La Société des Transports Intercommunaux de Bruxelles naît officiellement le 1 janvier 1954.

Elle profitera des grands travaux routiers en vue de l’Expo58 pour développer son réseau de tramways et leurs créer des sites propres.

<https://www.stib-mivb.be/irj/go/km/docs/WEBSITE_RES/Attachments/Corporate/Rappport_Financier/mandataires2020/2022/2021%20-%20Annexe%201%20-%20Pr%20sences%20en%20r%20union.pdf>

<https://www.stib-mivb.be/article.html?l=fr&_guid=80f863e4-58c9-3a10-4fb0-f14926c30950>

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 lui 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. Visual Basic 4 si ma mémoire est bonne. J’ai fait un peu de Delphi mais cela n’est pas allé très loin.

1999-2001 Institut Paul Lambin

Puis graduat en tant qu’analyste programmeur de gestion à l’Ephec.

Au niveau de ma carrière, j’ai une carrière d’indépendant complémentaire durant laquelle j’ai essentiellement fait du dépannage informatique mais j’ai également eu l’occasion de créer plusieurs programme soit pour ma propre entreprise CyberDev que ce soit un CRM pour la gestion de mes clients de mes factures de mes projets, soit quand j’étais par exemple chez Eurocities j’ai créé un petit programme en Delphi qui faisait l’inventaire de la machine – inventaire très léger : c’était le nom de la machine, les rams, CPU, disques durs - et il renvoyait ça dans une base de données en via une page en PHP pour stocker le tout dans une base de données en MySQL

J’ai fait pour une petite société un add-on sur Exchange pour la gestion des contacts.

Voilà mon expérience en tant qu’analyste-programmeur indépendant a surtout été pour supporter l’activité de dépannage.

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

Sinon à la Stib il n’y a pas si longtemps j’ai créé des outils pour la gestion des valideurs orange. Quand il y avait encore ces valideurs orange à l’entrée des métros, dans les trams et les bus ces valideurs étaient reliés en tout cas pour les métros reliés à des pc en OS2 qui centralisaient les données il y en avait une centaine à Bruxelles et donc j’ai créé une base de données avec tout ce qu’il fallait pour pouvoir me connecter à distance pour pouvoir rapatrier les données, pour vérifier qu’ils réagissaient bien de façon à gagner du temps parce que pour anecdotes les nuits de changements d’heure moi j’étais réveillé à 3h du matin pour appuyer sur certains boutons pour les arrêter et les redémarrer parce qu’il y avait un problème dans le changement d’heure. Donc heu beaucoup d’expérience par rapport à ça.

VVH : Et tu as me semble-t-il été employé dans des boîtes de consultance et via elles tu as travaillé alors pour des sociétés comme UCB. C’est bien ça ?

PT : Oui tout à fait. Quelque part aux alentours de 2012 en tant que j’étais consultant en tant qu’employé et donc je m’occupais d’un logiciel mis à disposition et fourni vendu par Symantec - Client Management Suite – et cela faisait la gestion du parc informatique. Donc en gros il y avait 10.000 pcs à travers le monde et il fallait aller vérifier que les utilisateurs pouvaient commander des logiciels et il fallait que ces logiciels s’installent de manière silencieuse à la demande - pour la gestion des licences, la gestion de outils nécessaires à leur travail et pour avoir des pcs standards. S’ils demandaient ordinateurs, ils devaient pouvoir recevoir 5 ordinateurs identiques.

Pierre Matterne : quelle raison aussi t’as incité à changer ton parcours, à changer ton orientation ?

PT : heu… les sous ? ^^

PM : ça intéressera très fort ses camarades

VVH : oh mon dieu oui, ils veulent aller travailler chez Oddo parce qu’ils auront un bon salaire… Oui mais à part ça qu’est-ce qui t’attire ? ben le salaire. D’accooooord et donc c’est toi qui te prétends être front-end en ayant écouté qu’il ne fallait pas mettre une fonte illisible rouge sur un fond bleu foncé. Va chez Odoo…

PT : perso j’ai un problème avec Odoo : c’est qu’ils sont partis d’un logiciel open-source qu’ils ont revendu – et ça je ne supporte pas. Parce que l’open source c’est la bonne volonté des gens de se mettre ensemble pour faire un truc sans gagner d’argent et ce n’est pas pour qu’il y en ait un autre qui reprenne le truc lui mette un joli papier cadeau et qui le revende. Ça je ne supporte pas.

VVH : c’est pour ça qu’on est amis !

PT : alors je sais que la situation est un petit peu plus compliqué mais le basique c’est quand même ça.

VVH : alors je regarde parce que tu as fait tellement de choses et que je te connais depuis tellement longtemps…

PT : si tu veux j’avais préparé des réponses aux questions que tu avais envoyées.

VVH : je regarde un peu pour parler de toi Patrick Tellier tu as été indépendant

PT : oui mais ça c’est après dans les questions

VVH : oui mais moi je préfère toi. Dévoile-toi : Qui es-tu Patrick Tellier, toi qui plaisais tant à ma grand-mère ^^

[Et là on n’a même plus enregistré… puis là un peu mais je ne vais pas le retranscrire, yen a en classe qui pourraient se vexer ^^]

PT : alors des conseils.

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. C’est vraiment une des choses dans ma carrière où je me dis rétrospectivement jusqu’à il n’y a pas longtemps je me dis qu’est-ce que je voudrais continuer à faire -. J’ai trois tracks en parallèle et je ne sais laquelle choisir.

La première qui est plus orientée pognon dans le sens où c’est Microsoft c’est du .Netcore sur du C# avec du ATT Framework management, des API, et des brols comme ça mais en C#, du Razor en front-end et des API en backend

Ou alors tu as le mode open-source avec du python pour les API et pour le front du Angular/JS

Et puis tu as le côté un peu historique qui est le PHP MySQL parce que c’est mes premiers amours. C’est avec ça que j’ai débuté.

Et donc j’ai ces 3 tracks là et je ne sais pas lequel des 3 choisir donc j’avance un petit peu dans chacun des 3 mais c’est un petit peu… Donc il y a un moment il ne faut pas avoir peur de prendre du temps et de prendre un cours sur UDEMI sur pluralsites sur nornonline. 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. Je veux dire Virtual Studio Code et c’est gratuit. En 2002-2003 il n’y avait pas. C’est vers 2005 je crois qu’il y a commencé à avoir une base un peu stable niveau IDE.

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

Le deuxième conseil c’est la SECURITE ! tu peux créer la plus magnifique application que tu veux, si elle n’est pas sécurisée, il ne faut surtout pas la vendre. Et aujourd’hui tout est orienté sécurité. Donc si t’essaies de vendre ton truc et qu’il y a des failles de sécurité, tu te fais rembarrer.

Il faut penser à l’informatique en mode récession. Il faut 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. 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. Et pourquoi c’est tout ce qui va autour du développement ? parce qu’en fait là je n’ai pas parlé d’une seule ligne de code mais que de ce qui va autour.

Bah de 1 parce que c’est mon boulot au day-to-day et de 2 c’est parce que aussi aujourd’hui tu vas sur Stack Overflow tu vas trouver 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 tri-bulles. Et bien tu vas trouver un tri-bulles en Pascal, en C#, C++, et tous les exemples possibles. En gros tu prends et tu fais copier/coller.

Et aussi depuis que Microsoft à racheter GIT, ils ont sorti Github Copilot, et en fait l’idée c’est que ce truc propose des exemples de codes pour toi. Tu dois décrire d’une certaine façon un peu ton algorithme et puis lui il va aller chercher dans toute sa base de données et il va te proposer du code. 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, mais on arrive on se rapproche et on va vers ça.

Donc c’est pour ça que tout le reste est important. Quand tu te pointes vers un employeur et que tu lui dis que tu sais faire du Domain Driven development que tu sais faire du CI/CD et que tu es orienté sécurité et que tu sais comment sécuriser ton API, le fait que tu ne sois pas un expert en C# dans la librairie X ou Y ce sera moins intéressant. Par contre si t’as le mindset il va être beaucoup plus intéressé par tes centres d’intérêts.

VVH : qu’est-ce que c’est CI/CD ?

PT : Continuous Integration/Continuous Deploiement. L’idée de base c’est que tu prends ton code, tu le push sur GIT. En général c’est un code compilé donc sur ton pc local tu vas faire Run dans ton IDE. Et ton CI/CD qu’est-ce qu’il va faire ? et bien au lieu que ce soit ton ordinateur qui va aller compiler c’est un serveur qui va aller détecter que tu as fait un commit en GIT. Il va prendre ton code, il va lui en faire un exe et il va faire tout une série ça peut être du test. Il va commencer si c’est un minimum bien fait tu as déjà l’audit du code.

L’audit du code c’est vérifier si tu n’as pas fait trop de copier/coller. Il y a des statistiques qui sont faites pour détecter si tu as fait beaucoup de copier/coller. Parce que si tu as fait beaucoup de copier/coller, c’est que tu as mal codé. Normalement plutôt que de copier/coller, tu dois faire une fonction qui reprend l’opération que tu veux faire et tu vas l’utiliser plusieurs fois.

Tu as des règles de bonnes pratiques concernant l’indentation, tu as des règles comme quoi ça doit être documenté dans ton code. Tu dois avoir des commentaires qui disent cette fonction fait ceci ou fait cela.

[pause]

Il y a un premier outil qui va aller prendre ton code, qui va le lire et vérifier s’il remplit certains critères. Ça peut être ben j’ai un ami thomas et lui on lui avait fait la réflexion qu’il avait du code trop long. Parce qu’en fait lui il avait mis du code directement dans son HTML et ils avaient refusé au début que ces 3 lignes étaient trop longues. Parce qu’il y a un audit qui est fait automatiquement par ces systèmes.

Donc tu as un audit au niveau de la qualité, tu as un audit niveau sécurité pour vérifier que certaines fonctions sont utilisées correctement.

Puis c’est compilé et une ça créé un exe une DLL un APK ce que tu veux. Ensuite tu vas le déployer de manière automatique sur un serveur. Souvent tu as dans les environnements les plus classiques dev test act prod.

Dev c’est toi en tant que développeur tu codes dessus. Tu crées dessus.

Ensuite quand tu estimes que le code rempli un minimum les conditions qu’il est prêt tu peux… (*rien derrière sur l’enregistrement)*

Il va prendre l’exécutable qui doit être le même dans tous les environnements il va prendre le dev et puis il va le copier en test. Mais il doit aussi adapter tous les paramètres du style connexions-string à ta base de données, des droits d’accès car tu vas utiliser des utilisateurs différends parce que sinon si tu veux te connecter à une base de données il te faut un login mais tu ne vas pas prendre le même login pour tous tes environnements, tout doit être ségréguer pour des raisons de sécurité.

Donc il y a des variables. Au début de ton code tu as login password pour la base de données. Login password pour te connecter à du Graylog à du redis à du RabbiteMQ donc différents systèmes de base de données de stockage et échanges d’informations. Et toutes ces informations-là elles varient d’un environnement à l’autre. C’est comme si je te disais j’ai 4 ordinateurs le premier je fais n’importe quoi avec, le deuxième c’est celui que je vais donner à quelqu’un pour qu’il puisse tester l’application, le troisième c’est celui où le business va vraiment tester et donner son accord avant de le copier sur la production. Et donc ce sont des ordinateurs séparés les uns des autres

Normalement c’est une boucle infinie. C’est-à-dire que tu vas analyser, coder, déployer, tester, tester, déployer, récupérer des informations de tes utilisateurs, de ton système de monitoring parce que si jamais ta base de données est sur le point d’exploser il faut voir pourquoi. Si t’as déployé le 15 et que le 16 elle a doublé de taille puis le 17 triplé de taille etc. Ensuite tu récupères ces informations tu cherches pourquoi tu recorrige et tu redéploies la version suivante, et c’est un cercle sans fin. Tu verras. Si tu cherches sur internet CI/CD tu vas voir une boucle infinie avec ces méthodes.

Et donc ça c’est une qualité de travail. Parce que quand tu paies quelqu’un 1000€ la journée en tant que consultant, tu ne veux pas qu’il perde son temps à chaque fois faire des copier/coller, à chaque fois adapter les chaînes de connexions. Tu cliques sur un bouton et ça doit être déployé d’environnement en environnement.

Je prêche un peu contre ma chapelle puisque que mon boulot c’est de faire du copier/coller, de recevoir des documents Word et d’installer dans les différents environnements. C’est une partie de mon boulot. Mais bon d’un autre côté quand tu vois l’efficacité qu’on a l’équipe de développement chez nous qui s’occupe du déploiement qui gère le département de vente, ils ont poussé les trucs assez loin avec des scripts and cibles pour migrer d’environnement en environnement avec vraiment pour garantir que les environnements auront toujours la bonne structure.

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 de jolis post-it qui représente 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 toi. Parce qu’avant d’avoir un document qui décris bien … ben ce n’est pas facile du tout. Surtout si tu ne sais pas ce qu’est une vente, c’est difficile de coder un système de vente. Si tu ne sais pas ce qu’un produit à la STIB que tu ne sais pas qu’il y a 10 sortes de tickets différents autant d’abonnements etc. les tarifs différents. Si tu ne sais pas tout ça, c’est quasiment impossible de coder une application qui gère ça.

Donc discussions avec les architectes avec les analystes pour essayer de comprendre comment tu dois coder ton truc et 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 puis tu codes …

Si tu as l’occasion vas sur [www.lesjoiesducode.com](http://www.lesjoiesducode.com) – mort de rire ^^ Au début tu ne vas pas comprendre puis tu vas comprendre et tu vas te marrer deux fois. Et des fois tu pleureras devant to ordinateur quand tu comprendras 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 bien aux demandes. Tu peux faire aussi des réunions de présentations des fonctionnalités à tes collègues, tu as les échanges d’informations avec les collègues c’est-à-dire que tu vas être bloqué tu vas demander de l’aide, il est bloqué donc il vient te demander de l’aide, et 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. Par exemple au Ministère des Pensions ils ont créé un Framework interne.

Donc tout ça c’est beaucoup de recherches. Automatiser, tester, écrire les tests, discuter avec les testeurs pour savoir ce qu’ils doivent tester et comment ils peuvent le 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 c’est énormément de communication. Pour savoir ce que tu dois coder ce que tu dois fournir.

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 bcss est bloquée on ne sait pas pourquoi. Ben… oui ok.

T’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 passait en weekend complet, on commençait le vendredi soir vers 10-11h quand ils fermaient les derniers canaux de vente les boutiques les Kioks. Puis on arrêtait tous les services on mettait tout à jour puis on commençait à tester

Donc déjà pour te donner une idée les premières fois que j’ai fait ces mises en prod Sales j’avais 24 documents Word de 5/6 pages chacun avec :

Prendre le fichier qui se trouve à tel endroit

Le recopier à tel endroit

Ouvrir le document XML

Modifier la valeur X par Y

Modifier la valeur Z par …

La première fois que j’ai fait ça ça a été une de mes pires expériences parce que j’avais 20 développeurs qui étaient en train d’attendre que je fasse des copier/coller pour eux pouvoir commencer à tester. Une fois que j’avais terminé les 200 copier/coller, que j’avais mis les fichiers qu’il fallait, on redémarre les services et là ils commencent à tester. Puis là ils se rendent compte qu’il y a des trucs qui ne marchent pas car j’ai foiré mes copier/coller. Donc tu n’en mène pas large.

Ça c’était les premiers jours. On arrêtait, on recommençait le samedi matin. Et le samedi matin c’était le Business qui commençait à tester. Donc là on rouvrait les boutiques …….

On ouvrait les premières Bootiks et les Kioks et les gens du business commençaient à teste les nouvelles fonctionnalités. Test de régression - vérification pour voir que ce qui marchait hier marche toujours aujourd’hui.

Et puis Dimanche c’était là qu’on attendait qu’il y avait beaucoup moins de choses ouvertes le dimanche donc et puis le lundi c’est là que tu te prenais la vague d’incidents parce que tout ce qui a été testé du weekend ben il y avait encore des trucs qui ne marchaient pas, ou des trucs qui ont cassés pendant la nuit.

Et donc on a remplacé ce weekend complet avec 20 développeurs, tous sur site évidement ça coutait une blinde à la STIB, par en automatisant ça bah 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 cote je rappelle qu’en informatique c’est la base c’est l’automatisation.

Donc depuis la découverte du charbon et des locomotives à vapeur, jusqu’à aujourd’hui on n’a jamais fait autre chose qu’automatiser pour avoir des km et des km2 de pomme de terre qu’il y a un péquenaud dans un tracteur qui fait le boulot de l’informaticien ce n’est rien d’autre que ça. C’est éviter qu’il y ait un guichet ouvert 24/7/an en mobilisant des dizaines voire des centaines de personnes pour 3 clients, et que on puisse au client à 2h du mat sur son pc ou son Gsm et pouvoir commander ses courses faire ses comptes taper ses impôts. Alors quand j’en discute avec des amis parce que je suis évidemment inquiet parce que fatalement a un moment il va y avoir des bons programmeurs qui vont bien faire le truc et puis entre le cloud au niveau de l’infra où ça tient à peu près la route et les programmeurs qui font à peu près bien leur boulot grosso modo on fait à peu près la même chose. 85% des sites web c’est du WordPress tu m’excuses des gens qui publient les mêmes trucs, 10% ça va être des sites de vente de n’importe quoi (Amazon etc.). Une fois que tu as enlevé d’Internet tout ça il te reste quoi ? Netflix, Disney+ tu enlèves encore ça il reste quoi ?

Donc à partir du moment où tu as WordPress et Joomla qui ont 85% du business 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. Et donc là où je suis inquiet c’est l’avenir à ce niveau-à, et puis mes amis qui me disent t’en fait pas on invente on réinvente 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 donc je ne suis pas inquiet pour Xavier je ne l’ai pas été pour mon frère

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

VVH : ça ça me parle par contre ^^ Ya pas que dans le web.

Tu sais 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, l’installer et faire en sorte que ça continue à marcher. Donc 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, s’il y a un problème avec les canaux de vente, avec le système de régulation métros, système de régulation des trams et des bus,

VVH : Et moi il me semble aussi que quand tu avais commencé à la STIB c’était comme consultant et tu avais aidé à construire te mettre en place tout ce qui concernant la nouvelle app pour smartphone

PT : alors je l’ai suffisamment décortiquée par ce que – une petite anecdote avec ça il y a une espèce de tuuuuuuuuuuut du cote de Louvain qui a décidé que comme il travaillait dans une société de marketing et que ce serait sympa de connaitre les temps d’attente des différents véhicules aux différents arrêts. Donc visiblement il savait bidouiller 2/3 petits trucs donc il a réussi à saturer complètement nos serveurs. Espèce de tuuuuuuuuuuut

Donc on s’est retrouve sur Facebook avec des commentaires comme « ouais à la STIB ils ne savent pas faire un site avec des temps d’attente » etc. moi je me bats en interne en disant « les gars il y a un problème. On me dit non. Si il y a un problème. On me dit non. Je dis si il y a un problème. Ils m’ont montré les statistiques Google il n’y a pas de problème on reçoit autant de requête qu’avant. Je vais demander à la sécurité. Je dis on est victime d’une attaque. On me dit que non on a regardé dans les logs des firewalls. Je vais demander au développeur. Le développeur me dit ma base de données est saturée mais pfff je ne sais pas heiiiiin moi je suis le développeur. (humm agacé de Patrick).

Et puis il y a un moment où j’ai été chercher dans les logs des serveurs mêmes. Alors ce qu’il faut savoir ce qu’en réseau il y a ce qu’on appelle un load balancer. Un load balancer c’est un truc tu mets 4 serveurs tu le mets devant il repartit la charge entre les 4. Il reçoit 4 demandes il envoie une demande à chaque. Du coup au niveau sécurité c’est bien au niveau stabilité c’est bien mais quand tu dois trouver pourquoi il y a un problème c’est nul. Parce que quand tu dois essayer de comprendre d’où viennent les demandes ben les logs des demandes que tu reçois c’est depuis le load balancer. C’est là que j’ai utilisé un agrégateur de logs et un analyseur de log et là j’ai pu comparer ce qu’il y avait quand ça marchait bien et après quand ça marchait plus. Et là on voit x10 des appels au webservices. Donc rebelotte je prends mon petit bâton de pèlerin je vais retourner voir le responsable qui m’a dit que selon Google il n’y a pas eu d’augmentation des appels. Là je lui prouve par A+B qu’il avait tort, il me dit ah bon donc on tombe à peu près d’accord que ces statistiques ne sont pas parfaites… là on remonte voir le gars de la sécurité qui m’avait dit qu’on n’était pas victime d’une attaque. Je lui demande d’aller voir dans le firewall d’aller voir les connexions et là oooooooooooh c’est bizarre c’est toujours la même adresse IP qui nous appelle. 50x/seconde. Et qui fait toutes les demandes pour tous les lignes pour tous les arrêts (rehummm très agacé de Patrick).

VVH : ce n’est pas une attaque çaaaaaaa

PT : nan ce n’est pas du coup de force pas du tout ce n’est pas du déni de service.

Donc c’est assez rare que je sois directif mais j’ai dit au gars de la sécurité tu bloques l’IP. Il m’a dit je ne sais pas si… TU BLOQUES L’IP. Il a finalement fait et là bizarrement ça allait mieux. On a retracé l’IP quelque part du côté de LLN. En plus le gars faisait depuis son entreprise t’avait le site internet on avait retrouvé les records DNS concernant le serveur SNTP. On est +/- remonté à la personne mais ce qu’il a fait n’était pas illégal.

Donc voilà ça c’est du reverse engineering. Et comme je te l’ai dit c’est de la communication et pour l’anecdote on est allé on a remonté aussi grâce au firewall et là on a vu qu’en fait il appelait les pages directement du back-end. Donc en gros tu as ta page de présentation et c’est celle-là sur laquelle ils avaient mis les statistiques Google. Mais lui allait attaquer les pages de backend derrière donc il n’y avait pas de statistique Google. Ça explique pourquoi quand je suis allé voir le responsable du site il me disait ben non on a tout à fait tout le temps le même nombre de trafic qu’avant.

VVH : on avait eu la même chose chez Proximus quand il avait lancé les nouveaux webservices MyProximus dans l’heure du lancement ça avait été pareil. Et c’est donc à ce moment-là que j’ai découvert qu’on pouvait aller « par derrière ». c’est à ce moment précis que j’ai commencé à découvrir une autre facette du web. Parce que j’étais pas du tout là-dedans.

PT : on le découvre souvent je ne vais pas dire quand c’est trop tard mais quand il y a un problème. C’est là que tu commences à chercher.

PM : on est en train de faire des formulaires en ce moment et ce que tu viens de dire résonne tout particulièrement. On a trop de requêtes API pour nos formulaires. On a une requête par champ. Et quand on a une dizaine de champs on a une dizaine de requêtes. Quand o a une dizaine de formulaires, on a une centaine de requêtes. Et quand on a une autre entreprise qui fait aussi ses tests pare qu’elle est en train de developper un outil, nos API ne suivent pas

PT : d’où Kubernetes et consorts qui ont pour but justement d’augmenter ton nombre de serveurs backend pour permettre la charge mais il faut que tout soit ok parce que c’est bien beau d’avoir de API supplémentaires mais si ta base de données ne suit pas non plus ça ne fait jamais que déplacer le problème.

Mais il y a de nouveaux outils qui permettent de résoudre ça mais c’est cher.

[pause ? je pense]

Mais ouai la première app elle était bien elle était simple c’était un APK pour Android qui allait lire un fichier sur une API et qui affichait les temps d’attente. C’était bien (en aigu de Patrick) c’était simple. Bon évidemment le problème c’est que niveau architecture ils avaient tapé ça dans une base de données Oracle. Les temps d’attente, qui changent toutes les secondes, qui ont une durée de vie d’une minute – dont tu te fiches royalement de savoir ce que hier avant-hier avant-avant-hier ou même il y a 3 minutes, c’était. Mais plutôt que d’avoir une In-Memory-Data base Ben ils ont été tapé ça dans du Oracle. C’était la version précédente. La nouvelle ils ont fait du in-memory-database et puis voilà on shut-down le serveur on perd les données mais on s’en fout puisque de toute façon dans une minute les données seront de nouveau bonnes.

VVH : - anecdote sur les tickets à acheter en ligne. 36minutes et 34secondes

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 tu vas être contacté à n’importe quelle heure du jour ou de la nuit, ils vont tous t’appeler quand ça les arrange eux. Le niveau de vie est plus élevé et tu paies des gadgets. Donc si tu veux t’acheter une grosse voiture, des GSMs tous les mois une montre connectée bah c’est très bien. Si tu veux t’acheter une maison c’est une catastrophe.

PT : élément que je modifierais : ne pas avoir peur de perdre du temps, rester focalisé sur une seule chose et la terminer. Parce que le problème c’est que c’est moins utile de savoir voir quelque chose en 4 langages différents que de connaitre un langage à fond. Je peux te programmer à peu près la même chose en python, en C#, en Java et en PHP mais ça ne sert à rien. Enfin ça sert mais ce n’est pas très utile de savoir le faire en 4 ou 5 langages. Par contre de savoir faire quelque chose de A à Z et de dire je sais le faire comme ça,

Ne pas avoir peur de demander de l’aide. Ça c’est parmi les choses – avoir peur de passer pour un imbécile et/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.

Est-ce que c’était différent avant ? oui. Pour moi c’était beaucoup chacun pour soi et on parle en 2000 hein. Tu n’avais pas Stackoverflow avec des exemples de code, et l’open-source ou tous les développeurs partagent leurs connaissances. Donc avant beaucoup de chacun pour soi et maintenant beaucoup plus de partage de connaissance et d’aide.

Ce qui m’anime ? l’aspect créativité mais surtout l’aspect gain de temps et de pouvoir se concentrer sur les vrais problèmes. Ça c’était vraiment – c’est mon cote vieux con – quand on me dit « c’est une révolution » parce qu’on a des iPhones on a des téléphones un avec peu plus de fonctionnalités qu’avant ce que je réponds c’est « tu prends le plus cher des iPhones, tu désactives le wifi la 4G et le GPS – ça te sert à quoi ? plus rien ». tu prends un pc qui a 12 ans avec un Linux mais tu le connectes avec un récepteur GPS et de la 3G tu as accès à des ressources quasi illimitées. Donc je suis plus orienté sur la fonctionnalité. Et donc le fait qu’après tu gagnes du temps quoi. T’as le GPS ça t’évites de faire des kilomètres inutiles de consommer de l’essence.

Et l’aide aux utilisateurs finaux, parce que le but c’est ça d’avoir des clients qui… j’avais un fait logiciel pour une client avocate pour qu’elle enregistre sa facturation. Donc elle encode au fur et a mesure ses prestations et puis elle appuie sur un bouton et la facture sort avec un numéro de facture unique et des choses comme ça et donc ça lui fait gagner du temps. Du temps qu’elle peut utiliser soit pour ses passions soit pour travailler et gagner des sous.

C’est surtout ça qui m’animes

Tournant dans ma carrière ? bah début 2000. Quand tu es entouré de mauvais d développeurs qui te disent que le problème vient d’ailleurs que chez eux. A sa décharge ne venait pas de son code, le problème venait 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 ’99-2000 ce n’est pas là ou 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 1 qui s’occupait de tout. Donc tu ne commençais pas à dire oui mais non mais bon.

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. T’as un problème ce n’est pas venir te taper sur la tête – c’est venir te voir et te demander ce qu’on 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. Pour te faire virer de la STIB faut vraiment faire des grosses conneries. Donc le but c’est de rester là. Ce n’est pas la peine d’essayer de te faire virer. Même si tu as fait n’importe quoi tu seras s pas viré. Par contre faut qu’on règle le problème. Donc maintenant comment on fait pour régler le problème qui a été créé ? on fait des backups, on rachète des licences, on met quelqu’un à la porte MAIS pour tenir la porte. Donc orienté solution. Le tournant de ma carrière c’est beaucoup ça. Au tout début j’ai des très mauvais souvenirs d’un développeur Delphi…

Les soft-Skills : l’informatique c’est un milliard de choses simples qui ont tous un rapport les uns 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. Faut être paresseux. Parce qu’il ne faut pas faire les choses deux fois – c’est pour ça que tu automatise. 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 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.

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 tout le restant tu le remettras en route. Il est évident que si tu travailles dans certains domaines – quand je travaillais pour UCB tu avais des machines qui ne pouvaient pas être éteintes qui ne pouvaient pas être redémarrées parce qu’elles étaient reliées à un thermomètre et elles enregistraient en permanence les températures de vaccins. Ils ont eu des cas ù l’ordinateur a été éteint comme ils ne pouvaient plus prouver que la température n’avait pas variée et ben ils ont dû jeter tout jeter. Donc 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 e 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 e dit fait comme ça que tu dis 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 super le vent en poupe et c’est super important parce que tu te mets dans la peau de l’utilisateur final et quand tu commences – t’as pas encore écrit une ligne de code et tu décris ce que tu veux faire et comment tu veux que ça marche. Et ensuite tu commences à coder ce que tu as écris que tu allais tester. Mais il ne faut pas que ce soit toi qui testes ce que tu as écris. Mais l’aspect testing est super important parce que tu peux avoir…

Alors là j’ai 10paragraphes : est-ce que je suis passionné par ce que je fais ?

VVH : non, ça s’entend…

PT : la question était est-ce qu’il faut être passionné et le faire en permanence et se tenir au courant. Pour moi j’ai cette idée en tête que depuis 2000-2001 (tu avais l’ADSL) 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. Sur ton téléphone, 85% des sites ce sont des gens qui publient des informations et le reste ce sont quasi que des magasins. Ce n’est rien d’extraordinaire. On refait la même chose. Donc ce n’est pas que tu dois t’imaginer que tu dois devoir tout le temps relire et réinventer. Enfin si justement c’est fatiguant (pause qq minutes. Il va chercher un livre de 2001-2002 – un gros pavé Framework.net). tout ce dont je t’ai parlé et que tu vas apprendre se trouve déjà là. Maintenant tu vas l’apprendre en .NetCore au lieu de .Net mais ce sont les mêmes fonctionnalités. Tu vas des informations dans une base de données structurée, tu vas l’afficher pour un utilisateur. Ça n’a pas changé depuis 20 ans. Donc… ok il y a une série de fonctionnalité et l’IA. Même la reconnaissance de caractère les OCR tu mettais un document tu le scannais tu le faisais passer un dans un scanner il te le ressortait. Alors ok la qualité du document a changée mais les fondamentaux sont toujours là. Donc faut-il passer ses journées à se former, vraiment pas. Par exemple quand tu regardes les différences entre NetCore 3 5 et 7 (à vérifier les versions) en gros dans les réalisé notes ont te dit on a augmenté la vitesse, on a résolu des bugs. Hein quoi des fonctionnalités ? vous voulez des fonctionnalités ? ah on a remis des trucs qui existait auparavant. Parce que si tu veux le .netcore est arrivé en parallèle du .netFramework mais il n’avait pas tout ce qui est windowsform. Donc dans la version 5 ils ont rajoutés que tu pouvais ravoir des choses qui existait il y a 10 ans.

Alors ok ton GSM il y a l’aspect miniaturisation mais sinon… ce n’est pas une révolution.

(petite aparté)

PT : important : beaucoup de société on simplement automatisé des processus internes et les ont publiés sur internet. Sauf que ça a montré que les processus qui étaient bancals en interne se sont retrouvés bancals pour les clients.

En 2002, vous aviez déjà des OCR, reconnaissance vocale, films, photos, musiques, livres, forums (IRC ou phpbb), moyen de communication (Skype, icq) 20 ans plus tard, les fondamentaux sont toujours là et n'ont pas changé. Jeux 3D, simulateurs d'avions, voitures… tout existe depuis fin 1990. On a amélioré les choses, c'est plus joli, plus rapide, "moins cher" Les jeux vidéo en 3d existaient déjà il y a 20 ans – c’était beaucoup moins bien on est d’accord, mais c’était déjà là et ça n’a en réalité pas tellement changé. C’est juste que c’est mieux fini maintenant.

Après, si vous voulez vous adresser à une large population pour faire de l'argent, alors là, il faut toute la machine de guerre marketing et front end. Et attention que Prenez deux applications, une avec un interface dos de 1995, l'autre une magnifique interface web réactive. Si les fonctionnalités ne sont pas là, ça ne sert à rien. Les gens iront quand même sur le DOS.

Les méthodologies ont pas mal changé. Donc avant c’était de l’investissement on achète un logiciel on essaie de le rentabiliser autant que possible. Maintenant on commence avec des fonctionnalités de base et on en rajoute au fur et a mesure. D’où le principe de location. Pour anecdote, les vendeurs informatiques on un point commun avec les dealers : c’est qu’ils considèrent leurs clients comme des drogués. Parce que tu vas toujours vouloir la dernière version du produit.

VVH : et tu aimes toujours le développement ?

PT : oui mais c’est parce que je n’en fais pas trop dans mon boulot ^^ C’est

VVH : mais tu fais quand même du développement le soir pour ton plaisir ?

PT : ouais là je suis plutôt sur du python et de l’Angular parce que j’aime bien ce cote libre – dans le sens ou ce n’est pas Microsoft qui se lève un matin en se disant on va faire plus de fric sur ceci ou cela. Parce que le nombre de décisions qu’ils ont prit pour SharePoint ou … parce que ce n’était pas rentable, et si toi tu as investi du temps dedans… alors oui t’as appris une technologie et la logique est là, mais bon

VVH : toute dernière question : qu’est-ce que tu me conseillerais à MOI. On se connait depuis plus de 20 ans. Tu sais comment je suis. Tu sais que 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 je puisse alors garder mon code pour le réutiliser plus tard

PT : les conseils d’avant sont toujours d’actualité.

PM : c’est penser à ton toi futur. Quand tu vas écrire ton code et que tu devras le relire dans 6 mois et te rendre service en l’écrivant bien.

PT : 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 d jour de 8h a 22h, ben on m’appelait tout le temps. Moi ce qui m’a manqué c’était d’avoir du temps a moi. De me dire ok il est 9h du mat, j’ai bien dormi hé ben maintenant je vais me concentrer sur une problématique, un programme. Et arriver à décortiquer un truc chaque petite chose ça peut être pour beaucoup de gens. C’est arriver à 8h du matin et couper mon téléphone pour pouvoir me concentrer. décortiquer à ton rythme sous l’aspect utilisateur, sous l’aspect sécurité, sous l’aspect développement. Si tu avances petit à petit ….

Faut pouvoir se concentrer sur des choses minuscules une couleur – parce que le bleu c’e n’est pas du bleu. Ce n’est pas idiot de passer 2jours à choisir 2 couleurs qui vont bien. 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 la que tu verras les experts. Ce sont eux qui auront tout décortiquer en tout petit.

PM : j’ai aussi entendu un bon avocat – contre les commentaires parce que ceux-ci sont souvent inutiles, souvent mal rédigés, et si le code est rédigé de façon lisible tu n’en as pas besoin. Alors j’aimerais savoir ton avis à toi sur les commentaires parce que moi j’ai l’habitude dans mon JavaScript de découper tout en les plus petites fonctions possibles et j’ai une mer de petites fonctions et il n’y a pas besoins de commentaires pour expliquer parce que c’est redondant avec la fonction elle-même. Mais dans ton expérience à toi ?

PT : si tu es dans du JavaScript tu vas retrouver avec du code que tu peux lire mais si tu as des éléments que tu vas-tu vas utiliser des méthodes fournies pas quelqu’un d’autre tu vas avoir accès à son code donc tu vas forcément avoir besoin d’une documentation. Donc la documentation si elle est dans les commentaires elle est directement liée au code. Donc c’est au plus proche. Apres si je pouvais si tu as comme règle de documenter tout ce que tu fais et de le mettre dans un Java Doc qui va lire et généré un code HTML que tes collègues pourront lire, alors ouais. Et si tu tombes sur des gens qui commentent mal alors autant ne rien mettre, ne commente pas. Ce sont des choses qui semblent faire perdre du temps et être inutile mais quand c’est bien fait ça fait gagner un temps considérable et ça facilite la vie de chacun.