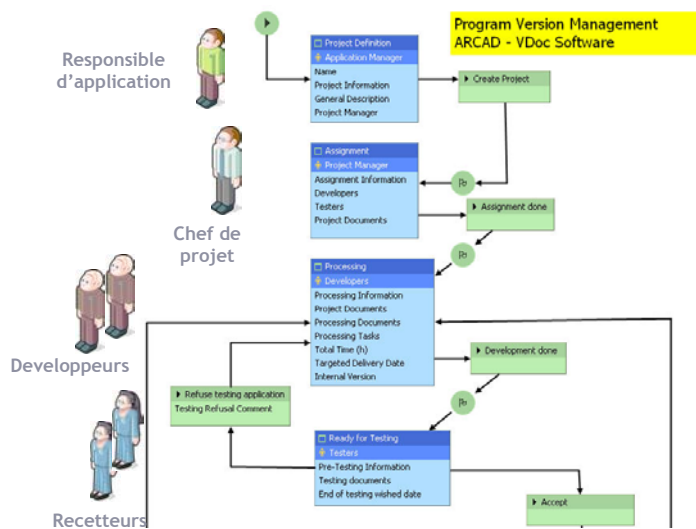


8- Workflow





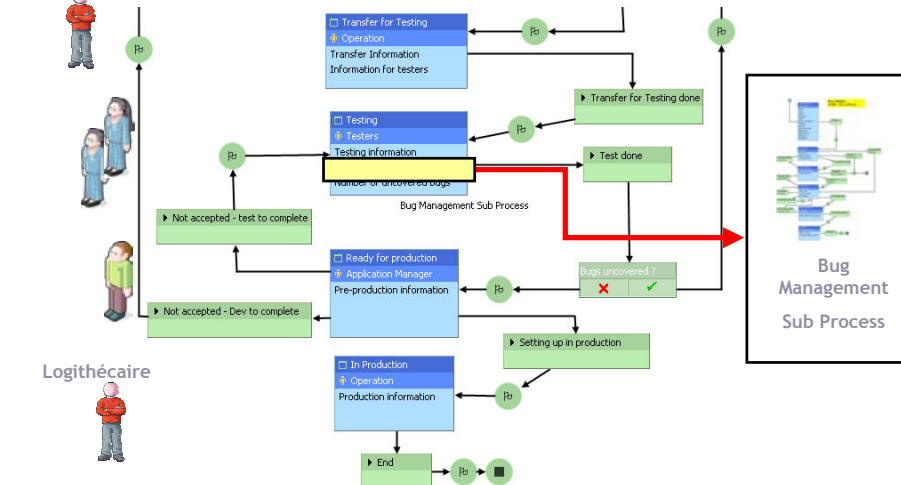
- Définition des acteurs, des tâches, des changements de contexte
 - Transfert d'information automatique
 - Validations
-
- Techno: Vdoc Process





Process de gestion de version (2ème Partie)

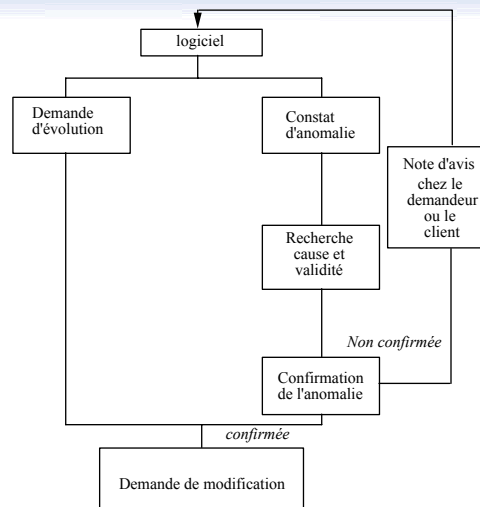
Logithécaire



9- Gestion des demandes utilisateurs

- Formaliser l'ensemble des demandes
 - Principe de traçabilité
 - Eviter le "copinage"
 - Décider des changements importants en comité

- Evaluer les charges
- Mesurer la fiabilité



Etablissement d'une demande modification
(extrait de la norme afnor Z 67-130)



- Renseigne le détail des modifications à effectuer après analyse de l'incident ou de la demande d'évolution
- Gravité/urgence
- Catégorie (corrective, évolutive)
- Assignation d'un responsable
 - Du développement
 - Du test
- Gestion de projet
 - Date au plus tard, au plus tôt
 - Charge de travail



- Reliée aux composants durant les phases de changement
- Renseigne le détail des modifications apportées
- Renseigne les infos pour le suivi de projet
- Subdivision de la notion de version
- Le lien entre la technique et l'utilisateur
- Constitue la mémoire de l'évolution de l'application
- C'est le "pourquoi ?" de la gestion de versions (qui a fait quoi, quand, pourquoi)



10- Gestion des changements de sources



Principes

- Allocation de composants ("check-out")
 - Jeton (exclusif ou non)
 - Visible via le référentiel
 - Copie du composant dans l'espace de travail
- Gestion des modifications concurrentes
 - Comparaison/fusion
 - Reports de modifications



- **L'Integrated Development Environment**
 - Est le socle de base de l'acte de changement
 - Il fournit des standards de dialogue avec les outils de GCL
 - Le niveau d'intégration des outils de GCL dans l'IDE doit être le plus élevé possible
- **IDE les plus standards**
 - Eclipse (Wdsc, Rational Application Developer, etc.)
 - Visual Studio



- La copie de l'application entière sur le PC
- Un jeton supplémentaire exclusif
- La synchronisation
 - Remontée des modifications sur le serveur
 - Redescente des modifications des autres développeurs sur la version
- Archivage local



- Le numéro de version comme instrument de dialogue et de coordination entre les équipes
- Une hiérarchie naturelle des maintenances parallèles



11- Fabrication des exécutable (build)





- Automatiser à 100% la création du code exécutable
- => prendre en charge toutes les particularités de configuration
- => de l'outillage



- Lancement de commandes avant ou après la phase de compilation
 - Ex: création d'un fichier temporaire ou Override d'un fichier
- Préservation des attributs et des droits
- Préservation des données pour les fichiers
- Gestion de compilations dépendantes
 - Ex: fichiers logiques, programmes
- Dans Arcad = Pré-compilateur + Commande unique de compilation



- Outils d'intégration continue
 - Maven, Buildforge, ...



12- Gestion des mises en production





- Allocation des composants
- Archivage des versions n - 1
- Transport des composants
- Application de la politique de sécurité
- Mise à jour du référentiel
- Clôture du lot (version)

- *Déroulement des procédures manuelles*

- Information des utilisateurs



- Définir l'ensemble des tâches à réaliser
 - Sur les composants,
 - Sur les données.
- Discerner les tâches automatisables
- Pour les tâches non automatisables => écrire des procédures

- Une tâche non automatisée est une tâche "à risque"



- Mécanismes de reprise en cas d'anomalies
 - Capacité d'interaction en temps réel sur le processus
- Mécanismes de retour à un état stable antérieur
 - Impossibilité matérielle d'analyse de l'anomalie => décision de retour arrière
- Traçabilité
 - Logs globales
 - Logs d'anomalies
 - Historique des composants manipulés



- = partie critique du processus
- Sauvegarde/restauration
 - Attention aux bases de données larges
- Mécanismes de reprise spécifiques
- Nouvelles techno = nouvelles contraintes
 - Triggers
 - Contraintes référentielles
 - Journalisation



- Langages de scripts
 - CLP
 - ANT
 - PERL
 - Etc.
- Outils dédiés
 - Arcad Process manager
 - Arcad List Manager



- Penser global (i.e: Σ plateformes)
- Viser le 100% d'automatisation
- Raisonner "éditeur"
- Si 100%, alors planifiable de nuit



13- Gestion des déploiements



Multi-environnements de production

- Impossibilité de faire migrer tous les environnements de production simultanément
- Obligation de faire de la corrective sur des plusieurs niveaux simultanément



- = Σ des versions entre le niveau d'un site et le niveau à envoyer
 - Constitution automatique de lots agrégés
 - Fusion des process de reprise



- = un lot de composants corrigés
- Le lot est constitué à partir de la tâche en amont
- Si un composant a été modifié pour plusieurs tâches, les composants des tâches sous-jacentes doivent être embarqués
- Compatible avec un certain niveau de version
 - => gestion rigoureuse des niveaux (particulièrement des niveaux de base de données)
- Les cumulatives relivrent les PTF déjà livrées
- La petite porte qui peut faire des courants d'air



- Notion de groupes d'environnements
- => Découper les tâches
 - Préparation
 - Envoi
 - Installation
- Permet de synchroniser les installations
- Rollback multi-sites
- Accusés d'installation = pilotage central



- => Process d'installation flexibles
- Mise à jour manuelle des accusés d'installation
- Multi-environnements sur site distant
 - => mises en production "locales"

Conclusion

- Identifier les faiblesses actuelles
- Prendre de la compétence
 - Formation
 - Prestation externe
- Communiquer



- www.cmcrossroads.com
 - LE site de référence sur la GCL
- www.cmcrossroads.com/cgi-bin/cmwiki/view/CM/
 - Un wiki sur le sujet
- www.cmresourceguide.com/
 - Un guide de référence
- www.itsmf.fr