

## Interview développeur :

*Actuellement Développeur freelance.*

*Pendant 8 mois j'ai travaillé dans une banque d'investissement (pari sportif, type winamax).*

*Avant cela j'ai travaillé pour France Connect.*

### Quel est ton parcours d'étude ?

*Classique bac S spé science de l'ingé avec plusieurs spécialisation pour partir vers le numérique.*

*Après le lycée je suis entré à Epitech Rennes.*

*L'apprentissage chez Epitech est différent des autres car apprentissage par la pratique en 5 ans, titre d'expert car plus de pratique. Je ne possède pas de diplôme d'ingénieur.*

*J'ai commencé à mon compte en auto-entrepreneur durant sa première année d'étude chez Epitech. Désormais, mission en télétravail pour rester libre.*

### Qu'est-ce que tu maîtrisais avant d'entrer en entreprise ?

*Premier stage de 4 à 6 mois en fin de première année. Bases sur le C, suffisent à apprendre d'autres langages web c#, C++, ... Avant tout cela permet d'apprendre l'autonomie.*

*Les entreprises sont toujours contentes des étudiants Epitech car les stagiaires sont toujours prêts à se débrouiller et à apprendre.*

*Le principal but est l'autonomie. Former professionnellement et à la méthodologie de travail en entreprise. Travailler en équipe, projets en groupe, Développement des soft skills, organisation des réunions, organisations des sprint..*

### Quelle est la place de la méthodologie Agile dans ton travail ?

*Pendant les études plutôt méthodologie Kanban, un simple Trello peut permettre d'avoir du recul sur les projets. La méthodologie agile est venue particulièrement en entreprise mais n'est pas à appliquer à 100 %. En fonction de : temps disponible pour la mise en place, taille de l'équipe, délais à respecter et souhaits du client.. Les ESN n'ont pas forcément le temps de mettre en place une méthodologie Agile complète.*

*A contrario de chez France Connect où le SCRUM était respecté à 100 % afin de rendre une nouvelle version de produit fini. Manifeste Agile respecté ++.*

*Cela reste une bonne méthodologie car c'est la plus utilisée, mais avec précaution car n'est pas toujours possible en réalité. Il est important de la connaître.*

### Quelle est aujourd'hui ta stack technique ?

*Je suis un développeur plutôt polyvalent : Maîtrise d'une stack web : Nest et Next.js MongoDB, node.js, ... J'ai par le passé beaucoup bossé sur Angular mais je suis passé sur React. Angular permettait des projets organisés et complets. Le problème sur des gros projets est que l'organisation est difficile sur React. React est plus difficile pour un projet en groupe.*

*Pour moi, Next.js a donné un véritable coup de pied dans la fourmilière.*

*Ensuite le langage C++ car j'adore le réseau, les software embarqués.. Mais peu d'argent dans ce milieu en freelance. Les contrats sont nettement plus juteux en développement Web quand freelance.*

*J'aime aussi Rust mais c'est un langage de développeur car ce n'est pas possible de l'apprendre en premier.*

*Ce livre, c'est vraiment ma bible. -Nous montre à ce moment « The Rust Programming Language ».*

### Gestion de projet :

*Dans l'équipe il peut y avoir un Lead Dev et un Scrum Master.*

*Ils ont tout deux des rôles très importants dans l'impact et l'efficacité de l'équipe.*

*Le Lead Dev est souvent en collaboration avec le Scrum Master. Son rôle n'est pas de développer à fond mais de superviser et de représenter les différents devs.*

### Missions quotidiennes, routines en entreprise :

**9:00 :**

*Je commence ma journée vers 9h00, en premier a lieu la réunion daily pour faire un retour de la veille, faire remonter les infos, demander de l'aide, ... Cela sert surtout à donner à l'équipe vision sur son avancement. Vous comprenez bien qu'il est difficile chaque jour de connaître l'avancement des 12 devs de l'équipe, le daily permet d'avoir un regard global.*

**9:30 :**

*Ensuite je vais sur mon PC pour environ 30m de communication : Slack, mails, ... J'étais en entreprise responsable de la certification TLS, donc beaucoup de questions de Devops et autres, ...*

**10:00 :**

*Si ma tâche de la veille est terminée, je vais chercher dans le backlog du sprint et prends le ticket du haut et je me l'assigne, puis je commence à réfléchir puis coder sur cette tâche. J'adopte une méthodologie de développement appelée test-driven development. Les tests sont réalisés en premier puis on vérifie si le code colle aux tests.*

*Le dev, ce n'est finalement pas ce qui me prends le plus de temps. Je dirais que cela représente 80 % de test pour 20 % de dev pur. La qualité doit être là donc très important de passer par beaucoup de tests.*

**13:30 :**

*L'après-midi reprise de la tâche laissée. Chaque tâche dure un à deux jours, trois jours max sinon elle est doit être découpée. Si l'on voit que c'est un gros chantier, il faut détailler chaque fonctionnalité du chantier et y consacrer un sprint.*

**16:00 :**

*Vers 16h , je soumet la merge request Git, deux dev de l'équipe review du code, font des commentaires qui servent surtout à éviter les erreurs d'inattention ou apporter leurs connaissances. Il existe également un intégrateur, son rôle est de merge les request et de faire des tests pour analyser les documents. Lui doit mesurer la performance du produit en fonction des différents merge effectués. Dans l'IA c'était en terme de pourcentage, il trace les améliorations ou dégradations apportées des Merge afin de voir si un merge a réduit la performance, le pourcentage*

*de bonne réponse, .. Son but est finalement d'analyser la qualité des features, cela permet de voir si les tickets expérimentaux sont utiles ou non.*

### Retour sur méthodo agile :

*Pour revenir sur la méthodologie Agile, je l'ai vraiment appréciée ! Au début je me disais que c'était pour les puristes, mais en tant que développeur ça permet vraiment de ne pas avoir de pression sur la deadline. Lorsqu'il y a un problème il vient de l'équipe et pas d'un dev spécifique, personne ne ressent de pression sur sa personne.  
C'est plus difficile quand produit avec deadline fixe et un client avec demande rapide.*

### Soft skills :

*La première c'est vraiment l'aisance à l'oral. Rencontrer des gens que l'on ne connaît pas, collaborer avec eux et partager une même vision du projet, mettre les quelques différents de côté, devoir se mettre en avant, chercher de l'aide, ... c'est vraiment formateur. Je n'ai jamais vraiment eu de mauvaises expériences avec mes collègues.*

*La deuxième, c'est la sympathie, je suis toujours tombé sur des entreprises cools, où les patrons te permettent de bosser dans une ambiance fun. Souvent tu tombes sur des gens avec le même type de profil, des centres d'intérêts communs..*

*Troisièmement, il faut également apprendre à écrire des messages soignés et avoir une communication propre. C'est important afin de parler aux clients et aux autres équipes. Les mails doivent être concis et clairs, surtout pas insultants et pas impératifs, ça ne passe pas en entreprise.. Il faut expliquer, étayer, essayer de se mettre à la place de l'autre lorsque l'on envoie un message.*

*Après autre chose, c'est l'anglais ! Je n'étais pas particulièrement fort mais dès les premières réunions tu es obligé d'apprendre, ça s'impose à toi !*

*C'est pareil, pour les réunions il y a un savoir être. Il faut trouver la juste mesure entre trop participer et ne pas assez participer. Lorsqu'on prends la parole il faut être clair, être un orateur.*

*Tout ça c'est beaucoup l'expérience qui le permet, pas forcément la formation..*

*Il ne faut pas hésiter aussi à parler de ses états d'âmes, ne pas être tout seul dans ses problèmes, les collègues sont souvent bienveillants et des soutiens. Il ne faut pas garder de zone d'ombre dans son travail, le boss doit tout connaître. Ce n'est pas forcément possible de rester bloqué sur une tâche deux jours d'affilés. Il ne faut pas s'enfermer sur soi même car c'est à ce moment qu'on se fait détruire par le boss. Vraiment, je pense qu'il faut apprendre à donner des nouvelles et tenir au courant, montrer que l'on est bien en train de travailler.*

*Être un dev expérimenté ce n'est finalement pas le plus important en entreprise, c'est la collaboration qui prône.*

### Quelles sont les désillusions, ou les difficultés que tu as pu rencontrée en accédant à ce métier ?

*La difficulté au début c'était vraiment la communication, j'étais quelqu'un d'assez réservé, donc quand on m'a demandé de présenter à des clients et des responsables commerciaux des nouvelles*

versions de produits, j'ai vraiment commencé à paniquer! Je devais présenter devant quasi 100 personnes, c'est une pression très importante, j'ai vraiment appréhendé. Maintenant il n'y a plus vraiment de problème avec tout ça, je suis assez détaché, pareil pour les discussions avec les responsables. Au début c'était assez difficile de gérer la hiérarchie, de parler naturellement à ton N+, mais maintenant ça va mieux. Il faut être avenant avec les collègues et les inconnus et ça passe.

Finalement c'est pareil que pour l'anglais, c'était difficile au début mais s'est vite venu. C'est quasi-indispensable dans le métier.

Pour le côté organisation et gestion de projet ça m'a fait plus bizarre, je n'étais pas du tout habitué à gérer des projets en avance, à planifier. C'était difficile de faire confiance aux gens, j'adore que tout soit fait par moi et à ma façon, mais c'est impossible d'avoir cette vision en entreprise. Les projets sont trop gros et il faut apprendre à déléguer. A l'école on n'est pas habitué non plus à travailler de cette façon. Il faut apprendre à faire confiance. Aujourd'hui je travaille en freelance avec 4 autres personnes, et je suis très serein d'être avec ces personnes avec qui je collabore.

### Avantages et inconvénients du métier :

Un avantage c'est de travailler sur des super projets. C'est super intéressant de voir un projet grandir puis être utilisé par d'autres. Je me souviens le premier projet fini c'était une API pour envoyer un fax depuis un mail et inversement. Le jour du déploiement je voyais le fichier de log et c'était super satisfaisant de voir que le produit était utilisé.

Les projets sont challengeants, à chaque projet une ou des expériences supplémentaires. On n'a pas vraiment le choix sur les tickets et les délais, ça t'oblige vraiment à te dépasser pour produire ce qui est demandé.

Il faut aussi être dans une optique d'amélioration continue, c'est quand même un luxe que l'on a chez les des d'avoir assez de temps pour faire bien les choses et de se former aux technologies, il faut garder cette envie. Quand on est puriste sur la qualité c'est très satisfaisant.

Pour le négatif, on est sur l'ordinateur toute la journée. Parfois il y a un manque de contact humain et donc c'est important de prendre des pauses avec les collègues. Il faut aussi rattraper derrière avec un sport.

Quand on bloque une journée sur un bug c'est aussi négatif car on se remet beaucoup en question, mais très satisfaisant lorsque c'est résolu. Sur des projets perso ça va, lorsque l'on est en entreprise pression de ne pas faire avancer le projet.

Le métier de dev est vraiment bien, autre point positif c'est l'argent. C'est assez facile d'être à l'aise financièrement.

Un autre point négatif les ESN : Ce sont des boîtes à éviter car pour moi ce sont des boîtes d'exploitation. Les ESN veulent se différencier et elles sont occupées à faire leur promo alors que c'est un travail à la chaîne et on est considéré comme une simple ressource. Les rush sont importants, les deadlines sont courtes, bref ce n'est pas très intéressant sur le long terme. Par contre le point positif c'est la formation très rapide sur les technologies.

Pareil il n'y a pas ou peu de code review, pas d'avis à donner, pas de grande valeur dans l'entreprise.

Dans des starts up ou autres groupes, il y a quand même moins de pression au quotidien.

Est-ce que tu as des conseils à nous donner ?

*C'est compliqué à première vue, surtout ne pas rester sur ses acquis, toujours avoir l'envie de se former car c'est un métier à vivre entièrement. Il faut se renseigner sur actualité informatique, les frameworks, les mises à jours logiciels, .. ça permet d'éviter de se faire dépasser par le métier.*

*Il faut aussi essayer de faire le maximum. Donner des choses qualitatives c'est important personnellement mais également pour le projet, ça permet de revenir dessus plus facilement, ..*

*Il ne faut pas hésiter à s'ouvrir, à apprendre de nouveaux langages, de nouveaux domaines de la tech.*

*Personnellement j'ai commencé la freelance dès ma première année d'école. Aujourd'hui je n'ai plus de pression sur le fait de se sentir bien dans une entreprise. Finalement, c'est le dev qui impose son rythme à une entreprise. Le dev c'est une ressource précieuse, tu n'as pas à avoir le stress du burn-out ou autre..*

### Conseil de se lancer en freelance ?

*Les entreprises sont très formatrices au début, c'est quand même bénéfique.*

*Un dev junior c'est environ 38k par an, après impôts et autre c'est 2100 € par mois. En freelance tu n'es imposé qu'à 25 %, donc c'est quand même beaucoup moins. En plus dans mon cas je facture à la journée de travail, c'est 400€ par jour. Là actuellement je peut me prendre un mois de vacances sans me prendre la tête. Je paie même des formations super chères à côté. Pareil pour le matériel , je me permet d'acheter un set-up super confortable.*

*Après l'entreprise ça permet quand même une sécurité sur l'emploi, des congés payés, les arrêts maladies, ...*

*Mais pour moi la freelance garde quand même un avantage, je m'ennuie généralement assez vite, donc je n'ai pas vraiment de remord à quitter une boite. Quitter une boite c'est souvent pris un peu comme une trahison par les équipes, la boite.. là ça me permet de ne pas avoir d'attache fixe.*