



# Comment faire fuir les bons candidats : un cas d'école

---

22 septembre 2019



# Table des matières

1.	Les faits . . . . .	1
1.1.	L'entreprise X . . . . .	1
1.2.	La prise de contact . . . . .	2
1.3.	L'attente . . . . .	2
1.4.	Le « challenge » . . . . .	5
2.	Un râteau passé à la loupe . . . . .	7
2.1.	Le recrutement tech est un jeu de séduction... à double sens ! . . . . .	7
2.2.	Prendre soin de son image . . . . .	8
2.3.	Comprendre et respecter les candidats . . . . .	9
2.4.	Tester un candidat, ça ne se fait pas à coups de truelle . . . . .	9
3.	Épilogue : une erreur de jeunesse . . . . .	11

Salut !

Aujourd'hui, je vais vous raconter une mésaventure que j'ai vécue cette semaine. Cela sera une très belle occasion de faire un petit point et prendre du recul sur le recrutement des ingénieurs en informatique, à travers un exemple qui serait drôle s'il était isolé.

En ce moment, je passe des entretiens de recrutement à droite à gauche, à la recherche d'un job de rêve. Après tout, quand on est ingénieur, il est sain de se frotter au marché de l'emploi pour voir ce qu'il a à offrir, s'auto-évaluer et éventuellement découvrir de belles opportunités.

Étant donné que j'ai la chance d'être dans un secteur (ingénierie informatique, développement — notamment en Python) où l'on s'arrache les bons profils et où il n'est pas rare de crouler sous les demandes de contact, j'ai choisi des critères extrêmement sélectifs, en ne m'ouvrant qu'à des *niches* d'activité : si je dois changer de job, je veux que ce soit dans un domaine qui vend du rêve, comme celui de la musique, du divertissement, des arts, ou du jeu vidéo.

Tout se passait parfaitement bien, jusqu'à ce que je sois contacté par l'entreprise X...

## 1. Les faits

### 1.1. L'entreprise X

X est une jeune *start-up* d'environ deux ans qui fournit un service web dans le domaine de la promotion musicale.



**Je limite la description pour éviter que cette société soit reconnaissable.** Mon but n'est absolument pas de leur causer du tort.

## 1. Les faits

Pour développer et faire croître leur service, X recrute plusieurs profils, dont un *ingénieur backend*. Je note dès ici que le profil recherché est, selon leurs dires, celui d'un ingénieur **expérimenté**. D'ailleurs, l'équipe étant très jeune, ils se déclarent avides de recruter des gens chevronnés pour les aider à mettre en place et/ou améliorer leurs processus internes, car ils sont conscients que tout ceci est très nouveau pour eux, et désirent, bien sûr, évoluer dans la bonne direction.

### 1.2. La prise de contact

La semaine dernière, je reçois un mail, très bien rédigé (certes un peu générique), qui me présente l'entreprise X dans un langage "jeune et dynamique" : tutoiement, emojis et points d'exclamation sont de mise. Le mail se conclut sur une invitation à discuter rapidement au téléphone. Je réponds très vite : le domaine correspond, les postes et technos semblent convenir, ce que fait l'entreprise me parle. Le rendez-vous est pris.

Au téléphone, je tombe sur **Machin**<sup>1</sup>, cofondateur de X. Machin est jeune, il utilise un langage décomplexé, avec des réponses comme « Top ! » pour exprimer son approbation. Son discours est maîtrisé et bien huilé : il sait présenter son entreprise (« *l'aventure* » dans la parlance des jeunes pousses), il sait pourquoi il l'a créée, il sait où il veut l'emmener, il a l'air conscient des défis (les « *challenges* ») que X doit relever du point de vue humain et technique.

Je me présente à mon tour, lui brosse un résumé rapide de mon expérience en me concentrant sur les infos qui me semblent pertinentes dans son contexte. La discussion se conclut sur un *match* (dans la parlance moderne, on parle de *match* ou de *fit* quand les candidats et profils recherchés semblent bien correspondre).

Machin m'explique la suite du processus :

Top !

Nous allons t'envoyer un **challenge technique**, qui servira de base à de futurs entretiens, nous permettra de te situer techniquement, et te permettra également de te rendre compte de ce que l'on attend chez X.

Par contre, nous en sommes au tout début de notre recrutement, donc le challenge n'est pas encore tout à fait fini, mais notre *lead backend*, Bidule, est en train de le finaliser, et ça sera prêt dans le courant de la semaine prochaine.

*Machin, au téléphone*

J'attire votre attention sur la terminologie : il est question d'un **challenge technique**. Intérieurement, je tique un petit peu, mais je lui réponds que ça me va.

« Top ! Merci beaucoup pour cet échange. À très bientôt. Salut ! ».

### 1.3. L'attente

Très correct, Machin m'envoie un mail juste après l'entretien téléphonique pour récapituler (*débrief* dans le jargon) :

---

1. Il ne s'appelle pas comme ça en vrai, bien sûr. D'ailleurs, ce billet n'est pas une attaque personnelle : il faut voir y voir Machin comme une *persona* abstraite. La personne "réelle" derrière cette histoire a, depuis, appris de ses erreurs.

## 1. Les faits

Hello [nohar],

C'était un plaisir de pouvoir échanger avec toi. Je reviens vers toi semaine prochaine avec un challenge technique qui est une base d'échange pour la suite.

Encore merci, À très vite, Best,

*Machin, par mail*

Fin de semaine dernière, une petite relance pour me rappeler qu'ils pensent à moi.

Hello [nohar],

J'espère que tu vas bien! Un court mail d'update pour te dire que les tests techniques seront finalisés semaine prochaine ([Bidule] dans la boucle). On t'envoie ça au plus vite. Surtout n'hésite pas si tu as des questions d'ici là!

Encore merci, Best,

*Machin, par mail*

J'accuse réception (« Bien noté, merci, à la semaine prochaine »).

Le week-end passe, puis finalement, Mardi, je reçois un nouveau mail et je commence à pénétrer dans la quatrième dimension :

Hello [nohar],

J'espère que tu vas bien!

Le [X] challenge tech (Back & Ops) est finalisé. Le challenge est à faire en - de 4h sur une plage horaire de ton choix selon tes dispos. Il servira de base d'échange à un futur entretien avec [Bidule] sur la partie tech.

Est-ce que tu as un créneau de disponibilité dans la semaine sur une plage de ~4h (max) pour que je t'envoie le challenge au début et que tu nous le renvoies à la fin?

Encore merci, Best,

*Machin, par mail*

... **QUATRE HEURES ?!**



FIGURE 1.1. – Ma réaction en lisant ce mail

Je suis choqué, quelque chose ne tourne définitivement pas rond, mais j'y reviendrai plus bas. Je prends la décision de jouer le jeu. Je pose mon jeudi après-midi en RTT.

Mercredi soir, changement de dernière minute :

Hello [nohar],

J'espère que tu vas bien! On a refait le point avec [Bidule] suite à notre appel et on a revu le test technique en conséquence.

## 1. Les faits

On pense en effet également que 4h est un slot trop long. On a donc adapté le test en un test d'1h (plus concis et pertinent). Il s'agit comme mentionné au téléphone d'une première formalité qui est une super base d'échange pendant l'entretien à venir!

N'hésite pas si tu as la moindre question!

Encore merci pour ta motivation, À très vite, Best,

*Machin*

Ce mail me plonge dans la confusion : il fait référence à **un appel qui n'a jamais eu lieu**. Je ne les ai pas eus au téléphone depuis la semaine dernière. Selon toute vraisemblance, il me confond avec un autre candidat qui s'était plaint...

Il est trop tard pour annuler mon congé, mais soit. Va pour un test d'une heure.

### 1.4. Le « challenge »

Le jour même, je reçois un mail avec en pièce jointe une archive au format TAR.

L'archive contient :

- Un fichier `docker-compose.yml`,
- Un répertoire `testapp`,
- Un fichier `README.md` dans lequel le but du jeu est plus ou moins clairement défini : je dois réaliser un code qui passe les tests unitaires écrits dans `testapp`, commiter tout ça et le pousser dans un dépôt git.
- ... et c'est tout.

Le code de `testapp` contient des tests écrits avec `pytest`, qui interagissent avec une application dont l'URL est `yourapp`, et qui vérifient le fonctionnement de plusieurs appels.

- Un Hello World,
- Un appel qui est supposé inverser, et concaténer, une liste de chaînes de caractères, avec en commentaire l'instruction de ne pas utiliser un raccourcis syntaxique ou une builtin de Python,
- Un appel `nestedsum` qui doit réaliser la somme des nombres compris dans une structure imbriquée, genre `[1, [2, 3, [4, 5], 6]]`.
- Un appel qui est supposé associer une clé à une valeur dans une base Redis.
- Un appel qui est supposé récupérer la valeur associée à une clé dans un Redis.
- Un appel qui est supposé associer une clé à une valeur, si la clé n'existe pas déjà, dans un Redis, et retourner des codes d'erreur HTTP adaptés.

Comme proposé dans le mail, j'envoie un message pour clarifier et être sûr que j'ai bien compris que je suis censé :

- Écrire à partir de zéro une application web,
- La conteneuriser dans un Docker,
- Adapter la configuration `docker-compose`,
- Implémenter les appels correspondants dans l'application web,
- Versionner tout ça,
- Le pousser sur Github.

Au bout de trois quarts d'heure, un mail de Bidule.

## 1. Les faits

Hello [nohar],  
Effectivement, tu as tout compris!  
Lis testapp et tu devrais avoir une idee de l'organisation attendue.

*Bidule, par mail*



FIGURE 1.2. – "C'est très bien, very good !"

Je respire un bon coup, et j'envoie finalement cette réponse.

Ok, désolé mais je préfère renoncer, et voici pourquoi.

Visiblement ce challenge sert à tester la capacité d'un développeur (junior) à comprendre le code qu'il lit et à passer des tests unitaires :

- retourner "Hello World",
- inverser une chaîne de caractères sans utiliser une builtin de python (je n'en vois absolument pas l'intérêt, mais soit, faisons une boucle for sur range(len(input), -1, -1))
- calculer la somme d'une structure imbriquée (piège, la plupart des gens vont faire une fonction récursive au lieu de définir une stack explicite, ce qui est un anti-pattern en Python)
- savoir faire des opérations de base (get/set) dans un Redis, et transformer une erreur en un code de retour HTTP approprié le cas échéant.
- implémenter un "set if not exists" avec Redis.

Ce sont des opérations triviales, qui ne mobilisent aucune connaissance en algorithmique. Le seul challenge réel, est de réaliser une configuration pour redis-compose et une micro-application (avec flask ou autre...), et créer un repo sur github en moins d'une heure de manière à se laisser assez de temps pour écrire du code : ce sont des opérations qu'on ne réalise qu'une seule fois par projet, et jamais dans l'urgence, et si jamais on rencontre un



## 2. Un râteau passé à la loupe

bug au niveau du docker compose, on finit par passer une heure à déboguer docker et lire de la doc plutôt que s'attaquer au fond du sujet.

Le candidat doit donc savoir faire tourner et déboguer docker-compose en prenant moins d'une heure, mais on n'attend de lui aucun skill en termes de développement (un peu d'algorithmique, estimation de complexité?), d'architecture (comment distribuer une application? faire de la réplication? garantir la disponibilité des données? comment équilibrer la charge entre serveurs?), de conception (utiliser des patterns?).

J'en déduis que je n'ai pas le profil recherché : de toute évidence, ce test est taillé pour un développeur junior, on est très loin de la valeur que je pourrais apporter à une toute jeune entreprise.

Sans rancune, mais je suis désormais convaincu que c'est un *mismatch*, et donc je préfère arrêter là pour que nous ne nous faisons pas perdre mutuellement notre temps.

Bonne continuation,

*Ma réponse*

Plus tard dans la journée, Machin me rappelle au téléphone. Dans son laïus, il m'explique que ce test était en fait un test *de screening* (on y renviendra plus bas).

Qu'à cela ne tienne, je conclus cet épisode en lui envoyant, par mail, deux exemples différents de stratégies de *screening* utilisées dans la vraie vie, de manière à « alimenter leur réflexion sur le recrutement », et je me promets intérieurement de ne plus jamais avoir affaire à cette entreprise.

## 2. Un râteau passé à la loupe

Cette histoire (malheureusement vraie, malheureusement banale) illustre à peu près tout ce qu'il ne faut pas faire quand on est une « *nouvelle startup innovante et disruptive* ». Voici une analyse de ce qui ne tourne pas rond.

### 2.1. Le recrutement tech est un jeu de séduction... à double sens !

Dis-moi, Machin. Est-ce que ça te viendrait à l'esprit de te rendre à un premier rencard en sortant de la salle de sport, histoire de faire profiter ton *date* de l'odeur de ta transpiration sous tes belles fringues ?

... Non ? Sans blague !

La première chose à noter, c'est que dans le domaine des nouvelles technologies, **les profils qualifiés ont l'embaras du choix**. Nous ne sommes pas *en chien* lorsque nous cherchons un job. D'ailleurs en informatique, aujourd'hui, les candidats n'écument plus les annonces eux-mêmes : si un développeur compétent place son profil sur un *medium* de recherche d'emploi "classique" (comme Monster), il s'expose à un harcèlement téléphonique permanent, jusqu'à ce qu'il finisse par supprimer purement et simplement son CV.

Bref, si tu veux que quelqu'un vienne bosser pour toi dans ce domaine, et s'il est parfaitement logique de t'assurer que cette personne a le profil recherché, sache que **ça va dans les deux sens**, et tu as plutôt intérêt à **lui donner envie**. Il ne te suffit pas de déclarer « *si tu es blonde, à forte poitrine, ça m'intéresse* » pour *pécho*.



FIGURE 2.3. – "Tournez ménages" (les Inconnus, 1990) était un sketch, PAS un tuto de recrutement !

Et pourtant, Machin, c'est exactement ce que tu es en train de faire...

## 2.2. Prendre soin de son image

Alors certes, tu as un discours et un ton bien huilés pour présenter ta boîte et montrer que tu y crois. On sent que tu t'es bien préparé pour te présenter aux investisseurs et lever quelques millions d'euros, mais là je te parle de tes actes. Une fois que tu as eu un premier *call* avec un candidat, tu n'as encore **rien** accompli. Tu n'as rien sécurisé : il t'a donné 30 minutes de son temps, c'est tout, mais il a encore tout le loisir de changer d'avis et il a une vue en premier plan sur tout ce qu'il y a *derrière et tout autour* de ton discours.

- En m'annonçant que « le challenge n'est pas encore prêt », tu me montres directement que tu abordes ton recrutement à l'arrache : les tests, ça s'anticipe, et ça se crée en même temps que la fiche du poste que tu cherches à pourvoir,
- En changeant au dernier moment la durée de quatre à une heure, tu *crois* que tu me montres que tu sais te remettre en question, mais tu me hurles qu'en fait, tu n'avais même pas réfléchi à la façon dont tu allais tester tes candidats et tu me confirmes que tu y vas à l'arrache : tu commences par proposer un resto chic, et tu changes la veille pour le lendemain en faveur d'un MacDo,

## 2. Un râteau passé à la loupe

- En me confondant par mail avec un autre candidat, tu me montres que tu te contentes de copier-coller tes communications sans même les relire, signe que tu es probablement en train de pêcher au chalut, pas à la ligne,
- Et je ne parle pas encore du contenu du test...

### 2.3. Comprendre et respecter les candidats

Peut-être que les jeunes ingénieurs sortis d'école sont moins regardants que moi sur ce sujet, mais sache qu'un ingénieur, c'est quelqu'un de *rigoureux*, dont on attend qu'il fasse preuve de logique, de soin, de cohérence et de précision **et qui n'en attend pas moins de ses futurs collaborateurs.**

- Appeler le candidat 10 minutes en retard sans t'excuser, c'est lui manquer de respect.
- Lui demander de mobiliser **QUATRE HEURES** avant même d'avoir rencontré qui que ce soit chez toi, et donc lui faire poser une demi-journée (ou bien le mobiliser entre 20h et minuit ?!) alors qu'il ne sait même pas encore s'il veut bosser avec toi, c'est lui manquer de respect.
  - Tu as une idée du nombre de *processes* que les candidats suivent en parallèle ?
  - Comment ils font si tout le monde fait comme toi ? Ils prennent une semaine de congés ?
- Changer d'avis au dernier moment pour qu'il ait finalement posé sa demi-journée dans le vent, c'est lui manquer de respect
  - J'avais accepté le deal de 4h, même si tu réalises entre temps que c'est une connerie, **assume-la jusqu'au bout !**
- Répondre avec un délai de trois quarts d'heure pendant un test d'une heure, c'est lui manquer de respect.

### 2.4. Tester un candidat, ça ne se fait pas à coups de truelle

Annoncer au début un **challenge technique**, puis conclure sur « c'était du **screening** », c'est absolument contradictoire... Mais au fait, tu connais la différence entre les deux ?



FIGURE 2.4. – Ce que je visualise derrière les « Top ! » et les « Best, »

Eh bien pour commencer, ce sont deux choses **contraires** et **complémentaires**.

## 2. Un râteau passé à la loupe

### 2.4.1. Le screening

Le *screening*, c'est le fait de *trier le bon grain de l'ivraie*, à savoir écrémer les candidats pour en éliminer le plus grand nombre, **avec le moins d'efforts et d'investissement possible**. En informatique, il est très facile d'éliminer un candidat sur deux critères :

- Il ne connaît pas les technos, ou pas assez,
- On ne pige rien à ses explications.

L'enjeu d'un bon *screening* est de :

- **prendre le moins de temps possible** : un appel téléphonique de 30 minutes, ou encore un test écrit de 20 minutes suffisent largement, parce que le temps des candidats est aussi précieux que celui du recruteur,
- **rassurer les bons candidats sur la qualité de l'entreprise**.
- **s'adapter au profil recherché**.
- **tester aussi bien les connaissances techniques que la qualité de l'expression**.

Un test de *screening* trop facile n'est pas bon signe : si on ne se sent pas un tout petit peu *challengé*, alors on en ressort avec l'impression que la boîte n'est pas compétente. A contrario, un test avec des questions qui dépassent notre niveau aide à épicer les choses, à montrer que l'auteur du test est quelqu'un de bon, et que l'on aura probablement des choses à apprendre dans cette société.

**Il est hors de question de demander à une personne de réaliser un programme pendant un *screening***. C'est beaucoup plus d'investissement qu'il n'y en a besoin, alors qu'il suffit de demander à la personne de nous expliquer rapidement la différence entre un *thread* et un *processus* pour en éliminer la moitié.

Toujours dans cette optique de séduction, avec l'expérience, on peut même réaliser ce *screening* sans même préparer un test ou des questions, mais simplement en demandant au candidat de nous parler de l'un de ses projets, creuser les questions techniques que l'on trouve intéressantes, et essayer de toucher la limite de ce à quoi il sait répondre, *sur le ton de la conversation*.

### 2.4.2. Les challenges techniques

Lorsqu'il ne nous reste plus que quelques candidats à départager *sur le plan technique* (je ne parle même pas de l'humain, mais ça devrait venir avant), on passe alors aux tests techniques plus poussés. En général, on préfère réaliser ces tests en face à face, devant un tableau, en soumettant le candidat à des problèmes plus corsés, mais il est également possible de lui faire faire ce test *chez lui*, en lui demandant de réaliser un programme, soit sur une durée **courte** (une heure), soit sur une durée **illimitée** (« fais-moi parvenir ton code dès que tu auras eu le temps de le finir, on se donne une deadline à telle date »).

Ici, l'enjeu "séduction" est encore plus fort qu'en *screening* : on veut vraiment que le candidat soit impliqué dans la résolution du problème, l'amuser, l'intéresser et le stimuler pour qu'il passe un bon moment.

Pour les profils de développeurs senior, il n'est pas rare de les confronter à des problématiques d'architecture système, en plus des questions d'algorithmique.

### 3. Épilogue : une erreur de jeunesse

#### 2.4.3. Tester ce qui est pertinent

Qu'il s'agisse de *screening* ou d'un test technique plus poussé : il faut faire extrêmement attention à ne tester que des compétences utiles, sous peine d'introduire un biais dans l'évaluation des candidats. Typiquement, quand Bidule s'attend à ce qu'on ait poussé un dépôt git ou réalisé une configuration pour `docker-compose`, il est irrémédiablement **à côté de la plaque**.

Comme je leur ai expliqué dans le mail, si un candidat n'a jamais touché à ces outils, *on s'en fout* ! Ce sont des choses qu'il peut acquérir en deux heures le jour où on en a besoin, et surtout, ce n'est clairement pas quelque chose que l'on est amené à faire sur une base quotidienne. *A fortiori* sur un test en temps limité, où l'outil peut boguer, ou encore ne pas être disponible sur l'ordinateur du candidat, ce qui résulterait en un **faux négatif**, à savoir un échec complètement indépendant de ses compétences.

D'ailleurs, sur un tel test, il est bon de fournir au minimum le *boilerplate*, c'est-à-dire qu'au lieu de demander au candidat de créer les bons fichiers, on les lui crée à l'avance et l'on place des instructions en commentaire là où son code est attendu. De cette façon on lui évite cette tâche fastidieuse et inintéressante, pour mieux lui laisser le temps de se concentrer sur l'essentiel du test.

#### 2.4.4. Le test "parle" aux développeurs

En l'occurrence, ce que ce test m'a raconté (et que LinkedIn m'a confirmé), c'est que le fameux *lead backend* qui avait écrit ce test était inexpérimenté, et que comme tous les jeunes ingénieurs, il a cette propension à se gratter l'oreille droite avec le pied gauche, dans le but de "bien faire", car il n'a pas encore acquis cette leçon fondamentale : **KISS**.

Évidemment, Machin m'a pourtant assuré au téléphone que le test permettait d'écrémer « plus de 50% des candidats »<sup>2</sup>, sauf que ce dont il ne se rend probablement pas compte, c'est que même si c'était vrai, ce n'est pas des « 50% les plus mauvais » qu'il s'agit, mais plutôt, probablement, des « 25% les plus mauvais, et des 25% qui trouvent son test complètement débile », faisant ainsi fuir les candidats *de qualité*.

### 3. Épilogue : une erreur de jeunesse

Depuis la publication de ce billet, j'ai poursuivi mes échanges avec la personne réelle derrière le personnage de "Machin", car je souhaitais donner une fin positive à cette histoire : pour être tout à fait juste, je me dois de noter que le seul réel tort de cette entreprise est d'avoir **péché par jeunesse**, car ceci est leur toute première campagne de recrutement.

Dans notre échange, après que je leur ai expliqué tout ce que je viens de dire, Machin s'est excusé, et a corrigé le cap. Il a notamment :

- Changé complètement son processus de recrutement,
- Revu à la baisse l'expérience du profil recherché, parce qu'il pouvait difficilement savoir, *a priori*, que 10 ans d'expérience étaient *overkill* (voire néfaste) dans son contexte.

---

2. Je me demande d'ailleurs comment il a pu acquérir une stat fiable sur un jour et demie (mercredi et jeudi). En fait non, je ne me le demande pas, je SAIS qu'il m'a *bullshité*.

### 3. *Épilogue : une erreur de jeunesse*

Ce phénomène est assez rare, et est un signe suffisamment remarquable d'intelligence et de remise en question de sa part pour que je le mentionne ici.

Ce n'est, hélas, pas le cas de tout le monde...

---

Pour conclure, j'ai perdu une demi-journée pour rien. Mais ça encore, ce n'est pas grave.

Ce qui me révolte, en revanche, c'est que je vois de plus en plus de gens que je connais se faire *piéger* et traverser, au bout de quelques mois en poste, une cuisante situation d'échec au détriment de leur confiance en eux. Tout cela **parce que ces boîtes recrutent n'importe comment**.

Jeunes devs de ZdS, restez vigilents, et surtout **prenez soin de votre amour-propre** : ce genre de truc n'est ni normal, ni acceptable, même (et d'autant moins) pour un premier poste.

# Liste des abréviations

**KISS** Keep It Simple and Sexy. 11