AAA Programming

Pour les développeurs .Net

Table des matières

Pourquoi ce livre

Pendant plusieurs années j'ai été éditeur de logiciels pour les industries graphiques et plus particulièrement pour les éditeurs de magazines et les imprimeurs. J'ai développé en .Net une solution logicielle pour automatiser l'impression depuis l'éditeur de magazine jusqu'à l'imprimeur. Cette solution logicielle ne devait en aucun cas être à l'origine de l'arrêt de l'outil industriel de l'éditeur ou de l'imprimeur. Sachant que l'outil industriel d'un imprimeur se chiffre en moyenne à quelques dizaines de millions d'euros, la solution devait fonctionner 7j/7, 24h/24 sans aucun support compte tenu du fait que l'impression d'un magazine ou d'un quotidien se fait le plus souvent en dehors des heures ouvrées traditionnelles.

Pour satisfaire ce très haut niveau d'exigence, j'ai dû changer ma façon de coder.

Au fil des années j'ai mis au point un ensemble de techniques de programmation permettant de livrer rapidement une application sans bug et ne nécessitant aucun support une fois mise en production.

Puis en tant que consultant, j'ai partagé ces techniques avec d'autres développeurs lors des missions que j'ai menées à bien. A chaque fois j'ai été étonné de l'impact positif lié à l'application de ces méthodes:

- Amélioration de la lisibilité du code;
- Convergence plus rapide vers le zéro bug;
- Augmentation de la vélocité de l'équipe;
- Accroissement de la qualité du produit livré.

Partager ces techniques avec d'autres développeurs m'a aidé à les formaliser puis m’a incité à les présenter dans cet ouvrage.

A qui est destiné ce livre

Ce livre est destiné à un double public:

- A tous les développeurs .Net, du développeur débutant au développeur confirmé, qui ont l'ambition de développer des applications critiques ou grand public, qui ont l'ambition de fournir un code simple à comprendre, facile à lire, facile à maintenir, facile à faire évoluer;
- A tous les responsables qui ont les objectifs suivants pour leur équipe :
    - Augmenter la vélocité de l'équipe;

- Faire en sorte qu'un développeur puisse enrichir, modifier, maintenir le code d'un autre sans qu'on puisse distinguer qui à écrit quoi;
- Faire en sorte qu'une équipe de N développeurs agisse comme un seul développeur à la puissance N;
- Maintenir la maintenabilité;
- Répondre dans les plus brefs délais aux évolutions du métier ou du marché.

A propos des exemples de code montrés dans ce livre

Ce livre contient des exemples de code qui sont tous tirés d'applications réelles. Toutefois, ces exemples ont été retravaillés de façon à apparaître comme des "codes snippets" sans lien avec l'application et le développeur d'origine. Votre feedback est très important : je suis toujours à la recherche d'exemples de code réel à partir desquels je peux montrer comment appliquer les techniques décrites dans ce livre.

Pré-requis

Tous les exemples de code sont écrits en .Net C#.

Si vous souhaitez expérimenter vous mêmes les techniques montrées dans ce livre, je vous invite à installer Visual Studio 2015 Community Edition sur votre poste.

Work in Progress

Ce livre est en cours d'écriture. J'ai besoin de votre feedback pour l'améliorer : n'hésitez pas à commenter.

Code Companion

Vous pouvez voir en action les techniques du aaaProgramming en allant sur le projet GitHub associé à cet ouvrage. Un package NuGet est également disponible pour exploiter ces techniques directement dans Visual Studio.

Nommer correctement les choses

Nommer correctement les choses signifie donner un nom suffisamment évocateur à la variable que vous vous apprêtez à déclarer, à la méthode, à la classe, à la propriété, à l'interface, bref à tous les objets que vous allez manipuler au sein de votre application.

Cette activité de nommage permet de rendre son code expressif. Un code expressif raconte une histoire.

L'intérêt de raconter une histoire est qu'une relecture rapide permet de détecter très tôt les éventuelles incohérences, les éventuelles erreurs de conception , permet de se poser très rapidement les bonnes questions et permet d'échanger plus facilement avec les autres développeurs de son équipe.

L'objectif de ce chapitre est de vous montrer les méthodes qui vous permettront de trouver le nom le plus évocateur possible pour définir une variable, une méthode, une propriété ou une classe.

Trouvez le bon nom peut se révéler être une activité difficile ou chronophage.

Chaque fois que j'ai rencontré une réelle difficulté à trouver le bon nom, je me suis rendu compte que pour résoudre ce problème, il fallait modifier la conception ou les choix effectués pour développer l'application: dans tous les cas de figure il est apparu que la nouvelle conception ou les nouveaux choix étaient meilleurs que ceux initialement prévus.

Autrement dit nommer correctement les choses a une vertu inédite : détecter très rapidement si un choix technique ou un choix de conception est pertinent.

Si vous êtes responsable d'équipe, je vous recommande d'être à l'écoute des difficultés rencontrées par un développeur quand il doit donner un nom expressif aux différents éléments de son code : c'est un signal très fort de la présence d'un choix de conception ou d'un choix technique qui doit être retravaillé avec l'équipe.

Pour nommer correctement les choses, il est nécessaires de connaître et de respecter dans un premier temps les conventions de nom qui existent au sein du Framework .Net.

Ceci est un exemple, cliquez sur le lien de téléchargement pour obtenir le cours complet.