

Cartographier une organisation avec Org Topologies & se servir de la stratégie Kanban pour augmenter son agilité business

José Coignard
Coach Agile/Produit & ProKanban Trainer

Sur la base de Org Topologies (<https://www.orgtopologies.com/>) de Alexey Krivitsky
et Roland Flemm

et

de la stratégie Kanban de ProKanban (<https://prokanban.org/>)

Introduction

Toute organisation peut être analysée via la cartographie proposée par Org Topologies.

Sur cette cartographie le saint Graal se trouve en haut à droite, représentant une organisation capable de s'adapter ultra rapidement pour faire face à tous les périls semés sur son chemin. Une organisation par ailleurs en capacité de délivrer rapidement et avec une grande efficacité de la valeur à ses clients.

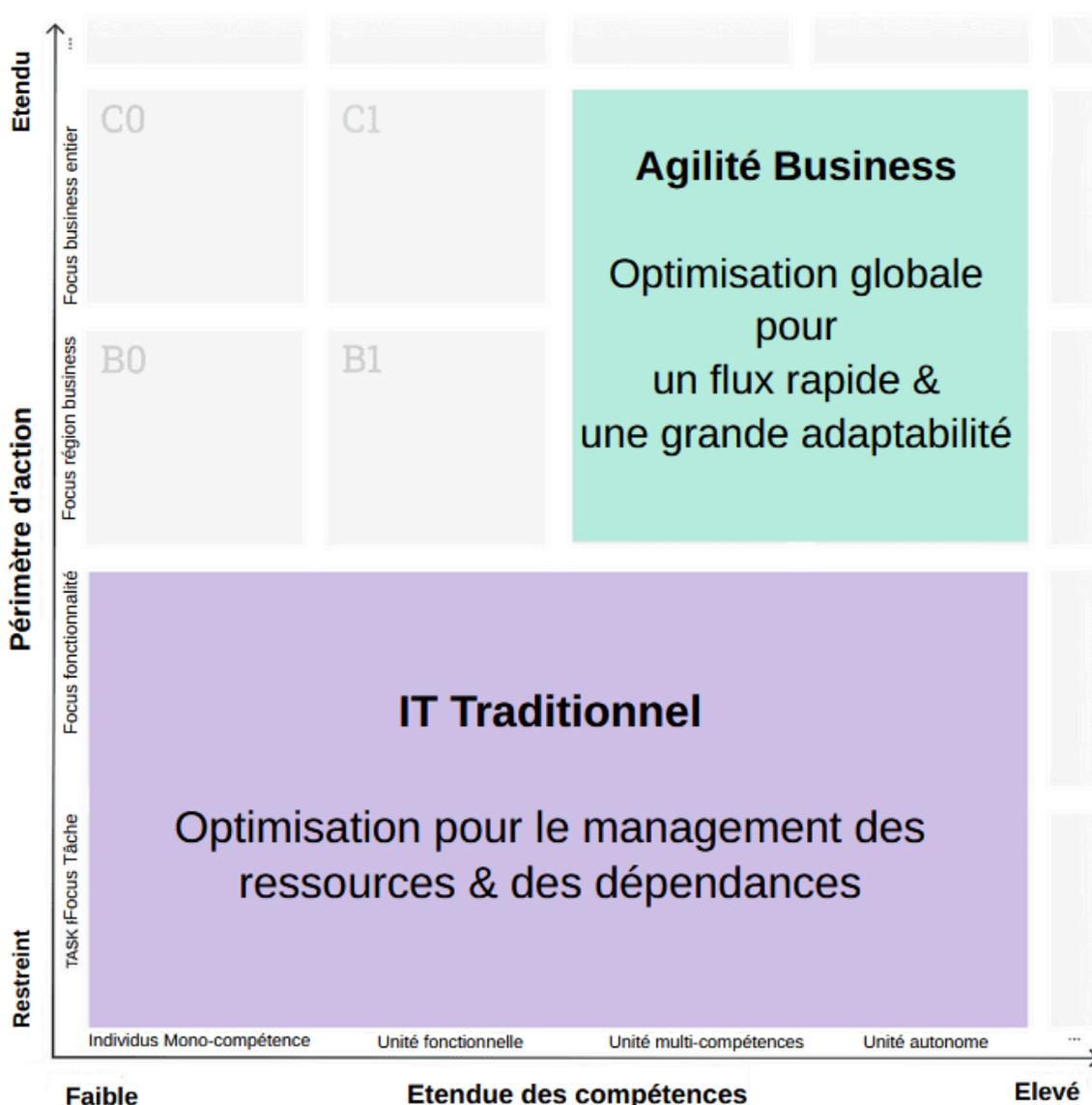


Figure 1 : La cartographie d'org topologies

Les startups se situent généralement dans cette zone et si ce n'est pas le cas, elles ne subsistent pas très longtemps dans ce monde de fortes compétitions, où il est crucial d'être à un très haut niveau de performance. Un niveau où la consommation

du peu de moyens disponibles permet de rapidement trouver le produit que des premiers clients adoreront et qui permettra de continuer l'aventure.

Lorsque la petite organisation s'agrandit, la tendance générale est à la segmentation des responsabilités. Des départements spécialisés dans différents domaines sont créés (marketing, commercial, après-vente etc.), entraînant une baisse de capacité d'adaptation et une baisse de capacité à délivrer rapidement et à moindre coût de la valeur.

L'accroissement d'une organisation tend à l'amener vers la zone basse à gauche, où les équipes et les individus la composant n'ont plus une vision holistique de ce qui est le mieux à faire pour la clientèle actuelle et pour continuer à prospérer en atteignant de nouveaux marchés par exemple.

Dans cet article, nous allons suivre l'histoire de Paul, un consultant spécialisé dans les tests d'application numérique au début de sa carrière. Paul rejoint lors d'une mission une organisation avec de multiples départements, des équipes et individus dispersés, mais qui doivent se coordonner pour in fine servir la clientèle de l'entreprise.

Dans cette histoire vous verrez comment cette organisation, et plus précisément la business unit que rejoint Paul, entreprend un chemin pour retrouver le saint-Graal, notamment en s'appuyant sur la stratégie Kanban.

Chapitre 1 : Le commencement

L'unité business que Paul rejoint, délivrant des services numériques à ses clients est un collectif d'environ 50 personnes. Ces 50 personnes couvrent l'ensemble des compétences permettant de délivrer à une partie des clients de l'entreprise un produit numérique (Business, IT, Marketing, Commercial, Hotline etc.).

Cette unité business se situe à l'intérieur d'une entreprise plus large servant d'autres produits et services, pour un panel plus large de clients.

En utilisant les 4 strates verticales de la cartographie Org Topologies (C, B, A ,Y), cela donnerait cette forme de représentation :

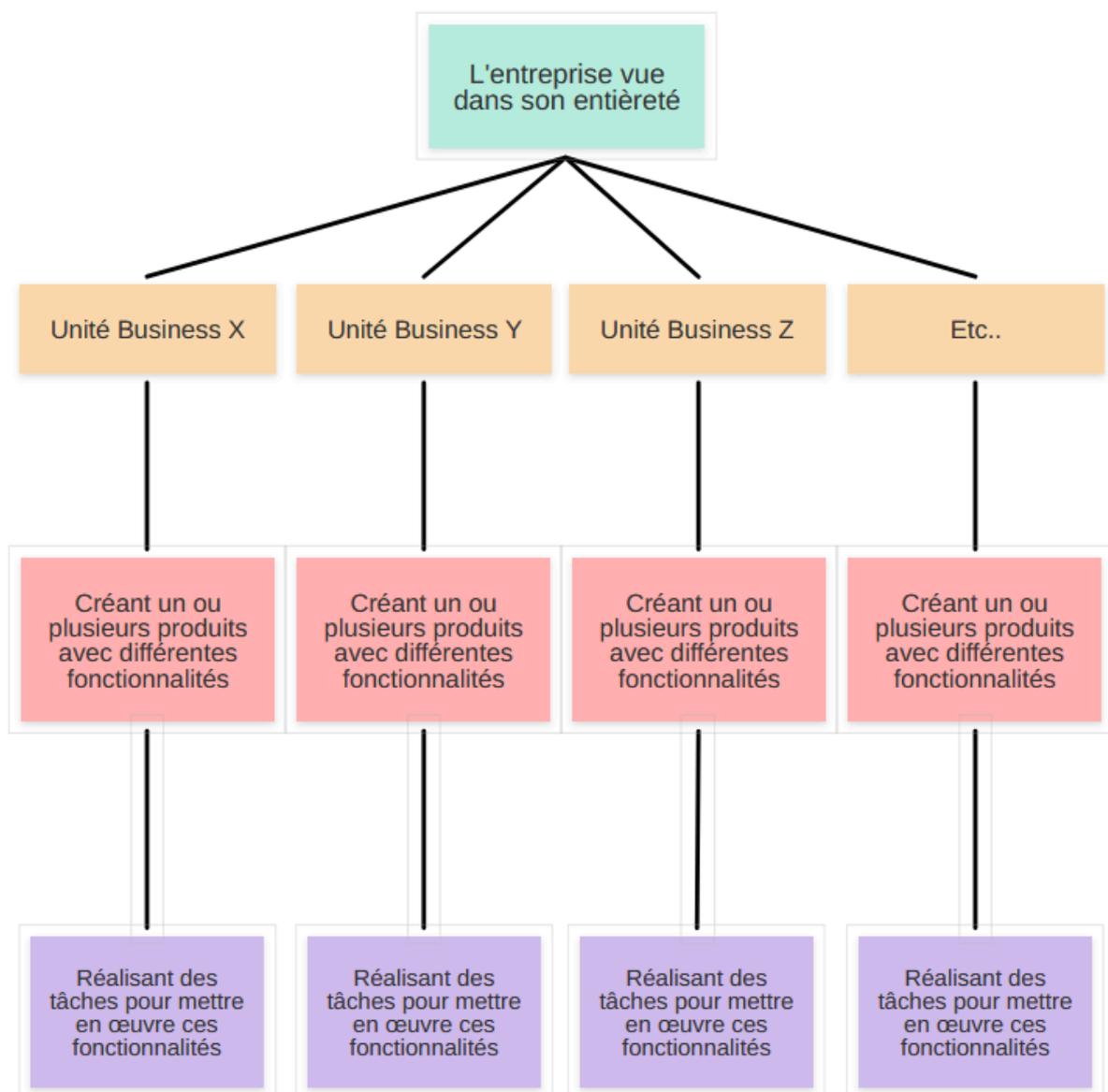


Figure 2 : L'entreprise vue dans son ensemble selon la hiérarchie verticale d'org topologies

L'histoire qui va être racontée tout au long de ces différents chapitres, fait le focus sur l'unité business X que Paul rejoint.

La situation organisationnelle de cette unité business au moment où Paul rejoint cette entreprise est représentée sur la cartographie org topologies suivante :

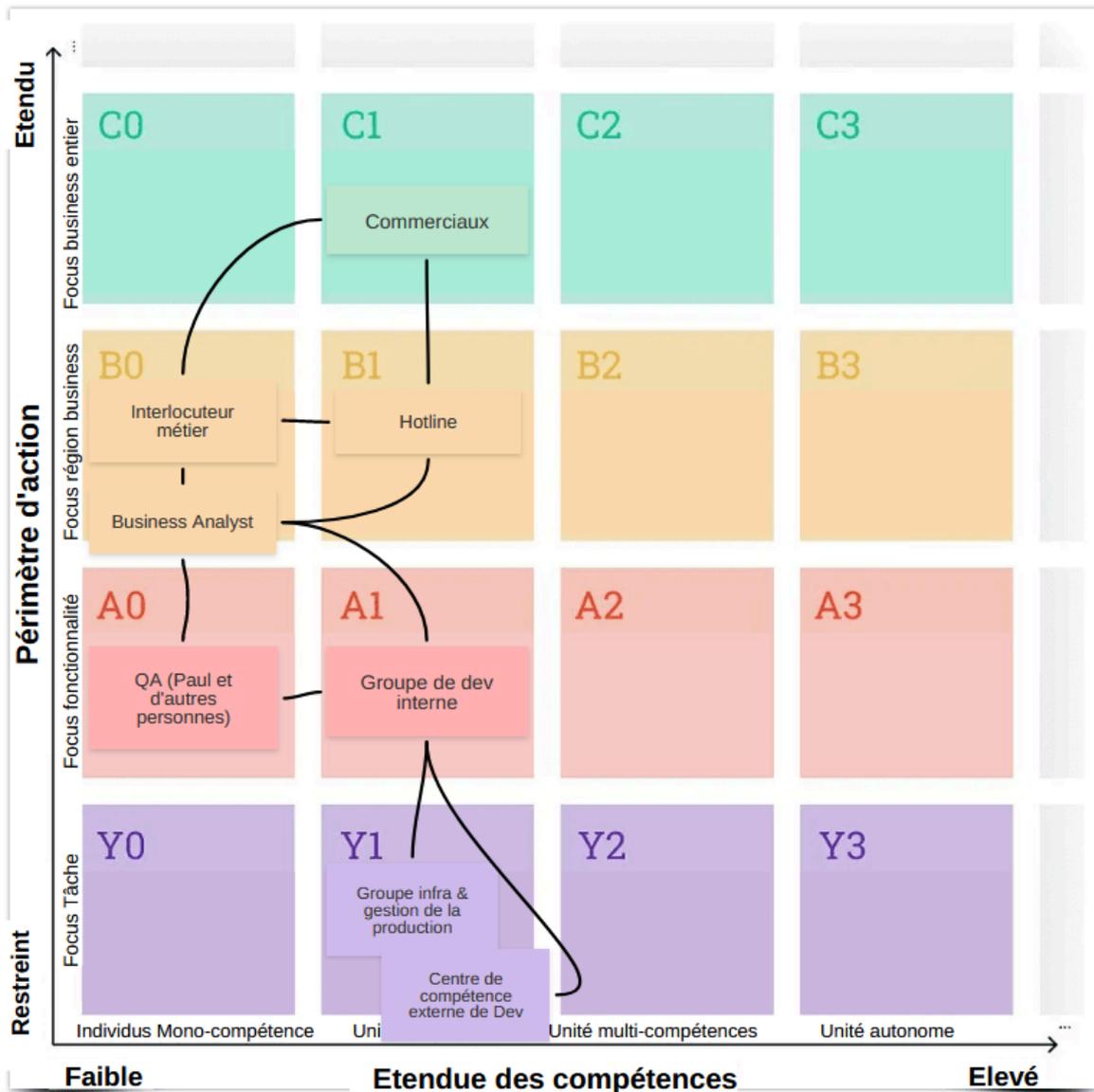


Figure 3 : Représentation de l'unité business que rejoint Paul avec la cartographie d'org topologies

Les liens entre les individus ou équipes (pseudo équipe) sont représentés par les connecteurs.

Le silotage au sein de cette organisation est très présent, jamais un membre du groupe de commerciaux (C1) pourtant au plus proche des clients n'échange avec le groupe développant le produit (A1).

Pour faire l'interface entre les commerciaux occupés sur le terrain à vendre les différents produits et services que l'entreprise propose et les personnes développant le produit, l'organisation a choisi de placer des interlocuteurs métier (B0), plusieurs, chacun avec son périmètre (chef de projet marketing pour les clients de type X, un autre pour les clients de type Y, etc.). Il n'y a pas une unité cohérente, partageant la connaissance, mais X interlocuteurs distincts avec sa connaissance business dédiée. Ces personnes se retrouvent sans une vision globale orientée client, mais une vision partielle centrée sur leur domaine business, leur domaine de clientèle. La hotline (B1) a une connaissance plus large, plus partagée, entre ses membres qui font face à tout type de clients et divers problèmes remontés, mais elle répond et débloque des situations au cas par cas sans avoir le pouvoir de décision et la capacité d'action pour apporter des changements pérennes bénéfiques aux clients.

Dans cette situation, beaucoup de coordination nécessaire entre ces différents acteurs et inévitablement de la déperdition d'informations utiles pour prendre les bonnes décisions pour le business et rendre au mieux service aux clients.

Ces personnes n'ont pas les compétences pour maintenir et faire évoluer le produit. Ils vont se reposer sur une filière informatique, généralement découpée en deux parties MOA et MOE, Business Analyst et Codeur.

C'est le cas ici, un groupe de personnes avec des compétences liant le business à l'IT vont analyser les problématiques et les besoins remontés par la Hotline ou par les interlocuteurs métiers. Il n'est pas rare non plus que ce groupe consiste en des personnes très spécialisées sur des domaines business, mais que tout le monde ne puisse adresser les sujets des différentes clientèles. L'organisation présentée ici est dans ce cas avec ses Business Analyst (B0) spécialisés avec des domaines de prédilection.

Paul est missionné et rejoint un groupe spécialisé dans les tests (A0). Ce groupe est ici en renfort pour libérer de la bande passant aux BA afin qu'ils se consacrent à la compréhension des besoins, des problématiques et gèrent les "gros" projets en tant que chef de projet SI. Un groupe de testeurs avec des missions souvent spécifiques autour de batteries de tests de non régression.

Un rythme effréné de tests laissant aucune place au partage de connaissances, amenant les personnes à se spécialiser sur des parties de l'application.

Paul a bien du mal à entrer dans sa mission, ses collègues étant très peu disponibles pour l'aider à comprendre le fonctionnement de l'application et toutes les subtilités. Heureusement, beaucoup de documentation existe et il arrive au bout de quelques semaines à prendre ses marques.

Au regard des évolutions à venir, un chef de projet SI lui demande de se focaliser sur une partie de l'application.

Évidemment l'organisation s'est dotée de développeurs informatiques (le produit numérique ne va pas se construire par magie !), ces développeurs forment une unité souvent eux-mêmes silotés en domaine de compétence spécifique (A1). C'est le cas dans cette organisation que rejoint Paul avec rarement du partage de connaissance, du pair-programming inexistant. Une unité où chacun selon sa spécialisation sur l'application prend des morceaux de spécifications écrites par les BA afin de coder l'évolution, la correction.

L'organisation a choisi de faire appel à un centre de compétence technique externe (Y1) afin de gérer la fluctuation des besoins de compétences en développement informatique. Cette entreprise est dans une stratégie de contrôle de l'accroissement de la masse salariale sur ces compétences techniques, craignant une baisse d'activité autour de ces compétences dans le futur et ne voulant pas avoir des dizaines de collaborateurs auxquels elle n'aurait plus d'activités à confier. Entre contrat mal ficelé et contraintes autour de la sécurité des SI, ce centre de compétence se retrouve à réaliser des tâches plutôt correctives, qu'évolutives et à déposer des livrables qui doivent être ensuite intégrés par l'équipe de développement interne.

Enfin pour livrer l'application en production, réaliser le monitoring technique et assurer la maintenance et l'évolution de l'infrastructure, l'organisation s'est dotée de spécialistes et les a regroupés dans une unité dédiée (Y1).

Paul regrette de ne pas être en lien avec les interlocuteurs métiers ou la hotline. Ce sont les BA, les chefs de projet SI qui le sont et reçoivent les problématiques pour finalement lorsqu'ils ont besoin les faire redescendre à l'unité dans laquelle se trouve Paul.

Il pense sincèrement qu'il pourrait mieux comprendre les anomalies constatées et plus facilement les reproduire pour aider les développeurs à les corriger. Il a déjà évoqué plusieurs fois le sujet, mais rien ne change.

Une organisation qui réalise de nombreux comités pour coordonner les différentes équipes et personnes, nécessitant de nombreuses heures de préparation de reporting et de nombreuses heures passées dans ces réunions.

Paul doit remonter fréquemment, les compteurs d'anomalies, le pourcentage de cas de tests restants, une appréciation du temps nécessaire restant pour boucler les tests qu'on lui a poussés, chose pour lui compliqué car s'il détecte une anomalie, il est assez souvent coincé pour continuer. N'ayant aucune idée de combien de temps prendra le correctif, il s'engage sur d'autres périmètres de tests à réaliser. Cela l'amène à prendre pas mal de temps à se replonger quelques jours plus tard sur ce qui avait été interrompu.

Au moins, se dit-il : *"Je monte en compétence de plus en plus sur tout le périmètre applicatif."*

Il se demande sur quelle base les chefs de projet SI communiquent des dates d'atterrissage sur les différents chantiers. Cela lui semble être vraiment fait au feeling plus qu'avec des éléments très factuels. Certes il y a des Gantt, des plannings, mais il y a toujours des imprévus, notamment des bugs ou des régressions, qu'il détecte et qui viennent chambouler les plans.

Un type d'organisation que vous avez peut-être connu, ou dans laquelle vous êtes actuellement à certains détails près.

Une organisation peu efficace, peu efficace et aucunement prédictible.

Pour cette organisation, les décalages dans les plannings sont constants. Des annonces de corrections et de nouveautés sont communiquées aux clients et ne sont quasi jamais tenues. Le mécontentement client est de plus en plus présent, à un niveau qui devient critique pour le business, ce qui ne laisse pas de marbre la direction. Il est absolument vital pour cette organisation d'accroître la capacité à répondre plus rapidement et de manière plus prédictible.

La suite des aventures de Paul dans cette organisation, dans le prochain chapitre...

Chapitre 2 : Projet de refonte et première modification d'organisation

L'organisation qui était loin d'être performante avait de plus entraîné l'application numérique dans une dette technique importante. Les clients en subissaient très fortement les conséquences, avec des incidents critiques qui étaient de plus en plus fréquents. Paul lui aussi en faisait les frais, de plus en plus de tests de non régression étaient poussés à l'unité de testeur, espérant ainsi éviter d'amener des problématiques importantes sur l'application jusque dans les mains des utilisateurs.

La direction, au regard de cela et ayant de plus en plus d'écho que l'organisation n'était pas efficiente, décida de lancer un vaste projet de refonte.

Elle en profita pour faire quelques modifications d'organisation afin de rapprocher des personnes qui avaient pour objectif de mener à bien ce projet. Les BA/Chef de projet SI fusionnèrent avec les personnes en expertises de tests, pour former une unité fonctionnelle (B1) (Pour la suite, cette unité sera appelée l'unité de BA).

Paul qui avait acquis beaucoup de connaissance sur le périmètre se vit offrir une possibilité de rejoindre l'équipe et en finir avec sa vie de consultant. L'idée lui plaisait d'autant plus qu'au niveau interlocuteur métier, la direction avait entrepris des actions pour qu'une unité de connaissance client se fédère autour d'une personne, un chef de projet métier (C0).

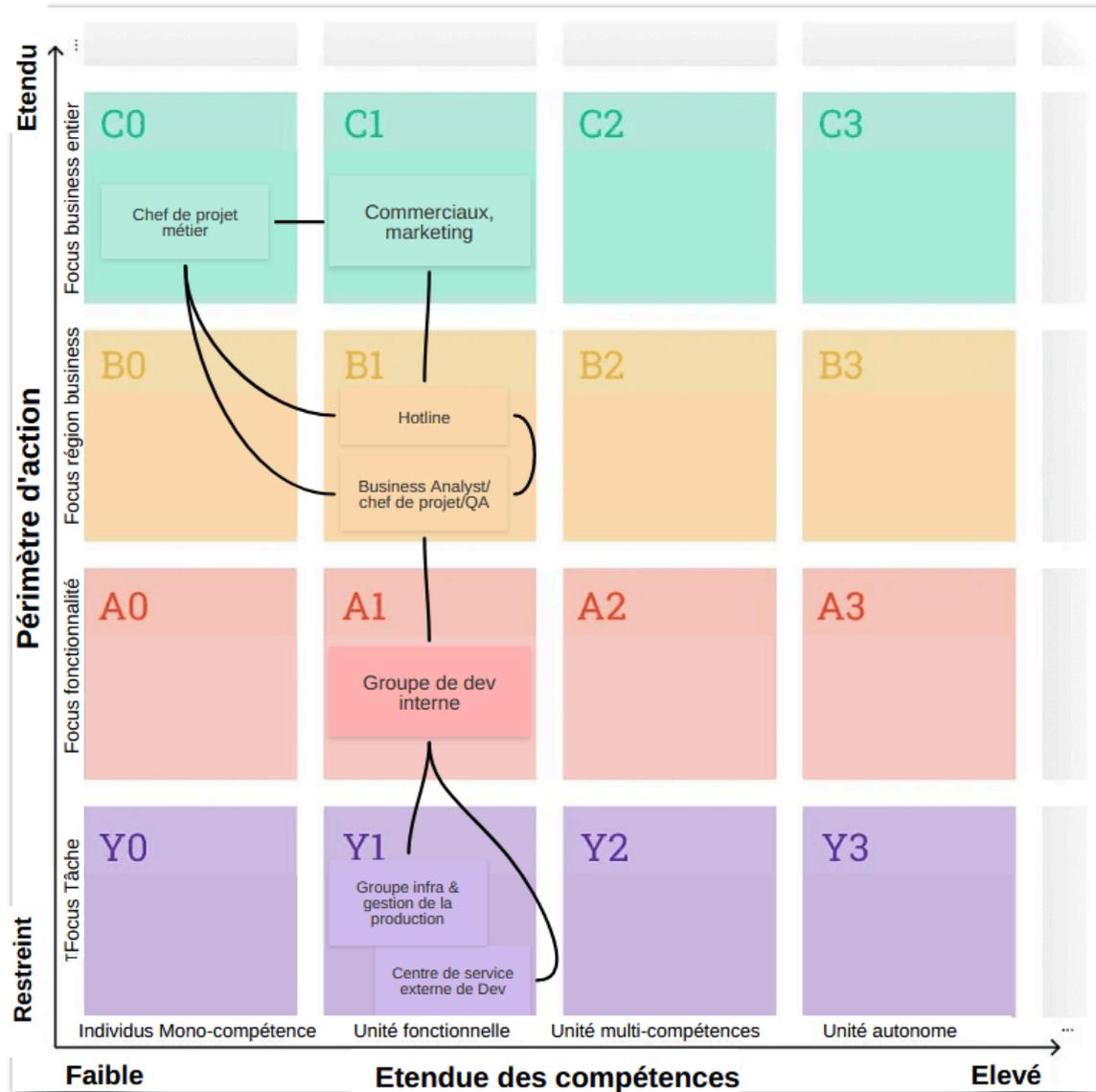


Figure 4 : Représentation de l'unité business après la première modification

Cette personne par son travail avec les commerciaux, les personnes du marketing, de la hotline et bien évidemment des clients devait rapidement acquérir une connaissance élargie des attentes et problématiques des clients. Elle allait travailler par ailleurs étroitement avec l'équipe dont Paul faisait partie pour amener du sens au travail réalisé et prioriser les développements les plus pertinents par rapport à ce qui était attendu par les clients.

Paul n'avait jamais eu l'occasion d'avoir un interlocuteur lui faisant aussi bien prendre conscience des attentes des clients, sa motivation ainsi que celle de ses collègues devint rapidement bien plus importante.

Paul avait vu dans ses précédentes expériences des équipes qui utilisaient un management visuel physique pour gérer leurs projets. Il proposa au reste de son unité de BA de mettre ce management visuel en place.

L'idée de Paul et la collaboration avec le reste de l'unité les amena à mettre en place un simple workflow "A faire", "En cours", "En attente", "Terminé".

Côté unité de développement, ils avaient eu eux aussi écho de cette approche et Paul se souvient qu'ils avaient à peu près mis la même chose en place.

Paul découvrit un peu plus tard qu'ils avaient initié la mise en place d'une stratégie Kanban, cependant :

"Nous n'étions clairement pas dans une stratégie Kanban, tout simplement parce qu'à cette époque personne ne connaissait ce qui s'était mis en place chez Corbis et l'émergence de Kanban pour le domaine du product management. Si cela avait été le cas nous n'aurions certainement pas eu cette colonne ("En attente").*

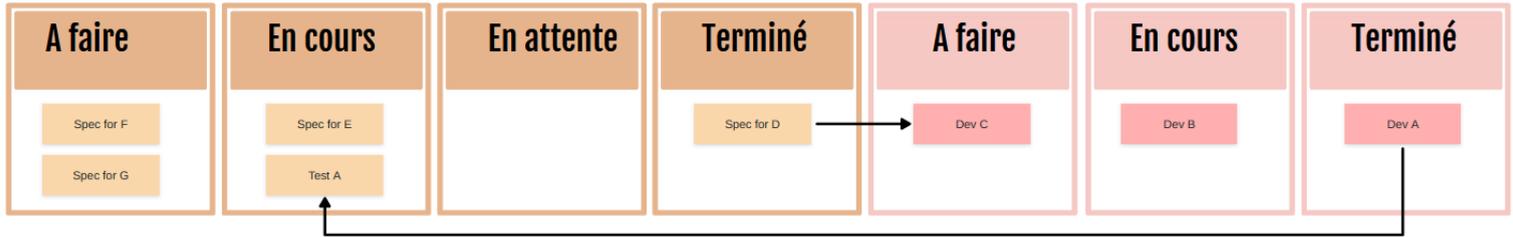
Nous n'avions par ailleurs pas de règles de passage très explicites entre nos états, il y avait beaucoup d'implicite autour de ce workflow. Enfin il nous manquait tous les autres éléments d'une véritable stratégie Kanban. Comment limiter l'encours? Des règles de tirage explicite, un niveau de service sur lequel l'unité se challengent etc.

Nous aurions pu et dû à cette époque fusionner nos workflow afin d'avoir une vision globale de ce qui se passait, car au finale quand nous (BA) terminions quelque chose, c'était une spécification, une explication d'une anomalie que nous donnions à l'équipe de dev et qui donc pour eux aller apparaître dans leur "A faire".

Quand ils terminaient, cela revenait chez nous (BA) pour que nous testions. Nous faisons alors entrer un ticket de recette qui traversait notre workflow (d'où la colonne en attente que nous utilisions lorsqu'un bug était trouvé et que nous attendions la livraison d'un correctif pour reprendre les tests)."

*Si vous voulez en savoir plus sur la naissance de Kanban pour le domaine du product management qui s'est passé chez Corbis (une compagnie de Bill Gates), l'histoire est très bien raconté dans le guide de poche Kanban (<https://prokanban.org/kpg/>)

Si ces deux unités avaient juxtaposer leur workflow, ils auraient obtenu quelque chose comme :



Etat quelques jours/semaines plus tard ...

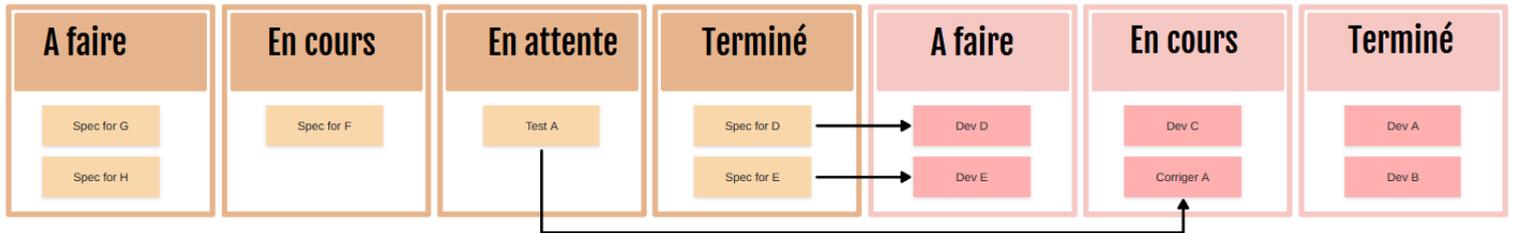


Figure 5 : Le workflow initial mis en place et l'évolution d'élément au bout de quelques semaines

Difficile par cette juxtaposition d'avoir une vision de bout en bout pour savoir combien de temps ce collectif mettait pour terminer un sujet qu'attendait des clients où qui était utile dans le cadre de la refonte de l'application.

Paul raconte : *"On avait parfois jusqu'à 4 ou 5 aller et retour avec des corrections à réaliser et des nouveaux tests à faire, autant de post-it qui circulaient sur notre management visuel. On avait ajouté des dates de création de sur nos tickets mais si on voulait savoir combien de temps avait pris le développement, le test et la correction du Dev A on était en difficulté pour avoir cette information.*

La simple mise en place de cela avait tout de même un gain non négligeable par rapport à la visibilité dans chaque unité de ce qu'il y avait à faire, de ce qui était en cours. La collaboration au sein de nos unités était devenue meilleure, les personnes étaient de plus en plus capables d'intervenir globalement sur l'application existante, mais aussi sur la nouvelle que nous construisions"

Paul relate par contre, que le chef de projet métier s'agaçait régulièrement car il était impossible pour lui d'avoir une information claire de quand un sujet allait être bouclé. Il voyait qu'il y avait beaucoup de travail réalisé, qu'il y avait des avancements, mais s'il voulait communiquer à la direction ou aux clients quand la fonctionnalité F allait sortir, il ne pouvait toujours pas avoir de réponses fiables. Ce qui l'amenait à devoir revenir sur sa communication régulièrement.

Le temps réellement pris pour terminer une fonctionnalité était bien souvent largement en écart à l'estimation initiale qui avait été faite... et malheureusement pas dans le bon sens, plusieurs semaines assez fréquemment.

La direction de son côté commençait à douter de la capacité à mener à bien le projet de refonte, de nombreux décalages de planning survenaient encore et toujours.

Paul : *"L'agacement de plus en plus fréquent du chef de projet métier et le doute de la direction (qui nous redescendait sous forme de pression) nous fit ressentir fortement que nous avons un axe d'amélioration important à mettre en place rapidement.*

En discutant avec les collègues de l'unité de BA et ceux de l'unité de développement, nous faisons surgir le besoin de casser nos silos et de devenir une seule et même unité multi-compétente, qui allait s'aligner sur un même objectif à court terme afin de rassurer sur notre capacité à mener à bien ce projet. Nous espérions aussi que cela allait nous permettre d'avoir une vue d'ensemble nous permettant d'être plus fiable dans nos réponses à la question du quand qui nous était souvent posée?"

Chapitre 3 : La naissance d'une Scrum Team

Cette volonté des deux unités distinctes de se rapprocher pour améliorer globalement la performance de la création du produit numérique fut quelque chose acceptée par la direction. Non sans un effort tout de même conséquent, notamment de Paul, pour arriver à convaincre que cette expérience valait le coup d'être tentée. Cette nouvelle unité rassemblée était bien évidemment attendue sur les résultats escomptés de cette expérience, notamment autour du fait que ce rassemblement devait améliorer; le Time to Market (T2M).

Ce T2M était clairement le temps de cycle, mais l'équipe utilisa cette appellation mieux comprise de la direction pour vendre l'expérience qu'elle souhaitait réaliser.

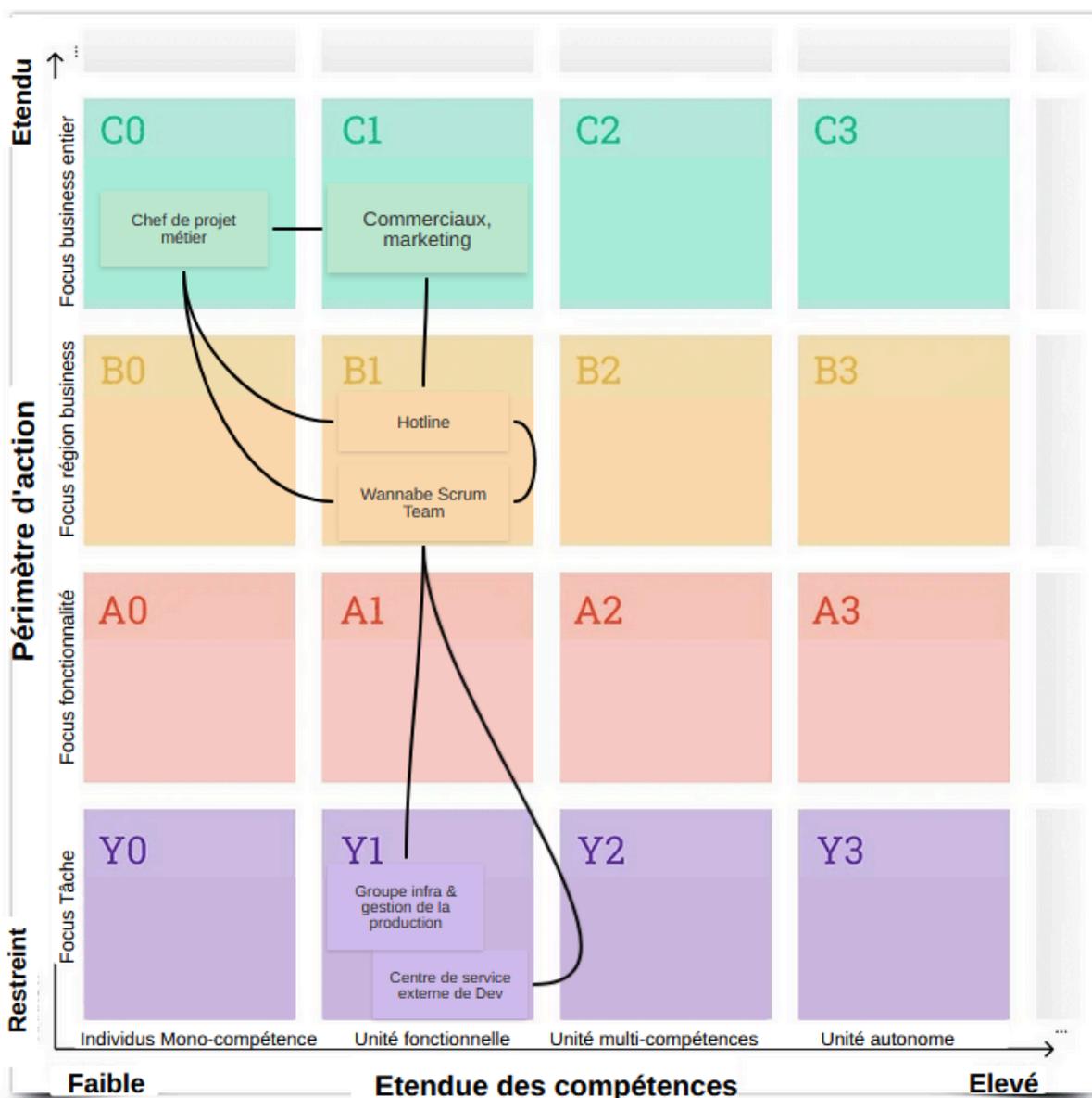


Figure 6 : Représentation de l'unité business après la seconde modification

Paul relate que cette évolution d'organisation amena deux choses :

"Le chef de projet "métier" intégra plus ou moins l'équipe dans une redevabilité de Product Owner. Je dis plus ou moins, car on était au tout début de Scrum chez nous. Si aujourd'hui a encore pas mal d'endroits, Scrum n'est pas correctement mis en place, notamment autour de ce rôle de PO, je vous laisse imaginer ce qu'à l'époque (et on parle des années ~2010) ce rôle de PO avait de différents avec celui d'un chef de projet (spoiler alert : rien ou presque).

Nous (BA) formions avec les développeurs internes, l'équipe de développement Scrum (oui c'est comme ça que ça s'appelait à l'époque, donc j'ai le droit de le dire !)

Nous nous dotions d'un super outil de ticketing. Super est ironique, car il est certainement responsable de pas mal de travers que nous avons pris lorsque nous définîmes notre workflow d'équipe unifiée :

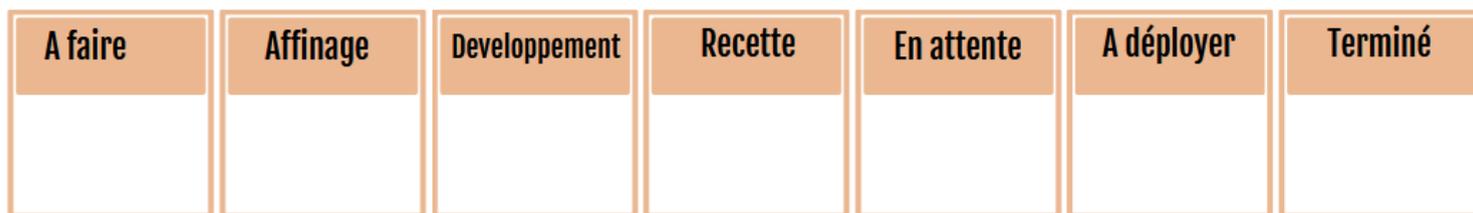


Figure 7 : Workflow unifié suite au changement d'organisation

Nous avons enfin une vue plus globale, unifiée de ce que nous faisons pour créer de la valeur."

Vous avez peut-être remarqué que cette colonne "en attente" était toujours présente avec cette unification des workflows par la fusion des deux entités.

Paul explique pourquoi : *"Eh bien pour nous en tout cas, c'était très clairement parce que nous n'étions pas encore une équipe, mais seulement un groupe de personnes avec quasi toutes les compétences nécessaires, mais toujours à parler de MOA, MOE et à se repasser la patate chaude. Donc oui, on avait toujours cette colonne en attente signifiant principalement qu'on attendait des personnes de la MOE de corriger un problème détecté."*

Une bien mauvaise pratique, en tout cas pour espérer atteindre un flux performant ! (Mais tellement courante dans une organisation avec des silos).

En effet, chaque silo peut être considéré comme une dépendance, par laquelle chaque élément de valeur potentielle va devoir passer pour pouvoir atterrir enfin dans les mains de l'utilisateur final.

Chaque dépendance est un point où le flux se rompt avec potentiellement (et fréquemment) des éléments qui s'empilent.

En conséquence, ces silos entraînent un accroissement du nombre d'éléments en cours, car par exemple la "MOA" démarre des travaux supplémentaires en attendant soit des développements, soit des corrections. Les changements de contexte sont nombreux ce qui entraîne également une perte d'efficacité, liée aux coûts non négligeables en termes de temps à se replonger dans des éléments. Une étude sérieuse sur le sujet des changements de contexte a montré qu'en changeant de contexte entre 2 sujets, il y avait une perte d'environ 20 % du temps, 40 % de temps perdu en poursuivant 3 sujets en même temps. (voir : <https://www.scrum.org/resources/blog/financial-cost-task-switching>)

Malgré tout, ce rapprochement après quelques semaines d'expérimentation commença à montrer des signes positifs. A la fois sur le T2M, des mesures que tenait le Product Owner en devenir, une tendance baissière était visible. Mais également sur la qualité de ce qui était produit. En effet, ce rapprochement avait entraîné parfois (malheureusement pas encore régulièrement) des ateliers de travail collaboratifs entre les différentes compétences pour comprendre l'attente, la problématique à résoudre et trouver ensemble la meilleure solution. Les compétences techniques pouvaient donc mieux saisir ce qui était attendu, ce qui leur permettait de faire des propositions (et non plus coder sans chercher à comprendre le pourquoi), ce qui collectivement permettait de mettre en œuvre des incréments plus qualitatifs.

Cependant, il y avait un point d'inefficacité notable encore dans cette organisation, il se situait autour de cette unité de développement externe provenant d'un centre de service.

Paul le souligne d'ailleurs très bien :

"En fait dans ce workflow, si un zoom était fait dans "Développement" on voyait que des choses n'étaient plus au sein de notre équipe unifiée, mais du côté de ce centre de services. Les travaux qu'ils faisaient, devaient être intégrés. Nous avons ajouté une étape d'intégration avant l'étape de recette pour rendre cela transparent. Les données collectées montraient clairement un point d'inefficacité, cela nous aida à appuyer auprès de la direction le besoin de traiter la problématique. "

Au détour de l'opportunité d'un changement de centre de service, Paul et ses collègues réussirent à convaincre la direction de faire quelques modifications au contrat et au processus interne. Non sans mal, ils réussirent à faire en sorte que cette force de coding supplémentaire rejoigne leur équipe Scrum et puisse intervenir au même titre que n'importe qui sur le produit.

Une dépendance en moins, un alignement aussi de ces personnes sur un objectif commun, des nouvelles cellules grises pour réfléchir ensemble aux meilleures solutions possibles. In fine, un prestataire et des intervenants plus engagés, car face à des problèmes à résoudre qui avaient du sens pour des utilisateurs finaux (au lieu de "pisser" du code sans en comprendre la raison).

Un changement qui fut clairement payant, vous le découvrirez dans le chapitre suivant.

Chapitre 4 : La montée en puissance sur la stratégie Kanban

Avec un peu de patience, d'effort de dialogue, Paul et son équipe scellèrent un esprit collectif et d'unité.

Ils avançaient sur leur compréhension de Scrum, mais aussi de la stratégie Kanban. Ils commencèrent à mesurer des temps de cycle, à faire des expériences de limitation de WIP dans une recherche d'optimiser leurs flux.

En créant la DoD de l'équipe, ils devaient malheureusement encore s'arrêter avant la mise en production. Cette dernière étape était toujours réalisée par d'autres personnes en dehors de leur équipe.

Paul raconte : *“Nous ne considérons plus les éléments de travail comme quelque chose que nous nous transmettions les uns aux autres, mais comme quelque chose que nous devons terminer aussi vite que possible ensemble. Nous nous aidions mutuellement, que ce soit sur le plan fonctionnel ou technique, avec toutes les compétences dont nous disposions individuellement. Par exemple, les personnes fonctionnelles avaient appris à réaliser des tests automatisés, et pas seulement au travers des IHM, nous allions jusqu'à l'automatisation de tests au niveau des API. Les personnes aux compétences techniques étaient invitées à participer plus tôt à la conception de la solution et à apporter leurs idées. Certains réalisaient des tests, lorsque cela était préférable au regard de l'état de notre flux de travail. Nous étions maintenant dans la même pièce, la collaboration était très forte, la programmation en*

binôme était devenue une coutume régulière et des binômes souvent fonctionnels & techniques.”

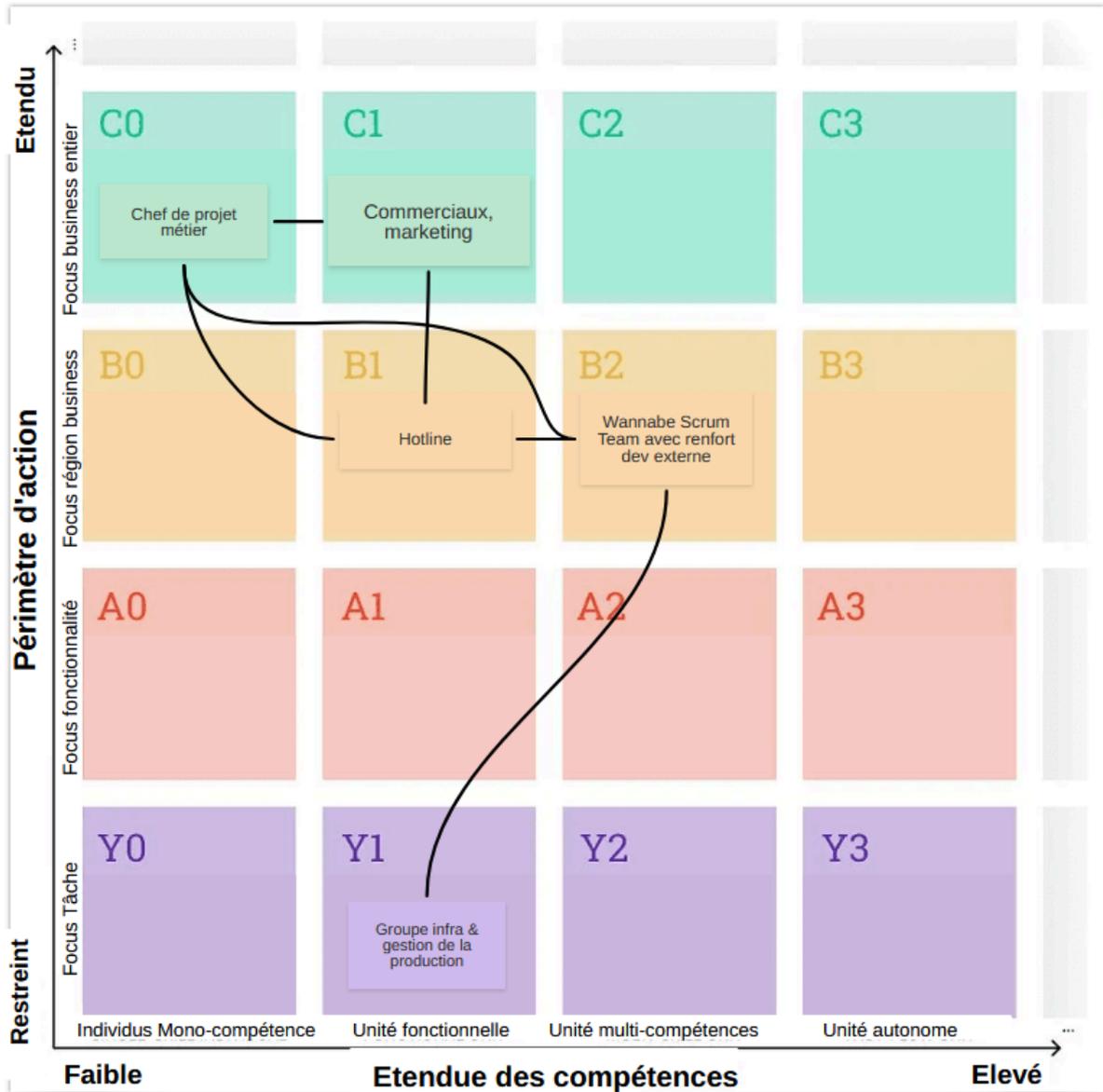


Figure 8 : Représentation de l'unité business après la troisième évolution

Le workflow* de l'équipe Scrum avait donc évolué et était maintenant :

SLE Fonctionnalité (Affinage > A déployer) : 20 jours ou moins avec 85% de probabilité

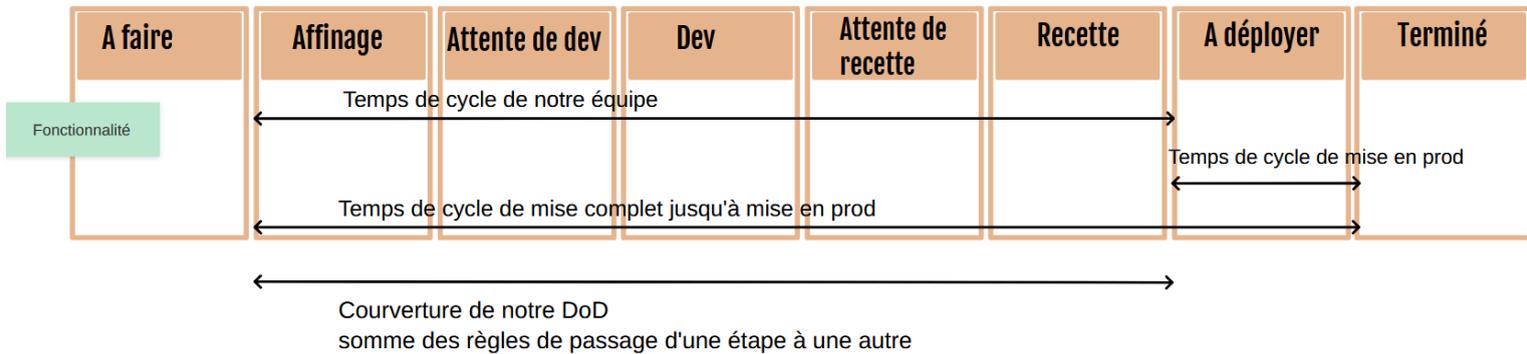


Figure 9 : Evolution du workflow suite à cette troisième évolution

* Par souci de lisibilité, tous les éléments d'une définition de workflow d'une stratégie Kanban professionnel ne sont pas reportés ici (se référer au guide Kanban pour comprendre ce qu'il manque).

En prenant des précisions auprès de Paul, l'équipe avait instauré des limites de WIP sur les étapes d'affinage, de dev et de recette. Les limites de WIP devaient aider à améliorer la focalisation de l'équipe en évitant de démarrer un trop grand nombre de travaux en parallèle, mais aussi à éviter que quelque part dans le workflow une "pénurie". L'équipe avait donc instauré des limites de WIP optimum, c'est-à-dire à la fois une limite haute à ne pas dépasser, mais également une limite basse à laquelle se maintenir. Le but in fine était de rechercher par expérimentation les meilleures limites qui feraient gagner en performance.

Paul : "L'erreur que nous fîmes au début, fut de ne pas limiter aussi les étapes d'attente de dev, d'attente de recette (ou par regroupement d'étapes avec l'étape précédente de travail actif)"

Une erreur assez classique, qui génère bien souvent un empilement de travaux en attente, et in fine plus de travaux globalement en cours que le niveau optimum pour l'équipe.

Paul relate que ce problème fut assez rapidement détecté et corrigé, il ajoute : "Tout cela nous amena à une progression nette de notre rapidité à traiter les sujets. On ne le mesurait pas très bien avant, mais avec les données que le chef de projet, hmm pardon notre Product Owner communiquait et les différentes replanifications, nous étions grosso modo entre 30 jours et 90 jours pour boucler une fonctionnalité. Les données que nous collections maintenant nous indiquaient une division par deux du temps de cycle maximum et une variabilité plus réduite (~25-45 jours)."

Une équipe qui avait une volonté d'amélioration continue qui s'ancrait de plus en plus en elle. Ces mesures maintenant fréquentes des temps de cycle et la recherche de l'amélioration de leur performance, les amena à s'intéresser à une métrique

encore plus intéressante que le temps de cycle, à savoir l'âge des éléments en cours dans leur flux.

Pourquoi encore plus intéressante ? Parce qu'elle les informait bien plus tôt que le temps de cycle (calculé seulement une fois l'élément ayant atteint la fin du workflow), ce qui leur permettait d'avoir les conversations nécessaires plus rapidement pour gérer au mieux la performance de leur flux.

Paul nous raconte comment cela est venu : *“Nous avons découvert cette métrique au travers du billet de blog : <https://caroli.org/en/latency-and-banana/>. L'idée nous avait paru géniale, mais on préféra tout de même compter le nombre de jours sans accrocher de peaux de bananes sur nos post-it...*

Un peu plus tard, nous apprenions à visualiser le temps de cycle de nos différents éléments sur un graphique en nuages de points et, en y ajoutant des percentiles, nous découvrions la répartition du temps de cycle de nos éléments... 70 % en dessous de 24 jours, 85 % en dessous de 33 jours.

Ceci nous conduira à définir un niveau de service attendu (SLE) basé sur nos données historiques et en nous challengeant à faire mieux : 20 jours ou moins à 85 % du temps.

Nous couplions cela avec le suivi de l'âge de nos éléments en cours et ce fut un game changer.”

Les limites de WIP que l'équipe avait expérimentées, avaient apporté une amélioration de performance assez minime. Les expériences menées leur avaient apporté parfois des améliorations, mais aussi parfois des régressions, sauf que cela prenait un peu de temps à se voir. Le contrôle de l'âge et le challenge autour du SLE qu'avait choisi l'équipe par contre apporta une amélioration conséquente. De meilleures limites de WIP se dessinèrent même assez naturellement au travers de cette pratique du contrôle de l'âge et du focus à essayer de ne pas dépasser le SLE. Après quelques mois, l'équipe avait largement rempli son challenge et pouvait même se challenger à améliorer encore son SLE.

Elle entreprit cependant une autre amélioration, celle de l'augmentation de la qualité d'utilisabilité de son produit, avec la montée en compétence d'une partie de l'équipe sur l'UX Design.

Paul nous raconte cependant une mésaventure sur cette montée en compétence UX : *“Nous avons réussi à bien vendre l'apport que cette compétence pouvait avoir pour in fine le bonheur de nos clients et donc le business. La direction et notre représentant métier (Product Owner) avaient du coup très hâte de voir l'apport de ce type de compétences pour le produit, à tel point qu'ils ne nous laissèrent pas appliquer de nous même la compétence que nous avons fraîchement acquise. Ils firent appel à un cabinet de consulting spécialisé et malheureusement cela ajouta une unité distincte de notre équipe et donc une dépendance”*.

Chapitre 5 : Une bonne idée qui se transforma en dépendance pendant quelques temps

Cet ajout amena une dépendance et malheureusement un ralentissement global du système.

En effet, cette équipe avait, elle aussi, son propre workflow déconnecté de celui de la Scrum Team. Le Product Owner travaillait en amont avec cette équipe UX externe pour comprendre plus précisément les attentes et les problématiques des clients. Ils réalisaient des prototypes, faisaient des tests utilisateurs avec ces prototypes. Cela malheureusement, prenait pas mal de temps.

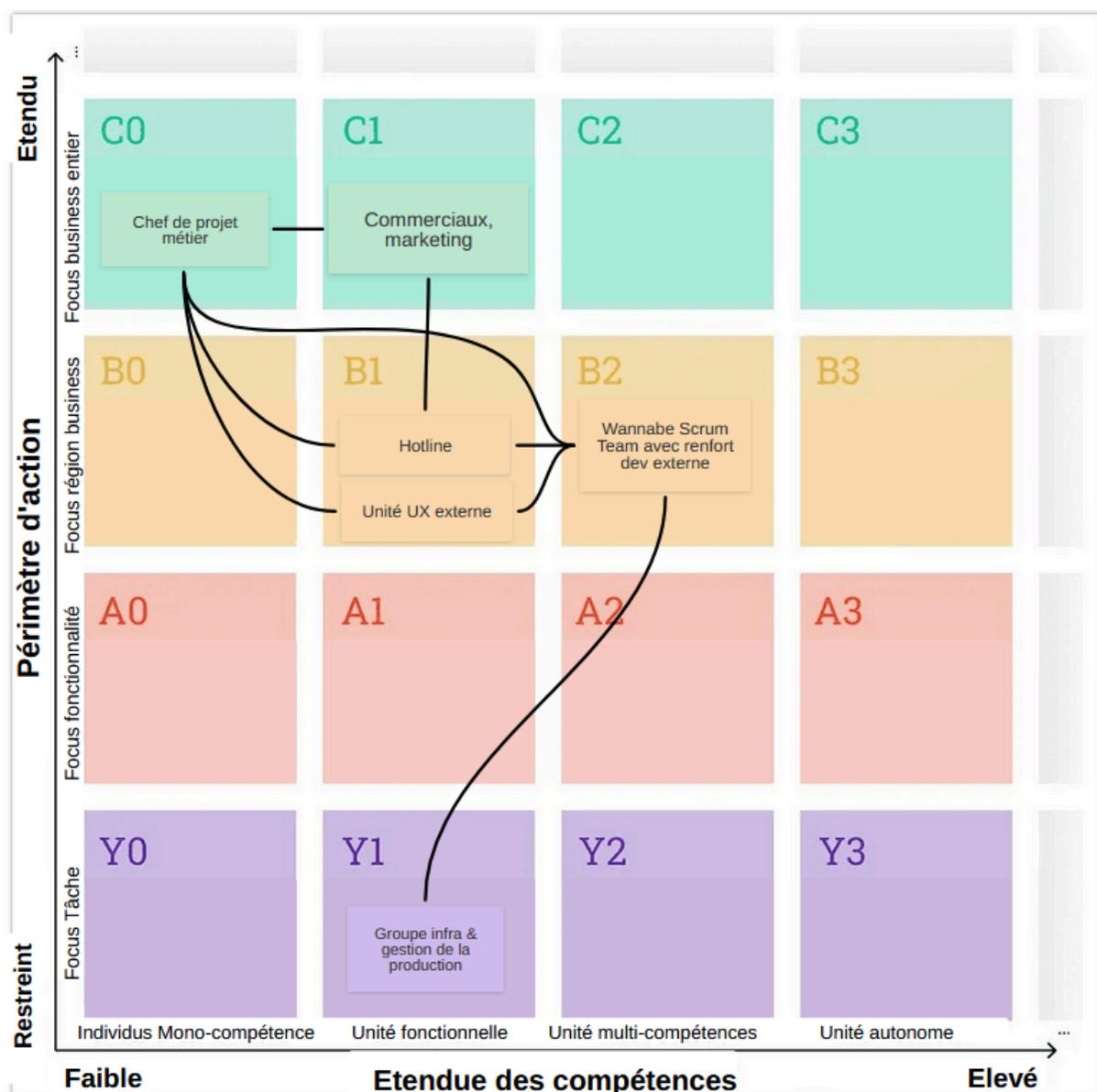


Figure 10 : Cartographie org topologies après la quatrième évolution

Paul se souvient bien de cette période, frustré de ne pas mettre à profit ses nouvelles compétences acquises :

“Ils étaient assez haut niveau dans la macroconception de la solution et nous avons toujours besoin lorsque nous récupérons leurs travaux de faire de l’affinage, de découper les grosses features en plus petits morceaux pour avancer itérativement et incrémentalement comme nous avons pris l’habitude de faire. De plus, il n’était pas rare que les prototypes qu’ils avaient testés avec les utilisateurs, nous posaient problème en termes de faisabilité pour les mettre en œuvre. Nous étions dans un silo discovery/delivery.”

Cette volonté de mieux cerner les attentes des clients était louable, car il est vrai qu’il est dommage de développer des choses qui ne sont pas attendues, qui ne vont pas être utilisées, qui vont poser des problèmes d’utilisabilité, mais tout cela avait un coût, et finalement le T2M, le temps de cycle pour livrer des évolutions sur le produit était beaucoup plus court avant.

Paul et l’équipe avaient réussi à savoir combien de temps il s’était écoulé avant qu’un sujet ne leur parvienne :

“Il faut voir que grosso modo avec l’ajout de cette équipe UX/UI externe c’était ~ 2 mois de temps de cycle passé avant que nous récupérons la patate chaude (enfin tiède, voire presque froide, au vu du délai déjà passé).

La véritable connaissance de ce que nous apportons comme valeur était du coup retardée de ~ 2 mois, mais nous avons potentiellement quelque chose avec un plus haut niveau d’utilisabilité et quelques hypothèses à valeur faible écartées.

La principale problématique à mon sens était la déconnexion du discovery et du delivery, accentuée par le fait que ce n’était que des travaux amont et jamais cet UX Team ne s’appuyait sur le concret en production et dans les mains des utilisateurs pour fermer la boucle d’apprentissage amenant à de nouvelles réflexions UX pour faire évoluer le produit.”

Dans ce positionnement finalement uniquement orienté “business”, dans le sens où ce prestataire répondait à la commande de “discovery” initiale du Product Owner mais sans refermer la boucle d’apprentissage avec le véritable produit et le comportement des utilisateurs en situation réelle, l’organisation était loin de tirer bénéfice de l’UX design.

Paul et les autres personnes de l’équipe qui avaient été formés, avaient bien conscience que le prestataire aurait pu ou aurait dû proposer plus.

Ils avaient perdu une bataille avec l’arrivée de cette équipe UX externe, mais décidèrent de ne pas en rester là. Ils installèrent un outil leur permettant de réaliser de l’analytique produit et ils commencèrent discrètement à analyser les parcours, l’utilisation des fonctionnalités, les points de décrochage, ils mirent des heatmap en place pour voir plus finement dans les pages où circulaient les utilisateurs...

Paul nous raconte : *“Toutes ces informations étaient extrêmement précieuses, car source d’apprentissage de l’utilisation réelle de notre produit. Je ne comprends toujours pas pourquoi la société d’expert UX n’a pas proposé cela et s’est contentée de ce “discovery” initial, mais bon cela finit finalement par faire notre bonheur. En effet, un jour, nous décidâmes avec quelques autres membres de l’équipe que nous avions suffisamment de billes via les analytiques produits pour ouvrir des discussions sur des fonctionnalités inutilisées, des parcours à améliorer, etc. Pour les parcours que nous avons détectés à améliorer, nous avons fait quelques maquettes de faible fidélité (peu coûteuses) pour proposer des améliorations et avoir un support de discussion. Des éléments qui paieront, car très appréciés des parties prenantes, de la direction et qui permirent d’asseoir notre expertise également en la matière”.*

La mission du prestataire UX se termina quelques semaines après et la direction, le Product Owner décidèrent de s’en remettre à l’expertise que l’équipe avait prouvée avoir acquise, en tout cas suffisamment pour ne plus voir le gain à payer une prestation.

Ceci amena une évolution du workflow de l’équipe sur 2 aspects.

Premièrement, l’équipe (y compris le Product Owner) décida que le point de fin n’était plus le passage en production pour que le client ait le produit dans les mains. Le point de fin serait maintenant au-delà de cela et se situerait lorsque la collecte du feedback de l’utilisation de la fonctionnalité serait jugée suffisante. Soit pour décider de terminer, soit pour décider de terminer, mais en apportant un nouvel élément en entrée du workflow lié au feedback et à l’apprentissage obtenu de l’utilisation du produit.

Deuxième évolution du workflow sur les aspects de “discovery”. Paul avait suivi récemment la formation Professional Scrum With UX (PSU) et avait compris clairement le concept de dual track agile. En résumé ce concept est le mélange délicat et en continu entre ce qui est nécessaire d’apprendre en amont pour éviter de développer des choses qui ne seraient pas utilisées (ou trop peu) et le développement rapide, se basant sur cet apprentissage, de petits incréments, dans un but également d’apprentissage et de réalimentation de la boucle à l’aide de ces nouvelles connaissances.

Ce n’est pas comme beaucoup ont pu le comprendre (et le comprennent peut-être encore), deux processus parallèles, menés par des personnes différentes et qui s’alimentent en discontinuité, en séquentiel (les travaux de discovery des 2 dernières semaines, alimentant le delivery des 2 semaines suivantes), sans tenir compte en continue des apprentissages réalisés au travers de chaque “prisme”.

Avec ces connaissances nouvellement acquises, Paul amena la proposition d’une expérimentation à l’équipe pour visualiser au niveau du workflow ce travail de discovery.

“En product management agile, il est crucial de savoir le plus rapidement possible si on a tort, il était donc hors de question pour moi d’amener l’idée d’avoir des éléments de discovery qui circuleraient à côté des éléments de delivery et à faire avant de pouvoir progresser sur le delivery. La majorité du temps, il fallait que ces travaux soient liés au même élément de valeur, pour que cela soit vu comme un tout, où toute l’équipe restait responsable de faire progresser l’élément le plus rapidement possible vers la fin de notre workflow.

Mais d’un autre côté, parfois il pouvait s’avérer nécessaire de faire des études plus poussées et plus longues pour s’assurer que l’hypothèse de valeur était à priori “viable”.

J’amenais donc l’expérience suivante à tenter :

- *La création d’un nouveau type d’élément de valeur qui circulerait dans notre workflow que j’appelais “Etude (discovery)” et qui allait correspondre à ces travaux de discovery plus longs à mener.*
- *Pour le reste, je proposais au reste de l’équipe de considérer l’apport des pratiques UX dans les différentes étapes de notre workflow :*
 - *En affinage, par exemple en ayant rapidement fait une interview de quelques utilisateurs, ou en ayant prototypé rapidement sur le papier quelque chose pour le confronter également à quelques utilisateurs. Ceci nous permettait de démarrer le développement informatique avec quelques améliorations ou ajustements de la solution qu’on envisageait.*
 - *Pendant le développement et quand cela faisait sens (notamment quand émergeait des interrogations sur l’utilisabilité), on allait continuer à explorer si la solution était adéquate, par exemple sur la base d’un prototype de plus haute fidélité.*
 - *Lorsque nous arrivions à l’étape de recette, on allait approcher des utilisateurs finaux pour les faire manipuler la nouvelle fonctionnalité et vérifier notamment des points où nous avons des doutes sur l’utilisabilité.”*

Ces propositions furent jugées intéressantes et l’équipe s’embarqua dans cette expérimentation.

Pour le nouveau type d’élément “Etude discovery”, l’équipe se challengea avec un SLE associé. Ils ne voulaient pas eux même ramener la mauvaise expérience vécue avec le cabinet d’expertise UX externe et décidèrent que cela devait être réalisé la majorité du temps (80%) dans un laps de temps ne dépassant pas les 15 jours.

Le workflow* pouvait se représenter de cette façon :

SLE Etude : 15 jours ou moins avec 80% de probabilité
 SLE Fonctionnalité (Affinage > A déployer) : 20 jours ou moins avec 85% de probabilité

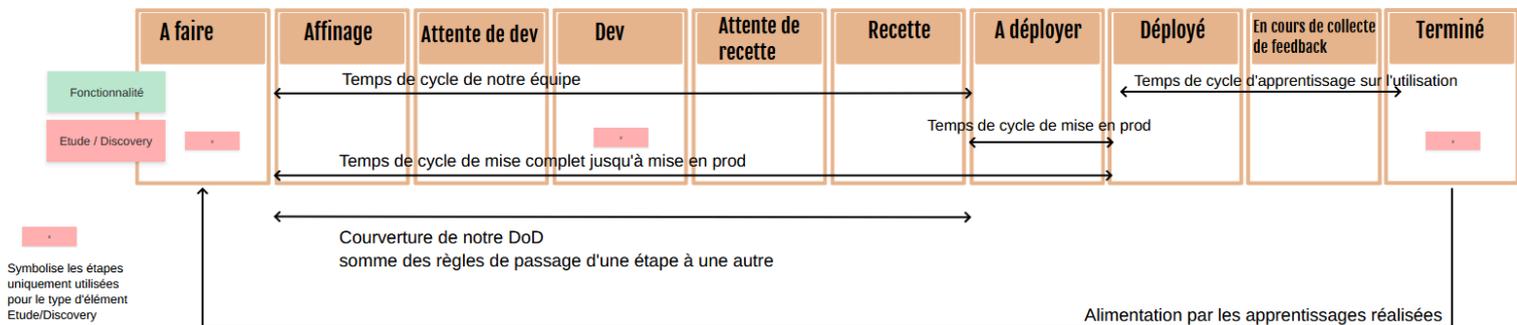


Figure 11 : Définition du Workflow après la cinquième évolution

* Par souci de lisibilité, tous les éléments d'une définition de workflow d'une stratégie Kanban professionnel ne sont pas reportés ici (se référer au guide Kanban pour comprendre ce qu'il manque).

L'expérience fut très concluante et renforça encore les liens et la pluridisciplinarité de l'équipe. En effet, des membres qui ne s'étaient pas formés à l'UX Design découvrirent et montèrent en compétence en travaillant de concert avec leurs collègues, l'expertise n'était pas au même niveau, mais cela permettait de réaliser des travaux d'UX ne demandant pas une très grande expertise (récolter des impressions, des feedback sur les maquettes, les proto, par exemple) par quasiment n'importe qui de l'équipe.

Toutes ces évolutions, rapprocha l'équipe de plus en plus des utilisateurs finaux, la confiance de la direction dans l'équipe était excellente, son efficacité s'était accrue drastiquement et sa prédictibilité également. Le Product Owner (et une bonne partie de l'équipe) avait été formé sur l'utilisation de leurs données de débit (factuel) pour réaliser des prédictions probabilistes via une simulation de Monte Carlo, qui donnait des résultats beaucoup plus justes et pertinents.

La confiance du marketing et des commerciaux s'en trouvait également largement améliorée et les tensions qui pouvaient exister dans le passé, avaient disparu totalement.

Le Product Owner était de plus en plus intégré également à l'équipe, fort de cette expérience et des résultats obtenus, il se vit offrir une opportunité très intéressante dans une autre entreprise qu'il saisissa.

La direction savait qu'elle pouvait compter sur Paul pour le remplacer et lui offrir cette opportunité, qu'il ne se fit pas prier pour accepter.

A ce moment là, l'organisation pouvait se décrire de cette manière :

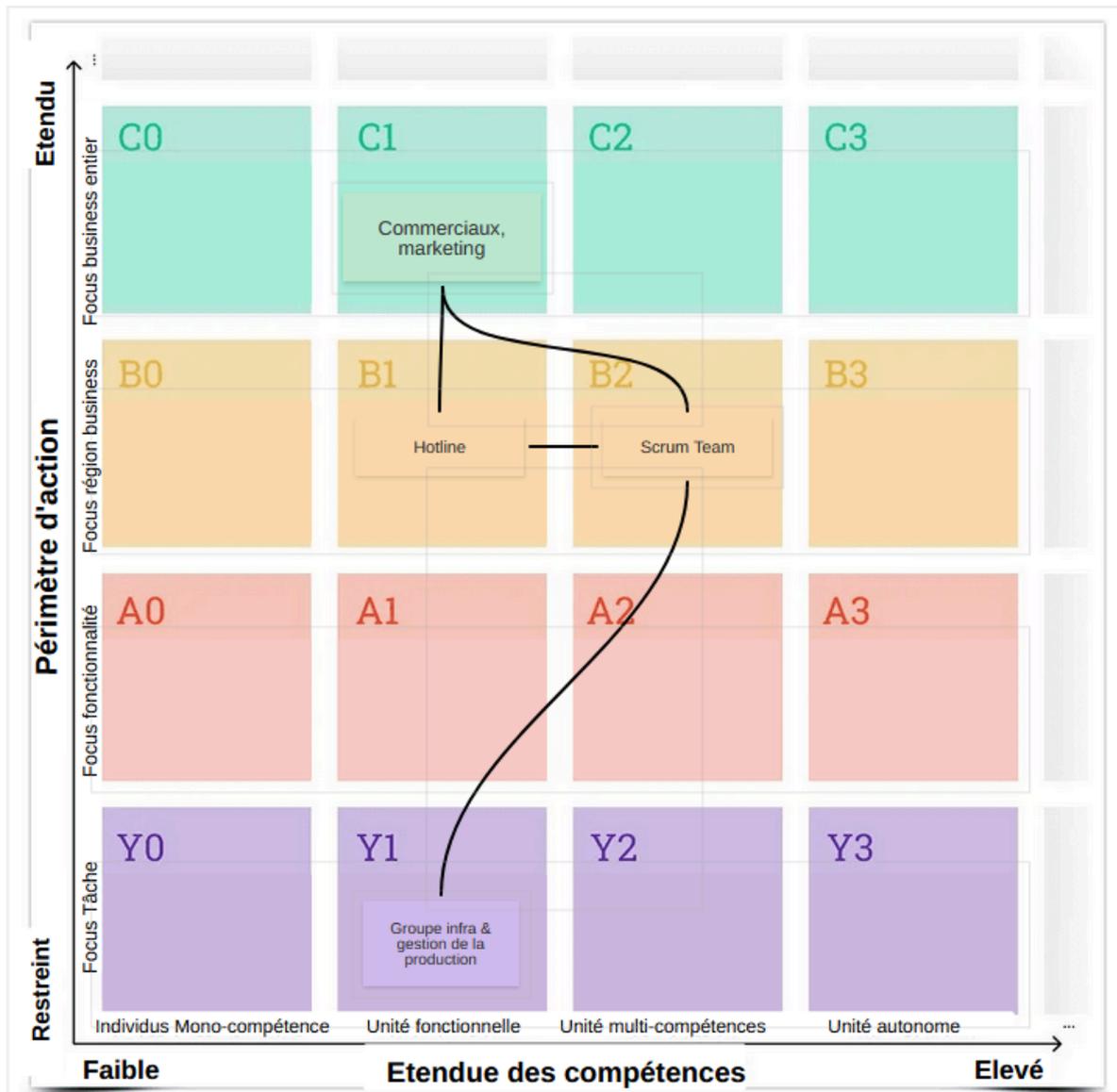


Figure 12 : Cartographie org topologies après la cinquième évolution

Paul, depuis ce jour, soutenu par le reste de l'équipe tente de résoudre une difficulté freinant encore assez amplement l'efficacité de l'équipe à délivrer de la valeur : l'existence de cette unité de gestion de la production et de l'infrastructure technique.

En effet, l'équipe Scrum est toujours dépendante de cette unité pour pouvoir voir les développements réalisés disponibles en production, ce qui leur fait perdre quelques jours avant que les nouveautés ne soient disponibles et qu'ils puissent en tirer des apprentissages.

Lorsqu'ils auront réussi à avoir le pouvoir, les droits, les habilitations pour eux même livrer leurs produits, ils seront réellement devenus une Scrum team autonome et pouvant délivrer en flux continu et rapide des nouveautés et améliorations sur leurs produits.

Il est à espérer pour eux qu'ils y arriveront et lorsque cela sera le cas l'organisation dans cette business unit pourra se cartographier de la façon suivante :

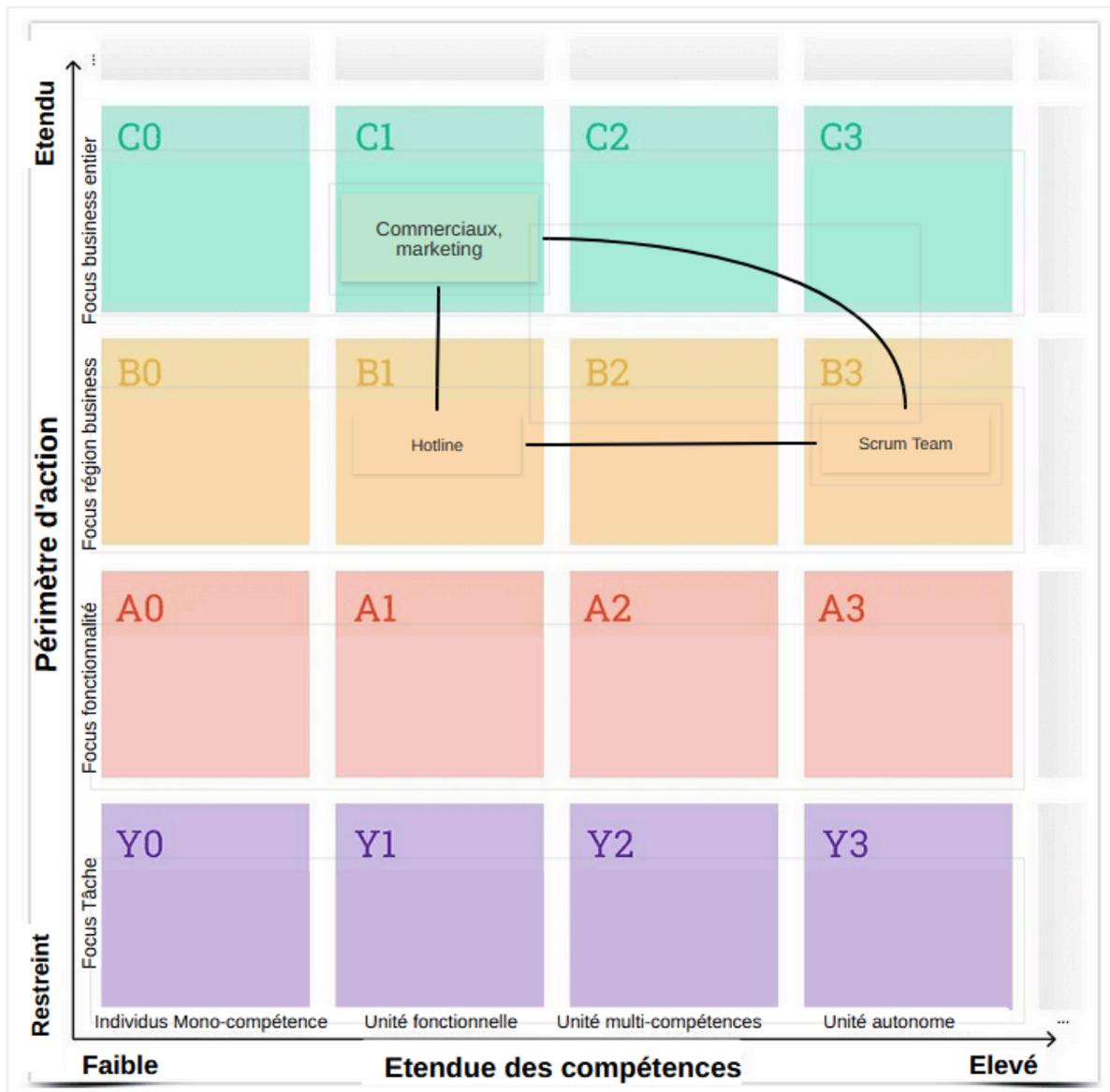


Figure 13 : Cartographie org topologies après la sixième évolution peut-être à venir

THE END... ou pas, ça dépendra de votre réaction à vous les lecteurs ;-)

Un grand remerciement à Alexey Krivitsky et Roland Flemm pour avoir créé Org Topologies et leur incroyable classe que j'ai eu la chance de suivre sur Paris en Mars 2024, ainsi qu'à leurs différents feedbacks, précisions et éclaircissements à apporter à cette histoire avant que je ne la publie.

N'hésitez pas à vous intéresser à leur travail ici : <https://www.orgtopologies.com/> et à aller en apprendre plus sur Org Topologies avec eux dans l'une de leur formation.