

Les 7 questions qu'un développeur doit poser en entretien

Après notre article "7 questions à poser à un développeur en entretien RH", on se place de l'autre côté du bureau (du canapé ou autre) pour revenir sur les 7 questions que les développeurs et développeuses doivent poser au cours d'un entretien pour s'assurer de l'adéquation de leur profil à celui de l'entreprise.

Temps de lecture : minute

22 novembre 2021

Republication du 16 décembre 2019

C'est souvent la même histoire, lorsque je vois un candidat un entretien, il arrive souvent que la réalité de son quotidien soit bien différente de son impression de départ. Il pensait pourtant avoir une idée claire de ce qu'il voulait. *Quel est le moment où tout a dérapé ?* Ici on ne parlera pas de taille de société, de salaire, de proximité géographique de baby-foot, ou encore d'une énième place au classement Great Place to Work.

On parlera de ce qui fait l'essence du travail d'un développeur ou d'une développeuse, à savoir : l'équipe, le projet et les bonnes pratiques de développement qui sont de mises. Sans que ce soit évidemment une liste exhaustive, voici quelques points essentiels, qui résument les questions que les développeurs et développeuses DOIVENT poser en entretien.

Dans le process d'entretien vais-je échanger

avec des développeurs·euses de l'équipe ?

C'est fondamental de pouvoir rencontrer ses collègues. Il existe encore des boîtes où dans le process on ne rencontre que RH et responsable... sans avoir aucune idée des personnes avec qui on va échanger au quotidien.

Quelles sont les méthodologies et rituels mis en place ?

Challengez plutôt la capacité de l'équipe à faire preuve de pragmatisme et à se remettre en question plutôt que de l'Agile by the book. En définitive il ne s'agit pas de pointer du doigt une méthodologie plutôt qu'une autre. Cherchez plutôt si le mindest va vers une amélioration continue des pratiques et des usages. Et aussi challengez vos interlocuteur·rice·s pour voir quelles sont les rigidités de la structure - et il y en a toujours !

Comment vous assurez-vous de la qualité du code ?

Là aussi, si c'est un sujet qui est important pour vous - ça devrait l'être ? - challengez les approches mises en place, TDD (Test-Driven Development), Pairing, Code Reviews, session de Clean Code, etc.



À lire aussi

7 questions à poser à un développeur en entretien RH

Assurez-vous de comprendre comment se fait l'onboarding d'un point de vue technique ; challengez vos interlocuteur·rice·s pour comprendre comment se fait la montée en compétence. Êtes-vous tout·e seul·e dans votre coin, profitez-vous de la documentation mise en place ? Êtes-vous accompagné·e par un profil sénior avec qui vous faites du Pair Programming ?

Quelle est l'importance de la dette technique, comment est-elle traitée ?

L'important ici n'est pas tellement de constater sa taille mais plutôt la direction prise. Est-ce qu'elle se résorbe ? Quelles actions sont mises en place ? De manière générale, à part être dans une startup ou une cellule de R&D où on part souvent d'une plage blanche ; dans la majorité des cas il y a un existant, l'idée n'est pas de savoir s'il y a de la dette technique, mais plutôt quel est le soin que l'équipe prend à la traiter, quelle est la

direction en termes de qualité de code.

Quelle est la fréquence de vos mises en production ? Est-ce automatisé ?

Si on vous parle de Scrum mais qu'il y a trois releases par an... fuyez ! Constatez aussi l'inclinaison de l'équipe au DevOps et à la conteneurisation. Il faut aussi avoir en tête que le DevOps est avant tout un process ; rendre tout le process automatisé est possible, et même souhaite à bien des égards, mais encore faut-il qu'il y ait une qualité suffisante pour permettre l'automatisation... encore une fois sans qualité il est difficile d'avoir une usine de build performante sur le long terme.

Et enfin, cela va sans dire – mais ça va mieux en le disant –, demandez de manière précise les technos, frameworks, et outils utilisés. Cela peut paraître bête, mais combien de développeurs se sont retrouvés devant des RH qui leur ont parlé des technos récentes, et finalement étaient sur une version bien plus ancienne ! Cela revient à dire qu'il FAUT challenger la fiche de poste. Chez Code Insider, quand on discute avec un client, on creuse tous les sujets. Combien de fois on a pu voir la notion de TDD (Test-Driven Development) présente alors qu'en réalité il n'y a aucune inclinaison aux tests unitaire dans l'équipe.

Si on devait conclure en parlant du futur de la programmation – en paraphrasant Uncle Bob on pourrait dire que les développeurs doivent se responsabiliser, et prendre conscience des " Technical Practises " qui doivent être respectées afin de ne pas reproduire un Louvois bis.

César Mourot est cofondateur de Code Insider

