



Modélisation des champs des champs des formulaires de LesFurets.com

FITOUSSI Jonathan

Projet de fin d'études

Université Pierre et Marie Curie

Encadrant Université : CHAILLOUX Emmanuel

Encadrant Entreprise : DEGERBAIX Benjamin

Aout 2015

Table des matières

1	LesFurets.com	5
1.1	L'Assurance	5
1.2	Les comparateurs d'Assurance	5
1.3	Historique de l'Entreprise	6
2	Sujet de stage	7
2.1	Technologie	7
2.2	Objectif	8
3	Problématique	9
3.1	Contexte	9
3.2	Besoins	9
4	Analyse du problème	11
4.1	Eclipse Sirius	11
4.2	Application WEB	11
4.3	Donnée Business	11
4.4	Autre utilisations	12
4.4.1	Plan du site	12
4.4.2	CRM - Envoi de mail	12
5	Solution	13
5.1	Eclipse : Sirius	13
5.2	Page de maintenance Web : Javascript	13
5.3	Cytoscape.js + Features	13
5.3.1	Zoomer, Déplacer les éléments, Se déplacer	13
5.3.2	Choix du tunnel	14
5.3.3	Changement du layout	14
5.3.4	Recherche d'un champs et autocompletion	14
5.3.5	Champs du formulaire	14
5.3.6	Disposition du graph	14

5.3.7	Modification des elements et des dependances	14
5.3.8	Modification du label du champs	14
5.3.9	Modification type du champs	14
5.3.10	Suppression d'une dependance	14
5.3.11	Sauvegarde etat du graph	15
5.3.12	Charger un graph	15
5.3.13	Affichage des differences	15
5.3.14	Not User-friendly	15
5.4	Données Business	15
5.4.1	Récupération des données	15
5.4.2	Affichage des données	15
6	Gestion de projet	17
6.1	Agile	17
6.2	Kanban	17
6.3	Kanban chez lesFurets	18
6.4	Integration continu	18
6.5	Deploiement continu	18
6.6	Feature Branch	19
6.7	Ma methodologie	19

LesFurets.com

1.1 L'Assurance

Le secteur de l'assurance fait partie du secteur économique tertiaire qui regroupe les industries de services essentiellement immatériel. Il s'agit d'un secteur d'activité où les acteurs économiques vendent une protection contre le risque avec un poids très important au sein du paysage économique mondial et, plus particulièrement, au sein du paysage économique français. Sa particularité réside dans son activité et sa diversité, puisque il rassemble une multitude d'acteurs : compagnies d'assurance, courtiers, mutuelles, acteurs internet, etc. Mondialement, le marché de l'assurance est actuellement dominé par les états-Unis suivi par le Japon. La France occupe la 4ème position, derrière le Royaume Uni, avec 6,5% des cotisations mondiales, soit près de 211 235 millions d'euros. (FFSA et Gema)

1.2 Les comparateurs d'Assurance

Un comparateur de prime d'assurance est un outil informatique permettant de comparer toutes les offres d'assurances répondant à un ensemble de critères mentionnés par le client. Avec un marché fortement concurrentiel et très complexe, les comparateurs ont un rôle de plus en plus important auprès des clients en matière de comparaison, de choix et de souscription de produits d'assurance. Aujourd'hui, le marché est occupé par des concurrents connus tels que LeLynx.fr, LesFurets.com, Assurland.com, Hyper-assur, automotocompare.fr, kelassur.com, etc. En effet, ces comparateurs présentent des environnements ergonomiques permettant aux visiteurs d'effectuer un comparatif de contrats auto, moto, multirisques habitation, emprunteur ou santé, et de sélectionner les offres les mieux adaptées à leurs attentes. Pour répondre aux besoins des clients, chaque acteur maintient son propre modèle qui repose sur plusieurs aspects : le tarif, la qualité des services proposés, les services après-vente, les garanties, les franchises, les prestations, etc. Ainsi, chaque comparateur présente des offres d'assureurs différents. Le référencement web et la publicité présentent des investissements lourds pour l'ensemble des concurrents qui cherchent à obtenir une visibilité conséquente auprès de leur clients. Les nouveaux arrivants doivent forcément investir massivement pour trouver une place parmi les acteurs historiques sur ce marché.

1.3 Historique de l'Entreprise

Courtanet, fondée en 2005 par Jehan de Castet, est la société éditrice du site AssureMieux.com. Avec ses sites www.assuremieux.com, www.creditmieux.com, et son logiciel Bénéfit, la société est devenue rapidement l'un des plus importants fournisseurs français de solutions de comparaison d'assurances. Plus de 1 500 courtiers indépendants français étaient équipés de ses solutions. Depuis Novembre 2010, Courtanet est majoritairement détenu par BGL Group créé en 1992, intermédiaire britannique d'assurances et propriétaire également du leader de la comparaison d'assurances en Grande Bretagne via www.comparethemarket.com, qui a emporté 35 est aujourd'hui structuré autour de quatre « piliers » : Entreprises intermédiaires, Entreprises pilotées par marque (LesFurets.com), comparethemarket.com et services juridiques. En Avril 2012, la plateforme de comparaison Assuremieux change d'identité et dévoile une nouvelle marque baptisée www.LesFurets.com, le comparateur qui « simplifie » l'assurance. LesFurets.com est un comparateur indépendant et impartial d'assurance auto, moto, santé, habitation et crédit. Sa mission est d'aider les consommateurs à trouver l'assurance la mieux adaptée à leurs besoins au meilleur tarif. La mission du site est d'apporter plus de transparence et de simplicité dans l'assurance. Ainsi, il permet de comparer facilement les tarifs, les garanties, les franchises et les services de grands assureurs auto, santé, moto, habitation et crédit. Son objectif est d'aider à trouver l'assurance la mieux adaptée aux besoins des clients au meilleur tarif. Dans le but d'offrir un service de qualité, LesFurets.com s'associe avec les plus grands assureurs du marché : Direct Assurance, Amaguiz, IdMacif, Eurofil, AllSecur, Aon Assurances, AcommeAssure, SOS Malus, Assurpeople, L'olivier Assurances, 4assur, EuroAssurance, ActiveAssurances, AssurOnline, assurbike, aloa Assurances, SwissLife, Alptis, etc.

Sujet de stage

D'un point de vue technique l'équipe de lesfurets.com a décidé d'abstraire les champs du formulaire ainsi que les dépendances à l'aide d'un méta-modèle. A l'heure actuelle, l'application web est capable de transformer ce modèle de champs au format XMI pour pouvoir les exporter dans un logiciel de modélisation comme MagicDraw, Rational. Le stage débutera par un passage à d'autres formats de modélisation (EMF, JSON) traités à l'aide d'outils open-source qui permettront aux équipes de mieux visualiser le graphe des champs des formulaires ainsi que leurs dépendances. Il est prévu ensuite de superposer des analytics sur ces graphes. Il est aussi envisagé de garder une trace des différences entre les versions de l'application pour visualiser les modifications et/ou les régressions en fonction des développements. Enfin au même titre qu'un IDE classique, l'outil devra pouvoir proposer des fonctions d'édition, de sauvegarde et de partage des données traitées.

2.1 Technologie

Dans le cadre du projet, il me faudra explorer plusieurs pistes : La première consistera à me familiariser avec un outil de modélisation basé sur Eclipse : Sirius (<https://eclipse.org/sirius/overview.html>). Sirius est un projet Open Source de la Fondation Eclipse. Cette technologie permet de concevoir un atelier de modélisation graphique sur-mesure en s'appuyant sur les technologies Eclipse Modeling, en particulier EMF et GMF. L'atelier de modélisation créé est composé d'un ensemble d'éditeurs Eclipse (diagrammes, tables et arbres) qui permettent aux utilisateurs de créer, éditer et visualiser des modèles EMF. Le projet est une initiative française et la communauté autour du projet est très active. Je me mettrai en relations avec les créateurs du projet pour pouvoir aborder les questions que j'aurais tout au long de mon stage. Il me faudra aussi explorer la piste de la création d'une application web en charge d'afficher la modélisation. Il existe de multiples bibliothèques en JavaScript pour l'affichage des données sous la forme de graphe orienté. De plus il s'agit d'un graphe orienté faiblement cyclique qu'on pourra traiter, côté serveur avec des algorithmes implémentés en Java, Scala, Python ou JavaScript. Enfin aujourd'hui le graphe est généré sur un fichier au format XMI mais on pourra imaginer que l'outil introspecte les objets directement dans les binaires.

2.2 Objectif

Les utilisateurs de l'outil seront d'une part les architectes logiciel de la société, les développeurs mais aussi les business-analyste (concepteur fonctionnelle) qui pourront mieux cerner les impacts des modifications ainsi que les besoin des utilisateurs du site lesfurets.com. Ainsi il faudra imaginer une interface qui pourrait s'apparenter à des wireframes qui ne sera plus l'apanage des seuls ingénieurs.

Fonctionnalités futures Une fois l'outil développé et adapté aux besoin des formulaires, les équipes souhaiterait aussi modéliser l'arbre de dépendances des pages du site entre elle et de les liée aux donnée SEO déjà présentent.

Problématique

3.1 Contexte

L'application du site lesfurets.com est construite sur une architecture JAVA SE packager a l'aide de l'outils de gestion et d'automatisation de production des projets logiciels Maven. Maven s'occupe de gerer les dépendances de code générer certaines classes ainsi que de compiler le projet. Chaque formulaire est représenté au seins du site par un tunnel, toute la logique est est gerer par GWT (Google Web Toolkit) qui s'occupe de générer du code javascript depuis du code java. L'intérêt du framework est de permettre aux développeur java de l'entreprise d'aborder leur code coté client et coté serveur avec la même approche. De plus GWT met l'accent sur des solutions efficaces et réutilisables aux problèmes rencontrés habituellement par le développement AJAX : difficulté du débogage JavaScript, gestion des appels asynchrones, problèmes de compatibilité entre navigateurs, gestion de l'historique et des favoris, etc. Au seins de l'application il existe déjà un projet pour générer des fichiers mdxml et csv représentant les champs du formulaire c'est dans cette partie du code que l'on développera l'outils de modélisation. Pour la modélisation en elle même Il s'agira essentiellement d'identifier les entités logiques et les dépendances logiques entre ces entités. La modélisation des données est une représentation abstraite, dans le sens où les valeurs des données individuelles observées sont ignorées au profit de la structure, des relations, des noms et des formats des données pertinentes. Pour simplifié la problématique, l'équipe de lesfurets.com a décider d'abstraire les champs du formulaire ainsi que les dépendances à l'aide d'un méta model. Un méta modèle décrit la structure des modèles et permet de raisonner sur les modèles comme sur les connaissances de premier niveau.

3.2 Besoins

Du coté des équipes les besoins était double dans un premier lieu donnée une visibilité plus forte a la mécanique déployer pour les utilisateurs au seins des tunnels. Mettre en relations les données récolté sur les parcours utilisateurs et le parcours utilisateurs lui même. Et d'autre part la visualisation des modifications possible du graph lui même

Analyse du problème

Dans un premier temps ma démarche a été de trouver une solution pour modéliser les champs du graph et de pouvoir les modéliser sous la même forme qu'il apparaisse sur le formulaire lui-même. C'est à dire que l'idée était d'observer le lien entre un champ texte "Avez vous déjà souscrit une assurance auto ?" et un champ date "Quand avez vous obtenu votre permis de conduire"

4.1 Eclipse Sirius

Pour cela il m'a été suggéré d'utiliser un framework d'Eclipse : Sirius. Comme exposé plus haut Sirius est un outil spécialisé dans la modélisation de toute forme de problématique que peut se transposer en problème informatique.

4.2 Application WEB

Pour ma part j'ai aussi proposé une seconde solution qui correspondait peut-être plus à mon background. La solution était d'utiliser des librairies javascript pour visualiser les champs les layouter avec du HTML/CSS et de les propulser dans l'espace de maintenance de l'application en même temps que le packaging de base

4.3 Donnée Business

Pour ma part j'ai aussi proposé une seconde solution qui correspondait peut-être plus à mon background. La solution était d'utiliser des librairies javascript pour visualiser les champs les layouter avec du HTML/CSS et de les propulser dans l'espace de maintenance de l'application en même temps que le packaging de base

- Récupérer les données à l'instant où la page est affichée
- Mettre en cache à des périodes les données nécessaires
- Réagir de manière réactive aux données mises à jour pour afficher une évolution des données en fonction des changements dans l'architecture des tunnels

4.4 Autre utilisations

Construire à l'aide de metamodel est une pratique courante chez lesfurets.com ainsi il m'est possible de réfléchir à d'autres utilisations de l'outil sans reprendre beaucoup la structure et le code utilisé.

4.4.1 Plan du site

Les pages du site suivent une logique propre ainsi qu'un modèle prédéfini avec comme information la JSP utilisée, la version mobile et la version desktop ainsi que les CSS utilisées il m'est ainsi possible de mettre en place un graph de toutes les pages avec les données sus-mentionnées. De plus si les liens proposés dans le site qui sont aujourd'hui des strings étaient remplacés par des valeurs dans le code on pourrait en plus visualiser le mail complet du site sans parcourir tout le code de chaque page à la recherche d'attribut href.

4.4.2 CRM - Envoi de mail

Lesfurets.com disposent d'un outil leur permettant d'envoyer des mails de services à leur client. Tous les champs utilisés dans ces mails sont présents dans le code et on pourra proposer dans l'outil un moyen de visualiser les mails et leurs dépendances vers les champs existants.

Solution

5.1 Eclipse : Sirius

Les premiers début du développement d'un modele pour Eclipse Sirius fut très fastidieux, après avoir passé un bout de temps sur le tutorial présent sur le site je me suis confronté a un manque d'autres exemples me permettant de vraiment prendre en main Sirius.

5.2 Page de maintenance Web : Javascript

Ma premier demarche a été de trouver une bibliotheque capable de dessiner un gros graph orienté cyclique. Mon premier choix s'est porté sur d3.js qui est surement la bibliotheque la plus populaire pour tracer des graphes. Seulement celle-ci est performante pour creer des graph batton, camembert, blabla J'ai ensuite fait le choix de m'orienter vers google pour definir la library a utiliser, qui m'a rediriger vers une question similaire sur stackoverflow. J'ai eu le droit a un comparatif complet des existent sur le net, avec leur avantages et inconvenient. Mon choix s'est porter sur cytoscape.js. Cytoscape avait l'avantage de realiser rapidement un graph similaire a mes besoin avec une integration avec jQuery et/ou Angular pour la manipulation du DOM. Il est de plus sous licence GPL ce qui en faisait un projet open source interessant pour moi. De plus Cytoscape propose par default une dizaine de layout utilisable pour explorer de differentes maniere le graph

5.3 Cytoscape.js + Features

Comme explicité precedement Cytoscape permet d'utiliser jQuery et/ou AngularJS dans la manipulation du dom. C'est ainsi qu'une fois que j'ai reussi a generer le graph de dependance des champs du formulaire sous forme de tableau JSON interpretable par Cytoscape j'ai pu mettre en place des mecanismes asynchrone pour regenerer le graph.

5.3.1 Zoomer, Déplacer les éléments, Se déplacer

Une fois le graph afficher il m'a paru essentiel de pouvoir laisser l'utilisateur zoomer, se déplacer et déplacer les éléments du graph

5.3.2 Choix du tunnel

La premiere fonctionnalité que j'ai voulu implementer etait de pouvoir cliquer sur un bouton pour afficher le formulaire rechercher et de recalculer le graph sans changer de page. Je me suis servi des mecanismes proposer par la bibliotheque et Angular pour faire un appel serveur asynchrone pour recharger le graph.

5.3.3 Changement du layout

De la meme maniere que pour l'affichage d'un tunnel different, pour le recalcul en fonction du layout, je me suis servi des mecanismes de AngularJS

5.3.4 Recherche d'un champs et autocompletion

Pour pouvoir rendre l'outil plus user-friendly, j'ai decider de proposer aux utilisateur de pouvoir rechercher une question parmi les question proposer et ainsi pouvoir zoomer sur celle-ci sans avoir a les rechercher visuellement

5.3.5 Champs du formulaire

Dans un premier temps l'idée etait de pouvoir differencier les champs du formulaire et les faire ressembler aux champs du site

5.3.6 Disposition du graph

Par la suite il a fallu reprendre le graph pour le faire ressembler a un formulaire du site

5.3.7 Modification des elements et des dependances

Une fois le graph ressemblant au formulaire present sur le site il a fallu mettre en place une fonction de modifications des elements du graph. Et une fonction de modifications des dependances entre les champs.

5.3.8 Modification du label du champs

La premiere fonction était sans doute celle qui avait le moins d'incidence sur le formulaire. On peut juste changer le label afficher sur le formulaire

5.3.9 Modification type du champs

On peut à l'aide d'une autocompletion avoir accès a tous les types de champs present sur le tunnel et modifier le type du champs sur lequel on est

5.3.10 Suppression d'une dependance

Il s'agit juste de mettre en evidence les champs touché par la suppression d'une dependances. Les dependances ne sont pas interdependantes.

5.3.11 Sauvegarde etat du graph

Une fois les fonctions de modifications mise en place il a paru evident qu'il a fallait pouvoir sauvegarder l'etat du graph modifié pour pouvoir l'envoyer par la suite. La premiere approche simple était de pouvoir faire une capture d'ecran de l'etat du graph dans une resolution assez grande pouvoir voit toute les details du formulaire. La seconde approche fut de sauvegarder un fichier JSON avec toutes les données modifié ainsi que les position des champs

5.3.12 Charger un graph

Une fois les graph sauvegardable il a fallu pouvoir les reafficher a l'aide d'un module d'import. Une balise input file et un recalcul du graph nous a permis d'implementer cette fonction rapidement

5.3.13 Affichage des differences

Une fois le graph importer nous avons voulu aussi proposer une visualisation des differences avec la version presente et une version importer.

5.3.14 Not User-friendly

Les furets mette leur site en production de maniere quotidienne, il nous a paru pertinent comme fonctionnalité de pouvoir affichage la presence de difference entre le graph en de la version du site en production et la version a mettre en production sans passer par l'interface graphique. Ainsi on proposera une liste au format texte des differences entre ces deux versions.

5.4 Données Business

5.4.1 Récupération des données

Apres differentes discussion avec les equipes en charge de la conception de l'application nous avons decider de ne pas faire des appels SQL a chaque affichage du graph. L'entreprise etant en train de migrer les données sur cassandra toute intervention sur la base serait inutile. Cependant des batches sont deja en place pour recuperer des informations a intervalles regulier, nous avons donc opter dans un premier par recuperer les informations a chaque fois que les batches fait leur jobs. Ainsi nous avions deja un traitement de l'information. Il nous a fallu quand meme definir quels informations recuperer et trouver une moyen de les superposer au graph. Nous avons decider de tracer le graph dans un premier temps et mettre en mecanisme en place an angular qui permet d'ajouter les données business a la demande l'utilisateur

5.4.2 Affichage des données

Une fois le traitement et la recuperation des données réalisé nous avons decider de mettre ses données en evidence sous forme d'un indicateur allant du rouge au vert pour le taux d'abandon du formulaire, avec une affichage plus détaillé lors du clique.

Gestion de projet

Après une semaine passé à essayer de mettre en place une solution technique élégante, il est apparu évident pour mon tuteur que nous devons définir une methodology pour construire l'outils de modélisation. Au seins de l'équipe IT de LesFurets.com, une methodology Agile type Kanban à été mise en place.

6.1 Agile

Les méthodes agiles sont des groupes de pratiques de projets de développement en informatique (conception de logiciel), pouvant s'appliquer à divers types de projets. Elles ont pour dénominateur commun l'Agile manifesto. Rédigé en 2001, celui-ci consacre le terme d'« agile » pour référencer de multiples méthodes existantes. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du client en priorité aux termes d'un contrat de développement. Les méthodes agiles reposent sur une structure (cycle de développement) commune (itérative, incrémentale et adaptative), quatre valeurs communes déclinées en douze principes communs desquels découlent une base de pratiques, soit communes, soit complémentaires. Les méthodes pouvant se qualifier d'agile depuis la publication de l'agile manifesto sont le RAD (développement rapide d'applications) (1991) avec DSDM, la version anglaise du RAD (1995) ainsi que plusieurs autres méthodes, comme ASD ou FDD qui reconnaissent leur parenté directe avec RAD. Les deux méthodes agiles désormais les plus utilisées sont : la méthode Scrum qui fut présentée en 1995 par Ken Schwaber puis publiée en 2001 par lui même et Mike Beedle pour enfin être diffusée mondialement par Jeff Sutherland ainsi que la méthode XP Extreme programming publiée en 1999 par Kent Beck. Un mouvement plus large (management agile) couple les valeurs agiles aux techniques de l'amélioration continue de la qualité (plus particulièrement le Lean). On constate un élargissement de l'utilisation d'agile à l'ensemble de la structure de l'entreprise/

6.2 Kanban

Kanban est un terme japonais signifiant « fiche » ou « étiquette ». Cette méthode a été initialement mise en place dans les usines Toyota fin dans les années 60. Parmi les outils agiles, Scrum et Kanban sont aujourd'hui les plus utilisés dans le cadre de la réalisation de projets informatiques. Ce sont tous deux des

outils de processus mais néanmoins bien différents. Il est à noter que Kanban est plus adaptatif que Scrum qui fixe un cadre de travail plus rigide.

L'approche Kanban consiste globalement à visualiser le Workflow (Le processus de traitement d'une tâche). On met en place un tableau de bord des items (demandes). Chaque item est placé à un instant donné dans un état. L'item évolue jusqu'à ce qu'il soit soldé. Chaque état du tableau peut contenir un nombre maximum prédéfini de tâches simultanées (défini selon les capacités de l'équipe) : on limite ainsi le WIP (Work In Progress). Il est primordial, pendant l'exécution des tâches, de mesurer le "lead-time". Il s'agit du temps moyen pour compléter un item. Cette durée sera progressivement de plus en plus courte et prévisible.

Les intérêts de la mise en place de cet outil Kanban sont principalement :

- Possibilité de mise en place progressive de la méthodologie Agile (moins directif que Scrum)
Les points de blocages sont visibles très tôt. La collaboration dans l'équipe est encouragée pour résoudre les problèmes de manière corrective, le plus tôt possible.
- On peut se passer de la notion de sprint. Un sprint d'une ou deux semaines n'est pas envisageable dans certains cas de figure (réactivité supérieure exigée). La méthodologie Kanban est donc utilisée dans des services de support au client (gestion des tickets d'incidents).
- Facilité de communication sur l'état d'avancement du projet.
- La Définition du Done (Ensemble de critères permettant de considérer la tâche comme traitée) permet de garantir un niveau de qualité constant et défini de manière collective.

6.3 Kanban chez lesFurets

Depuis 3 ans lesFurets utilisent la méthodologie Kanban. Toutes les tâches sont gérées depuis JIRA, un logiciel disponible depuis un client web. JIRA permet la création de tickets qui correspondent à des tâches. Ces tâches peuvent être attribuées aux personnels de l'IT à traiter, peuvent changer d'état. Le processus décrit dans le tableau KANBAN est automatisé pour les tickets et une vue sous forme de tableau KANBAN est disponible. Mais pour autant il existe un tableau physique représentant les tâches et leur état pour respecter le concept de management visuel.

6.4 Integration continu

La culture desFurets est de très Test Driven Development c'est à dire que des tests sont développés avec chaque fonctionnalité déployée. C'est ainsi qu'à chaque fois qu'un développeur termine une fonctionnalité, les modifications doivent passer les batteries de tests complètes. C'est l'intégration continue, ce processus est permis grâce à l'automatisation.

6.5 Déploiement continu

De plus lesFurets ont aussi adopté une attitude très DevOps. DevOps est un mouvement visant à réduire la friction organisationnelle entre les "devs" (chargés de faire évoluer le système d'information) et les "ops" (chargés d'exploiter les applications existantes). Ce que l'on pourrait résumer en travailler ensemble pour produire de la valeur pour l'entreprise. Dans la majorité des entreprises, la valeur sera économique mais pour d'autres elle sera sociale ou morale. C'est ainsi que les mises en production sont réalisées par

les developpeurs, il m'est arriver de mettre en productions des fonctionnalités que j'avais developpés à plusieurs reprises

6.6 Feature Branch

Enfin pour assurer une developpement optimale de chaque fonctionnalité la methodologie utilisé est Feature Branch. Chaque nouvelle fonctionnalité developé est developpé à partir de la derniere version du master. Une fois le developpment terminée la branche doit passer par une mecanisme d'octopus qui permet de detecter des conflits present avec d'autre developpement en cours.

6.7 Ma methodologie

En vue de toute l'architecture mise en place chez lesFurets il m'a fallu m'adapter pour pouvoir developper l'outil tout en respectant l'eco systeme mise en place et pouvoir profiter des avantages produitzs par le workflow mise en place.

Tout d'abord lors d'un BBL (une conference pendant la pause dejeuner) j'ai entendu parler du Kanban "Perso" qui etait exempt de certaines regles du Kanban. Par la suite une des personnes faisant partie de l'equipe dont je faisait partie m'a pris en main pour poser les bases de mon workflow. C'est ainsi que sur un tableau kanban personnel (informatiser à l'aide de Trello) je me suis imposer des colonnes

Backlog | To Do | Doing | Done