Dans mon entourage, il y a ce charmant garçon avec qui je suis amie depuis nos 18 ans. Bref, ça remonte. On ne se connaissait pas encore qu’il était déjà mordu de développement informatique. Quand moi je m’interrogeais sur mon choix d’études, lui allait en cours en journée et bossait en soirée pour sa propre société « Cyber-Dev ». J’aurais mis 23 ans à comprendre le nom de sa boîte.

S’il est trop modeste pour se qualifier ainsi, je peux moi affirmer sans hésitation que c’est un authentique Full-Stack. Il aurait certainement pu gagner des fortunes, il a préféré un salaire moins mirobolant à la STIB avec des collègues plus compétents que les précédents et une vraie qualité de vie – sauf les nuits de changements d’heure apparemment.

Vendredi 09/09/2022 à Berchem-Sainte-Agathe

VVH : Hello Patrick. Peux-tu nous raconter comment tu es tombé dans l’informatique.

PT : Je commence mon aventure dans la programmation vers 1994 lorsqu’on m’offre un livre sur comment programmer en Turbo Pascal.

Ensuite, lorsque j’avais environ 15 ans, j’ai pris des cours du soir à l’EPFC (à l’EPHEC ?) en Visual Basic. J’ai aussi fait un peu de Delphi. De 1999 à 2001, je suis allé à l’Institut Paul Lambin puis j’ai enchaîné avec un graduat en tant qu’analyste programmeur de gestion à l’EPHEC.

Au niveau de ma carrière, mon expérience en tant qu’analyste-programmeur indépendant a surtout été en complément de mon activité de dépannage informatique. J’ai également eu l’occasion de créer plusieurs programmes que ce soit un CRM de gestion de clients, factures, projets etc. pour ma propre société CyberDev, ou par exemple un petit programme en Delphi, pour la société Eurocities, qui faisait l’inventaire des machines – inventaire très léger : c’était le nom de la machine, les rams, CPU, disques durs - et il renvoyait les informations dans une base de données via une page en PHP pour stocker le tout dans une base de données en MySQL. J’ai également fait pour une petite société un add-on sur Exchange pour la gestion des contacts. Ce genre de choses.

En tant qu’employé, j’ai plus fait du développement pour m’aider dans mes tâches. Par exemple j’ai été intégrateur pour le logiciel Synexis. Quand ils sont passés de la version 3 à la version 4, les fournisseurs n’offraient pas d’option pour migrer les données. La version 3 n’avait de système d’export. Donc j’ai moi-même dû faire du reverse-engineering pour créer une application exportant les données sous un format compatible à l’import dans la version 4.

À la STIB, du temps où il y avait encore les valideurs orange à l’entrée des métros, ceux-ci étaient reliés à des pcs en OS2 qui centralisaient les données. Il y en avait une centaine dans tout Bruxelles. J’avais créé une base de données pour pouvoir me connecter à distance et rapatrier les données, vérifier que les valideurs réagissaient bien, de façon à gagner du temps.

Parce que pour anecdote les nuits de changements d’heure, moi je devais me réveiller à 3h du matin pour appuyer sur des boutons pour les valideurs et les redémarrer pour les mettre à la bonne heure.

Donc pas mal d’expérience à ce niveau-là.

*VVH : Et tu as été employé dans des boîtes de consultance via lesquelles tu as travaillé pour des sociétés comme UCB. C’est bien ça ?*

*PT : Oui tout à fait. Quelque part aux alentours de 2012 j’étais consultant chez UCB et je m’occupais d’un logiciel fourni par Symantec - Client Management Suite – qui faisait la gestion du parc informatique. En gros il y avait 10.000 pc à travers le monde et il fallait aller vérifier que les utilisateurs pouvaient commander des logiciels. Il fallait que ces logiciels s’installent de manière silencieuse à la demande - pour la gestion des licences, la gestion des outils de travail et pour avoir des pc standards. S’ils demandaient 5 ordinateurs, ils devaient pouvoir recevoir 5 ordinateurs identiques.*

VVH : Aurais-tu des conseils pour ceux qui désirent se lancer dans le web-développement ?

PT : Mon premier conseil c’est de ne pas avoir peur de perdre du temps dans le sens expérimenter, lire. Ce n’est pas une perte de temps que d’essayer un truc, de se rendre compte que ça marche ou que ça ne marche pas. Tu ne sais pas à quel moment ça va te servir. Donc il y a un moment il ne faut pas avoir peur de prendre du temps et de prendre un cours sur UDEMY, sur Pluralsight sur NORN-online. Des plateformes il y en a 13 à la douzaine, sur YouTube il y a pléthore de trucs, il ne faut pas hésiter à perdre du temps surtout qu’en plus aujourd’hui comparé à il y a 20 ans il y a beaucoup plus de facilités à avoir des cours, du contenu, des exemples, des compilateurs, des IDE*.*

Donc premier conseil ne pas avoir peur de perdre du temps – intelligemment hein – pour expérimenter, lire.

Le deuxième conseil c’est la SÉCURITE ! tu peux créer la plus magnifique application que tu veux, si elle n’est pas sécurisée, il ne faut surtout pas la vendre. De toute façon, si tu essaies de vendre ton app et qu’elle a des failles de sécurité, personne n’en voudra. Tout doit être orienté sécurité.

Il faut aussi penser à l’informatique en mode récession. Faire plus avec moins. Tout ce qui est structural-code CI/CD, tout ce qui est automatisation pour les déploiements, tout ce qui est tests. Très important le testing.

*Économiser sur les tailles des données. Et encore que ça, l’économie des données c’est plus concernant le Cloud : comme avec le cloud tu paies chaque ressource que tu prends, si tu arrives à avoir une base de données plus petite, elle sera plus économe pour ton patron ou ton client.*

Voilà, ne pas avoir peur de perdre du temps, sécurité, CI/CD au niveau du déploiement. En fait c’est tout ce qui va autour du développement. Pourquoi je dis ça ? car en fait là je n’ai pas parlé d’une seule ligne de code mais uniquement de ce qui va autour.

De 1 parce que c’est mon boulot au day-to-day. De 2 car aujourd’hui si tu vas sur Stack Overflow, tu trouveras des milliers d’exemples de code*.* Tu as plusieurs sites dont rosettastone.com où tu as des centaines de bouts de codes et l’idée c’est dire ta problématique —genre tu veux un menu de navigation — et bien tu vas un menu en Pascal, en C#, C++, et tous les exemples possibles. En gros tu prends et tu fais copier/coller.

Et aussi depuis que Microsoft a racheté GIT, ils ont sorti Github Copilot. L’idée c’est que ce programme te propose des exemples de codes. Tu dois lui décrire ce que tu désires obtenir. Lui va regarder dans son immense base de données et il va te proposer du code correspondant à tes demandes. Donc tu n’as plus besoin de savoir coder pour coder puisqu’il va te proposer du code. Alors bon c’est souvent un peu pourri et il faut l’adapter, mais on se rapproche du moment où l’ordinateur te proposera un code parfait. On va vers ça.

VVH : à quoi ressemble une de tes journées-type ?

PT : Une journée type à la STIB ? En tant que développeur, tu vas commencer par Le Scrum meeting c’est 15 minutes 2 à 3 fois par semaine (selon les équipes) où tout le monde va dans la même pièce et tu as des post-it qui représentent ce qu’on doit faire. C’est là que chacun va dire moi « je suis là je dois avancer » et tu vois d’environnement en environnement si tu es bloqué à certains endroits, si tu as besoin de quelqu’un. C’est là qu’on va en discuter.

Ça doit être vraiment court, straight to the point. Tu dois décrire avec précision ce qu’on t’a demandé, si tu es bloqué et si oui où et de quoi tu as besoin pour avancer.

Ensuite, les réunions avec les architectes et les analystes pour savoir ce qu’on attend de moi. Parce que ce n’est pas facile d’avoir un bon cahier des charges qui décrit précisément les choses. Surtout si tu ne sais pas ce qu’est une vente, alors c’est difficile de coder un système de vente. Si tu ne connais pas les produits à la STIB, que tu ne sais pas qu’il y a 10 sortes de tickets différents, et autant d’abonnements qu’il y a différentes zones avec DeLijn, le TEC, etc. Si tu ne sais pas tout ça, c’est quasiment impossible de coder une application qui gère correctement tous ces produits et tout ce qui en découle.

Donc discussions avec les architectes et les analystes pour essayer de comprendre comment il faut coder le truc. Puis tu codes, et tu es seul face à ton ordinateur et puis tu te bats et puis tu codes et puis tu te bats et puis tu codes …

Si tu as l’occasion vas sur [www.lesjoiesducode.com](http://www.lesjoiesducode.com). Au début tu ne vas pas comprendre. Puis tu vas comprendre et tu vas te marrer deux fois plus. Et des fois tu pleureras devant ton ordinateur quand tu comprendras que tu es en train de vivre ce que tu as vu dans les blagues.

Après tu codes, tu testes, tu fais des réunions avec les business pour vérifier que tu as bien compris et codé ce qu’on t’a demandé et que le résultat correspond aux demandes. Tu peux faire aussi des réunions pour montrer les nouvelles fonctionnalités à tes collègues.

Tu vas échanger avec eux, c’est-à-dire que si tu es bloqué tu vas demander de l’aide ou si ton collègue est bloqué il va venir te voir. Vous allez avoir des discussions sur comment faire au mieux, quel algorithme utiliser, quelle méthode utiliser, quelle librairie utiliser, quelles sont les librairies au sein de la STIB etc.

Donc tout ça c’est beaucoup de recherches. Automatiser, écrire les tests, discuter avec les testeurs pour savoir ce qu’ils doivent tester et comment ils peuvent le tester, tester, discuter avec ceux du monitoring pour vérifier comment ils vont monitorer comme quoi ton application fonctionne bien. Quelle méthode tu dois implémenter pour fournir l’information aux autres.

La communication est primordiale à tous les stades pour savoir ce que tu dois coder et ce que tu dois fournir aux autres.

Aider les gens de l’opérationnel quand tu as un truc qui foire parce que quand on t’appelle en te disant ah le système de vente est bloqué parce que la communication avec la banque de données est bloquée on ne sait pas pourquoi. Ben… ok je vais regarder ce qui se passe.

Tu as les nuits de mises en production. Ça aussi c’est toujours comique. Au début pour te donner une idée les premières mises en production de Sales, on y passait un en weekend complet. On commençait le vendredi soir vers 22-23h quand ils fermaient les derniers canaux de vente physiques. Puison arrêtait tous les services on mettait tout à jour et on commençait à tester

Quand l’équipe IT avec ses 20 développeurs avait fini, on s'arrêtait. Ça c’était le vendredi.

On recommençait le samedi matin. On ouvrait les premières Bootiks et les Kiosks et les gens du business commençaient à tester les nouvelles fonctionnalités. Test de régression - vérification pour voir que ce qui marchait hier marche toujours aujourd’hui.

Le dimanche on faisait le point. Et comme il y avait beaucoup moins de points de vente ouverts, on en profitait pour faire les ajustements.

Puis le lundi, tu te prenais la vague d’incidents parce que bien que tout ce qui avait été testé durant weekend, il y avait encore des trucs qui ne marchaient pas, ou des trucs qui avaient cassés pendant la nuit.

Et donc on a remplacé ce weekend complet avec 20 développeurs, tous sur site évidement — avec les coûts que ça engendre — en automatisant tout ça. Maintenant ça prend 40 minutes et on est à 4.

VVH : donc vous avez fait virer des gens

PT : Non ! On leur a permis de travailler sur d’autres choses plus intéressantes.

Mais d’un autre côté, je rappelle qu’en informatique la base c’est l’automatisation. Depuis la découverte du charbon et des locomotives à vapeur, jusqu’à aujourd’hui on n’a jamais fait autre chose qu’automatiser tout ce qui est possible pour augmenter les rendements et faciliter le travail. C’est par exemple éviter qu’il y ait un guichet ouvert 24/7/an en mobilisant des dizaines voire des centaines de personnes pour seulement 3 clients, et également permettre au client de pouvoir commander ses courses, faire ses comptes, taper ses impôts à 2h du mat sur son pc ou son Gsm.

Alors quand j’en discute avec des amis parce que je suis évidemment inquiet sur l’avenir des développeurs car fatalement a un moment donné il va y avoir des bons programmeurs qui vont bien faire le truc et qui *vont rendre les autres devs obsolètes.* Donc à partir du moment où tu as WordPress et Joomla qui ont 85% du business avec des gens qui publient les mêmes trucs, 10% qui sont des sites de vente genre Amazon, et les sites de streaming comme Disney+ et Netflix, il ne reste pas grand-chose. Du coup comment tu veux vendre un site web ?alors oui il y a le Front-End parce que là tu vas vraiment pouvoir te différencier tu as le mais ça c’est une autre partie avec une grosse machine marketing etc.

Et puis mes amis me disent « ne t’en fait pas on invente on réinvente mais on fait toujours la même chose. Heureusement on a des politiciens qui changent les lois pour tout ce qui est systèmes bancaires qui rajoutent des taxes, et ben il faut refaire des systèmes qui vont calculer les taxes, pareils pour les pensions ».

À la STIB je ne sais pas trop comment ça va évoluer mais ils essaient de faire mobilité ASA-Service donc on arrive quand même à réinventer pas mal de trucs. Surtout quand ils s’arrangent toujours tous les 4 ans pour te faire refaire la même chose.

VVH : ça, ça me parle beaucoup. Ce n’est pas que dans le web.

Pourrais-tu me repréciser ton poste à la STIB ?

PT : Je suis officiellement Application Engineer. Donc je reçois un programme à installer, un mode d’emploi sur comment l’installer, je dois l’implémenter et faire en sorte que ça continue à marcher. Là actuellement (ndlr : vendredi 09/09/2022 à 20 :30) je suis de garde. On peut m’appeler s’il y a un problème avec le système des portillons, les canaux de vente, le système de régulation métros, le système de régulation des trams et des bus.

VVH : Quelles sont les avantages à travailler comme indépendant ?

PT : quelles sont les bénéfices à être indépendant : aucun, parce que tu as autant de patrons que de clients – tous tes clients sont tes patrons. Et ils vont tous t’appeler quand ça les arrange eux à n’importe quelle heure du jour ou de la nuit. Le niveau de vie est plus élevé et tu peux te payer des gadgets. Donc si tu veux t’acheter une grosse voiture, des GSMs tous les mois, une montre connectée - c’est très bien. Par contre si tu veux t’acheter une maison, c’est une pas terrible.

VVH : Changerais-tu quelque chose dans ton parcours, dans tes choix ?

PT : ne pas avoir peur de perdre du temps, rester focalisé sur une seule chose et la terminer. Car c’est moins utile de savoir un peu programmer en 4 langages différents que de connaitre un seul langage à fond. Je peux te programmer à peu près la même chose en python, en C#, en Java et en PHP mais ce n’est pas très utile.

Ne pas avoir peur de demander de l’aide. Ça c’est parmi les choses que je regrette – avoir peur de passer pour un imbécile ou un incompétent, ben ça m’a fait passer pour un imbécile et un incompétent. Alors que si j’avais posé la question, on m’aurait répondu et j’aurais avancé.

VVH : Selon toi, les choses ont-elles évoluées dans la manière de travailler dans le web ? dans le bon sens ou dans le mauvais ? Est-ce que c’était mieux avant ?

PT : Est-ce que c’était différent avant ? oui. Pour moi dans les années 2000 c’était beaucoup chacun pour soi. Et tu n’avais pas Stack Overflow avec des exemples de code, et l’open-source où tous les développeurs partagent leurs connaissances.

Donc avant beaucoup de chacun pour soi et maintenant beaucoup plus de partage de connaissance et d’aide.

VVH : Qu’est-ce qui ce qui te plait dans le développement ?

PT : Ce qui me plait c’est l’aspect créativité mais surtout l’aspect gain de temps et ainsi pouvoir se concentrer sur les vrais problèmes. Je suis plus orienté sur la fonctionnalité des choses, sur leur utilité et les gains que ça permettraient. Par exemple le GPS - ça évites de faire des kilomètres inutiles et donc tu gagnes du temps, tu évites de consommer une certaine quantité d’essence donc économie écologique et financière.

Et puis aussi l’aide aux utilisateurs finaux, parce que le but c’est ça d’avoir des clients qui sont satisfaits. J’avais fait un logiciel pour une avocate pour qu’elle enregistre sa facturation. Donc elle peut maintenant encoder au fur et à mesure ses prestations, et simplement en appuyant sur un bouton la facture sort avec une référence unique et ce genre de choses. Ça lui fait gagner du temps - temps qu’elle peut utiliser soit pour ses passions soit pour son travail.

C’est surtout ça qui m’animes

VVH : Il y a eu un moment marquant dans ta carrière, car tu as beaucoup évolué et fais des choses fort différentes. Y-a-t-il eu un tournant précis ?

PT : Un tournant dans ma carrière ? bah début 2000.

Quand tu es entouré de mauvais de développeurs qui te disent que le problème vient d’ailleurs que chez eux. Au tout début, j’ai des très mauvais souvenirs d’un développeur Delphi. À sa décharge, le problème ne venait pas de son code mais du serveur de base de données qu’il utilisait. Mais en tant que développeur c’était quand même sa responsabilité surtout qu’on était en 1999-2000 - ce n’était pas l’époque où tu avais 5 experts avec 1 qui s’occupe du réseau 1 qui s’occupe de l’OS 1 qui s’occupe de la base de données 1 qui s’occupe du frontend et 1 qui s’occupe du réseau. Non, il y avait une seule personne qui s’occupait de tout. Donc tu ne commençais pas à dire « mais oui mais non mais c’est pas moi ».

Donc être entouré de mauvais développeurs, être entouré de mauvaises personnes. Et ce qui a changé c’est d’être entouré de gens qui sont orientés solution. Tu as un problème, on ne va pas venir te taper sur la tête – on va venir te voir et te demander ce qu’~~on~~ il y a comme solution. Ya un problème ok. Comment est-ce qu’on le règle.

Et ça mon frère l’a rencontré beaucoup plus dans des boites américaines et moi je l’ai rencontré beaucoup plus ici à la STIB.

Les tournants de ma carrière, je dirais que c’est ça.

VVH : selon ton expérience, quelles sont les Soft-Skills importants ?

PT : l’informatique c’est un milliard de choses simples qui ont toutes un rapport les unes avec les autres. Chaque chose peut avoir un impact sur n’importe lequel de tous les autres éléments. Donc il faut avoir un esprit de synthèse.

Il faut être paresseux. Parce qu’il ne faut pas faire les choses deux fois – c’est pour ça que tu automatises.

Il faut aimer les puzzles, il faut aimer décortiquer. C’est une des choses pour lesquelles je ne suis pas sûr que je ferais un bon programmeur sur du long terme – c’est que tu dois dois vraiment découper les choses en toutes petites petites petites parties, parfois les regrouper. Et c’est parfois un peu fatigant. C’est pour ça que je dis aimer les puzzles, les Lego, aimer les trucs qui s’emboitent les uns dans les autres.

[il manque qq chose]

Les plus gros problèmes sont avant tout humains, ils sont rarement techniques.

Un problème technique tu vas le régler. Sauf si tu as perdu des données parce que tu as fait un drop de la mauvaise table de la base de données de production et que tu n’as pas de backup. Mais à part ~~ça~~ ce genre de cas très particuliers, tout le restant tu pourras le remettre en route. Il est évident que si tu travailles dans certains domaines tu dois être beaucoup plus prudent– quand je travaillais pour UCB, il y avait des machines qui ne pouvaient pas être éteintes, qui ne pouvaient pas être redémarrées parce qu’elles étaient reliées à un thermomètre et qu’elles enregistraient en permanence les températures de vaccins. Ils ont eu des cas où l’ordinateur marchait mal et a été redémarré. Comme ils ne pouvaient plus prouver que la température n’avait pas variée. Ils ont dû tout jeter. Donc ça veut dire qu’on arrête l’usine de médicaments pour un pc qui a redémarré. Tu jettes pour 100.00€ de vaccins médicaments etc. puis tu remets la machine en route. Donc il n’y a pas mort d’homme mais tu as quand même fait perdre quelques centaines de milliers d’euros à la boite qui t’emploie.

Mais pour ça il y a une solution. Elle n’est pas gratuite mais il y a une solution.

Par contre si tu as un chef qui te dit non, ça c’est un problème humain. Quand t’as un collègue qui te dit non, c’est un problème humain. Quand t’as un collègue qui te dit fait comme ça et que tu lui ~~dis~~ réponds ce n’est pas la bonne façon de faire, c’est un problème humain. Et ça il n’y a pas de solution.

Au niveau Hard Skills, c’est tout ce qui est testing – savoir tester correctement. Test Driven developpment. Ça a le vent en poupe et c’est super important parce que tu te mets dans la peau de l’utilisateur final. Quand tu commences – avant même de commencer à coder - tu vas décrire ce que tu veux faire et comment tu veux que ça marche. Et seulement ensuite tu commences à coder ce que tu veux tester. Mais attention : il ne faut pas que ce soit toi qui testes ce que tu as écris, il faut que ce soit quelqu’un d’autre comme s’il était l’utilisateur final c’est-à-dire sans savoir à l’avance où cliquer, où regarder, quel champ remplir et comment le remplir.

Mais l’aspect testing est super important parce que tu peux avoir… [il manque des choses]

PT : Je regarde dans les questions que tu m’as envoyées… Alors j’ai 10 paragraphes pour celle-ci : est-ce que je suis passionné par ce que je fais ?

VVH : non, ça s’entend clairement…

PT : la question était est-ce qu’il faut être passionné, se tenir au courant et le faire en permanence. Pour moi j’ai cette idée en tête que depuis 2000-2001 : tu avais l’ADSL grâce auquel tu pouvais télécharger de la musique, des films, écrire des sites des blogs, tu pouvais télécharger des programmes, tu pouvais chatter sur internet, publier des photos, tu avais Skype. Tous les basiques étaient là.

Et on refait les mêmes choses. Sur ton téléphone, 85% des sites, ce sont des gens qui publient des informations et le reste ce ne sont quasi que des magasins. Rien d’extraordinaire. Donc tu ne dois pas t’imaginer que tu vas devoir tout le temps relire et réinventer. Tu vas mettre des informations dans une base de données structurée, tu vas l’afficher pour un utilisateur. Ça n’a pas changé depuis 20 ans. Ok il y a une série de nouvelles fonctionnalité et l’IA. Mais même la reconnaissance de caractère, les OCR etc. ça existe depuis 20 ans. Alors certes la qualité a évolué mais les fondamentaux sont toujours là.

VVH : toute dernière question : qu’est-ce que tu me conseillerais à moi en particulier. On se connait depuis plus de 20 ans. Tu sais comment je suis. Tu connais mes qualités et mes défauts. Tu sais que je suis paresseuse dans le bon sens du terme, que j’aime que les choses soient bien faites et propres plutôt que simplement tape-à-l’œil, et que du coup je pourrais garder mon code pour le réutiliser plus tard.

PM : Le conseil que moi je vais te donner, c’est de penser à ton toi futur quand tu vas écrire ton code et que tu devras le relire dans 6 mois, et du coup te rendre service en l’écrivant bien.

PT : oui. Et les conseils que j’ai dits tantôt

Faut s’y mettre, faut se trouver un bon rythme. Parce qu’avant comme j’étais dans l’opérationnel à aider beaucoup les gens à n’importe quelle heure de 8h à 22h, ben on m’appelait tout le temps. Moi ce qui m’a manqué c’était d’avoir du temps à moi. De me dire ok il est 9h du mat, j’ai bien dormi hé bien maintenant je vais me concentrer sur une problématique précise. Et arriver à décortiquer un truc chaque petite chose ça peut être difficile pour beaucoup de gens. C’est arriver à 8h du matin et couper ton téléphone pour pouvoir te concentrer. Décortiquer à ton rythme sous l’aspect utilisateur, sous l’aspect sécurité, sous l’aspect développement. Avancer petit à petit.

Il faut pouvoir se concentrer sur des choses minuscules comme une couleur – parce que le bleu ce n’est pas du juste bleu. Ce n’est pas idiot de passer 2jours à choisir 2 couleurs qui vont bien ensemble. Faire un calendrier avec toutes les contraintes – tu réalises ? faire un beau graphique ? tu peux y passer des semaines. Il ne faut pas avoir peur d’essayer et de perdre du temps. Le UX est très important. Ou mettre un bouton ? c’est très important. Par contre ton chef va arriver et te demander : c’est prêt ? faut trouver un équilibre entre tout ça.

Tu dois profiter de tes 14 mois de formation pour tester dans tous les sens et avec sérénité. C’est là que tu verras les experts. Ce sont eux qui auront pris le temps de tout décortiquer en tout petit.