

Eric DEWITTE¹

Valider un LIMS : quelques réflexions

RÉSUMÉ

Cet article présente quelques réflexions pratiques sur la mise en œuvre d'un projet de validation lié à l'implémentation d'un système LIMS (Laboratory Information Management System). Le propos de cet article n'est pas de décrire la méthodologie de validation dans son ensemble. L'accent est ici mis sur la valeur ajoutée de la validation, nous abordons ce sujet selon une approche pragmatique en présentant des exemples concrets autour de quelques points clés de la démarche de validation.

MOTS-CLÉS

LIMS, validation informatique, GAMP

I - Validation informatique : quelle approche ?

La validation informatique doit permettre d'établir qu'un système répond aux besoins des utilisateurs mais aussi aux contraintes réglementaires attachées au secteur d'activité de l'entreprise. Si tel n'est pas le cas, les conséquences peuvent être désastreuses pour le client final (patient par exemple) et pour l'entreprise (refus de distribution, arrêt de production).

Les laboratoires soumis aux exigences réglementaires et ayant comme projet la mise en place d'un système de type LIMS (Laboratory Information Management System) se doivent donc d'intégrer très en amont de leur réflexion les aspects liés à la validation de leur future application.

Cette réflexion doit être menée dès les premières étapes du projet pour les raisons suivantes :

1) Le choix même de la solution peut être dicté par le résultat d'un audit fournisseur, cette démarche étant d'ailleurs préconisée par les différentes instances réglementaires.

2) Les interactions étant très fortes avec les étapes classiques de mise en place d'un LIMS, la validation doit être intégrée au planning très en amont.

3) Bien évidemment et compte tenu de ce dernier point, les étapes de validation auront un très fort impact en terme de budget et ressources qu'il est indispensable de prévoir et budgéter en amont.

4) Enfin, les activités de validation et d'implémentation font généralement appel aux mêmes ressources internes clés et de surcroît au même moment.

Cette intégration de la validation, la motivation et la démarche qui vont être choisies reposent généralement sur deux types d'approches : l'une dictée par la « forme », l'autre motivée par le « fond » (figure 1).

1) La « forme » : cette approche généralement motivée par la peur de l'inspection a tendance à oublier la définition même et les fondements de la validation.

2) Le « fond » : cette approche est généralement

motivée par la qualification des fonctions du logiciel. Elle a également tendance à oublier la définition même de la validation et peut s'avérer au final insuffisamment documentée.

Nous préconiserons dans la suite de cet article une autre voie : celle du pragmatisme. Cette dernière vise à limiter l'effort de validation tout en garantissant la validité de la démarche et du système lui-même.

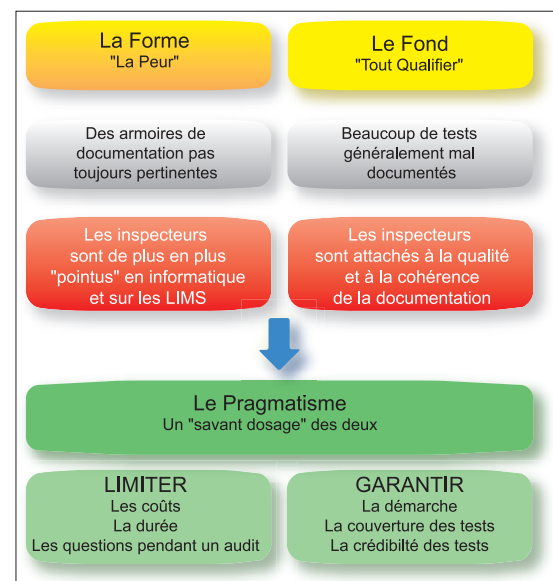
Le poids généralement admis de la validation par rapport au coût de mise en place (implémentation) du logiciel est de l'ordre de 20-40 % :

- 35 à 40 % si le projet ne s'accompagne pas d'achat de licences ni de matériels ;
- 20 à 25 % dans le cas contraire.

Un des objectifs clés des sponsors du projet sera donc de limiter au maximum ce coût inhérent à la validation tout en garantissant une démarche cohérente et crédible, facteur de réussite pour un audit futur.

Figure 1

Les approches de la validation.



¹MB Développement – 5, rue Baldung Grien – 67000 Strasbourg – E-Mail : eric.dewitte@mbdev.fr – Tél. : 09 62 28 12 01 – www.mbdev.fr

Projet

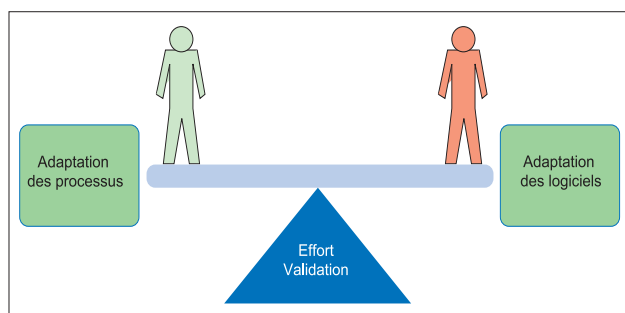
Valider un LIMS : quelques réflexions

Avant d'aller plus loin dans notre propos, il nous semble de bon aloi de revenir à la définition même de la validation. Une définition étant : « *L'établissement de la preuve, en conformité avec les principes de bonnes pratiques de fabrication, que la mise en œuvre ou l'utilisation de tout processus, procédure, matériel, matière première, article de conditionnement ou produit, activité ou système permet réellement d'atteindre les résultats escomptés.* (2) »

Nous nous focaliserons ici sur le terme « résultats escomptés » appliqué à un LIMS. Il s'agit en fait de s'assurer que l'application finale répond aux besoins métiers et réglementaires exprimés par les utilisateurs. L'objectif, même si il n'est pas toujours bien compris, est clair. Par contre les moyens et les approches pour y parvenir sont multiples. Compte tenu de cette définition, nous pouvons d'ores et déjà mettre en avant deux facteurs clés de réussite d'un projet de validation :

- 1) Pour pouvoir être validé un besoin se doit d'être « SMART » : Spécifique, Mesurable, Acceptable (audit), Réaliste et établi dans le Temps (priorité).
- 2) Moins il y a de besoins complexes et donc moins il y a d'adaptation du progiciel et plus l'effort de validation sera limité (figure 2).

Figure 2
La balance des besoins.



II - Les points clés des réglementations

Une fois ces deux fondements établis, regardons de plus près les réglementations en vigueur en termes de validation informatique. Même si les différents textes évoluent et n'ont pas toujours la même approche sur la forme, quelques idées directrices en ressortent et doivent être traitées avec toute l'attention possible :

- La définition et la gestion des besoins utilisateurs ;
- L'analyse de risques ;
- La qualification de conception ;
- La gestion des changements avant, pendant et après le démarrage ;
- Le maintien de l'état validé de l'application.

La notion de risque étant présente à chacune de ces cinq étapes.

Sur la forme, certaines divergences subsistent que ce soit sur les termes ou sur les documents à fournir. Ainsi certains préconisent le terme de vérification plutôt que validation ou qualification. Certains parlent de tests systèmes et d'UAT (User Acceptance Testing) pendant que d'autres parlent de Protocoles de Qualification d'Installation,

Opérationnelle et de Performances.

Peu importe, il faut que la démarche soit fondée, pertinente et cohérente. Une des responsabilités des instances réglementaires de l'entreprise est de mettre en place les procédures internes consolidant les éléments issus des réglementations. Dès lors, la validation se fera en conformité avec ces procédures internes.

III - Validation : Le pragmatisme en quatre suggestions concrètes

Nous allons ici nous focaliser sur quelques points précis en donnant des pistes concrètes pour mettre en place une démarche pragmatique.

1. Les besoins utilisateurs

Comme indiqué précédemment ils doivent être « SMART ». De plus chaque besoin doit comporter les éléments suivants :

- Un identifiant unique « manuel » qui sera référencé dans les autres documents ;
- Une catégorie (rapport, sécurité,...) ;
- Un demandeur ;
- Une description et éventuellement une référence à un document détaillant le besoin ;
- Un impact cGxP (bonnes pratiques de fabrication, laboratoire, clinique, distribution,...) ;
- Un classement (priorité) ;
- Utiliser un tableur plutôt qu'un éditeur de texte qui permettra de faire des tris et regroupements.

La partie la plus ardue consiste toutefois à rester le plus proche possible du progiciel standard. Ainsi, une bonne communication et un processus de conduite du changement bien anticipés par les responsables devraient permettre de satisfaire ce point critique. Il faut gagner la confiance des utilisateurs clés du projet (figure 3).

Ces phases de recueil des besoins sont en général menées avec le chef de projet, un responsable

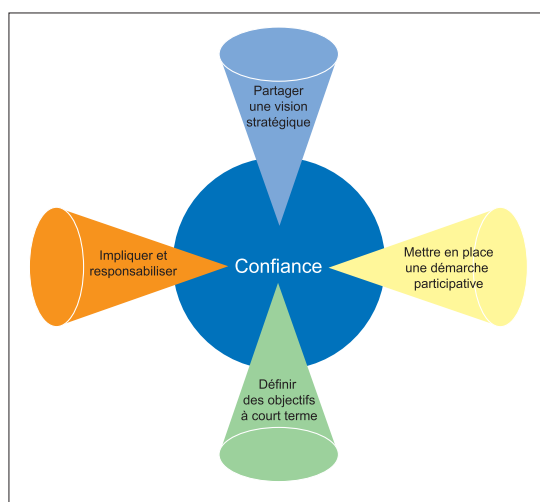
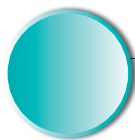


Figure 3
La conduite du changement.



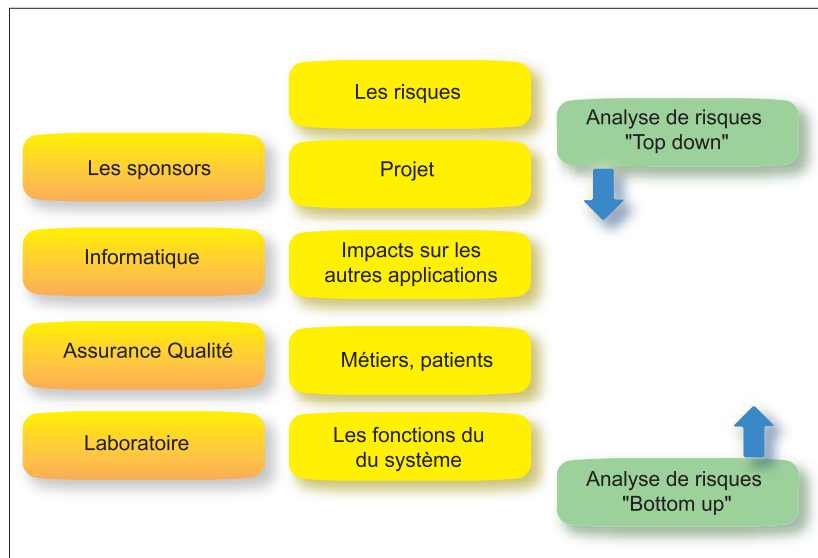
qualité et les « Super-Utilisateurs ». Ces derniers, se trouvent de ce fait responsabilisés et deviennent une des pièces maîtresse du projet.

Le « Super-Utilisateur » tient donc de la « perle rare », du « mouton à cinq pattes ».

Il doit nécessairement présenter les compétences/qualités suivantes :

- Une maîtrise du fonctionnement (« workflow ») du laboratoire dans son ensemble ;
- Une connaissance des aspects qualité ;
- Une attirance pour l'informatique ;
- Savoir prendre des décisions (pour les autres...);
- Un relationnel à toute épreuve ;
- Une « tête bien faite » (esprit de synthèse, ne pas se noyer dans les détails) ;
- Savoir se remettre en question ;

Il doit donc être choisi avec précaution.



2. L'analyse de risques

Elle doit suivre une approche descendante (top-down) et remontante (bottom-up) (1) (figure 4) :

- Identifier et classer les risques liés au projet ;
- Réaliser une étude d'impacts sur les autres applications ;
- Identifier les données métiers critiques ;
- Identifier en détails les risques liés à l'application et les classer ;
- Définir le premier niveau GAMP5 (Good Automated Manufacturing Practice) de priorité des risques ;
- Définir les actions nécessaires (ajout de tests, procédures,...).

L'étude des données métiers critiques sera essentiellement menée avec les responsables qualité. Selon la réglementation applicable, cette étude s'attachera à relever les données critiques utilisées et générées par le système et ayant des conséquences potentielles vis-à-vis du patient final. Par voie de conséquences, ces risques métiers pourraient s'avérer financiers pour l'entreprise.

Cette étude pourra être facilitée en utilisant les outils et les méthodologies proposées par exemple par l'ICH (International Conference of Harmonisation). Les sessions de travail peuvent alors être découpées selon l'ordre et les sections proposées par l'ICH. Il est important de noter que l'ICH Q9 fournit désormais des outils et procédés liés à la gestion des risques.

Nous suggérons également, d'ajouter une étape de pondération des risques après la classification de type GAMP5 pour définir un niveau de priorité « final ». La classification GAMP5 se focalise sur la gravité, l'occurrence, la détection pour déterminer une priorité du risque. Une fois cette priorité établie, une pondération ultérieure, dont le facteur chiffré est déterminé par le client, pourra se faire par exemple sur les points suivants :

- Fonction standard/non standard ;
- Fonction testée dans un « kit » de validation du fournisseur ;
- Niveau de test SAT/FAT (System/Factory Acceptance Testing) suffisant ;

- Déjà testée dans une version précédente ;
- Fonction non impactée par une migration ;
- Procédure (SOP - Standard Operating Procedure) existante,...

L'étude de ces points et un calcul formel permettront d'établir une priorité de risque finale, cette dernière déterminant directement le nombre et le niveau de tests (Qualification Opérationnelle) à réaliser.

Il est important de noter que cette analyse de risque doit être clairement établie et intégrée dans le plan de validation, mais également au processus de gestion des demandes de changements et en phase de maintenance de l'application.

3. La matrice de traçabilité

La matrice de traçabilité permet de corréler et de référencer les informations depuis les besoins utilisateurs jusqu'aux tests utilisateurs permettant ainsi de s'assurer de la cohérence de la démarche de validation. Cette exercice peut s'avérer fastidieux et le document final très lourd à gérer. Ainsi, nous proposons quelques solutions concrètes :

- Créer des références « manuelles » dans tous les documents (pas de numérotation automatique) ;
- Utiliser un tableur plutôt qu'un éditeur de texte ce qui permettra de faire des tris et regroupements ;
- Intégrer dans la matrice des colonnes relatives à la « gestion de projet » (date workshop, traitée/repousée, date approbation) ;
- Scinder le document en plusieurs parties (onglets) pour en faciliter la gestion et la lisibilité ;
- Utiliser ce document dans le cadre de la Qualification de Conception.

4. Le déroulement des tests

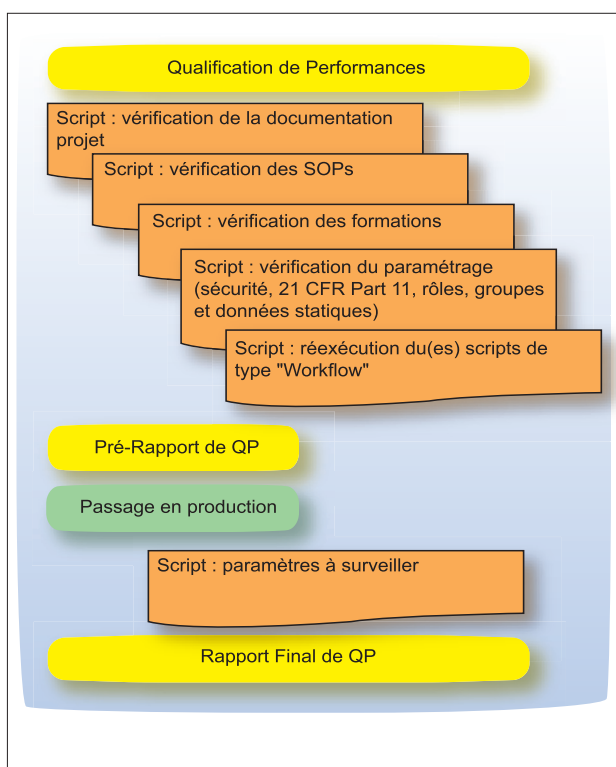
Le déroulement des tests généralement à la charge des utilisateurs doit être facilité au maximum afin d'en limiter la durée. De plus, tout doit être mis en œuvre afin d'éviter le déclenchement d'un processus formel de gestion des anomalies alors qu'il

Figure 4

L'analyse de risques.

Valider un LIMS : quelques réflexions

Figure 5
La qualification de performances.



s'agit de problèmes relatifs au test lui-même ou au jeu de données associé.

Ainsi, nous suggérons de :

- Utiliser au maximum les compétences de l'intégrateur pour aider à la préparation des données de test.
- Automatiser au maximum le chargement de données de test pendant l'exécution de la fiche de test. La partie relative à la création des données doit être limitée au strict minimum. C'est un gain de temps et cela évite des erreurs durant la création des données pouvant entraîner des « fausses » anomalies, voir bloquer la poursuite des tests.
- Référencer de manière explicite et unique les données utilisées et générées en y intégrant si possible la référence de la fiche de test et un numéro d'exécution. Ceci permet de retrouver facilement les données liées à l'exécution de cette fiche de test.
- Rendre les fiches de tests indépendantes les unes des autres permettant ainsi une exécution en parallèle et donc un gain de temps. Utiliser une « copie » de la fiche de test pour gérer les copies d'écrans et preuves générées.
- Séparer les fiches en grandes familles : les données statiques, les données dynamiques, le 21 CFR Part11, les éditions,...

5. La Qualification de Performances (QP)

La limite entre la qualification opérationnelle et la qualification de performances est parfois floue. Une proposition est de scinder cette qualification en deux étapes. Une avant le passage en production donnant lieu à un rapport de QP provisoire et à une autorisation de mise en production. Puis l'autre partie de ce protocole sera exécutée après le démarrage et sur une période permettant de surveiller le sys-

tème ; elle permettra grâce à la collecte d'informations pertinentes de statuer définitivement sur son utilisation. Elle donnera lieu à un rapport final de QP (voir figure 5).

IV - Validation : qui fait quoi ?

Si le client décide de faire appel à des ressources externes pour l'aider dans la mise en place et la validation de son système, il aura généralement en tête la locution bien connue : « *On ne peut être juge et partie* ». Ainsi, il sera tenté de faire appel à deux prestataires distincts. Cette démarche est compréhensible lorsque l'éditeur de la solution propose des services de validation associés. Elle l'est moins lorsque le client fait appel à un intégrateur indépendant proposant les deux gammes de prestations. Dans ce cas l'intégrateur devra proposer une équipe intégrée ayant les deux compétences mais avec une répartition claire des responsabilités sur les activités d'implémentation et de validation.

Au travers de quelques étapes, nous allons donc voir que cette locution n'est pas nécessairement un bon fondement dans la constitution des équipes projets.

1. Le recueil des besoins (URS – User Requirements Specifications)

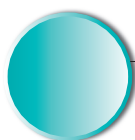
Cette étape est commune à tout projet informatique mais est également le point d'entrée du processus de validation. L'intégrateur accompagné de l'équipe projet du client devra s'assurer du bon référencement des besoins, de leur expression « SMART » et de la prise en compte des aspects réglementaires. La connaissance du produit constitue ici une valeur ajoutée permettant très en amont d'identifier et d'alerter sur un besoin complexe, voire une impossibilité, par rapport aux capacités du produit sélectionné.

2. L'analyse de risques

L'intégrateur de part sa double compétence métier sera à même d'assister l'assurance qualité sur les aspects réglementaires de cette analyse de risques. Sa compétence informatique et sa connaissance de la solution permettront par ailleurs de traiter les aspects fonctionnels et système avec les utilisateurs clés et le chef de projet. Enfin, en participant activement à cette étape, l'intégrateur aura les éléments qualitatifs et quantitatifs permettant d'établir et d'anticiper la charge de validation.

3. La Qualification d'Installation (QI)

Cette étape est généralement menée par le service informatique, le chef de projet et l'intégrateur. Ce dernier s'assurera que cette étape sera menée « à blanc », puis sur la base de validation et sur la base de production. La valeur ajoutée de l'intégrateur est ici



la parfaite connaissance technique des composants informatiques ainsi que des modules ou fonctions qu'il a pu développer lui-même et livrer accompagnés de notice d'installation.

4. La Qualification de Conception (QC)

Cette étape doit être menée avec le chef de projet par un consultant n'ayant pas participé à la conception technique de l'application. Par contre, ce dernier ayant participé à la rédaction des besoins utilisateurs et ayant une connaissance fonctionnelle du produit sera très efficace dans cette étape de vérification de la conception de l'application par rapport aux besoins exprimés et aux spécifications fonctionnelles.

5. La Qualification Opérationnelle (QO)

Cette étape est généralement la plus pénalisante en termes de ressources et de planning. Il est donc souhaitable de l'anticiper au maximum et ce dès que les spécifications fonctionnelles et l'analyse de risques ont été approuvées. L'expérience montre qu'il est très difficile de rédiger des fiches de tests pertinentes et exhaustives en se basant uniquement sur le document de spécifications fonctionnelles et ce d'autant plus si ce document a été rédigé par une autre équipe. C'est ici que l'intégration des équipes implémentation et validation prend tout son sens. En effet, un travail en étroite collaboration permet d'obtenir une grande réactivité si des questions de détails subsistent sur les spécifications fonctionnelles ou si la rédaction et donc la pré-exécution de la fiche de test met en avant une incohérence de fonctionnement. La modification peut alors être traitée plus efficacement à ce niveau sans devoir recourir plus tard à un processus de gestion de changements formel et coûteux. Cette interaction étroite permet d'optimiser le planning et d'avoir une cohérence optimale entre les spécifications fonctionnelles, les fiches de tests et le résultat attendu.

Si la rédaction des fiches de test peut être confiée à l'intégrateur, l'exécution de ces fiches est généralement de la responsabilité des utilisateurs. Là encore, la connaissance du produit permettra à l'intégrateur de mettre en place des automatismes (données de tests préparées et chargées) permettant de limiter l'effort utilisateur lors de l'exécution des fiches de tests.

6. La Qualification de Performances (QP)

Des sections de ce protocole peuvent être confiées à l'intégrateur. De la sorte, sa valeur ajoutée technique pourra permettre de bâtir et de mettre en place une surveillance du système efficace ainsi que les aspects de vérification et contrôle du paramétrage utilisateur.

BIBLIOGRAPHIE

(1) LECOINTE C., CHANAL P., Mise en oeuvre des LIMS, 2007, *Spectra Analyse* 259, 45-47

V - Conclusion

Quelle que soit la méthodologie utilisée, la validation d'un système d'information de laboratoire doit suivre la voie du pragmatisme.

La validation ne doit pas être considérée comme une contrainte. En revenant à sa définition même, en impliquant les utilisateurs très en amont, en soudant les équipes implémentation et validation, elle procure une réelle valeur ajoutée en terme de satisfaction utilisateur.

Elle doit soulever des « anomalies » mais doit surtout permettre de les tracer, de les résoudre et ainsi de faire évoluer la qualité du logiciel et la satisfaction utilisateur.

La méthodologie utilisée doit être fondée sur une démarche logique, explicite et compréhensible. Enfin, le processus de validation doit être appliqué à l'intégralité du projet mais il doit se poursuivre et être intégré tout au long de la durée de vie de l'application.

A l'instar de la démarche qualité au sens général, l'exercice que représente la validation peut se révéler :

1) intellectuellement intéressant

Identifier et se focaliser sur les points clés, être capable de synthétiser un besoin, accepter de remettre en question sa façon de travailler, simplifier un processus, analyser un risque sont des exercices enrichissants.

2) ludique

Cela peut sembler curieux, mais l'on peut s'en amuser à condition de ne pas en abuser...

Ainsi la construction des fiches et des jeux de tests associés, la rédaction des manuels utilisateurs et surtout des exercices associés pour les habilitations, font appel à des notions et des qualités du domaine de la créativité.

3) constituer un vecteur de communication et un moteur de réussite

La validation d'une application impose un degré et un processus de communication à tous les niveaux du projet impliquant de nombreux acteurs (Support Informatique, Informatique scientifique, Utilisateurs, Assurance Qualité labo et informatique, Intégrateur). De ce processus de communication et de l'organisation mise en place dépendra la réussite du projet. L'entreprise et les sponsors se doivent de communiquer sur la réussite d'un projet informatique intégrant une validation formelle.

En effet, ce n'est pas tous les jours qu'une équipe de collaborateurs n'ayant pas les mêmes objectifs personnels, contraintes et valeurs parviennent ensemble à un même objectif commun : une application informatique répondant à la fois aux besoins des utilisateurs, aux exigences réglementaires, aux organismes qualité internes et externes et aux responsables informatiques en termes de budget de mise en place et de maintenance sur le long terme.

(2) Bonnes Pratiques de Fabrication Françaises, 2006