



## **Optimisation de la qualité des logiciels : Assurer l'équilibre entre l'évolution de l'entreprise et le risque**

*Michael Lundblad, directeur de programmes,  
logiciel Rational, IBM Software Group*

*Moshe Cohen, directeur commercial, logiciel Rational,  
IBM Software Group*

---

## Sommaire

---

- 2 Introduction**
- 4 Assurer l'équilibre entre évolution et risque**
- 11 Concrétiser les promesses**
- 13 Les solutions de gestion de la qualité**
- 16 Synthèse**

***Que nous en soyons conscients ou non, nous agissons au quotidien en interaction avec un grand nombre de logiciels.***

## Introduction

Une récente étude d'IBM menée auprès de directeurs généraux montre que 66 % des dirigeants s'attendent à voir leur entreprise aux prises avec de nombreux changements, largement influencés par l'innovation et l'évolution. Pour y faire face, les services de développement de logiciels doivent sortir plus vite un logiciel vital pour l'entreprise, ceci au prix toutefois d'une augmentation des risques et dans bien des cas d'une qualité insuffisante. La question est alors la suivante : Comment économiser du temps et réduire les coûts sans sacrifier la qualité ?

Bien que l'aviation soit plus que centenaire, la plupart d'entre nous restent émerveillés par la vue d'un avion qui décolle et atterrit. Et, plus incroyable encore, comment un logiciel de pilotage automatique peut-il contrôler et exécuter toutes les fonctions pratiquement sans intervention du pilote ? Etudions la maniabilité de l'avion de combat anglo-américain « JSF », également connu sous le nom de F-35. En ascension, il peut couper les gaz en plein ciel, évoluer en vol stabilisé, et reprendre son vol supersonique. Ces manoeuvres sont aussi commandées par un logiciel.

Et quoi dire de l'automobile ? Tous les véhicules d'aujourd'hui intègrent entre 100 et 150 calculateurs. Il s'agit pour l'essentiel de réseaux de calculateurs de roues, qui communiquent en permanence entre eux, et d'une technologie globale de services : télémétrie, système global de positionnement, sécurité et système diagnostic embarqués.

Nous interagissons au quotidien directement et indirectement avec un très grand nombre de logiciels. Les stocks de nos hypermarchés préférés sont gérés par logiciel ; nos comptes chèques et nos comptes d'épargne sont eux aussi contrôlés par un logiciel ; nos hypothèques, nos factures d'eau, de gaz, d'électricité, nos antécédents médicaux et les conditions à remplir pour la prescription de médicaments sont tous gérés par logiciels.

Ces avancées nous apportent confort et progrès, mais ils s'accompagnent aussi de risques et de dépenses inhérentes à ces avancées. Paul Ehrlich a dit : « L'erreur est humaine, mais pour vraiment la corriger, il vous faut un ordinateur ». Etudiez ces problèmes liés au logiciel, que le site devtopics.com considère comme faisant partie des 20 plus grandes catastrophes.<sup>1</sup>

---

Points clés

---

Le « Lundi noir » (19 octobre 1987), le Dow Jones affichait 508 points, perdant 22,6 % de sa valeur. Le S&P 500 chutait de 20,4 %. La plus forte perte en une seule journée que Wall Street ait jamais subie avait été causée par une panne de logiciel. Alors que les investisseurs liquidait leurs titres en masse, les programmes des transactions crachaient un flot d'ordres de vente, entraînant le plantage des systèmes et laissant les investisseurs dans le noir complet et dans l'incapacité d'effectuer des transactions.

En décembre 2008, un journal israélien de premier plan (*Yediot Acharonot*) titrait en première page : « Le bogue qui rend fous les hôpitaux ». L'article ne traitait pas d'une bactérie ou d'un virus qui rendait malade, mais d'un bogue informatique qui avait eu pour effet d'associer des résultats de prises de sang aux mauvais patients. Le cabinet du ministre de la santé israélien avait déclaré que le défaut se manifestait lors de la transmission électronique des résultats des laboratoires aux hôpitaux.

La livraison d'un logiciel de qualité n'a jamais été aussi vitale pour les entreprises. Elle est en fait vitale pour la survie de l'entreprise. Les directeurs des services informatiques doivent équilibrer méticuleusement l'évolution de l'entreprise avec la gestion des risques inhérents à son activité. Les exigences changent souvent, les projets prennent du retard sur le planning, les coûts sont passés au peigne fin, mais la qualité doit être améliorée ou les entreprises doivent cesser leurs activités. Cette tâche à priori impossible oblige les équipes de fourniture de logiciels à sortir des sentiers battus, étudier des méthodes réactives de développement et remettre en cause des méthodes traditionnelles bien ancrées. Elles abordent maintenant la gestion de la qualité différemment en la considérant comme une action continue qui exige un travail collaboratif en équipe, des techniques automatisées, et une communication de l'information ou la gestion d'un tableau de bord des métriques de façon précise et en temps réel pour faciliter une analyse concertée et bien menée.

Ce document étudie l'équilibre à trouver entre l'évolution et le risque. Il analyse les raisons pour lesquelles le développement d'un logiciel change les perspectives de l'entreprise, et traite des solutions à retenir afin d'améliorer la gestion de la qualité et optimiser la mise sur le marché, la réduction des coûts et la qualité du produit.

***Dans une entreprise où il faut atteindre un haut niveau de qualité pour un minimum d'investissement, gérer la qualité des logiciels est vital.***

---

Points clés

---

**Sur les 20 % du reliquat d'un budget informatique après les frais opérationnels, les trois-quarts sont investis dans la détection et l'élimination des défauts.**

**Assurer l'équilibre du « triangle de fer » est essentiel pour les sociétés qui veulent continuer à innover sans augmenter leurs risques.**

### Assurer l'équilibre entre évolution et risque

La qualité d'un logiciel impose aux équipes de trouver un équilibre entre trois dimensions : le champ d'application (les exigences), les coûts et le délai (figure 1). Selon l'étude que vous lisez, près de 80 % du budget informatique d'une entreprise concerne les frais opérationnels. Sur les 20 % restant, 70 à 80 % passe en recherche et en élimination des défauts des applications existantes. De ce fait, les finances affectées aux ressources nécessaires pour garantir la qualité du logiciel sont fixées au mieux.

L'expérience d'IBM acquise dans des entreprises de dimension mondiale et les données collectées par les chercheurs qui travaillent avec des centaines d'entreprises montrent à l'évidence que la plupart d'entre elles investissent 25 % ou plus de leur cycle de développement et des coûts dans l'assurance qualité. De plus, 30 % des coûts lié aux projets de développement de logiciels est associé aux opérations de reprise en fabrication, et 70 % de ce montant est lié aux erreurs dans les spécifications.<sup>2</sup> Avec l'ensemble des outils et des processus de développement en cascade dont on dispose actuellement, la qualité du logiciel nécessite plus de temps et génère plus de coûts que jamais. Pour rester compétitives, les entreprises doivent trouver les moyens d'améliorer la qualité tout en réduisant les délais de commercialisation pour un logiciel vital pour l'entreprise.

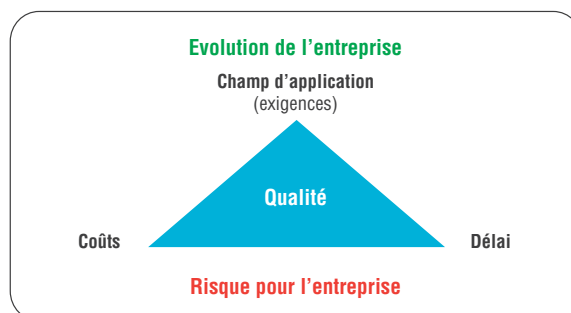


Figure 1 : « Le triangle de fer » - assurer l'équilibre entre évolution de l'entreprise et risque.

---

Points clés

---

***Des exigences mal définies ou mal gérées – ou modifiées sans un suivi précis – peuvent être à l'origine de graves défauts.***

***Les mises à jour de logiciel peuvent faire aussi le lit de défauts qui peuvent envahir le code source.***

Les exigences ne sont jamais figées... elles ne font que changer constamment. Une grande partie des défauts les plus graves s'infiltrer dans un logiciel au travers des exigences : définition vague, gestion insuffisante ou diffusion incomplète des exigences à tous les intéressés. Ces problèmes ne surviennent pas seulement au tout début d'un projet, mais pendant tout le cycle de vie des applications. Prendre ses désirs pour des réalités pour les dirigeants, c'était généralement définir les exigences et les figer de façon à rendre impossible toute modification par la suite. Ceci permettait certainement d'améliorer la qualité du logiciel, mais la réalité d'aujourd'hui est tout autre : les exigences évoluent sans cesse. Elles sont le reflet de l'accélération du monde des affaires, de l'urgence de rester compétitif et de la nécessité de se conformer à la réglementation imposée à un grand nombre de nos processus.

Étudions le cas d'un directeur de services informatiques qui se rend compte que la nouvelle réglementation va l'obliger à changer un certain nombre d'applications. Celui-ci et son équipe doivent définir de quelle façon ces changements toucheront le logiciel de l'entreprise, puis adapter les tests à ces changements. Dans ce laps de temps, l'étendue des changements, le délai nécessaire à tester manuellement ces changements et leurs coûts convainquent les entreprises du fait que le renforcement de l'automatisation, les tests de régression et les procédures itératives comme Agile sont le meilleur moyen de gérer ces effets indésirables qui accompagnent inévitablement les mises à jour de logiciel.

Analyse de l'impact des changements des exigences dans un environnement mondial. Poursuivons avec notre exemple de changement de la réglementation. La grande majorité des entreprises gèrent habituellement ces changements le plus naturellement du monde : elles font analyser par leurs équipes les impacts sur leurs applications et agissent en conséquence. Dans ce scénario, nous aboutirions probablement à différentes équipes effectuant simultanément les mêmes activités sur les mêmes produits. Il est toutefois courant pour les entreprises de mener de front plusieurs projets sur différents sites, en interne ou en externe ; tous doivent être actualisés pour leur mise en conformité avec la nouvelle réglementation.

---

Points clés

---

***Il est essentiel de collaborer à la protection contre les éventuels défauts : tous les membres de l'équipe peuvent ainsi communiquer lors de la définition des exigences et des mises à jour du logiciel.***

***Si une application est configurée de nombreuses façons, il est tentant de ne pas tester certaines configurations, mais on augmente ainsi le risque de livrer un logiciel défectueux.***

On ne peut que supposer que cette approche génère beaucoup de pertes et augmente aussi les risques. Qu'en est-il lorsque différentes équipes n'ont pas la même interprétation des nouvelles exigences ? Et qu'en est-il si elles mènent en fin de compte des tests élémentaires qui illustrent différents comportements alors qu'ils sont supposés identiques ? Gérer un grand nombre de projets sans augmenter de façon significative le risque nécessite une procédure de tests où il est possible de rattacher un ensemble unique d'exigences à différents projets. Il faut pouvoir définir des tests élémentaires dont on peut générer et / ou déduire des scripts tests. Il faut pouvoir partager et réutiliser les tests élémentaires pour l'ensemble des projets. Cela exige un programme de tests qui, comme les exigences, n'est pas un document figé ou une page Web structurée. Il s'agit d'un actif vivant, dynamique, sans cesse actualisé, permettant de définir et de partager des procédures de tests et des stratégies, d'un système de communication de l'information dans l'entreprise concernant les objectifs de qualité, et – tout aussi important – de faciliter la conduite des activités collaboratives de l'équipe, comme les révisions et les validations.

Les risques associés aux environnements d'exécution

Le logiciel d'aujourd'hui est d'une telle complexité qu'il est, très souvent, tout simplement impossible d'assurer le zéro défaut des applications. Dans d'autres cas, même si cela est techniquement possible, cela ne l'est pas au niveau financier. Prenons, par exemple, le cas d'un portail orienté client à tester. La première étape consiste pour l'équipe chargée des essais à développer des tests élémentaires, certains manuels, d'autres automatisés. Selon la complexité de l'application et de l'environnement auquel elle est destinée, un cycle de tests peut prendre de quelques jours à plusieurs semaines, voire des mois. Supposons que le portail n'accepte que deux langues et quatre navigateurs Internet ou versions, et qu'il fonctionne en s'appuyant sur trois bases de données et un serveur application existants. Ces seuls composants impliquent dans tous les cas une répétition du cycle de tests 24 fois (2 x 4 x 3 x 1). Autrement dit, ce qui demandait une heure de test en demande maintenant 24. Ce qui demandait une semaine de 40 heures demande maintenant 24 semaines, pratiquement 6 mois. Au fur et à mesure que le nombre de configurations à tester augmente, il est tentant de ne pas tester certaines d'entre elles, ce qui augmente le risque de livrer un logiciel défectueux.

---

Points clés

---

***IBM s'est rendu compte que le fait de tester un sous-ensemble réduit de combinaisons de configurations permet aux entreprises de détecter très rapidement un très fort pourcentage de défauts.***

IBM a recherché une façon prometteuse de réduire de façon significative le risque associé à un large éventail d'environnements d'exécution. Les résultats ont montré l'existence d'un sous-ensemble réduit de combinaisons capable de détecter très rapidement un très fort pourcentage de défauts. Il permet essentiellement une détection plus rapide des défauts. Comme l'illustre la figure 2, des tests menés dans différents environnements d'exécution suivent la courbe type « lente », lorsque des défauts sont détectés sans priorité spécifique. Leur détection à long terme se situe essentiellement dans la zone aléatoire, les plannings sont alors remis en question. Mais grâce à des optimisations intelligentes, la détection de défauts à long terme intervient dans la zone optimisée, où ils sont détectés beaucoup plus tôt au cours du cycle de tests, ce qui se traduit par une économie de temps et une réduction sensible du risque de livrer un logiciel défectueux.

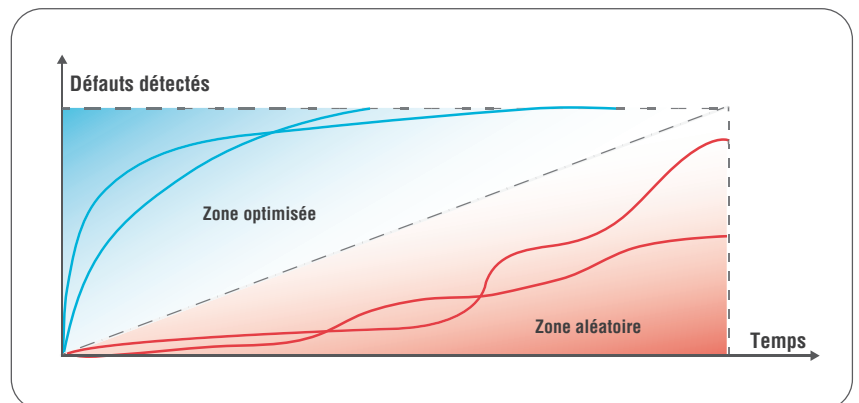


Figure 2 : Détection plus rapide des défauts grâce à des optimisations intelligentes

---

Points clés

---

**L'intégration de la gestion de la qualité dès le début du cycle de développement d'un logiciel permet aux entreprises d'éviter des coûts élevés d'élimination des bogues dans la phase ultérieure de développement.**

Un nouvel artisan de l'évolution dans la gestion de la qualité

La fourniture de logiciels de qualité oblige les entreprises à mettre en place un système de gestion collaborative s'appuyant sur des techniques automatisées et accompagnées d'une gestion réaliste des projets tout au long du cycle de vie du logiciel à livrer. Les contrôles traditionnels de l'assurance qualité (AQ) valident simplement le fait que le logiciel développé répond aux attentes de l'utilisateur en termes de fonctionnalité, de disponibilité et de performances avant son déploiement. Détecter les défauts en phase de contrôles AQ est bien plus onéreux et laborieux qu'au début du cycle de développement des applications (figure 3).

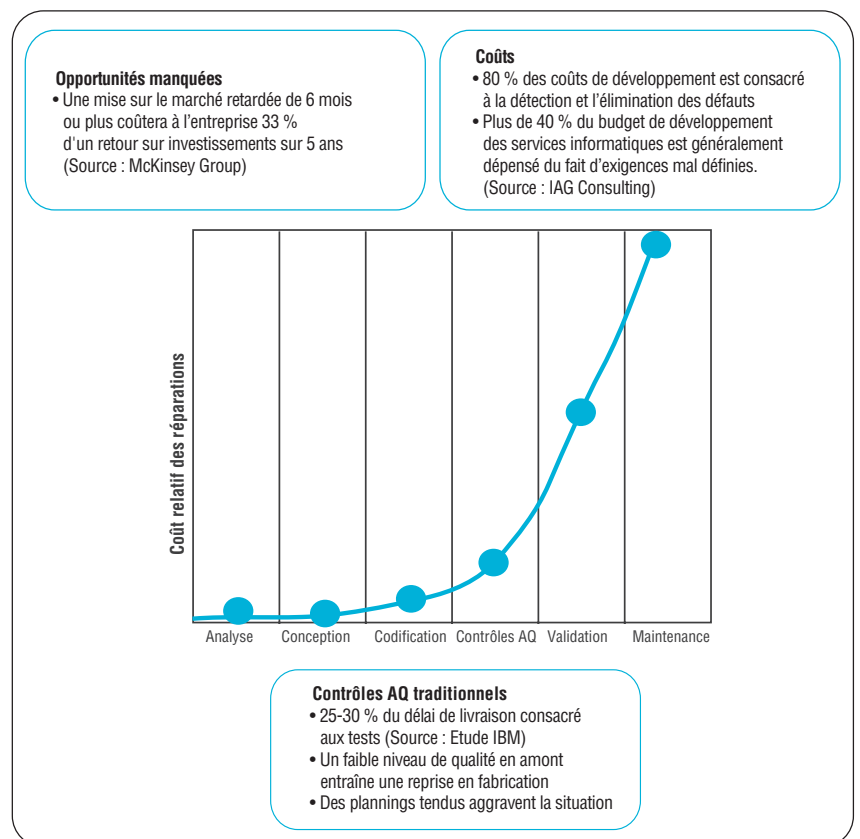


Figure 3 : Possibilités d'optimisation de la qualité et du délai de commercialisation sur tout le cycle de développement des applications.



---

Points clés

---

***La collaboration se heurte à trois principaux obstacles : dispersion des équipes sur différents secteurs géographiques, infrastructure incompatible et inadéquation des compétences***

La gestion de la qualité est devenue un sport d'équipe, mais la collaboration se heurte à de nombreux obstacles. Les *obstacles géographiques* bloquent la communication entre les équipes dispersés sur différents secteurs géographiques et sont à l'origine de lacunes, se traduisant par des reprises en fabrication. Les *obstacles liés à l'infrastructure* génèrent des problèmes d'intégration des procédures, des outils et des données qui ne permettent pas aux équipes de recevoir les informations en temps utile, et des plannings non respectés. Enfin, les *obstacles organisationnels* génèrent des problèmes au niveau de l'expertise du domaine, une gestion insuffisante du projet et un véritable casse-tête pour l'externalisation, comme la protection de la propriété intellectuelle.

Comment les sociétés disposant de la meilleure technologie améliorent-elles la qualité de leurs logiciels ? Une étude d'IBM Global Services<sup>3</sup> menée sur 846 projets chez un certain nombre de clients a fait apparaître un important différentiel de valeur en termes de méthodes collaboratives et automatisées. Des tests complets, des technologies intégrées sur tout le cycle de vie, des tests élémentaires aux normes industrielles et la réutilisation de scripts, ainsi que l'analyse pointue des défauts et des procédures de gestion de la qualité ont donné les taux d'amélioration suivants :

- *Amélioration de la qualité de 30 à 70 % ;*
- *Amélioration du cycle de 20 à 50 % ;*
- *Réduction des coûts de 15 à 60 %.*

Actif	Développement de procédures de tests reproductibles conformes aux normes industrielles			Analyse pointue des défauts	Développement de procédures de tests reproductibles répondant aux normes de l'industrie et pouvant s'appliquer à de futurs projets		Intégration des processus de bout en bout		Total
	Test élémentaires copiés	Scripts manuels copiés	Scripts manuels réutilisés	Empêcher la répétition des défauts	Données de base et migration de la documentation	Artéfacts de données de base	Bénéfice de la réutilisation des composants	Mises à jour dynamiques des équipements de tests	
Quantité	343	350	1 393	905	1 365	2 023	1 029	2 227	9 635
Heures économisées	167	175	696	1 755*	683	1 011	515	557	5 558
Valeur	\$16 690	\$17 514	\$69 633	\$175 452	\$68 254	\$101 125	\$51 459	\$55 673	
Total			\$103 387	\$175 452		\$169 379		\$107 132	\$555 799

\* Le nombre d'heures économisées suppose une moyenne de 4 heures pour détecter la reproduction des défauts. En réalité, la détection demande beaucoup plus de temps

Tableau 1 : Les valeurs économiques différenciées découlent d'une approche collaborative, automatisée, pilotée par des processus, de la gestion de la qualité.

Le tableau 1 récapitule un certain nombre de conclusions découlant de cette étude portant sur 846 projets et montre de quelle façon une économie d'un demi million de dollars a été réalisée pour chaque projet.

Dans quels domaines ces entreprises ont-elles réalisé des économies ?  
Dans leurs systèmes de gestion de la qualité.

- **Développement de procédures de tests reproductibles conformes aux normes industrielles.** Il paraît raisonnable que, dans n'importe quel secteur de l'industrie, les tests élémentaires et les scripts manuels soient devenus relativement similaires. Cela revient à dire que la reproduction de ces tests et de ces scripts communs pourrait dégager des économies substantielles. Dans cette étude, l'économie réalisée par projet a été en moyenne supérieure à 100 000 dollars US (16 690 + 17 514 + 69 633).
- **Analyse pointue des défauts.** Il n'est pas facile de détecter et de prévenir la répétition des défauts, mais il est très important de les détecter de façon précoce. A défaut, vous courrez le risque de voir différentes équipes travailler sur le même défaut sans le savoir, générant même souvent d'autres défauts. Leur détection automatisée non seulement améliore la qualité, mais réduit aussi le risque et les coûts. Dans cette étude, si l'on estime à quatre heures en moyenne le temps nécessaire pour détecter des défauts répétés, l'économie réalisée a été en moyenne de 175 452 dollars US par projet.

**Les entreprises peuvent gagner du temps et réduire leurs coûts en portant notamment leurs efforts sur la gestion de la qualité tout au long du cycle de développement et de livraison.**

- **Développer des procédures de tests reproductibles pouvant**

---

Points clés

---

**s'appliquer à de futurs projets.** *Le développement de nouvelles procédures d'essais pour chaque projet, de préférence à la réutilisation de systèmes existants, prend du temps et est inutile. La standardisation de procédures de tests reproductibles permet de gagner un temps considérable. L'économie moyenne réalisée par projet a été de 170 000 dollars US (68 254 + 101 125).*

- **Intégrer des procédures de bout en bout.** *L'intégration de la gestion des exigences grâce à des techniques de gestion de la qualité assure une parfaite traçabilité, dans les deux sens, de la phase de spécifications jusqu'aux tests élémentaires et aux résultats des tests. Elle permet aussi différentes études d'impact, notamment lorsque les exigences sont modifiées tout au long du cycle de développement. Elle est essentielle pour tirer parti des avantages concurrentiels dégagés d'une amélioration de la qualité et d'une réduction des coûts. Dans cette étude, l'économie moyenne réalisée par projet a été de 100 000 dollars US (51 459 + 55 673).*

En résumé, une méthode collaborative, basée sur des procédures et automatisée de la gestion de la qualité a non seulement permis d'appliquer des stratégies visant à limiter le risque et améliorer la qualité, mais d'économiser aussi, en moyenne, plus de 500 000 dollars US par projet !

### Concrétiser les promesses

Pour concrétiser les promesses d'innovation et de croissance – par exemple, la réactivité de l'entreprise à partir d'une architecture orientée service (service-oriented architecture - SOA) et le développement d'un logiciel basé sur des composants dans l'ensemble des équipes réparties sur différents secteurs géographiques - les organisations doivent fixer une stratégie de gestion de la qualité unifiée et réactive. Les équipes chargées de la conception et de la livraison de logiciels et de systèmes informatiques peuvent y contribuer par l'examen de trois secteurs essentiels.

**Une solide stratégie de gestion de la qualité facilite votre positionnement en termes d'innovation et de croissance.**

---

Points clés

---

***Les entreprises doivent définir une solide stratégie de gestion de la qualité qui englobe la collaboration, l'automatisation et la gestion efficace des informations.***

Collaboration

Les équipes doivent rester en phase avec des procédures et un système de gestion des activités simplifiés et dynamiques. Les exigences de l'entreprise, en constante évolution, doivent se répercuter sur la gestion de la qualité afin de s'assurer que les tests élémentaires sont mis à jour et que les développeurs saisissent les toutes dernières exigences. Les responsables AQ et les chefs de projet doivent faire en sorte que leurs équipes travaillent aux tâches hautement prioritaires. Les tests doivent être planifiés en continu et doivent intégrer une approche orientée vers les objectifs, avec des critères données d'entrée/résultats et des configurations d'environnement prioritaires.

Automatisation

Le développement de logiciels a vu l'automatisation d'activités de spécialistes, comme les tests fonctionnels et de performance. Les ingénieurs chargés aujourd'hui de l'élaboration de logiciels utilisent des scripts et un ensemble d'outils performants. Mais il faut faire encore plus pour automatiser le processus et les différentes étapes afin d'améliorer les performances de l'organisation, réaliser des économies et commercialiser plus rapidement le produit (par ex. la reconduction des tests élémentaires et des scripts-tests sur toutes les gammes de produits, le dimensionnement/l'analyse automatisé du laboratoire d'essais, et la conversion automatique des modèles pratiques en tests élémentaires).

Gestion efficace de la communication des informations

Les résultats de tous les types d'essais – incluant les tests de l'unité, les tests fonctionnels, d'intégration et d'extensibilité – doivent être immédiatement disponibles pour leur communication et l'analyse des tendances. L'intégration des informations techniques du projet au système d'analyse permet aux décideurs de niveau supérieur de peser davantage sur l'affectation et l'utilisation des ressources pour la mise en phase de l'informatique et du commercial. Les systèmes de gestion des tests et de planification doivent être étroitement intégrés dans l'analyse et la définition des exigences.

---

**Points clés**

---

Les équipes chargées des tests et de la planification des tests préliminaires doivent accéder en temps réel aux spécifications commerciales et fonctionnelles du projet, aux applications et aux contrats de services (SLA). L'intégration des informations sur la situation du projet aux méthodes d'analyse de l'entreprise permet aux analystes de comparer les modifications apportées au projet aux objectifs et contraintes de l'entreprise. L'examen de ces données à chaque phase permet aux équipes projet d'effectuer les ajustements et de peser les risques à traiter en priorité. Le résultat d'ensemble permet un meilleur contrôle du projet et, en cas de problèmes inhérents à celui-ci, une évaluation plus précise des risques pour l'entreprise. La bonne information au bon moment, ciblée et filtrée pour analyser les décisions à prendre, participe à la livraison d'un logiciel abouti et performant.

***Les directives, les meilleures méthodes et un ensemble d'outils intégré sont des éléments essentiels pour les solutions actuelles de gestion de la qualité.***

**Les solutions de gestion de la qualité**

Les solutions actuelles de gestion de la qualité fournissent des directives sur le processus de livraison de logiciels, les meilleures méthodes et des outils intégrés pour les équipes permettant aux entreprises d'améliorer leur productivité, les métriques de qualité et la valeur-temps pour les projets vitaux de l'entreprise.

Comme nous l'avons montré plus haut, la collaboration est essentielle. Elle peut se faire par l'échange de courriers électroniques ou de partage dense de fichiers, mais ces méthodes sont très lourdes et n'intègrent même pas la structure centrale de la fourniture du logiciel. Plusieurs entreprises ont développé des outils de gestion du cycle de vie des applications (application lifecycle management - ALM) qui assurent la transparence de la progression des équipes et régissent leur méthode de travail.

---

Points clés

---

**Les outils de gestion de la qualité intègrent des systèmes de collaboration, d'automatisation et de gestion de la communication de l'information pour aider les entreprises à livrer en permanence des produits de qualité supérieure.**

La plate-forme IBM Jazz™ intègre les outils de gestion ALM assurant la prise en charge de la gestion des exigences, des activités de développement et de la gestion de la qualité tout en aidant les équipes à surmonter les obstacles à la collaboration, qu'ils soient géographiques, liés à l'infrastructure ou à l'organisation. Les plates-formes de correspondance du Web permettent également le travail en collaboration afin de livrer un logiciel de qualité pérenne, ce qui constitue un atout stratégique pour l'entreprise. Les nouvelles solutions proposées permettent aux utilisateurs de :

- **Collaborer** au sein de toute l'entreprise, avec les équipes développement et les équipes chargées des tests et, grâce au déroulement dynamique des processus et des opérations basées sur l'activité pour la planification et l'exécution des tests.
- **Automatiser** les procédures nécessitant des ressources humaines importantes tout au long du cycle de vie du produit et appréhender les problèmes de qualité en amont, permettant une commercialisation plus rapide du produit, une réduction des coûts et une limitation des risques pour l'entreprise.
- **Communiquer** sur les métriques prioritaires spécifiques aux individus et aux équipes, facilitant une meilleure visibilité et permettant aux décideurs d'agir en toute confiance.
- **Mieux prévoir**, en rattachant les modèles de déploiement qui ont fait la preuve de leur réussite à des indicateurs de performances clé au niveau opérationnel.

Les entreprises qui tirent parti des nouveaux outils conçus pour la définition/gestion des exigences seront capables de :

- **Mettre en phase** les projets de développement avec les objectifs de l'entreprise afin de limiter les risques d'échecs qui représentent pour les entreprises des pertes se chiffrant annuellement en milliards de dollars.
- **Mieux formuler** les exigences et la validation entre les responsables commerciaux et techniques utilisant des techniques de visualisation et de collaboration qui ont fait leur preuve (schémas de processus d'entreprise, schémas d'interfaces utilisateur, storyboards, cas pratiques).
- **Gérer** plus efficacement les modifications apportées aux spécifications et à la portée d'un projet (analyse d'impact, par exemple).
- **Améliorer** le délai de commercialisation et de retour sur investissements en réduisant les reprises en fabrication du produit du fait d'exigences mal définies et manquantes.

Points clés

**Les outils ALM facilitent la création du produit et garantissent sa qualité par la modélisation de l'architecture, le développement, les tests fonctionnels et opérationnels du système informatique, les tests automatisés de validation de la fabrication, les tests manuels, les tests en charge et d'extensibilité et les solutions d'aide aux applications de production.**

Partie intégrante de la stratégie de la qualité ALM dans son ensemble, les outils de production de logiciels, comme ceux décrits dans le tableau 2, permettent aussi de garantir la qualité et la performance d'un logiciel essentiel pour l'entreprise.

Outils	Description
Modélisation de l'architecture	<ul style="list-style-type: none"> <li>• Valide les règles définies par l'utilisateur, qui représentent les contraintes d'architecture.</li> <li>• Détecte automatiquement les modèles de conception et les structures essentielles orientées-objet.</li> <li>• Détecte les défauts structurels de conception (croisements, structures papillon, nœuds) qui dégradent les performances.</li> <li>• Refactorise automatiquement les croisements grâce à des solutions rapides.</li> </ul>
Développement	<ul style="list-style-type: none"> <li>• Aide les développeurs à détecter une mémoire corrompue, une fuite, à profiler les performances et assurer la couverture des codes orientées-objet.</li> </ul>
Tests automatisés de validation du build du logiciel	<ul style="list-style-type: none"> <li>• Evite le développement de défauts dans le laboratoire de tests ou l'environnement d'essai du logiciel.</li> <li>• Permet aux développeurs de tirer parti des cycles d'heures creuses pour tester la stabilité et la fonctionnalité d'une application.</li> </ul>
Tests manuels	<ul style="list-style-type: none"> <li>• Encourage les meilleures méthodes comme la modularité et la possibilité de reproduire les tests pour faciliter le passage des tests manuels aux tests automatisés.</li> </ul>
Tests fonctionnels et opérationnels du système	<ul style="list-style-type: none"> <li>• Permet aux équipes de réaliser des tests manuels ou automatisés pour la détection d'une régression et de défauts fonctionnels.</li> <li>• Réduit les cycles des tests automatisés afin d'améliorer la qualité grâce à un spectre élargi et approfondi de tests.</li> <li>• Encourage l'exécution de tests plus précis, plus fiables et reproductibles.</li> <li>• Etend les tests fonctionnels à l'environnement d'applications Oracle, notamment OracleForms™.</li> </ul>
Tests en charge et d'évolutivité	<ul style="list-style-type: none"> <li>• Détermine les seuils de charge et d'évolutivité de technologies et d'applications comme Java™, Platform, Entreprise Edition (Java EE) ; Internet (notamment les portails d'accès) ; SOA ; Siebel ; Oracle et SAP.</li> <li>• Permet l'évolution et le fonctionnement d'une application logicielle pour sa conformité aux SLAs et aux attentes de l'utilisateur.</li> <li>• Permet aux responsables des essais de détecter avec précision l'origine des points de blocage, et de naviguer ainsi jusqu'au code source sans passer inutilement par de nombreux niveaux de code.</li> <li>• Assure un meilleur retour sur investissements du matériel par l'exécution de tests de planification des capacités de pré-déploiement qui calibrent les ressources des serveurs à prévoir pour assurer la performance et la capacité voulues.</li> <li>• Etend les tests de performance permettant de tester les solutions Oracle NCA telles que Oracle Business Suite®</li> </ul>
Solutions d'aide aux applications de production	<ul style="list-style-type: none"> <li>• Assure la collaboration entre les équipes de recherche opérationnelle et développement pour l'aide aux applications en boucle fermée, l'élimination des problèmes et les réparations.</li> <li>• Saisit le journal des transactions et l'historique des données, ainsi que les données complémentaires sur les ressources du système pour une définition plus fine des problèmes, réduisant ainsi les temps de réaction aux diagnostics d'aide aux applications.</li> <li>• Accélère la restauration des niveaux de service pour l'entreprise.</li> </ul>

Tableau 2 : Les outils ALM contribuent à assurer la qualité tout au long du cycle de vie du produit



## Synthèse

Nous avons assisté au cours de ces dix dernières années à une confrontation entre l'impératif d'évolution de l'entreprise par l'innovation, la nécessité d'améliorer la qualité des logiciels pour gérer le risque commercial, et la nécessité impérative de réduire les coûts pour sa survie économique.

L'évolution de l'industrie vers le travail d'équipe collaboratif entre équipes réparties sur différentes zones géographiques, associée à des processus et un système de communication de l'information automatisés pour assurer une gestion efficace, modifie actuellement les aspects de la livraison de logiciels de qualité.

La gestion de la qualité, ce n'est plus simplement des services assurance qualité disposant d'outils de tests automatisés facilitant la livraison traditionnelle des logiciels, ni des procédures de contrôle de la qualité qui ralentissent la livraison ou simplement détecter et éliminer les défauts. La gestion de la qualité, c'est intégrer la qualité dans un cycle de développement itératif et un programme proactif, en boucle fermée, d'aptitude à l'emploi d'un logiciel, complété par un ensemble d'outils intégrés, de données et un historique de métriques. Une gestion efficace de la qualité permet de mieux gérer et d'assurer plus en douceur la livraison, et de s'appuyer sur des équipes de gestion opérationnelle. Pour l'essentiel, elle brise le triangle de fer présenté plus haut en permettant aux chefs d'entreprise d'optimiser le champ d'application, de réduire les coûts et les délais tout en améliorant la qualité.

L'assistance à la gestion de la qualité tout au long du cycle de vie et la détection rapide des défauts permettent de réduire les coûts et d'améliorer la crédibilité. La communication constante au travers d'un ensemble d'exigences communes et bien comprises – et une très forte réactivité aux changements apportés à ces exigences – peut injecter dès le départ la qualité dans le développement du logiciel. De plus, une gestion dans les règles des processus de création et d'exécution des tests pour corriger le tir et affecter les ressources permettent aux entreprises de gagner en flexibilité, les aident à traiter les problèmes de mise en conformité avec la réglementation, à réagir plus vite à l'évolution des conditions du marché et, enfin, de guider la croissance de l'entreprise. Dans cette économie impitoyable, la réussite ou l'échec dépend souvent d'une seule chose : du produit de qualité optimale.

Pour en savoir plus...

Sur les solutions IBM de gestion de la qualité, contactez votre représentant IBM ou visitez le site :

[ibm.com/software/rational/offerings/quality](http://ibm.com/software/rational/offerings/quality)

Nous tenons à remercier tout spécialement M. Ron French, d'IBM Global Business Services pour ses conclusions d'un grand intérêt et pour le travail de son équipe.

© Copyright IBM Corporation 2009

Compagnie IBM France  
17 avenue de l'Europe  
92275 Bois Colombes Cedex

Imprimé en France  
Mars 2009  
Tous droits réservés

IBM, le logo IBM, [ibm.com](http://ibm.com) et Rational sont des marques commerciales ou des marques déposées d'International Business Machines Corporation, aux Etats-Unis, dans d'autres pays, ou aussi bien aux Etats-Unis que dans d'autres pays. Si ces termes et d'autres termes IBM de marques déposées sont signalés à leur première apparition dans la présente publication par un symbole de marque déposée (® ou ™), ces symboles indiquent des marques déposées ou des marques commerciales relevant du droit coutumier des Etats-Unis, qui sont la propriété d'IBM à la date de publication du présent document. Il est possible que ces marques commerciales soient aussi des marques déposées ou relevant du droit coutumier dans d'autres pays. Une liste actuelle des marques commerciales IBM est disponible sur Internet à la page « Informations sur les droits de reproduction et les marques commerciales » sur le site [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Java et toutes les marques commerciales tournant sur Java sont des marques commerciales de Sun Microsystems, Inc. aux Etats-Unis et/ou dans d'autres pays.

D'autres noms de sociétés, de produits ou de services peuvent être des marques commerciales ou des marques de service d'autres sociétés.

Les renvois dans la présente publication à des produits ou des services IBM n'impliquent pas l'intention d'IBM de les rendre accessibles à tous les pays où opère IBM.

Les informations contenues dans cette publication sont données uniquement à titre indicatif. En dépit de tous les efforts mis en œuvre pour vérifier l'exhaustivité et l'exactitude des informations contenues dans cette publication, elles sont fournies « telles que d'origine » sans garantie d'une quelconque nature, expresse ou implicite. Ces informations sont également basées sur les programmes et la stratégie produits actuels d'IBM, susceptibles d'être modifiés par IBM sans préavis. IBM ne sera pas tenu responsable du préjudice causé par l'utilisation ou lié à l'utilisation de ce document ou de tout autre documentation. Aucun élément contenu dans la présente publication n'est destiné, ni n'a pour effet de créer des garanties ou des représentations de la part d'IBM (ou de ses fournisseurs ou donneurs de licence), ou d'altérer les termes et conditions du contrat de licence en vigueur régissant l'utilisation des logiciels IBM.

- 1 <http://www.devtopics.com/20-famous-software-disasters>
- 2 Walker Royce, *Software Project Management : A Unified Framework*, Addison-Wesley Professional, Indianapolis, 1998.
- 3 IBM Global Services, SEANT (Systems Engineering Architecture & Test) ReUse Program, 2007.

RAW14125-FRFR-00