

## Chapitre 1

# Bienvenue dans le monde de la construction logicielle

### Sommaire :

- 1.1 La construction logicielle, qu'est-ce que c'est ? : page 3
- 1.2 Pourquoi la construction logicielle est-elle importante ? : page 7
- 1.3 Comment il faut lire ce livre : page 8

### Voir aussi :

- À qui s'adresse cet ouvrage : Préface
- Les objectifs de cet ouvrage : Préface
- Pourquoi j'ai écrit ce livre : Préface

Vous connaissez le sens habituel du mot « construction », lorsqu'on l'utilise pour autre chose que pour le développement de programmes : il s'agit du travail produit par les divers corps de métiers du bâtiment lors de l'édification d'une maison, d'une école ou même d'un gratte-ciel. Lorsque vous étiez enfants, vous construisiez des choses avec du papier. Dans le langage courant, le terme « construction » est lié à l'idée de bâtiment. Le processus de construction tient compte de divers aspects liés au planning, à la conception et à la vérification de votre travail mais « construction » évoque la partie manuelle de la création.

## 1.1 La construction logicielle, qu'est-ce que c'est ?

Le développement des logiciels d'ordinateurs peut être un processus complexe et, au cours des dernières 25 années, les spécialistes l'ont divisé en de nombreuses activités distinctes :

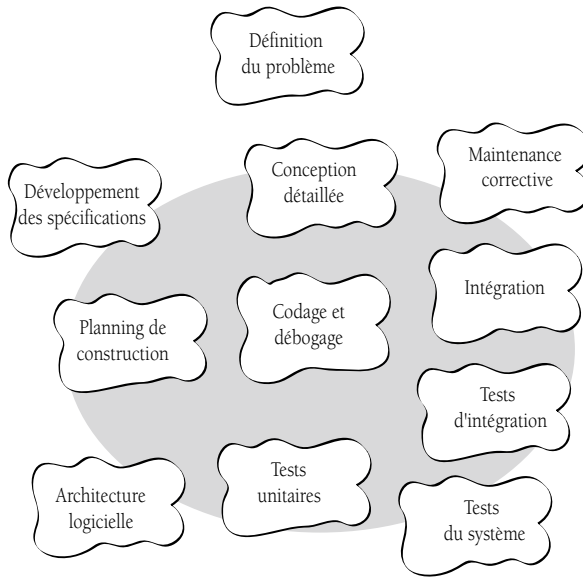
- La définition du problème
- Le développement des spécifications
- La planification de la construction
- L'architecture logicielle, ou conception de haut niveau

- La conception détaillée
- Le codage et le débogage
- Les tests unitaires
- Les tests d'intégration
- Les tests du système
- La maintenance corrective

Si vous avez participé à des projets informels, vous pensez peut-être que cette liste laisse prévoir des quantités de tracas administratifs. Si vous avez travaillé sur des projets trop formels, vous savez que c'est le cas. Il n'est pas facile de trouver l'équilibre entre trop et trop peu de formalisme : nous en reparlerons dans la suite de ce livre.

Si vous avez appris tout seul à programmer ou si vous avez travaillé essentiellement sur des projets informels, vous n'avez peut-être pas distingué les diverses activités qui interviennent dans la création d'un produit logiciel. Vous avez peut-être groupé mentalement toutes ces activités en une seule : la « programmation ». Si vous avez participé à des projets informels, l'activité principale à laquelle vous pensez à propos de création logicielle est probablement celle que les spécialistes ont appelée « construction ».

La notion intuitive de « construction » est assez précise, mais elle manque un peu de profondeur. En la replaçant dans son contexte, parmi les autres activités, on peut définir plus précisément les tâches réelles qu'elle recouvre et mettre en évidence les activités importantes qui n'en font pas partie. La figure 1-1 montre la place qu'occupe la construction dans l'ensemble des activités mises en jeu par le développement logiciel.



**Figure 1-1** Les activités de construction sont celles qui se trouvent à l'intérieur du cercle gris. La construction est orientée principalement vers le codage et le débogage mais elle comprend également la conception détaillée, les tests unitaires, les tests d'intégration et d'autres activités.

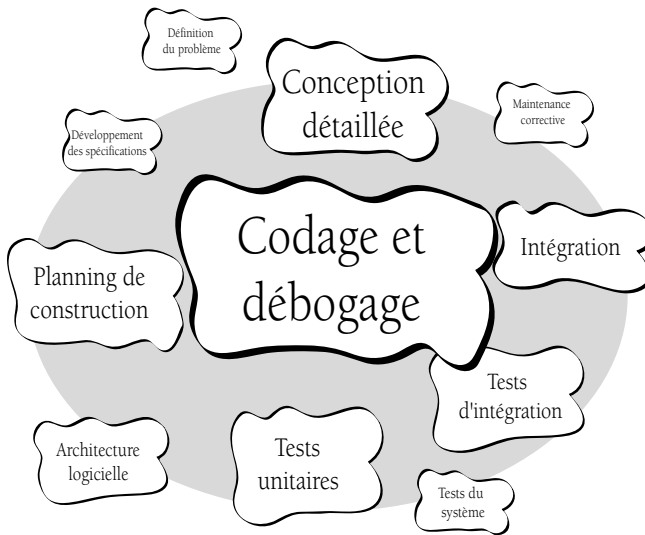


**POINT CLÉ**

Comme le montre la figure, les activités principales de la construction sont le codage et le débogage, mais elle comprend également, en tout ou en partie, la conception détaillée, la planification de la construction, les tests unitaires, l'intégration, les tests d'intégration et d'autres activités. Si ce livre devait traiter de tous les aspects du développement logiciel, vous y trouveriez les descriptions de toutes les activités nécessaires au processus de développement. Mais il est consacré aux techniques de construction : par principe, il lui accordera une importance tout à fait exagérée et ne parlera, à quelques rares exceptions près, que des sujets qui s'y rattachent.

La construction est quelquefois appelée « codage » ou « programmation ». Le terme de codage n'est pas le mieux approprié parce qu'il suggère une traduction en langage informatique d'une conception préexistante ; or la construction n'est absolument pas un processus mécanique : elle suppose de la créativité et du jugement. Dans la suite de ce livre, j'utiliserai indifféremment les termes programmation ou construction.

La figure 1-2 oppose la perspective globale adoptée par ce livre à la vue en deux dimensions de la figure 1-1.



**Figure 1-2** Ce livre traite du codage et du débogage, de la conception détaillée, de la planification de la construction, des tests unitaires, de l'intégration, des tests d'intégration et des autres activités à peu près dans les proportions indiquées.

Les figures 1-1 et 1-2 sont des vues générales des activités de construction ; mais qu'en est-il des détails ? Voici quelques-unes des tâches spécifiques nécessaires à la construction :

- Vérifier que le travail préparatoire a été effectué de telle sorte que la construction puisse se dérouler correctement.
- Déterminer quels seront les procédures de tests de votre code.
- Concevoir et écrire des classes et des sous-programmes.
- Créer et nommer des variables et des constantes nommées.
- Choisir des structures de contrôle et organiser des blocs d'instructions.
- Appliquer les tests unitaires et les tests d'intégration à votre code et, bien sûr, le déboguer.
- Examiner la conception de détail et le code réalisés par les autres membres de votre équipe ; leur faire examiner les vôtres.
- Améliorer la finition du code par une mise en forme et des commentaires soignés.
- Intégrer les composants logiciels créés séparément.
- Optimiser le code afin de le rendre plus rapide et moins gourmand en ressources.

La suite des titres des chapitres, dans la table des matières, vous fournira une liste encore plus complète des activités de construction.

La construction logicielle, semble-t-il, recoupe un bon nombre d'activités : il aurait peut-être été plus rapide de citer celles qui n'en font pas partie ! Ce n'est pas une plaisanterie, il y en a ! Entre autres activités importantes, la gestion du projet, le développement des spécifications, l'architecture du logiciel, la conception de l'interface utilisateur, les tests système et la maintenance ne font pas partie de la construction. Chacune de ces activités joue son rôle dans le succès final d'un projet au même titre que la construction, au moins dans le cas des projets qui emploient plus d'une ou deux personnes et dont la durée dépasse quelques semaines. Pour chacune de ces activités, vous pouvez trouver des ouvrages intéressants, mais aucune d'elles ne sera traitée ici.

## 1.2 Pourquoi la construction logicielle est-elle importante ?

Si vous lisez ce livre, c'est probablement parce que vous attachez une certaine importance à l'amélioration de la qualité du logiciel et de la productivité des développeurs. Aujourd'hui, dans tous les domaines, la plupart des projets intéressants utilisent largement l'informatique. Internet, les effets spéciaux au cinéma, les systèmes d'assistance vitaux en médecine, la conquête spatiale, l'avionique, les analyses financières en temps réel et la recherche scientifique en sont des exemples connus. Ces projets, comme d'autres plus conventionnels, peuvent tirer parti de la mise en place de méthodes améliorées : pour la plupart, leurs besoins fondamentaux sont identiques.

Mais si vous reconnaissez l'importance générale de l'amélioration des méthodes de développement, la question devient : en quoi la construction est-elle un point capital ?

Il y a plusieurs réponses :

***La construction est une partie importante du développement logiciel*** Selon la taille du projet, la construction représente en général de 30 à 80 % du temps total consacré au projet. Naturellement, une partie aussi importante ne peut qu'affecter le succès général du projet.

***La construction est l'activité centrale du développement logiciel*** Les spécifications et l'architecture sont définies avant la construction afin de favoriser l'efficacité de celle-ci. Les tests système (au sens strict de tests indépendants) sont effectués, quant à eux, après la construction afin de vérifier qu'elle a été réalisée correctement. De ce point de vue, la construction est le centre du processus de développement logiciel.

***Si vous faites de la construction une priorité, la productivité individuelle des programmeurs peut être nettement améliorée*** Une étude classique de Sackmann, Erikson et Grant a montré que la productivité individuelle des programmeurs pouvait varier d'un facteur de 10 à 20 au cours de la construction (1968). Depuis, ces résul-

tats ont été confirmés par de nombreuses autres études (Curtis 1981, Mills 1983, Curtis et al. 1986, Card 1987, Valette et McGarry 1989, DeMarco et Lister 1999, Boehm et al. 2000). Ce livre aidera les programmeurs à adopter les techniques employées par les meilleurs.

*Le produit de la construction, le code source, est souvent la seule description exacte du logiciel* Dans beaucoup de projets, la seule documentation accessible aux programmeurs est le code lui-même. Les spécifications et les documents de conception peuvent être périmés, mais le code, par définition, est toujours à jour. Par conséquent, il est impératif que le code source soit de la meilleure qualité possible. C'est l'application constante des techniques d'amélioration du code source qui fait la différence entre une bidouille à la Dubout et un programme correct, détaillé et donc instructif. Ces techniques ont leur meilleure efficacité pendant la construction.



**POINT CLÉ**

*La construction est la seule activité dont la réalisation est obligatoire* Un projet logiciel idéal commence par le développement des spécifications et la conception architecturale, avant de passer à la construction. Il doit se terminer par des tests système exhaustifs et contrôlés statistiquement, après la fin de celle-ci. Cependant, les projets réels, qui ne peuvent jamais être parfaits, oublient souvent les spécifications et la conception pour se limiter à la construction. Les tests peuvent également être négligés en raison du trop grand nombre des erreurs à corriger et du manque de temps. Mais quels que soient la précipitation et le manque d'organisation dont souffre un projet, il est impossible d'en omettre la construction : c'est seulement là que la réalisation devient effective. L'amélioration de la construction est donc le moyen de rendre efficace n'importe quel effort de développement logiciel, quelles que soient ses conditions de réalisation.

## 1.3 Comment il faut lire ce livre

Ce livre peut être lu du début jusqu'à la fin aussi bien que sujet par sujet. Si vous préférez le lire entièrement, passez au chapitre 2. Si vous souhaitez étudier des techniques de programmation spécifiques, vous pouvez commencer, par exemple, par le chapitre 6 qui traite des classes. Si vous hésitez sur le mode de lecture qui vous convient, commencez par la section 3.2.

### Points clés

- La construction logicielle est l'activité centrale du développement logiciel ; c'est la seule activité dont l'existence est réellement indispensable, quel que soit le projet.
- Les principales activités de la construction sont la conception détaillée, le codage, le débogage, l'intégration et les tests effectués par le développeur (tests unitaires et tests d'intégration).