Mardi 25 février 2025

Hier, Quentin et moi avons terminé l’installation tardivement dû aux différentes installations qui nous ont retardé tout au long de la journée. Ainsi, Quentin n’a pas eu le temps de couvrir les explications du fonctionnement de l’application de la fonction, du code, ni du déploiement dans Azure. Initialement, le travail de Quentin devait être reparti en 2 demi-journées, une consacrée au déploiement et une autre consacré aux explications, suivi et transfert des connaissances. Bien qu’il faisait partie des responsabilités d’Expertime d’expliciter les différentes installations nécessaires et de s’assurer que nous soyons en mesure de pouvoir les effectuer à temps, il était potentiellement notre responsabilité de leur faire part de l’importance que Gecina et sa DSI porte quant à la sécurité et aux limites informatiques auxquelles nous faisons face lors d’installation d’applications, software ou framework.

Ainsi, ce matin, Quentin m’a dédié une demi-journée afin de pouvoir couvrir les éléments mentionnés. Néanmoins, une demi-journée est loin (très loin) d’être suffisant pour correctement comprendre la plateforme Azure, qui a elle toute seule représente potentiellement 70 à 80% du métier de Devops.

Après avoir terminé avec Quentin, je suis allé voir Jérôme pour lui faire part de notre avancée, et Jérôme m’a expliqué des aspects intéressants : La dette fonctionnelle et la dette technologique.

La dette fonctionnelle en informatique consiste à mettre en production un POC, bien que certains aspects ne soient pas dans leur format final, généralement dû à une échéance serrée. L’exemple à donner dans cette situation est celle des installations lorsque que Quentin et moi étions en train de déployer la fonction dans Azure :

La raison pour laquelle nous devions faire ces installations est dû au fait que Quentin, sous les recommandations (ordres) de Kevin, a suivi les dernières best-practices d’Azure. Ces best-practices sont les suivantes : créer un github organisation avec un compte admin de service afin de sauvegarder le code dans le cloud. De cette manière, nous n’auront pas à enregistrer le code dans une librairies où un endroit dédié, Github est à lui-même la best-practice de l’enregistrement et diffusion de code. Aussi, Github permet de versionner le code, cela signifie qu’il est possible de push des changements du code, qui vont créer une nouvelle version, ou de pull, c’est-à-dire tirer du code depuis une librairie Github. Ensuite, une fois le code pulled, il fallait installer VS Code afin d’avoir une interface à travers laquelle on peut interagir avec le code. Différentes installations sont nécessaires pour pouvoir faire passerelle et déployer le code. Qu’est ce que ca veut dire ? Une fois que l’on a pull le code depuis github sur lequel il est enregistré, il faut ensuite utiliser vs code avec un ensemble de fonctionnalité afin de pouvoir exécuter une commande et pouvoir déployer le code en « one-shot » sur Azure. Comme mentionné plus tôt, Azure est une plateforme très sophistiquée, et quand Kevin a fait part à Quentin d’utiliser les best practices, c’était pour justement assurer un déploiement fluide, qui avec une seule commande puisse être couvert à 70%.

Ici, la dette fonctionnelle pourrait se présenter de la manière suivante :   
  
Nous avons une échéance serrée et nous devons assurer le déploiement -> le déploiement a été designé pour suivre les best-practices -> ces best-practices nécessitent des installations qui n’avaient pas été mentionnées dans les prérequis -> nous perdons du temps sur le déploiement -> nous compromettons la livraison du projet et ne respectons pas l’échéance

* Solution : faire le déploiement sans les best-practices mais l’assurer -> plus rigoureux mais fait.

La dette technique est plus simple à expliquer. Si nous prenons la DSI de Gecina en exemple, nous n’avons pas d’équipe de développeur dont le métier est de développer h24/7. Grâce à notre capacité financière, nous pouvons nous permettre d’engager des solutions externes à chaque fois que nous nécessitons de coder. Problématique (illustrée dans notre cas d’usages) : la technologie employée par la solution tierce peut nous être inconnu, difficile à comprendre ou à maîtriser. Ainsi, nous sommes dépendant de la solution tierce pour pouvoir agir sans prendre de risque.

Ainsi pour le déploiement de la fonction dans Azure, nous avons frôler la dette fonctionnelle et nous sommes actuellement en dette technique. Cela signifie que j’ai pris un risque sans m’en rendre compte en passant du temps à débloquer les installations pour suivre les best-practices et assurer le déploiement tel qu’il avait été prévu, et que nous sommes actuellement en dette technique car Azure est une plateforme qui nécessite une expertise que je n’ai pour l’instant pas.

Dans la continuité de la journée, je devais couvrir la 2ème partie du projet avec Expertime en intégrant la fonction déployer sur Azure dans le flow power automate. J’ai collaboré avec Maxime qui m’a guidé à travers le remplacement de brique, etc…

Je n’étais pas sans surprise lorsque j’ai fait face à la quantité de document que nous avions reçu depuis que le flow power automate avait était mis en OFF. Lorsque Maxime et moi avons réactivé les flows afin de couvrir un premier test, 150 documents se sont automatiquement mis en file d’attente d’Océrisation, ce qui m’a inquiété. Nous avons réussi à annuler le flow, changer des métadonnées de nombreux documents afin de les passer comment étant analysés afin qu’ils ne soient pas océrisés, et en nous avons fait de même pour les fichiers textes (une fois l’océrisation effectuée) afin de s’assurer qu’il n’y a pas 150 documents dans la synthèse de test.

En fin de journée, j’ai fait un topo à Jérôme, qui s’apprêtait à partir 2 jours en congés. Je lui ai fais part du fait que j’avais couvert avec Expertime, à savoir : le déploiement lundi, le transfert des connaissances mardi matin, l’intégration de la fonction dans le power automate mardi après-midi. Je lui ai fais part du fait que Maxime m’avait fait des recommandations intéressantes pour une V2, et il me demandait s’il devait nous fournir une documentation. Après avoir vérifier avec Jérôme le devis initialement envoyé par Expertime, la documentation du déploiement et de l’intégration faisaient toutes 2 parties des livrables. Je me suis donc permis de revenir vers Maxime pour lui en faire part, et après qu’il ait partagé sur le groupe teams surlequel se trouve l’ensemble des collaborateurs expertime avec qui gecina travaille sur ce projet. Parallèlement j’avais envoyé un message à Quentin pour lui demander les transcriptions des 2 réunions que nous avions partagées lui et moi (le déploiement et le transfert de connaissance). Après le message de Maxime, Quentin a lui aussi fait part du fait qu’il nous fournirait une documentation. Etant donné que Quentin avait passé une demi-journée de plus sur le projet qu’il n’était censé, Jérôme et moi étions d’accord pour ne pas lui demander la documentation pour éviter qu’il ne revoie le devis et nous facturent la demi-journée supplémentaire. Or, je me suis rendu compte que concernant l’intégration, 2 demi-journées y étaient consacrées, ce qui n’a pas été le cas (nous avons littéralement passé 1h dessus). Ainsi, il m’a semblé normal de demander la documentation car l’équilibre des demi-journée avait été effectué entre les parties de la mission.