# Entretien préliminaire C++

## Questions ouvertes

Ces questions ouvertes n’ont pas de bonne ou mauvaise réponse. Leur but est de nous permettre de mieux comprendre votre parcours et tout ce que cela vous a appris.

### Comment qualifiez-vous votre maitrise du langage C++ ? Pendant combien de temps et de quelles façons avez-vous utilisé ce langage ?

Si je comparais ma maitrise du langage C++ à un langage parlé comme l’anglais, je dirais que je suis fluent. J’ai débuté ma carrière avec le C++ il y a 17 ans dans un contexte de PME ou il a fallu apprendre très vite dans le cadre du développement d’un logiciel avec une architecture monolithe. J’étais entouré par un architecte, un responsable technique et 2 autres développeurs. Nous étions avec les MFC avec comme outil de gestion des sources Clearcase et une base propriétaire pour la gestion des bugs. Ceci a duré 7 ans avec pour finir une équipe réduite à l’architecte et moi-même.

Ensuite j’ai été engagé chez GE (Alstom racheté par GE) pour reprendre le développement de 2 logiciels codés en C++/Qt, l’un étant une passerelle entre 2 protocoles (IEC et OPC) et qui s’est avéré être un peu plus que cela et l’autre un outil de configuration destiné à nos équipes en interne. L’outil de gestion des sources est svn et nous utilisons JIRA pour les Bugs. Cette expérience est encore d’actualité 10 ans après. Lors de mon embauche nous étions 2 développeurs mais je me suis retrouvé seul 6 mois après. Ensuite l’équipe s’est étoffée.

### Quel est votre opinion sur ce langage : ses forces et ses faiblesses ? Comment se protéger de ses faiblesses ?

Je pense que le C++ est le langage orienté objet le plus complet . Il est certes difficile à appréhender mais une fois pris en main il permet de faire absolument tout ce qu’on veut. On peut effectivement lui reprocher d’être assez complexe dans sa syntaxe ou dans ses parties déclaratives, contrairement au C# où tout le « travail » est déjà fait dans des briques qu’il suffit d’inclure et d’utiliser. Le C++ demande de part ce fait une forte rigueur notamment en ce qui concerne la gestion des librairies externes ou de la gestion mémoire. C’est aussi un langage ancien qui est de moins en moins enseigné dans les écoles et qui manque donc un peu de modernisme mais le C++ 11 a atténué cela.

### Quel est le projet le plus ambitieux / complexe sur lequel vous avez travaillé ? De quoi êtes-vous le plus fier ?

Dans mes deux expériences professionnelles, chaque projet auquel j’ai participé était ambitieux car ils étaient toujours au centre de notre activité.

Lors de ma première expérience j’ai été amené à faire comprendre à la direction la nécessité de repenser l’architecture du logiciel car cette dernière nous empêchait de le faire évoluer en fonction des besoins des clients. Nous avons donc refait l’architecture. C’est ce dont je suis plus fier sur cette expérience.

Dans ma deuxième expérience ce dont je suis le plus fier a été de faire changer les mentalités de nos équipes sur la qualité/fiabilité du logiciel de passerelle IEC/OPC qu’on appelle le "Coupling". A mon arrivée au sein de cette organisation le Coupling était la bête noire de nos équipes de déploiement car ce dernier « crashait » assez souvent, on entendait à chaque fois : « Ah mais c’est à cause du Coupling ! ». J’ai su fiabiliser le code, le faire évoluer pour qu’il devienne la pièce maitresse de notre solution et pour que cette remarque ne s’entende plus dans les couloirs.

De manière générale, je dirais que ce dont je peux être fier dans mon parcours professionnel c’est d’avoir toujours su relever le défi et de toujours avoir su tenir la barre quand il le fallait.

### Sur quelles bases vous appuyez-vous pour juger de la qualité d’un code ou d’un projet ?

En ce qui concerne la qualité du développement, il est pour moi évident que le mauvais code peut se voir rapidement d’un point de vue syntaxique :

* Mauvais nommage des méthodes, des variables.
* Incohérence dans la topographie (majuscule, utilisation du ‘\_’)
* Code non commenté

En effet si déjà la syntaxe manque de méthode et de rigueur il est rare que le fond soit de bonne qualité. Un bon code se doit d’être clair, structuré et lisible. Ensuite en allant plus en profondeur, on s’aperçoit assez vite s’il s’agit de bon code avec notamment du code factorisé, même aujourd’hui il m’arrive souvent en me déplaçant dans du code de refactoriser des parties.

Pour moi un bon code est un code qu’un développeur arrive à comprendre ne serait-ce qu’en lisant le nom des classes, des méthodes et des variables.

Et bien entendu un code commenté est indispensable pour permettre d’approfondir cette compréhension.

### Qu’est-ce qui est important selon vous pour produire un logiciel de qualité ? Comment faites-vous pour atteindre cet idéal ?

Un logiciel de qualité est avant tout un logiciel qui fait ce pour quoi il est destiné et ce en étant le plus fiable possible. Une architecture bien pensée et en adéquation avec le système permettra aux développeurs de bien comprendre la structure et de pouvoirs s’interfacer les uns les autres et ainsi faire un travail de qualité. Si le contexte le permet je pense qu’un environnement agile est idéal pour le développement d’un logiciel. Comme tout le monde le sait, l’agilité permet de coller au plus près des exigences clients dans un temps le plus court possible ce qui permet de limiter les coûts de développement et de gagner en qualité.

### Qu’avez-vous appris à faire avec l’expérience et qui vous parait aujourd’hui essentiel, alors que vous ne le faisiez pas à vos débuts ?

Prendre du recul.

Prendre du recul est quelque chose que jeune développeur il m’était difficile de faire, toujours « la tête dans le guidon ». Or si on arrive à prendre du recul et remettre en perspective les problèmes, une solution plus simple et plus avantageuse apparaît souvent. Je me suis aperçus que les solutions m’apparaissent alors que je ne suis pas en train de coder mais en pause-café, pause déjeuner, discussion avec les collègues et même sous la douche.

Analyser avant de coder.

Au début de ma carrière et même pendant un certain temps j’avais tendance à coder directement pour résoudre le problème. Maintenant avec l’expérience j’agis autrement, je code toujours un petit peu au début pour lancer ma réflexion et commencer à architecturer ma pensée (« de quoi j’ai besoin ? ») ensuite je m’arrête et j’analyse plus la situation et commence a décrire comment je vais résoudre le problème.

### Quelle place accordez-vous au « process » et à la méthodologie par rapport à la programmation ?

La méthodologie et les process sont les garants de la qualité des solutions développées. Il faut prendre le temps de les mettre en place et de les appliquer mais aussi de les faire évoluer. Dans un contexte comme celui auquel j’ai été confronté avec GE (équipe R&D dans les opérations) il a été souvent difficile de mettre en place ces pratiques du fait de l‘urgence dans laquelle nous travaillions. Cependant nous avons réussi petit à petit à le faire et ainsi améliorer nos process, nos méthodes et donc notre qualité.

### Comment faites-vous pour progresser et vous améliorer dans votre pratique au quotidien ?

Je pense que pour progresser le plus important est de sortir des sentiers battus, de notre zone de confort. Pour cela il ne faut pas avoir peur de s’intéresser à d’autres projets, d’autres problèmes en collaborant avec ses collègues par exemple. Nous avons tous des connaissances différentes, des façons d’aborder tel ou tel problème et ma solution n’est pas forcément la meilleure.

J’essaie aussi de faire preuve d’humilité en me disant que on peut toujours faire mieux c’est pourquoi j’essaie de revoir mon code pour le rendre plus lisible avec des commentaires de la factorisation dès que je peux le faire.

Encore une fois j’essaie de prendre du recul pour avoir une analyse du problème en dehors du problème du code.

## Exercice de développement

Le but de cet exercice est de réaliser un petit programme que vous devrez ensuite livrer comme si vous répondiez à la commande d’un client fictif. Il ne s’agit donc pas seulement de programmer un bout de code, mais de s’assurer qu’il sera utilisable par le client.

Le programme à réaliser devra permettre de :

* Trier par ordre alphabétique une liste de mots reçus en entrée. Une option devra permettre, si elle est spécifiée, de trier dans le sens alphabétique inverse.
* Eliminer les doublons de la liste de mots reçue en entrée.

Chacune de ces 2 options doit pouvoir être utilisée seule ou en combinaison avec l’autre.

Vous avez la liberté totale de choisir votre environnement technique et vos outils de développement. Seul le langage C++ est imposé, ainsi que la méthode de livraison du résultat qui devra se faire sur une plateforme collaborative basée sur git, telle que GitHub ou GitLab.

Bon courage ☺