

## **La Réussite des Projets Logiciels Commence par la Gestion des Exigences**

Par Serge Charbonneau, Consultant Principal, Xelation Software Corporation (www.xelation.com).

Publié dans Direction Informatique le 18 septembre 2003

### **Introduction**

J'ai une question à vous poser: Pourriez-vous vous imaginer demander à un entrepreneur en construction domiciliaire de vous bâtir une maison sans lui spécifier quoi que ce soit? Ou de vous acheter une voiture sans en connaître les caractéristiques, ou même le prix? La réponse à ces deux questions est simple: Non.

J'ai une autre question à vous poser: Pourriez-vous vous imaginer demander à un entrepreneur de développement en logiciel de vous bâtir une application de gestion des payes sans lui spécifier quoi que ce soit? Ou de vous acheter une application de gestion des relations clients sans en connaître les caractéristiques, ou même le prix pour la déployer? La réponse à ces deux questions devrait être aussi simple que la réponse aux deux questions précédentes: Non.

Pourtant, combien de projets logiciels ne sont jamais complétés parce que les exigences sont mal gérées ou ne sont simplement jamais capturées de façon explicite? Et combien d'applications sont achetées mais ne sont jamais déployées, ou sont déployées mais ne sont jamais utilisées parce qu'elles ne rencontrent pas les attentes de leurs utilisateurs? La réponse à ces questions est la suivante: Beaucoup trop.

La meilleure façon d'éviter ce genre de problèmes dans le développement et l'acquisition de logiciels est d'implanter un bon processus de gestion des exigences. Mais quelles sont les caractéristiques d'un bon processus de gestion des exigences? Cet article tente de répondre à cette question.

### **Caractéristiques d'un Bon Processus de Gestion des Exigences**

Un bon processus de gestion des exigences doit avoir les caractéristiques suivantes:

- Des exigences exprimées de façon clair et explicite;
- Des attributs associés à chacune des exigences;
- Une traçabilité explicitement identifiée entre les exigences; et
- Un bon processus de gestion des changements.

Examinons chacune de ces caractéristiques en détail.

## **Des Exigences Exprimées de Façon Clair et Explicite**

Une exigence (requirement en anglais), souvent appelée requis, est définie comme étant une condition ou une fonctionnalité à laquelle un système doit se conformer.

Les exigences pour le développement ou l'acquisition d'une application logicielle doivent définir le 'Quoi' de l'application par opposition au 'Comment'. Dans le cas du développement d'une application, il est nécessaire d'exprimer clairement ce que l'application doit faire avant de spécifier comment elle va le faire. Le 'Quoi' est capturé durant la définition des exigences tandis que le 'Comment' est dérivé du 'Quoi' durant la conception (design). La conception est ensuite implantée en utilisant un langage de programmation (C, C++, C# ou Java). Dans le cas de l'acquisition d'une application, il est nécessaire d'exprimer clairement ce que l'application doit faire avant d'entreprendre le processus de sélection.

Chacune des exigences pour le développement ou l'acquisition d'une application logicielle doit avoir les qualités suivantes:

- Vérifiable → Est testable.
- Modifiable → Est modifiable.
- Traçable → Est dérivée d'une ou plusieurs exigences, et/ou une ou plusieurs exigences en sont dérivées.
- Compréhensible → Est compréhensible par tout le monde autant des gens techniques que non techniques.
- Correcte → N'est pas erronée.
- Complet → Décrit à elle seule, ou avec d'autres exigences, un ensemble complet sans éléments manquants.
- Consistant → Ne contredit pas d'autres exigences.
- Sans Ambiguïté → Veut dire la même chose pour tout le monde.

## **Des Attributs Associés à Chacune des Exigences**

Les attributs associés aux exigences permettent une gestion plus efficace d'un projet de développement logiciel. Voici certains de ces attributs:

- Etat: Identifie où l'exigence se trouve dans son cycle de vie. Une exigence peut être proposée, approuvée, incorporée ou validée.
- Priorité: Associe un niveau d'importance avec une exigence. Cet attribut peut être quantitatif (valeur de 1 à 10; 1 étant la plus haute priorité et 10 étant la plus basse priorité) ou qualitatif (élevé, moyen ou faible).
- Coût ou Effort: Identifie le coût (en dollars) ou l'effort (en jours personne) requis pour réaliser l'exigence.
- Ressource: Identifie la ressource responsable pour la réalisation de l'exigence.
- Itération Cible: Identifie l'itération durant laquelle l'exigence sera réalisée.

D'autres attributs peuvent être associés aux exigences.

Les attributs permettent aux gestionnaires de projets de planifier l'effort requis pour l'implantation d'une application logicielle en terme de temps, de coût et d'envergure. Si une application est développée de façon itérative, les exigences de priorité plus élevées devraient être implantées durant les premières itérations tandis que les exigences de plus faible priorité devraient être implantées durant les dernières itérations. De plus, le coût ou l'effort requis pour la réalisation d'un ensemble d'exigences peut être calculé à partir du coût ou de l'effort associé à chacune des exigences de l'ensemble.

## **De la Traçabilité Explicitement Identifiée entre les Exigences**

La traçabilité permet d'identifier de façon explicite la source d'une exigence ainsi que sa destination. Une exigence logicielle peut être dérivée d'une exigence de plus haut niveau et/ou peut être utilisée pour dériver une ou plusieurs exigences de plus bas niveau, un ou plusieurs éléments de conception (classes, attributs, opérations, sous-systèmes, etc.), ou un ou plusieurs cas de test.

La traçabilité présente deux avantages majeurs:

1. Elle permet d'assurer que toutes les exigences d'un certain type ou d'un certain niveau sont réalisées par des exigences d'un autre type ou de niveau inférieur; et
2. Elle permet d'effectuer de l'analyse d'impact plus efficacement et de façon plus objective d'une perspective de gestion de projet.

Lorsqu'une stratégie de traçabilité existe entre des exigences logicielles et des cas de test, il devient plus facile de s'assurer que des cas de test existent pour valider toutes les exigences d'une application et qu'aucune exigence reste non testée. De la même façon, si une stratégie de traçabilité existe entre différents niveaux d'exigences et autres éléments de conception, il devient aussi plus facile d'évaluer l'impact qu'aura un changement potentiel d'une exigence sur tous les éléments qui en sont dérivés.

## **Un Bon Processus de Gestion des Changements**

Pourquoi est-il nécessaire d'avoir un bon processus de gestion des changements dans un projet de développement logiciel? La réponse est simple: Parce qu'il va y avoir des changements. Développer du logiciel est une entreprise très complexe et risquée dans un champ d'activités très abstrait. C'est pourquoi il est pratiquement impossible d'implanter une application parfaitement en une seule itération. Ce qui implique qu'il y a des demandes de changement sur tous projets de développement qu'elles soient capturées de façon explicite ou non.

Un bon processus de gestion des changements implique que chaque demande de changement doit être capturée de façon formelle et explicite, analysée en

# ***Xelation***

■■■ Software Corporation ■■■

terme d'impact sur l'application existante et en terme d'effort d'implantation, et présentée à un individu ou un groupe d'individus qui décidera de son sort (approuvée, rejetée ou remise à plus tard). Lorsqu'elle est approuvée, la demande de changement doit être reflétée dans les exigences qu'elle impacte, et par la suite dans les éléments de conception et d'implantation que les changements sur les exigences impactent.

## **Conclusion**

Le génie logiciel est une discipline qui repose beaucoup sur le 'gros bon sens'. Malheureusement, plus souvent qu'autrement, le 'gros bon sens' n'est pas souvent appliqué. La gestion des exigences est un des aspects les moins techniques du génie logiciel qui repose presque entièrement sur le 'gros bon sens'. En effet, la gestion des exigences recommande simplement de 'spécifier ce que l'application doit faire avant de spécifier comment elle va le faire.' C'est simple, non? Alors, la prochaine fois que vous allez changer de voiture, pensez-y.

## **A Propos de Xelation Software Corporation**

Xelation Software Corporation est une compagnie fondée en 2003 qui offre à ses clients des solutions dans les disciplines suivantes du génie logiciel:

- Gestion des Exigences;
- Architecture;
- Analyse et Design Orienté-Objet;
- Développement;
- Contrôle de la Qualité;
- Gestion de Projets;
- Gestion de la Configuration et du Changement; et
- Ingénierie des Processus.

La mission de Xelation Software Corporation est d'accélérer le succès de ses clients dans l'amélioration de leurs pratiques de génie logiciel. Le site web de Xelation Software Corporation est [www.xelation.com](http://www.xelation.com).

## **A Propos de Serge Charbonneau**

Serge Charbonneau est consultant principal chez Xelation Software Corporation. Mr. Charbonneau a plus de 12 ans d'expérience dans le domaine du génie logiciel. Il a joué plusieurs rôles dans différentes organisations logicielles incluant analyste, concepteur, programmeur, chef d'équipe, représentant technique et consultant. Il possède une solide expertise en gestion de projets logiciels, gestion des exigences, analyse et conception orienté-objet, et ingénierie des processus.