

1

Établir sa feuille de route

Lorsque vous décidez de partir en voyage, surtout si c'est en famille, vous le faites rarement sans un minimum de préparation : choix de la destination, recherche d'hôtel ou d'appartement, ce qu'il faut emporter dans les bagages, recherche éventuelle d'un itinéraire, organisation éventuelle des étapes, j'en passe, et sans doute des meilleures.

Si vous travaillez de la sorte, c'est uniquement pour *mettre toutes les chances de votre côté* de passer des vacances aussi agréables et réussies que possible.

De même, si vous décidez de faire construire une maison, vous allez demander à un architecte (parce que c'est son métier) de vous faire des plans, de calculer le budget nécessaire, et bien d'autres choses encore, dans un seul but : vous permettre d'avoir au final une maison dans laquelle il fait bon vivre et que vous aurez plaisir à retrouver après une dure journée de boulot.

L'idée qui sous-tend ces deux exemples est la même : si l'on veut mettre toutes les chances de son côté de mener à bien un projet et d'en tirer le maximum, il est primordial de se faire une *feuille de route*, de prévoir les grandes étapes du projet et d'anticiper autant que possible les problèmes qui pourront se présenter.

Ce n'est généralement pas ce que vous avez prévu, ni même plus ou moins anticipé, qui pourrait faire de votre projet un véritable enfer, mais bien tout ce qui n'a été ni prévu ni anticipé : la réparation du trottoir ou les égouttages devant votre bungalow juste au moment où vous y êtes, l'orage qui a coupé toute alimentation électrique ou encore la faillite de l'entrepreneur qui devait placer votre toit.

Il suffira en effet que vous n'ayez pas anticipé le risque d'une panne mécanique ou d'un pneu crevé pour que vous vous retrouviez, purement et simplement, en rade sur l'autoroute à plusieurs centaines de kilomètres de votre destination. Si, par contre, vous êtes parti du principe que *cela n'arrive pas qu'aux autres* et que vous avez pris vos précautions *au cas où...*, les conséquences de votre pneu crevé pourront se limiter à une simple mésaventure vous ayant fait perdre un peu de temps mais que vos vacances aurons tôt fait de vous faire oublier.

De même, il suffira que l'architecte ait négligé de demander une étude de sol et que les bulldozers se retrouvent face à un sol dont la nature demande un surplus de travail, pour que les coûts s'envolent fâcheusement.

Il n'y a **aucune raison cohérente** pour qu'il en aille autrement en informatique.

Bien au contraire ! Un projet de développement n'est jamais qu'une abstraction, une *vue de l'esprit* qui ne s'appuie que sur d'autres abstractions, sans qu'il ne soit possible, à quel que moment que ce soit, de s'appuyer sur un objet que l'on peut manipuler, sur lequel on peut faire des tests de résistance ou de flexion comme il est possible de le faire avec tout élément physique représenté sur un plan d'architecte.

Nous pourrions, bien sûr, nous appuyer sur certaines bibliothèques externes, mais, même si la qualité de ces bibliothèques est excellente, et même s'il est possible de chiffrer le coût de leur utilisation, elles n'en restent pas moins quelque chose de tout à fait abstrait qui ne fait que représenter la *vue de l'esprit* qu'en ont les développeurs, et qui de plus vous sera imposée.

Enfin, contrairement à votre maison pour laquelle il est parfaitement logique de commencer par les fondations, si vous n'avez pas une idée un minimum précise de ce que vous voulez faire, vous ne saurez jamais ce qu'il vous faudra faire, par quoi commencer, ni dans quel ordre le faire.

1. Trop d'analyse tue l'analyse

Il faut cependant garder un principe fondamental en tête : **les besoins évolueront en permanence.**

Le meilleur moyen pour étouffer un projet dans l'œuf est sans doute d'attendre d'avoir une analyse des besoins strictement complète pour commencer à le développer. Simplement parce que, à vouloir analyser complètement les implications de chaque besoin que votre application devra rencontrer, vous avez de bonnes chances de définir plusieurs besoins supplémentaires à chaque besoin que vous analyserez, et donc autant de besoins qui devront, eux aussi, être analysés. Au final, vous risqueriez très fort de vous retrouver, peut-être après plusieurs années, avec un nombre de documents d'analyse ou de spécifications tel qu'il vous serait impossible d'en faire une synthèse suffisamment simple pour savoir par où commencer.

Le principe est donc d'avoir une idée générale *suffisamment précise de l'ensemble* de ce qu'il faudra faire, ainsi que de la priorité avec laquelle les choses devraient être faites, tout en concentrant vos efforts d'analyse et de développement sur des parties plus facilement manipulables.

Le but poursuivi par le fait d'avoir une *vue d'ensemble* générale est surtout de *prévoir les endroits* où les évolutions viendront se greffer sur l'existant, et donc d'anticiper, dans une certaine mesure du moins, les points auxquels il faudra apporter une attention particulière.

2. Faites le test

Avant d'aller plus loin, je vous propose un petit exercice, que je vous demande de réaliser correctement (comprenez : sans aller voir dans les pages qui suivent). Réfléchissez à la manière dont vous vous y prendriez si vous deviez développer un jeu d'échecs.

Comme la gestion complète d'un jeu d'échecs peut s'avérer complexe et que je présume que vous n'avez pas énormément de temps à consacrer à l'exercice, il serait sans doute malvenu de vous demander de réfléchir à la totalité du problème.

Je vous propose donc de réfléchir une petite demi-heure et d'aller aussi loin que possible dans votre réflexion avec comme objectif minimal la possibilité de gérer au moins deux types de pièces différentes et bien sûr tout ce qui peut tourner autour.

N'hésitez pas à présenter votre solution ou à poser des questions sur le forum C++ de developpez.com en cas de besoin. De nombreux intervenants se feront sans doute un plaisir de vous conseiller. Vous trouverez la solution complète de cet exercice, qui ira d'ailleurs bien plus loin que ce que je vous demande ici, dans la [dernière partie](#) de cet ouvrage.