Refonte technique : la formule magique pour ne plus en souffrir

Mickaël Wegerich

Préambule

Je ne vais pas montrer de code.

Je parle d’une refonte back, côté front les principes sont les mêmes, la réalisation est différente.

En novembre 2024, j’ai mené une étude sur les refontes techniques où j’ai récolté et analysé ~ 80 réponses. Dont 60% venaient de profil de type CTO.

Brisons la glace

Qui est dans un projet de refonte ?

Brisons la glace

Qui va débuter une refonte mais ne sait pas par où commencer ?

Brisons la glace

Qui ne souhaite pas devoir faire une refonte (compliquée) ?

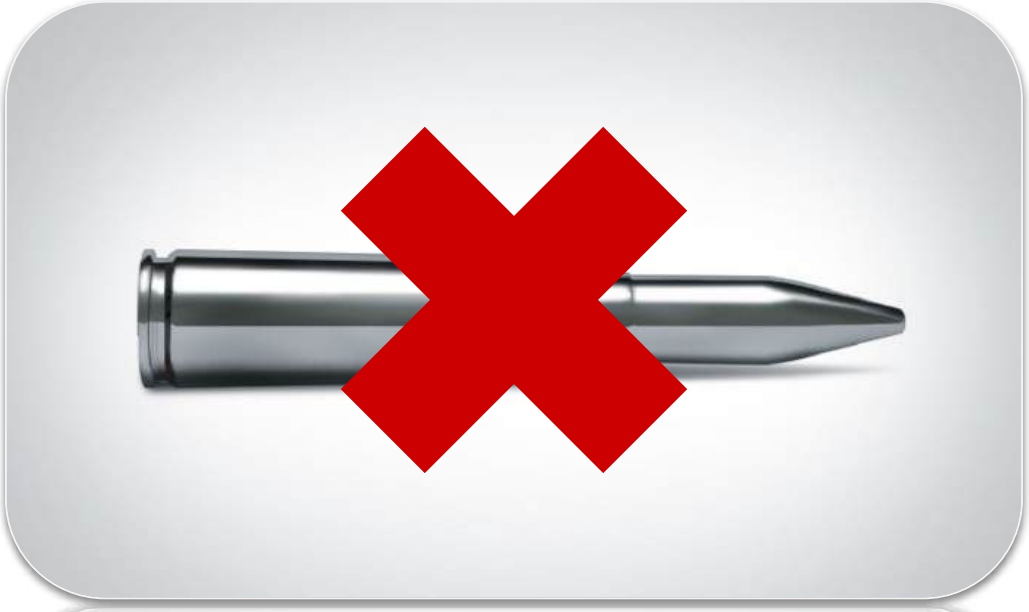
Mickaël Wegerich

PDG de 3 Lights Technology
Développeur et Coach Craft • Formateur

Plus de 10 ans dans le monde informatique
~ 5 ans dans la gestion de projet
~ 4 ans comme consultant senior chez OCTO Technology
Freelance depuis janvier 2022

Expatrié en République tchèque 🇨🇿

Il n’y a pas de Silver Bullet !



Contexte

Quelle est la situation avant le lancement de la refonte ?

Contexte

Situation initiale

Application B2B de gestion de planning

Contexte

Situation initiale

Application B2B de gestion de planning

Je suis arrivé 36 jours après le début du projet

Contexte

Situation initiale

Application B2B de gestion de planning

Je suis arrivé 36 jours après le début du projet

Deux équipes front et back

Contexte

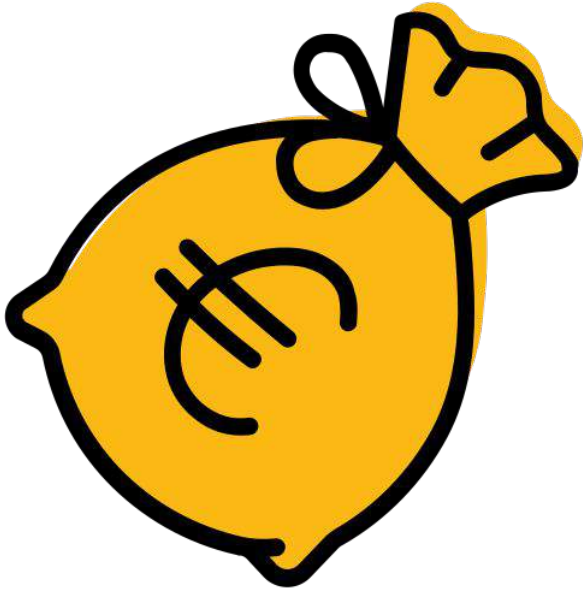
Situation initiale

Application B2B de gestion de planning

Je suis arrivé 36 jours après le début du projet

Deux équipes front et back

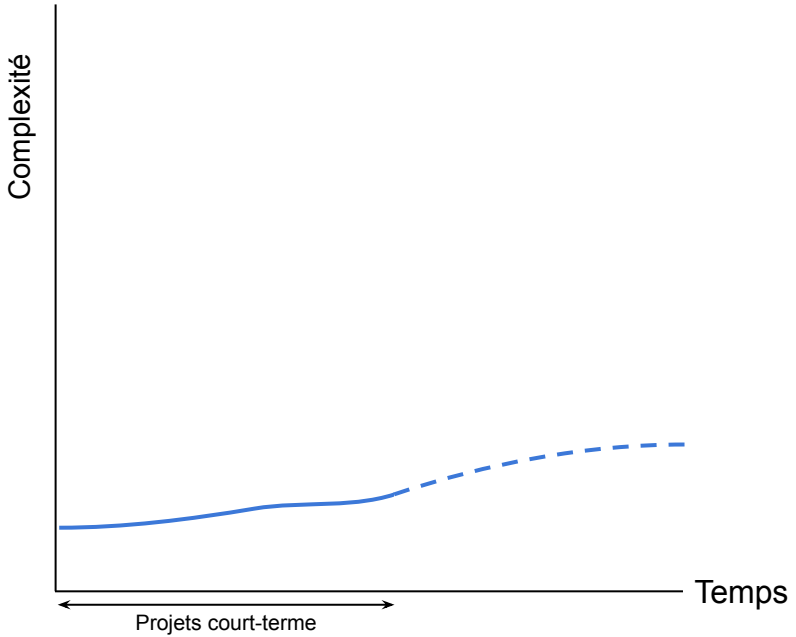
Projet fonctionne en Production sans anomalie majeure



Contexte

Élément déclencheur

En 2020 (après 2 ans), réunification des 2 équipes

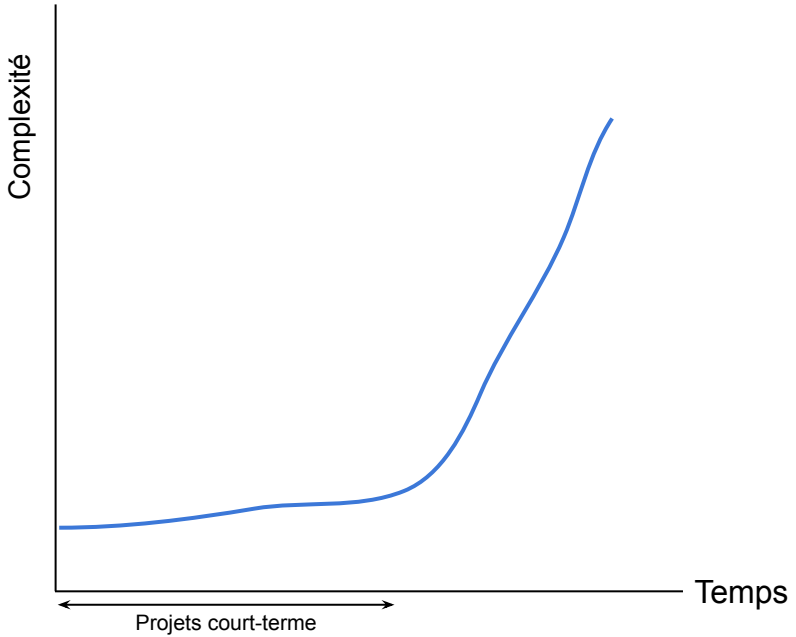


Contexte

Élément déclencheur

En 2020 (après 2 ans), réunification des 2 équipes

Le métier demande une fonctionnalité qui ne peut pas être implémentée facilement avec le code actuel



Lancement de la refonte

Quel est le plan ?

Lancement de la refonte

Se mettre d'accord sur le constat

Nouvelle fonctionnalité difficile à faire en l'état

Discussions avec toute l' équipe (PO inclus)

Il faut tout refaire

Le temps et le coût nécessaires à la mise en place de nouvelles fonctionnalités

est la source du déclenchement d'une refonte dans **35%** des cas.

Source : Refonte technique : Avez-vous toutes les billes pour prendre la bonne décision ?

Et pour vous ?

Le coût d’ajout d’une nouvelle fonctionnalité

est, **40%** du temps, un élément pour convaincre.

Source : Refonte technique : Avez-vous toutes les billes pour prendre la bonne décision ?

Lancement de la refonte

Comment et quoi annoncer au client ?

Le projet rapport de l'argent à l'entreprise.

Ne pas vendre une refonte Big Bang, ça fait peur et ça ne fonctionne jamais.

Nous devons tout refaire, petit à petit, pour pouvoir implémenter la nouvelle fonctionnalité.

MAIS nous allons également :

- faire d'autres fonctionnalités,
- corriger des bugs,
- moderniser le code, l'architecture, le projet, …

Ralentir aujourd’hui pour accélérer demain, sans engagement de temps.

Lancement de la refonte

Quelle relation avec le client ? ***Confiance***

Être transparent : décisions, problèmes, …

Lancement de la refonte

Quelle relation avec le client ? ***Confiance***

Être transparent : décisions, problèmes, …

Montrer que ça avance : Mettre en production régulièrement (plusieurs fois par semaine)

Lancement de la refonte

Quelle relation avec le client ? ***Confiance***

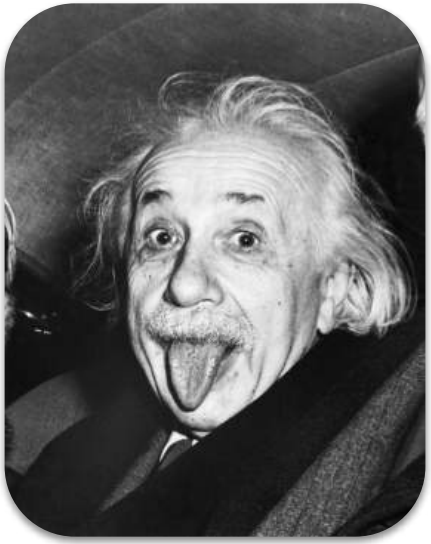
Être transparent : décisions, problèmes, …

Montrer que ça avance : Mettre en production régulièrement (plusieurs fois par semaine)

Avoir l'autonomie sur la gestion du quotidien : L'équipe gère ses priorités en fonction de ce qu'elle vit et celles données par le client

C’est parti !

“Nous ne pouvons pas résoudre nos problèmes avec la même pensée que nous avons utilisée lorsque nous les avons créés.”



- Albert Einstein

Environ **33%** du temps,

L’expérience et la stabilité de l’équipe sont indispensables pour la réussite d’une refonte.

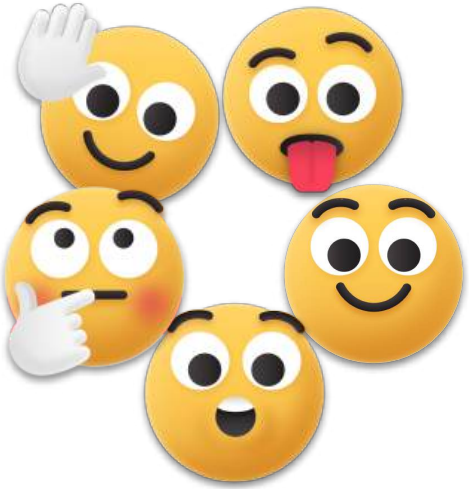
Source : Refonte technique : Avez-vous toutes les billes pour prendre la bonne décision ?

Composition de l’équipe

Client et ses équipes

1 PO

1 QA



Entre 4 et 6 devs (1 TL, 1 sénior, 2 confirmés et 1 à 2 juniors)

État des lieux

On fait peau neuve !

HAPI / JavaScript

Code testé

Début de Clean Architecture

Code entremêlé, difficile à lire et à refactorer

Express / TypeScript

DDD / Monolithe modulaire

Code testé et en français

Clean Architecture

Architecture événementielle / CQRS

💰 API existante fonctionne en l'état

À l'extérieur de l'équipe

OPS et dépendances externes

<u>OPS/Outils utilisés :</u>

- AWS / Kubernetes / Helm
- ArgoCD
- Prometheus / Grafana / Loki
- RabbitMQ

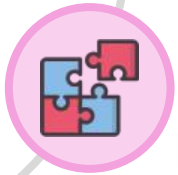
<u>Dépendances externes :</u>

- Auth0
- Business API
- Équipe mobile
- Un autre gestionnaire de planning

Strangler fig pattern

Stratégie

Première étape



Pratiques

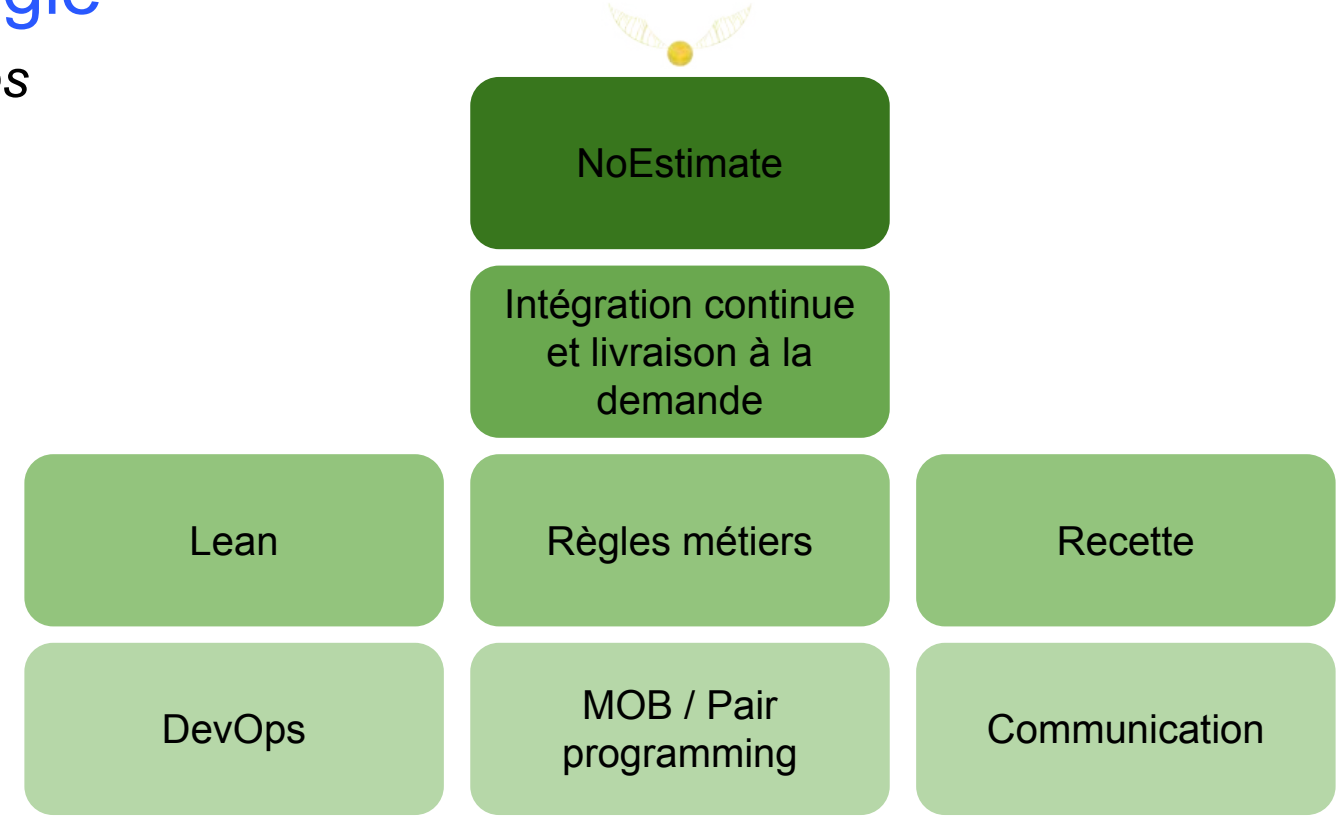
Stratégie

Strangler fig pattern

Le principe est de remplacer incrémentalement l'ancien par le nouveau, de manière à étouffer le Legacy.

Stratégie

Pratiques



Stratégie

Pratiques

Communication	Relation étroite avec les différentes équipes : AU, OPS et celle de l'API Business

Stratégie

Pratiques

Règles métiers

Voir directement la source et avoir les vrais règles pour les réécrire correctement

Stratégie

Pratiques

Recette

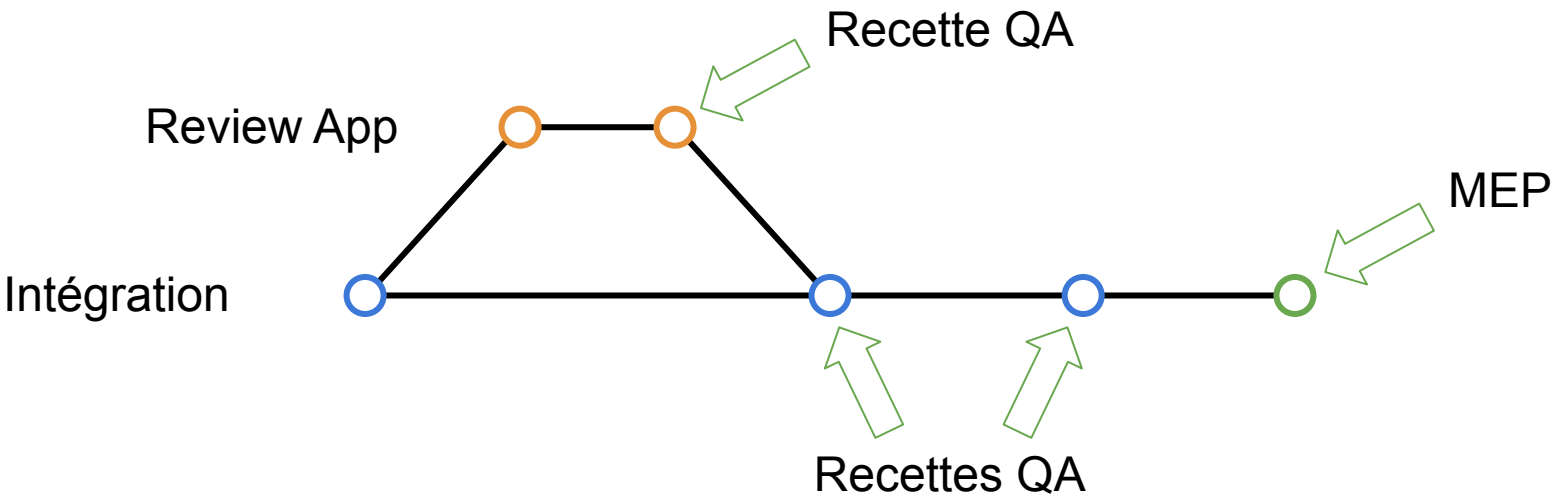
Uniquement le ticket

Chemin critique et tests de non régressions *(demandés par le dev)*

Flow Git et recette

Stratégie

Pratiques : Flow Git et recette



Qui a déjà dû faire vivre 2 APIs différentes en même temps, ou doit le faire ?

Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps

Nous ne voulons pas intervenir sur l'API v5 (Legacy) !

Elle fonctionne et il ne faut pas introduire de régressions.

Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps

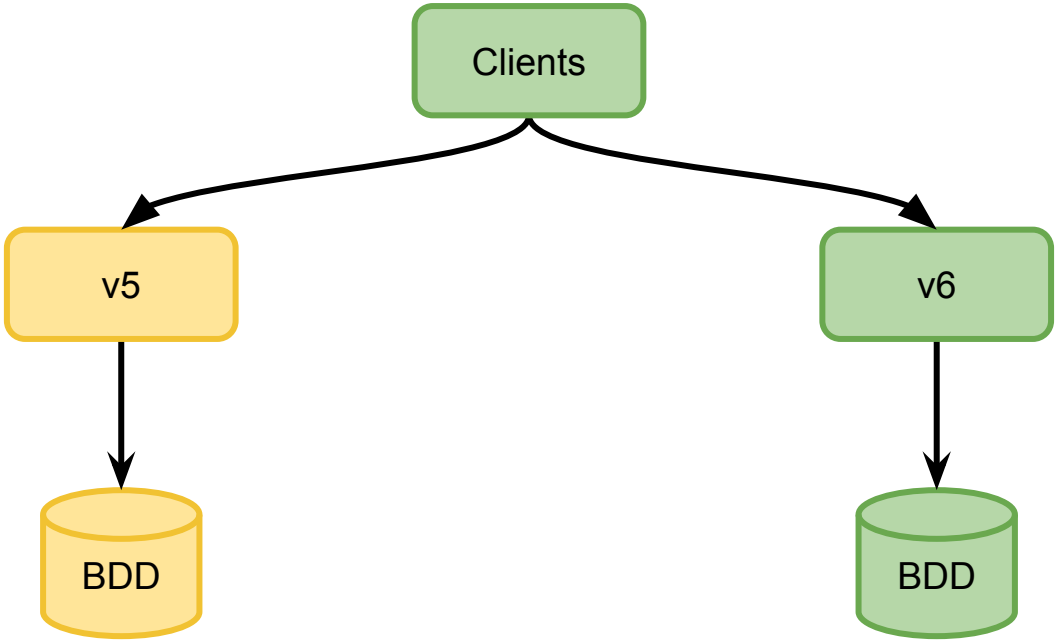
Nouvelle version de l'API, passage de v5 à v6

Nouveau dépôt de code

Nouvelle(s) base(s) de données

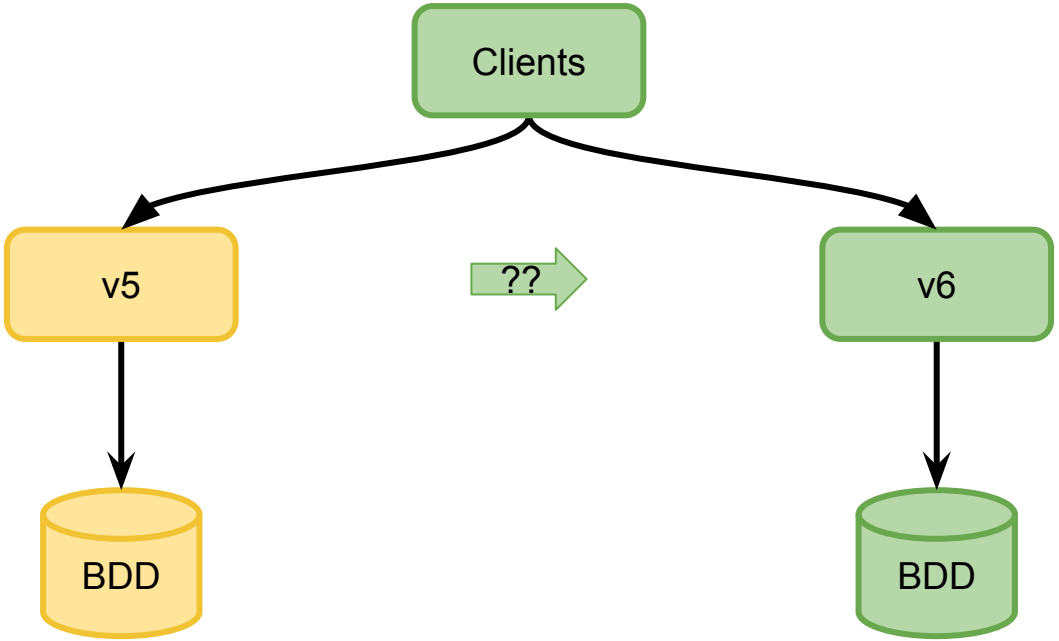
Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps



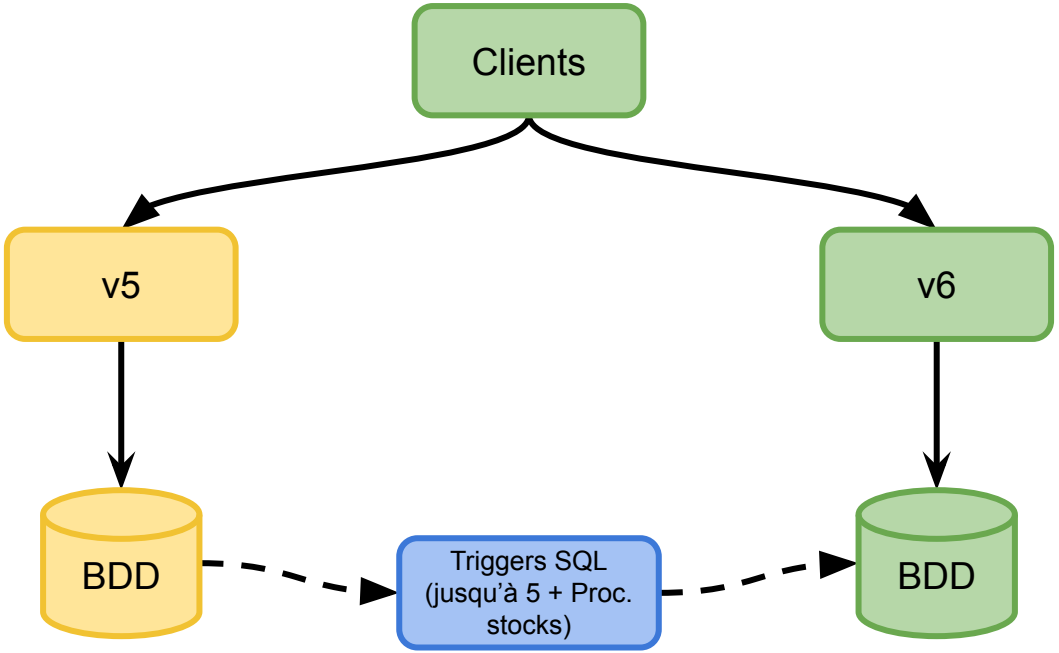
Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps



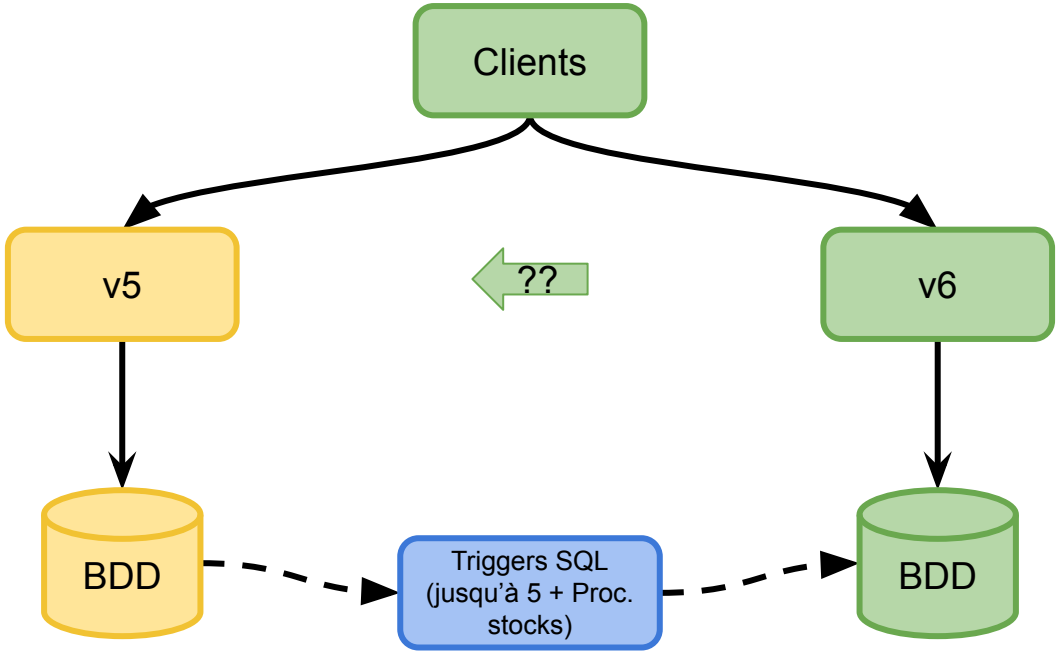
Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps



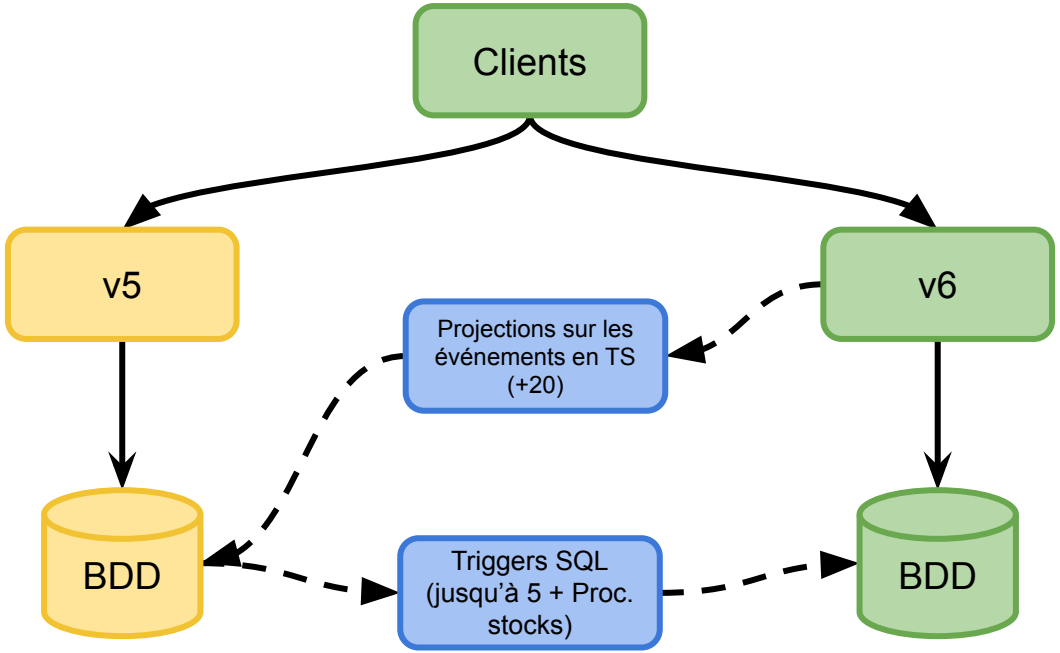
Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps



Stratégie

Pratiques : Faire vivre 2 APIs différentes en même temps



Refactoring de projet signifie que le projet sera forcément plus crade **pendant,** qu'il ne l'était **avant** et qu'il ne le sera **après**.

Stratégie

Première étape

Faire “Hello World” jusqu’à la prod !!!!

Rappel nous devons livrer en plus de la refonte des nouvelles fonctionnalités et des corrections de bug.

Chiffre : Initialisation du repository Git le 01/07/2020, première MEP le 03/07/2020.

Le suivi

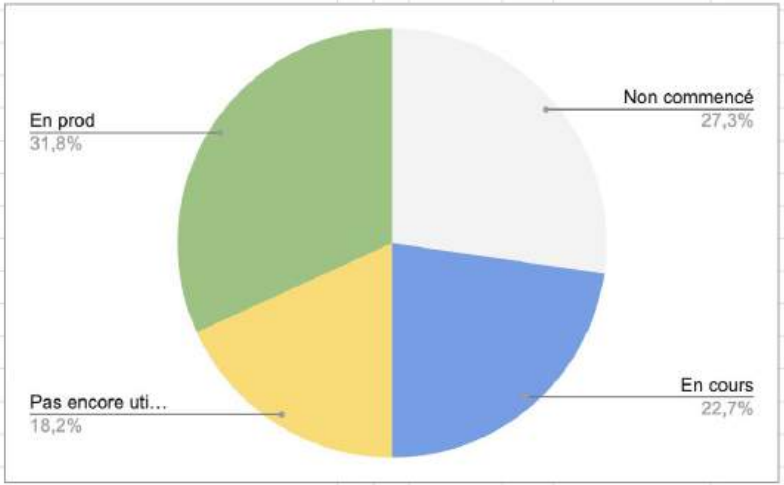
Suivre l’état d’avancement du décommissionnement, en se basant sur les points d’entrées et les contournements pris (ex. Trigger)

- Ce qui n’est pas encore fait
- Ce qui est en cours
- Ce qui est fait mais pas en prod
- Ce qui est fait mais pas utilisé en prod
- Ce qui est utilisé en prod
- Ce qui est complètement décommissionné

Le suivi

Sur legacy	Legacy	Action	Décomissionée	Commentaire
	Endpoint 1	GET	E	
	Endpoint 2	GET	E	
	Endpoint 3	GET	E	
	Endpoint 4	POST	E	
	Endpoint 5	POST	E	
	Endpoint 6	DELETE	P	
	Endpoint 7	POST	P	Commentaire
	Endpoint 8	DELETE	P	
	Trigger 3	Trigger	P	
	Cron 1	Cron	Y	
	Endpoint 11	GET	Y	
	Endpoint 12	GET	Y	
	Endpoint 13	DELETE	Y	
	Endpoint 14	POST	Y	
	Endpoint 15	DELETE	Y	
	Endpoint 16	POST	Y	Commentaire
	Endpoint 17	POST		
	Endpoint 18	POST		
	Endpoint 19	GET		
	Cron 2	Cron		
	Trigger 1	Trigger		Commentaire
	Trigger 2	Trigger		

6	Non commencé	6	GET	0	PUT	3	Trigger
5	En cours	7	POST	4	DELETE	2	Cron
4	Pas encore utilisé en prod						
7	En prod						



“It failed miserably and probably led to the closure of the company.”

(Cela a échoué lamentablement et a probablement conduit à la fermeture de l’entreprise.)

- Participant anonyme au questionnaire

Conclusion

Conclusion

Quelques chiffres du 01/07/2020 au 31/12/2022

C’est 640 jours ouvrés

<u>Fonctionnels :</u>

- ~ 1250 établissements clients
- ~ 15000 commandes saisies mensuellement
- ~ 10000 utilisateurs uniques quotidien sur les applications mobiles

<u>Techniques :</u>

- ~ 50 endpoints
- 161 MEP, soit 1,2 par semaine
- ~ 5 jours en moyenne pour mettre un ticket en production

Conclusion

- Nous n’avons jamais estimé
- Peu/pas de bugs
- Y aller petit à petit : Pas d’anticipation des problèmes
- Outillage et Monitoring
- Le code refait est toujours utilisé encore maintenant
- Plus d’API Legacy

- Difficile de toujours penser à tout
- Est-ce que tout le monde (devs) a bien été embarqué ?
- Beaucoup d’organisation
- + de 2 ans que pour le back (faire la même chose pour le front)
- Mettre en place des KPIs pour mesurer la réussite
- …

“Lors de la conception d’une nouvelle application, nous devons tout faire pour faciliter son future remplacement, afin qu’il puisse se faire de manière élégante.

Tout ce que nous faisons, c’est écrire aujourd’hui le logiciel hérité de demain.”

- Martin Fowler

Me suivre

/mickaelwegerich
/3-lights-technology

/3lights-technology

Me contacter

mickael.wegerich@3lights-technology.com

https://3lights-technology.com

Avez-vous des questions ?