

Rapport GPSTL 2025-2026

Ticket #1664 - POC Moteur de Recherche d'Images

Boudrouss Réda



Sorbonne Université
France

Contexte et objectif

Nous sommes le **30 septembre 2030**. En tant qu’Alice, alumni STL-INSTA (promotion 2025–2026) avec cinq années d’expérience dans les SI parisiens, je reçois de X. un courriel m’annonçant officiellement la constitution de l’équipe projet pour le **Ticket #1664** (POC d’un moteur de recherche d’images).

L’objectif de ce document est de fournir, en pièce jointe de ma réponse à X., un **plan d’action structuré** ainsi qu’un **calendrier prévisionnel** pour l’équipe projet.

1. Courriel à X.

Objet : [#1664] Plan d’action prévisionnel pour le POC moteur de recherche

Bonjour X.,

Merci pour ton mail, je suis très heureuse de faire partie de ce nouveau projet !

Vous trouverez ci-joint une proposition de plan d’action pour le POC du moteur de recherche. Il contient un calendrier prévisionnel et la répartition des tâches.

En partant du principe que nous débutons la semaine prochaine (soit début le **6 octobre 2030**), et en suivant ce plan d’action, nous estimons une **livraison prévue pour le 12 novembre 2030**.

N’hésite pas à revenir vers nous si tu souhaites des ajustements.

Bien cordialement,

Alice WONDROUS-IT

2. Équipe projet et rôles

- **Alice** (moi-même) : cheffe de projet technique, coordination globale, conception fonctionnelle, architecture, pilotage des risques.
- **Bob** : lead tech, arbitrages techniques critiques, conception des composants cœur du moteur, revue de code sur les parties sensibles. Très fort, inhumain, une vraie machine ; guru de la tech toujours tiraillé entre mille et une missions ; très peu disponible ; 10 ans dans les SI les plus connus du Silicon Sentier.
- **Cindy** : UX / UI, conception des wireframes et maquettes, cohérence de l’expérience utilisateur. Designer UX, libre comme une freelance depuis 2 ans ; avait commencé comme infographiste avant d’évoluer en designer technique.
- **David** : QA, stratégie de test, écriture de plans de test, pilotage de la non-régression. Associé à l’équipe depuis trois mois ; QA de la boîte depuis 1 an ; 3 ans d’expérience dans les SI.
- **Eddy** : développeur apprenti, implémentation incrémentale des fonctionnalités, automatisation simple, support tests. Apprenti en alternance SU UPMC.

3. Phases du projet

3.1 Cadrage fonctionnel et technique (Semaine 1)

- Relecture du ticket général (#1664) et clarification des contraintes métier.
- Ateliers rapides avec les métiers pour figer les quatre fonctionnalités clés du moteur.
- Définition des scénarios d’usage principaux (recherche par mot-clef, recherche RegEx, classement, recherche multiple).
- Définition des métriques de performance et des scénarios de montée en charge.
- Lancement de la collecte d’une base d’URLs d’images suffisamment volumineuse pour le POC.

3.2 Conception UX / UI et architecture (Semaines 2–3)

- Ateliers rapides avec Cindy pour dessiner les écrans clefs et les parcours utilisateurs.
- Spécification de l’API (endpoints, formats des requêtes / réponses, codes d’erreur).

- Conception du modèle de données (table d'index, métadonnées, structure de stockage des URLs).
- Choix de l'architecture WebAPI et des technologies principales.
- Mise en place de l'usine logicielle minimale (CI, repo, tests automatiques).

3.3 Développement itératif du moteur (Semaines 3–6)

- Implémentation incrémentale des endpoints :
 - /myImageFromKeyword/<str:name>/
 - /myImageFromRegEx/<str:name>/
 - /myImagesFromManyKeywords/<str:name>/
- Mise en place du mécanisme de classement (critère de pertinence déterministe).
- Récolte et alimentation d'une première **banque d'images** significative.
- Intégration avec la base d'images et gestion des performances (indexation, caches).

3.4 Stratégie de test et validation (Semaines 6–7)

- Définition des cas de test unitaires et d'intégration.
- David et Eddy construisent une batterie de tests fonctionnels et de charges.
- Mise en place de jeux de données pour la montée en charge.
- Exécution de campagnes de tests et corrections.

3.5 Tests de montée en charge (Semaine 7)

- Définition des scénarios de charge (volume de requêtes, volume de données).
- Exécution de scénarios de montée en charge et collecte de métriques (latence, débit, erreurs).
- Mesure des performances (latence, débit, consommation de ressources).
- Ajustement (indexation, mise en cache simple, paramétrage).

3.6 Finalisation du POC et restitution (Semaine 8)

- Stabilisation du code et des données.
- Gel de la version POC.
- Rédaction du rapport de tests de montée en charge.
- Rédaction d'un rapport de retours d'expérience, incluant les résultats des tests de montée en charge et les pistes d'industrialisation.
- Préparation de la démonstration et des supports (slides, résumé exécutif).

4. Calendrier prévisionnel et volume de travail

4.1 Répartition globale par phase

Phase	Alice	Bob	Cindy	David	Eddy	Total JH
1. Cadrage fonctionnel et technique	2	1	0	0	1	4
2. Conception UX / UI et architecture	3	2	5	0	2	12
3. Développement itératif du moteur	6	3	0	0	12	21
4. Stratégie de test et validation	2	1	0	4	3	10
5. Tests de montée en charge	2	1	0	3	2	8
6. Finalisation du POC et restitution	3	1	1	1	2	8
Total	18	9	6	8	22	63 JH

4.2 Vue par personne

Membre	Rôle principal	JH prévus	Commentaire
Alice	Pilotage + conception	18	Coordination globale + développements
Bob	Lead tech	9	Ateliers d'architecture et revue
Cindy	UX / UI	6	Conception et itération des écrans
David	QA	8	Stratégie de test + exécution
Eddy	Développement / support QA	22	Implémentation et automatisation

Ce volume reste cohérent avec un POC ambitieux mais réaliste sur environ huit semaines calendaires, en tenant compte des autres engagements de chacun (notamment Bob).

5. Diagramme de Gantt

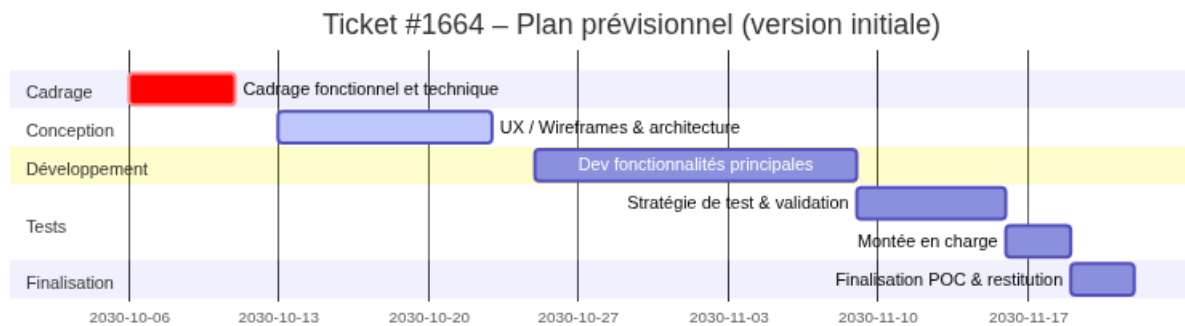


FIGURE 1

Contexte et objectif

Nous sommes le **7 octobre 2030**. Cindy, notre designer UX, quitte brutalement le projet pour des raisons indépendantes du Ticket #1664 (obtention inespérée de sa carte verte des US).

L'objectif de ce document est de :

- proposer à X. une contre-proposition constructive où Alice reprend le rôle UX de Cindy
- présenter une première version des wireframes du moteur
- fournir un Gantt mis à jour, intégrant une semaine de retard

1. Courriel de contre-proposition à X.

Objet : [#1664] Contre-proposition, prise en charge du rôle UX/UI

Bonjour X.,

Merci pour ta réactivité et pour l'information concernant Cindy. Je comprends la situation (félicitations à elle pour sa carte verte!), cependant, pour éviter un nouveau retard et préserver l'élan de l'équipe, je te propose de reprendre immédiatement le rôle UX/UI que tenait Cindy pour le Ticket #1664.

Je pense qu'il serait plus efficace de garder le rôle en interne plutôt que de chercher un prestataire externe.

D'une part, cela nous permettrait de préserver la continuité du projet : j'ai déjà coordonné l'analyse et rédigé un premier plan d'action (ci-joint), et le remplacement de Cindy par une ressource extérieure impliquerait forcément une phase d'onboarding (prise en main du ticket, réunions de transfert, mise en contexte...) qui risquerait de retarder encore le démarrage du développement. Sans parler des éventuels risques fonctionnels, notamment les écarts de vision produit et les allers-retours lents entre les équipes.

Ensuite, l'équipe actuelle est déjà prête et parfaitement alignée. Bob couvre tout l'aspect technique (performance et intégration), Eddy est disponible, formé sur les technologies du projet et extrêmement motivé, tandis que David gère le QA. De mon côté, je peux reprendre le lead sur la partie UX/UI ainsi que la coordination fonctionnelle.

Je joins les wireframes et le Gantt mis à jour à ce message. Je reste à ta disposition pour toutes questions supplémentaires.

Merci pour ta confiance.

Bien cordialement,

Alice WONDROUS-IT

2. Principes UX retenus

Les wireframes suivent quelques principes simples :

- Simplicité : un champ de recherche central, des options avancées repliées par défaut.
- Progressivité : l'utilisateur découvre d'abord la recherche simple, puis la recherche avancée et multiple.
- Lisibilité : résultats présentés avec l'URL, les mots-clefs associés et un indicateur de pertinence.
- Traçabilité : rappel visuel des mots-clefs ou de l'expression régulière utilisée pour la requête.

3. Gantt mis à jour après le départ de Cindy

Le départ de Cindy impose un décalage d'une semaine du lancement effectif du développement, le temps pour Alice de produire les wireframes essentiels. Le décalage est absorbé en reprenant le rôle d'UX tout en maintenant la dynamique de développement.

3.1 Répartition des JH mise à jour

Étape	Équipe	JH
Analyse et Architecture	Alice, Bob	4
Collecte et préparation BDD	Eddy	4
Dév API : Recherche simple et RegEx	Bob (Eddy support)	6
Dév API : Classement	Bob	3
Dév API : Recherche multiple	Alice, Eddy	4
UI/UX	Alice (ex-Cindy)	3
QA	David	4
Montée en charge et optimisation	Bob, David	4
Documentation et packaging	Eddy, Alice	3

Total JH par personne :

- Alice : **11 JH** (au lieu de 9 JH initialement, car elle reprend l'UX)
- Bob : **17 JH** (inchangé)
- ~~Cindy~~ : ~~3 JH~~ (partie)
- David : **8 JH** (inchangé)
- Eddy : **11 JH** (inchangé)

Total : 47 JH (au lieu de 63 JH avec Cindy, car Alice mutualise coordination + UX)

Livraison prévisionnelle : ~18 novembre 2030 (au lieu du 12 novembre, soit 1 semaine de glissement).

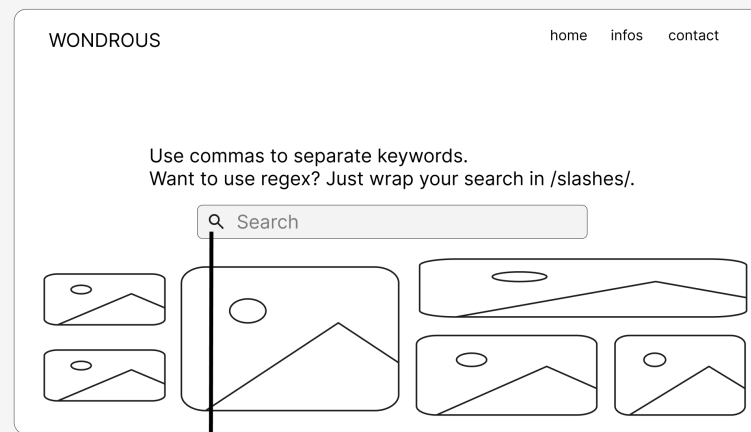
3.2 Principes du nouveau planning

- La conception UX est assurée par Alice sur les semaines 2 et 3, en parallèle partiel avec l'architecture.
- Eddy commence dès la semaine 2 la mise en place de l'infrastructure technique (CI, squelette de l'API).
- Le développement des fonctionnalités de recherche démarre une semaine plus tard, mais sur une base mieux cadrée.



FIGURE 2

L'effort UX est recentré sur Alice, ce qui évite une nouvelle phase d'onboarding pour un prestataire externe et limite le risque de décalage supplémentaire. La charge de développement reste concentrée sur Bob et Eddy, avec une fenêtre de tests toujours suffisante pour sécuriser le POC.



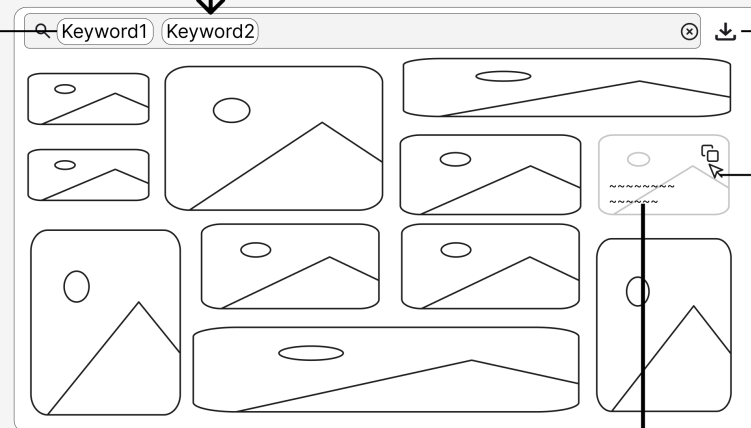
For better web handling maybe add a pagging system in the api ?

→ "Infinite" scroll (more like what we have in db)

On write for the first time

`/?search=<search_bar_content>`

Auto styling for keywords, when users inputs a comma, the keyword is wrapped



On click, downloads the list of images as a json file (will probably be just the response of the request)

→ "Infinite" scroll (more like what we have in the response)

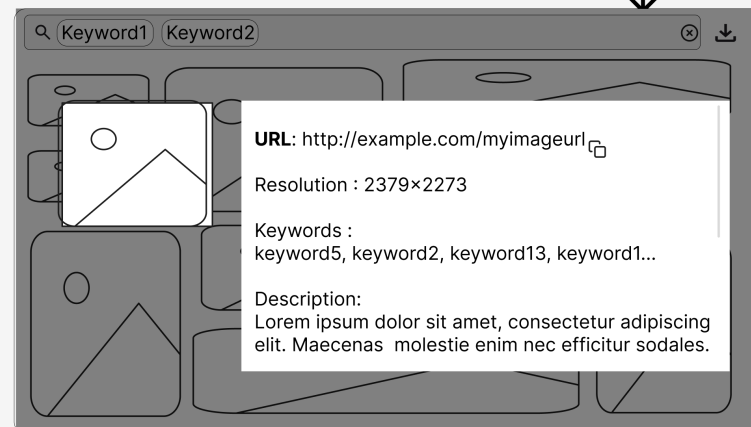
On image hover, "copy url" button shows + image information
On click, copies image url in clipboard

```
search_bar: on_change()
if input.contains(",")
  → Calls /api/myImagesFromManyKeywords/<input>
if input.starts_with("/") && input.ends_with("/")
  → Calls /api/myImageFromRegEx/<input>
else
  → Calls /api/myImageFromKeyword/<input>
```

(When loading, shows an animated placeholder for images)

On any image click

`/?search=<search_bar_content>&focus=<image_id>`



Contexte

Nous sommes le **14 octobre 2030** dans le TGV de retour vers Paris. Bob a proposé d’orienter le développement du Ticket #1664 en **TDD (Test Driven Development)**, avec l’arrivée de **Freddie**, apprenti QA sous la responsabilité de David.

L’objectif de ce document est de :

- proposer à Bob un plan d’action TDD intégrant Freddie dans l’équipe ;
- présenter une batterie structurée de cas de test (16 au total) pour le moteur de recherche d’images ;
- identifier un test volontairement rouge (qui échoue avec l’état actuel du code) afin de piloter la prochaine incrémentation.

1. Courriel à Bob

Objet : Ticket #1664 – Plan d’action TDD et intégration de Freddie

Bonjour Bob,

Suite à ta proposition de piloter le Ticket #1664 en TDD et à l’arrivée de Freddie dans l’équipe, je te partage ci-joint une première version de notre stratégie de tests, ainsi qu’un ensemble de cas de test structurés pour le moteur de recherche d’images.

L’idée est la suivante :

- partir des quatre fonctionnalités clés (recherche simple, recherche RegEx, classement, recherche multiple),
- définir pour chacune une batterie de tests unitaires et fonctionnels,
- utiliser les tests comme langage commun entre Dev et QA (toi, David, Freddie, et moi),
- faire écrire à Freddie une partie significative des tests (sous ta supervision et celle de David),
- intégrer Freddie sur la formalisation des cas de test et l’automatisation (tests API et tests de charge),

Je détaille ci-après l’état actuel du développement et une première liste de cas de tests. Si tu valides l’approche, je propose que Freddie et David la déclinent ensuite en scripts exécutables (par exemple via pytest + outillage de charge), avec revue par toi et moi.

Je t’ai également joint un Gantt mis à jour intégrant le temps nécessaire à la conception et à l’automatisation de ces tests.

Dis-moi si cette approche te convient ou si tu souhaites ajuster la granularité des tests ou leur priorisation.

Merci pour ton soutien,

Alice

2. État actuel du développement

- Un premier endpoint `myImageFromKeyword/<name>/` est disponible, avec recherche exacte sur la table d’index.
- La recherche RegEx est partiellement implémentée mais sans optimisation ni gestion poussée des erreurs d’expression régulière.
- Le classement des résultats est encore naïf (ordre de la base ou tri aléatoire).
- La recherche multiple par mots-clés (`myImagesFromManyKeywords/<name>/`) n’est qu’esquissée : séparation des mots-clés reconnue, mais sélection d’URLs distinctes non garantie.

3. Stratégie TDD globale

- **Niveau 1 – Tests unitaires :**
 - Comportement des fonctions de recherche (mot-clef exact, RegEx, recherche multiple).

- Gestion des cas limites (aucun résultat, doublons d’URL, mots-clefs inconnus).
- **Niveau 2 – Tests d’intégration :**
 - Appels HTTP aux endpoints WebAPI.
 - Format des réponses (schéma JSON, codes d’erreur).
- **Niveau 3 – Tests de non-régression :**
 - Jeux de données de référence pour vérifier que les modifications n’introduisent pas de régressions sur les fonctionnalités clés.

4. Batterie de cas de test (16 tests)

Chaque cas décrit l’objectif, une instance de test et les principaux `assert_true` associés (sur les réponses JSON/HTTP ou sur des fonctions internes du moteur).

ID	Objectif principal	Instances de test (exemples)	Assert(s) clé(s)
T01	Recherche simple – mot-clef exact, 1 résultat	name = “oyster”	1 URL retournée, contient mot-clef “oyster”
T02	Recherche simple – mot-clef inexistant	name = “unicorn”	0 résultat, code HTTP 200, liste vide
T03	Recherche simple – casse insensible (si spécifié)	name = “Sea_Bream” / “sea_bream”	Résultats identiques
T04	Recherche simple – plusieurs résultats triés	name = “crab”	Liste triée par pertinence non aléatoire
T05	Recherche RegEx – motif préfixe	name = “^sea”	URLs dont un mot-clef commence par “sea”
T06	Recherche RegEx – motif suffixe	name = “crab\$”	URLs dont un mot-clef finit par “crab”
T07	Recherche RegEx – motif invalide	name = “[abc”	Erreur contrôlée, code HTTP 400
T08	Recherche multiple – 2 mots-clefs distincts	name = “sea%20bream%1664oyster”	2 URLs distincts, bons mots-clefs
T09	Recherche multiple – 4 mots-clefs (exemple du sujet)	name = “sea%20bream%1664oyster%1664king%20crab%1664caviar”	4 URLs distincts, 1 mot-clef en trop
T10	Recherche multiple – mot-clef inexistant au milieu	name = “sea%20bream%1664xxx%1664king%20crab”	Erreur claire ou URL non documentée
T11	Classement – critère de pertinence stable	même requête exécutée 2 fois	Même ordre de résultats entre exécutions
T12	Performance – temps de réponse borne sur petit dataset	100 requêtes séquentielles	Latence < seuil défini
T13	Performance – temps de réponse sous charge modérée	1000 requêtes réparties	Taux d’erreur acceptable, latence raisonnable
T14	Robustesse – entrée vide	name = “”	Code HTTP 400, message explicite
T15	Robustesse – encodage spécial	name avec espaces (%20) et séparateurs (%1664)	Décodage correct, pas d’erreur interne
T16*	TDD – cas rouge : recherche multiple + contrainte forte	4 mots-clefs, banque partielle	Échec actuel attendu (voir ci-dessous)

T16* est le test rouge TDD : il spécifie un comportement cible que le code actuel ne satisfait pas encore.

5. Cas de test rouge (T16*) – TDD

Objectif : garantir que la recherche multiple retourne, pour chaque mot-clef demandé, un **URL distinct** parmi un ensemble limité d’URLs, même lorsque la banque d’images est partiellement couverte.

- **Instance :**
 - Banque d’images réduite à quelques dizaines d’URLs.
 - Requête : `.../myImagesFromManyKeywords/sea%20breem%1664oyster%1664king%20crab%1664caviar/`.
- **Comportement attendu :**
 - 4 URLs distincts.
 - Chaque URL contient exactement un des mots-clefs demandés dans ses métadonnées.
 - Pas de doublon d’URL même si un même URL pourrait contenir plusieurs mots-clefs.

Pourquoi ce test échoue aujourd’hui ?

L’implémentation actuelle de la recherche multiple (version naïve) :

- effectue des recherches successives par mot-clef ;
- concatène les listes de résultats sans contrainte forte d’unicité par URL ;
- ne garantit pas qu’un URL ne soit pas réutilisé pour plusieurs mots-clefs.

Le test **T16*** formalise précisément cette contrainte d’unicité et nous obligera à :

- introduire une logique de sélection d’URLs garantissant des résultats distincts ;
- ajouter une contrainte de sélection d’URLs distinctes dans la logique de recherche multiple ;
- documenter le comportement en cas d’impossibilité (par exemple, si la banque ne contient pas assez d’URLs distincts).

6. Localisation des asserts et schéma général

- Les **tests unitaires** seront écrits en priorité en Python (par exemple avec `pytest`) et exécutés en CI :
 - assert sur les valeurs retournées (listes d’URLs, codes HTTP, messages d’erreur) ;
 - assert sur l’ordre des résultats pour le classement.
- Les **tests d’intégration** s’appuieront sur des appels HTTP simulés :
 - asserts sur le schéma de réponse (présence des champs attendus) ;
 - asserts sur la tolérance aux entrées invalides.

Schéma TDD :

1. Écriture du test (par Freddie ou David, revue par Bob / Alice).
2. Exécution du test – il échoue (cas de T16* notamment).
3. Implémentation / ajustement du code.
4. Re-exécution du test jusqu’au passage au vert.
5. Intégration du test dans la suite de non-régression.

7. Gantt mis à jour avec Freddie (QA)

Freddie est intégré principalement sur la formalisation et l’automatisation des tests, en binôme avec David.

Ce Gantt montre que l’écriture des tests ne vient pas « après coup » : elle est intégrée au cœur du développement, avec une responsabilité explicite confiée à Freddie sous la supervision de David et de Bob.

Ce plan TDD donne à Freddie un rôle central dans la qualité du moteur, tout en gardant un cadre clair pour l’équipe de développement.

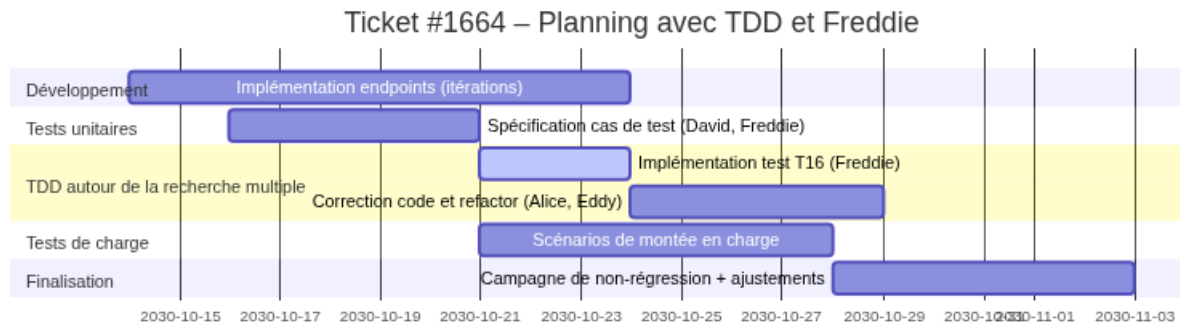


FIGURE 3

Contexte et objectif

Nous sommes le **4 novembre 2030**. Bob quitte le projet Ticket #1664 pour une promotion outre-Rhin. X. me propose deux options comme nouveau chef d'équipe : **Gaby** et **Harry**.

Aline, mon amie écrivain startupeuse, m'a suggéré d'analyser la situation à la lumière de la pyramide de Maslow afin de réfléchir à ce que chacun recherche vraiment dans ce projet : sécurité, reconnaissance, accomplissement...

Ce document vise à :

- situer **Alice**, **Gaby** et **Harry** dans la pyramide de Maslow ;
- formuler quelques critiques du modèle ;
- conclure sur un choix argumenté de chef d'équipe, sous forme de lettre à Aline.

1. Rappel synthétique de la pyramide de Maslow

La pyramide de Maslow est un modèle de motivation humaine qui hiérarchise les besoins en cinq niveaux :

1. **Besoins physiologiques** (survie, salaire de base).
2. **Besoin de sécurité** (stabilité du poste, environnement prévisible).
3. **Besoin d'appartenance** (équipe, réseau, reconnaissance sociale).
4. **Besoin d'estime** (statut, responsabilité, confiance reçue).
5. **Besoin d'accomplissement de soi** (sens, autonomie, impact).

Selon Maslow, les individus cherchent à satisfaire les besoins de base avant de progresser vers les niveaux supérieurs.

2. Profil d'Alice

- **Physiologique / sécurité** : déjà salariée, situation relativement stable.
- **Appartenance** : forte attache à l'équipe Ticket #1664 et à WONDROUS-IT.
- **Estime** : le rôle de coordinatrice technique sur un POC visible renforce son statut.
- **Accomplissement** : recherche d'un rôle de **lead technique** à moyen terme, envie de prouver sa capacité à piloter un projet de bout en bout.

En termes de Maslow, Alice se situe majoritairement aux niveaux **4–5** : elle a besoin de responsabilités réelles et de marge de manœuvre pour se projeter. Le Ticket #1664 représente pour elle une opportunité directe de satisfaire ses besoins d'estime et d'accomplissement.

3. Profil de Gaby

- **Physiologique / sécurité** : très expérimenté, à l'aise avec la dimension financière et politique.
- **Appartenance** : proche du comEx, réseau large, très sociable.
- **Estime** : déjà reconnu comme « monster techno-manager ».
- **Accomplissement** : semble rechercher des défis multiples, peut-être plus larges qu'un seul POC.

Gaby apparaît déjà très haut dans la pyramide : ses besoins d'estime et d'accomplissement sont en grande partie satisfaits. Le Ticket #1664 serait pour lui un projet parmi d'autres, pas nécessairement central.

4. Profil de Harry

- **Physiologique / sécurité** : bien né, réseau familial solide, peu inquiet sur la sécurité matérielle.
- **Appartenance** : intégré au comEx, posture très politique.
- **Estime** : recherche visible de reconnaissance, discours très travaillé, image à entretenir.
- **Accomplissement** : focalisé sur la dimension business et visibilité de la structure.

Harry semble surtout activé par les besoins d'estime (image, discours) et de pouvoir symbolique. Sa relation avec les équipes techniques est plus distante : il « parle bien », mais ne comprend pas toujours la profondeur technologique.

5. Critiques du modèle de Maslow

- Le modèle est statique, il suppose une progression linéaire, alors que dans un projet comme le Ticket #1664, les niveaux peuvent s'activer simultanément (sécurité de l'emploi, sens du projet, reconnaissance technique).
- Il est très individualiste, il ne tient pas compte des dynamiques d'équipe, des cultures d'entreprise ou des contraintes organisationnelles.
- Il manque de dimension contextuelle. À l'échelle d'un POC, certains besoins (apprentissage, expérimentation) peuvent être plus structurants que la hiérarchie des besoins classiques.
- Peu d'études scientifiques rigoureuses valident la hiérarchie stricte proposée par Maslow.
- En pratique, ce modèle est surtout utile comme grille de lecture qualitative, pas comme algorithme de décision.

Comme le dit Aline : « Kaizen, XP, AgileManifesto, JoelTest, ... ce sont tous des textes bibliques qu'on améliore et s'adapte chaque jour dans notre environnement. Les écoles de management enseignent toujours les mêmes fondamentaux chaque année. Seules les technos bougent vite et demandent une veille dynamique. »

6. Lettre à Aline – Choix du chef d'équipe

Chère Aline,

Comme promis, je me suis amusée à regarder notre trio Alice–Gaby–Harry avec les lunettes de Maslow.

Gaby coche beaucoup de cases : charisme, expérience, réseau. Mais c'est justement le problème : pour lui, le Ticket #1664 serait un dossier de plus dans un agenda déjà trop plein. Ses besoins d'estime et d'accomplissement sont déjà bien nourris ailleurs. Il serait un excellent sponsor, mais pas forcément un chef d'équipe présent au quotidien. Comme tu dis si bien, Qu'est-ce qu'il va faire avec moi dans les pattes devant nos stockholders, ce Gaby ? Chanter le micro de Bobby en me renvoyant au bumpkin country ?

Harry, lui, adore parler du projet et sait séduire les stockholders. Il nourrit clairement ses besoins d'estime et de visibilité. Mais son rapport à la technique reste décoratif, et je crains qu'il ramène tout à la communication externe plutôt qu'à la solidité du moteur de recherche d'images. Cependant, comme tu me l'as rappelé, Une SARL, c'est comme un PACS. Si l'amour réciproque est

hors de portée, vas-tu t'y mettre avec un gars que t'aimes à sens unique, ou plutôt avec un type qui t'adore ?

Moi, Alice, je suis peut-être la moins brillante sur le papier, mais c'est mon ticket, mon équipe, mon moteur de recherche. Mes besoins d'estime et d'accomplissement passent directement par la réussite de ce POC, pas par autre chose. C'est justement parce que ce projet compte vraiment pour moi que je suis prête à prendre la responsabilité du pilotage au quotidien.

Si je devais appliquer Maslow jusqu'au bout, je dirais que je suis celle pour qui le Ticket #1664 est le plus directement relié au sommet de la pyramide : le besoin de sens et d'autonomie dans mon travail.

Alors mon choix est simple : je propose à X. que Gaby soit sponsor / mentor stratégique, qu'Harry reste associé business (et qu'il m'apporte son soutien politique et sa capacité à convaincre les investisseurs), et que je prenne officiellement le rôle de cheffe d'équipe opérationnelle sur le projet.

À ton niveau, il ne faut jamais confondre la maîtrise technique et les compétences transverses (hard vs. soft skills). Je garderai un œil vif pour la veille technique (hard) et un œil plus conservateur quant au savoir être professionnel (soft).

Merci pour tes conseils, ma chère Aline. Tu es une vraie guru du temps moderne.

Alice

Contexte et hypothèses

Nous sommes le **6 novembre 2030** à Londres. Avec X., nous préparons un spin-off autour du moteur de recherche d'images issu du Ticket #1664, d'abord en B2B, avec une possible extension vers le B2C.

L'intrapreneuriat, c'est le terme qu'Aline a conseillé de me documenter pendant la semaine. À vrai dire, X. semblait satisfait, voire enthousiaste, lorsque j'ai fini par lui proposer le plan. « Parfait. Faisons comme ce que tu proposes. On va avoir un spin-off. Pour ce projet, je mets la moitié sur la table, à toi de trouver l'autre. » J'ai proposé 40% des parts à X., pour 50% des investissements dans la nouvelle structure, plus mon salaire actuel.

Objectif : présenter aux VC de la City un business plan chiffré et une courbe dépenses vs. recettes mettant en évidence le point d'inflexion (moment où les recettes cumulées dépassent les dépenses cumulées).

Hypothèses principales

- Monnaie : k€ (milliers d'euros).
- Horizon : 5 années (Année 0 = lancement).
- Modèle B2B initial (licences + services), ouverture progressive à des offres B2C.
- Structure du capital (ordre d'idée) : X. 40 %, Alice 25 %, Harry 25 %, VC de Londres 10 %.

1. Tableau prévisionnel dépenses / recettes

Année	Dépenses (k€)	Recettes (k€)	Dépenses cumulées	Recettes cumulées	Solde cumulé
0	300	0	300	0	-300
1	400	150	700	150	-550
2	450	450	1 150	600	-550
3	500	900	1 650	1 500	-150
4	550	1 400	2 200	2 900	+700
5	600	2 000	2 800	4 900	+2 100

Le **point d'inflexion** se situe entre les années **3 et 4** :

- fin Année 3 : recettes cumulées (1 500 k€) < dépenses cumulées (1 650 k€) → solde cumulé : -150 k€
- fin Année 4 : recettes cumulées (2 900 k€) > dépenses cumulées (2 200 k€) → solde cumulé : +700 k€

Détail des dépenses par année

- **Année 0** : Développement initial du POC, infrastructure, premiers recrutements (300 k€)
- **Année 1** : Équipe élargie, marketing B2B, infrastructure cloud (400 k€)
- **Année 2** : Consolidation de l'équipe, amélioration du produit, premiers clients (450 k€)
- **Année 3** : Expansion commerciale, R&D pour B2C, montée en charge infrastructure (500 k€)
- **Année 4** : Internationalisation, équipe commerciale renforcée (550 k€)
- **Année 5** : Optimisation, scalabilité, préparation levée série A (600 k€)

Détail des recettes par année

- **Année 0** : Aucune recette (phase de développement)
- **Année 1** : Premiers contrats B2B pilotes (150 k€)
- **Année 2** : Montée en puissance B2B, abonnements récurrents (450 k€)
- **Année 3** : Base installée B2B solide, premiers revenus B2C (900 k€)
- **Année 4** : Expansion internationale, offre freemium B2C (1 400 k€)
- **Année 5** : Maturité du produit, revenus récurrents stables (2 000 k€)

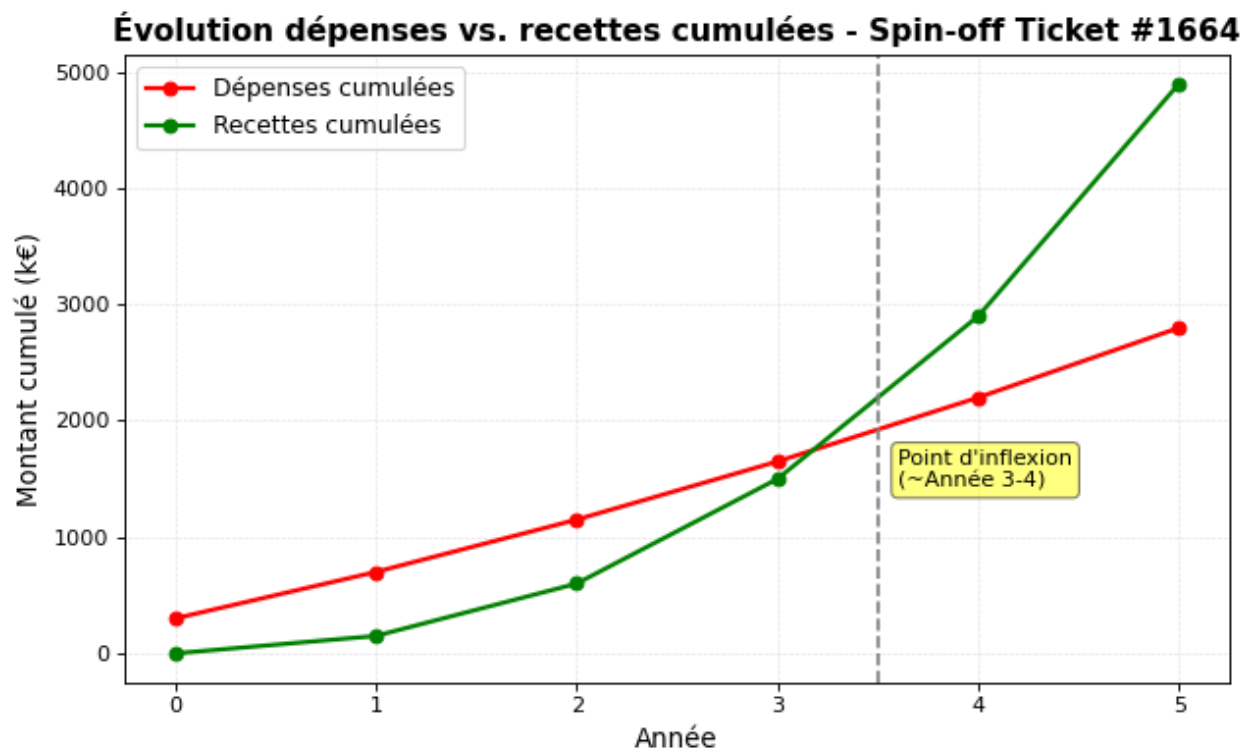


FIGURE 4

2. Courbe dépenses vs. recettes

3. Commentaire pour les VC de la City

Risques et opportunités

- Le **risque principal** est concentré sur les années 0 à 2 :
 - construction du produit,
 - obtention des premiers contrats B2B,
 - ajustement du modèle de pricing.
- À partir de l'Année 3, la base installée permet :
 - des **revenus récurrents** (abonnements, maintenance),
 - une mutualisation des coûts d'infrastructure.
- L'Année 5 illustre le potentiel de **scalabilité** du moteur, en particulier si l'on ouvre une offre B2C freemium adossée au moteur.

Avantages compétitifs

- **Technologie éprouvée** : le POC du Ticket #1664 a déjà démontré la faisabilité technique et les performances en montée en charge.
- **Équipe expérimentée** : Alice (lead tech), X. (sponsor stratégique), Harry (business development).
- **Marché porteur** : besoin croissant de solutions de recherche d'images pour les entreprises (e-commerce, médias, archives).

Stratégie de sortie

- Introduction en bourse. To the moon !