

Cahier des charges d'architecture

Projet de streaming vidéo interactif



GIBBERISH

Auteur(s) et contributeur(s)

Nom & Coordonnées	Qualité & Rôle	Société
Gérald ATTARD	Architecte logiciel	Gibberish

Historique des modifications et des révisions

N° version	Date	Description et circonstance de la modification	Auteur
0.1	08/12/20--	Création du document	
1.0	29/09/2022	Complément du document	Gérald ATTARD

Validation

N° version	Nom & Qualité	Date & Signature	Commentaires & Réserves
1.0	Alex Z Directeur technique		

Tableau des abréviations

Abr.	Sémantique
BDD	Business Driven Development (trad. Développement axé sur le métier)

Table des matières

I. Contexte et objectif.....	4
II. Objectifs et besoins fonctionnels.....	5
II.A. Objectifs stratégiques.....	5
II.B. Besoins fonctionnels.....	6
III. Stratégie architecturale.....	7
III.A. Stratégie de tests.....	8
IV. Parties prenantes.....	9
IV.A. Identification.....	9
IV.B. Capacité d'impacts.....	10
IV.C. Implication.....	11
V. Livrables.....	12
VI. Critères et procédures d'acceptation.....	12
VI.A. Critères d'acceptation.....	12
VI.B. Procédures d'acceptation.....	15
VII. Feuille de route.....	16
VII.A. Plan du projet.....	16



I. Contexte et objectif

Gibberish est une société de production de médias créée en 2009, mettant un point d'honneur à rester « à la pointe de la technologie ». Cette stratégie d'entreprise a d'ailleurs été payante puisque la société demeure aujourd'hui le leader incontesté dans ce domaine d'activité.

Aussi, afin de maintenir cette politique avant-gardiste toujours d'actualité, la société s'efforce sans cesse d'adopter les dernières technologies au travers d'achat de logiciels, toujours plus novateurs les uns que les autres.

En outre, afin de disposer continuellement des dernières nouveautés technologiques, la société dispose de sa propre équipe technique mettant en œuvre les innovations acquises selon les souhaits spécifiques de l'entreprise.

Cette modernisation perpétuelle s'accompagne intrinsèquement de l'application rigoureuse des normes en vigueur, en termes de Qualité, tels que la documentation et les bonnes pratiques relatives à l'Amélioration Continue, synonyme de fiabilité pour la société.

De plus, l'adoption de chaque nouvelle technologie s'accompagne systématiquement d'une campagne de recrutement d'experts de la technologie en question, dans un périmètre mondial.

Ainsi, dans le contexte mentionné ci-dessus, *Gibberish* a l'intention de produire une génération innovante de contenu de streaming interactif, afin de mettre à disposition du public une nouvelle manière d'appréhender le visionnage de contenu et les jeux vidéoludiques.

Fort de cet objectif, ce document définira la portée fonctionnelle, la vision architecturale ainsi que l'approche méthodologique employées durant ce projet, et fournira les éléments d'acceptation adéquats pour estimer sa progression au travers d'une feuille de route associée.



II. Objectifs et besoins fonctionnels

Suite à la description du contexte et de l'objectif définis précédemment, il est important de prendre en compte le consentement et l'appui des différents investisseurs et parties prenantes associés à ce projet.

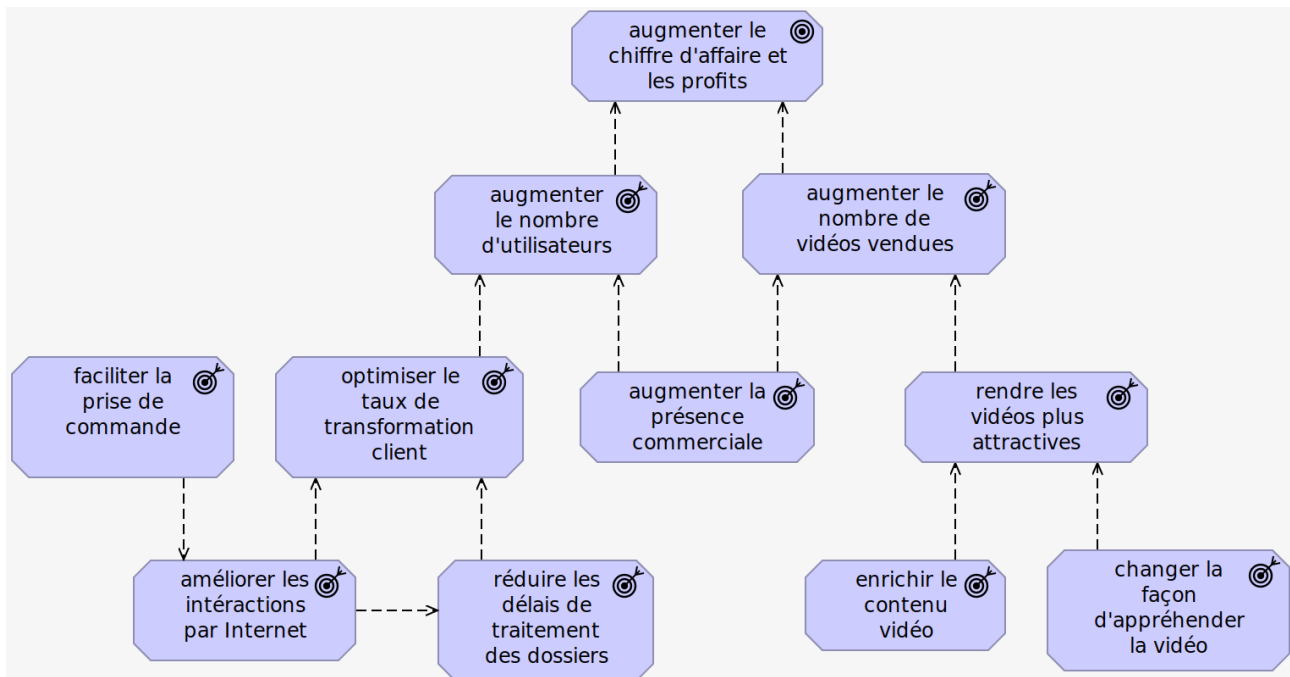
En effet, ces derniers sont intimement convaincus que l'ajout d'une interactivité créative et du streaming, aux productions médiatiques de la société, sont la prochaine grande étape à franchir pour conserver le statut de leader dans ce secteur.

II.A. Objectifs stratégiques

Ainsi, pour représenter le cycle d'architecture, tel que peuvent se le représenter les différentes parties prenantes décisionnaires, il conviendra de prendre en considération le diagramme d'objectifs permettant de :

- guider les changements à réaliser pour l'entreprise,
- quantifier les objectifs opérationnels,
- allouer ces objectifs aux bonnes parties opérationnelles,
- avoir une approche rationnelle pour prioriser ces objectifs.

Le diagramme d'objectifs est représenté ci-dessous :



II.B. Besoins fonctionnels

Ainsi, en prenant en considération l'évolution d'Internet, des télécommunications et des technologies de l'information, les fonctionnalités interactives attendues à l'issue de ce projet devront permettre aux utilisateurs d'interagir avec le contenu d'une vidéo, de manière positive et utile.

Ces fonctionnalités comprendront :

1. les téléchargements **de différents types de contenu** en utilisant les liens à l'intérieur de la vidéo. **Ces liens pourront renvoyer sur des scènes spécifiques de la vidéo elle-même, ou pointer vers des ressources externes ;**
2. **les prises de décision de l'utilisateur à des moments clés de l'histoire**, et donc, la modification de l'histoire elle-même, **en fonction des choix de l'utilisateur ;**
3. **la possibilité de modifier les angles de visionnage des scènes**, avec l'ajout d'une fonctionnalité de rotation interactive tridimensionnelle (vidéo à 360°), **tout au long de la vidéo ;**
4. **la prise en charge d'une fonctionnalité de navigation interactive, permettant à l'utilisateur de se déplacer chronologiquement à n'importe quel moment de l'histoire.** Ainsi, les éléments du menu ou de la table des matières devront permettre, lors de leur sélection, de passer directement au segment spécifique de la vidéo choisi par l'utilisateur ;
5. **prise en charge des retours d'expérience utilisateurs au travers d'une fonctionnalité de collecte et de publication d'avis**, comprenant la saisie de données et de commentaires des utilisateurs **pour chaque scène de la vidéo ;**
6. **possibilités d'enregistrer des séquences personnalisées, relatives aux choix décisionnels effectués par l'utilisateur, et donc, de générer dynamiquement des vidéos personnalisées qu'il sera ensuite possible de partager, via des liens externes, avec d'autres utilisateurs ;**
7. **création d'un index personnel, associé aux contenu privé de l'utilisateur, recensant tous les ajouts que celui-ci aura pu apporter au sein de l'histoire. Cet index sera accompagné de filtre permettant de sélectionner spécifiquement chaque type d'ajout personnel de l'utilisateur, afin de pouvoir les retrouver et les isoler du contenu d'origine de l'histoire ;**
8. **tous les ajouts spécifiques de l'utilisateur pourront être partagés ou conservés de façon privée et sécurisée avec un système de mot de passe personnalisable ;**
9. **la possibilité d'acheter de l'espace de stockage supplémentaire, associé au compte de l'utilisateur, pour pouvoir stocker davantage de contenu personnel associé aux licences de visionnage détenues ;**
10. **la possibilité d'acheter des licences de visionnage, spécifiques et temporaires, pour des périphériques autres que ceux déclarés par l'utilisateur.**

Nota : les éléments ci-dessus surlignés en jaune ont été ajoutés aux informations d'origine par l'Architecte Logiciel pour spécifier et/ou spécialiser les fonctionnalités d'origine.

Ainsi, tel que décrit ci-dessus, ce projet améliorera grandement la place de l'**Interactivité** et du **Streaming** au sein des productions vidéo de *Gibberish.net*.



III. Stratégie architecturale

Dans le cadre de ce projet, la stratégie qui sera mise en place sera alignée sur la politique d'innovation constante de *Gibberish*, éprouvée depuis la création de la société, à savoir :

- définition des caractéristiques et des interactions des utilisateurs de l'offre des services de streaming vidéo interactif ;
- acquisition des meilleurs logiciels éprouvés disponibles sur le marché pour répondre aux besoins de ce projet. Ces derniers devront prendre en compte les différents périphériques sur lesquels les utilisateurs utiliseront les services de diffusion vidéo interactive en continu, et fournir des options d'interaction au sein d'un environnement cloud ;
- préparation de la liste exhaustive des modules logiciels indispensables à la bonne réalisation du projet, dont ceux nécessitant un développement spécifique répondant à la stratégie d'entreprise relative au développement de logiciels personnalisés ;
- identification des interactions logicielles et intégrations des différents modules logiciels nécessaire aussi bien côté utilisateur, que dans le cloud ou sur les propres serveurs de l'entreprise ;
- établissement d'un plan de test au niveau des applications et des composants logiciels, selon la méthodologie BDD, permettant d'évaluer leurs fonctionnalités et leurs caractéristiques requises ;
- identification des compétences techniques requises pour le bon déroulement du projet, puis estimation de celles disponibles en interne et de celles nécessaires d'être sollicitées en prestation de services externes ; il est à noter que la recherche de prestataires externes sera considéré selon un périmètre mondial afin de maximiser les chances de recherche de profils adéquats.

III.A. Stratégie de tests

Associé à la stratégie architecturale générale décrite ci-dessus, un plan de test sera rédigé pour définir les modalités et les procédures associées. Aussi, ce plan de test devra :

- assurer que l'implémentation du projet respecte l'architecture cible préalablement définie ;
- fournir des listes de contrôle assurant la conformité, vis à vis de l'architecture cible, de :
 - l'ingénierie,
 - l'infrastructure informatique,
 - l'interface utilisateur,
 - des fonctionnalités et des caractéristiques propres aux services développés,
 - des API et/ou d'autres éléments à définir ultérieurement.
- Les listes de contrôle mentionnées au point précédent devront décrire des tests au niveau :
 - des composants pour s'assurer de l'adéquation des méthodes de maintenance associées à ceux-ci, comme par exemple, le maintien à jour de la bibliothèque de l'éditeur vidéo interactif ;
 - des applications nécessaires au système global pour s'assurer que chacune d'entre elles réponde bien au rôle qui lui ait attribué.
- préciser, pour chaque test, la procédure utilisée et les métrages nécessaires à l'évaluation du contexte testé ;
- proposer des indicateurs, associés aux métrages précédents, ainsi que leurs résultats attendus à l'issue de chaque test.
- décrire la méthode de collecte des données issues des tests.

Plus pragmatiquement, ce plan de test recensera les cas de tests associés à ce projet, afin de mesurer la bonne réalisation, ou non, du résultat attendu, selon une données ou d'un jeu de données de tests. Cette approche permettra de comparer la réalisé avec le résultat attendu, et comprendra, pour chaque cas de tests, trois informations :

- le type de test à exécuter et son scénario associé ;
- les données de tests à utiliser ;
- le(s) résultat(s) attendu(s).



IV. Parties prenantes

IV.A. Identification

Le tableau ci-dessous identifie les différentes parties prenantes au sein de ce projet, et définira, pour chacune d'entre d'elles, sa fonction et ses domaines d'intervention :

Nom	Rôle	Fonction	Domaine(s) d'intervention
Alex Z	Directeur technique	Responsable de la gouvernance du projet, il est le gestionnaire du portefeuille de projet et , plus précisément, de la gestation du projet, en termes de choix et de priorisation. Il fixe les spécificités attendues et les exigences que le projet devra respecter et suivre durant tout son cycle de vie. Il décidera de la poursuite du projet et des ressources allouées, suite à son acceptation.	Organisationnel (entreprise)
Marie M	Responsable ingénierie	Responsable de la gestion de l'ingénierie, elle est une experte scientifique et se tient régulièrement informée des évolutions de son domaine technique. Mandatée par le donneur d'ordre, elle est la garante de la vision globale du projet et de son accomplissement, d'un point de vue métier. Ainsi, au sein de son domaine métier, elle planifiera, coordonnera et contrôlera l'ensemble des tâches de clarification, d'élaboration, de production et d'exploitation des processus métier du projet.	Organisationnel (métier) Technique (métier)
Pierre PARKER	Administrateur du système	Personne susceptible d'intervenir au cours de l'étude en vu de profiter de ses services ou pour obtenir les autorisations requises.	Technique (projet)
Gérald ATTARD	Architecte logiciel	Mandaté par le donneur d'ordre, il est le garant de la vision globale du projet et de son accomplissement. Au sein du projet, il planifiera, coordonnera et contrôlera l'ensemble des tâches de clarification, d'élaboration, de production et d'exploitation du projet.	Organisationnel (projet) Technique (projet)

IV.B. Capacité d'impacts

Cette section permettra de développer une bonne compréhension des parties prenantes les plus importantes et d'enregistrer cette analyse pour référence.

Cette partie du document sera, également, à actualiser au cours du projet.

Fonction	Nom Prénom	Capacité à perturber le projet	Compréhension courante	Compréhension requis	Engagement actuel	Engagement requis	Support requis
Directeur technique	Alex Z	H	M	H	M	H	H
Responsable ingénierie	Marie M	M	H	H	M	H	H
Administrateur du système	Pierre PARKER	H	M	M	M	M	M
Architecte logiciel	Gérald ATTARD	H	H	H	H	H	H

Légende :

- H : *Haute*
- M : *Moyen*
- B : *Bas*

IV.C. Implication

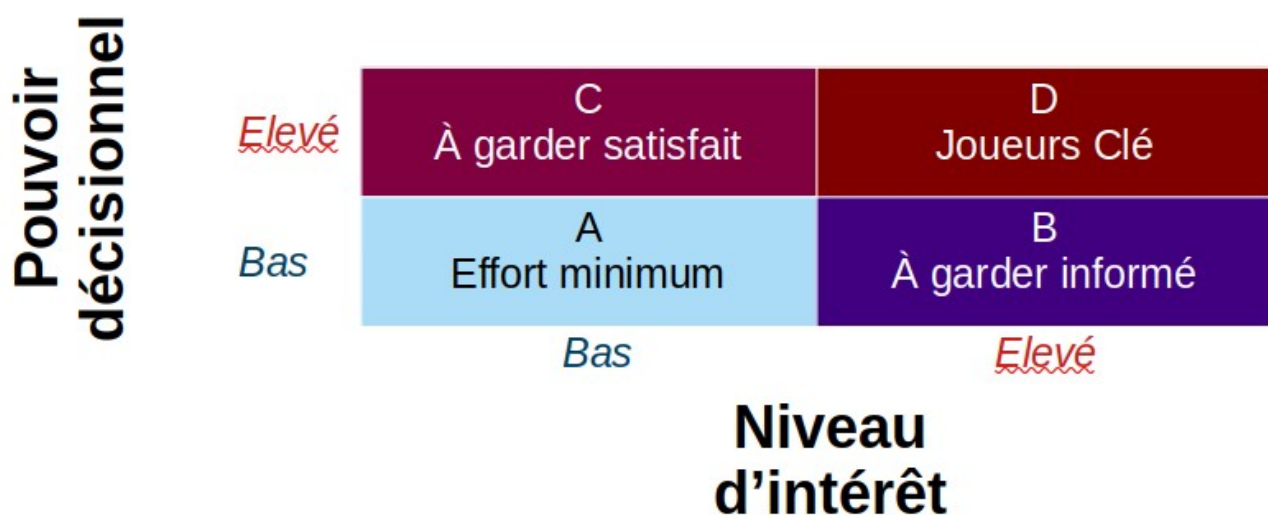
Les paragraphes précédents ont identifié une liste de personnes et d'organisations concernées par le projet d'architecture d'entreprise.

Certains d'entre eux peuvent avoir le pouvoir de bloquer ou d'avancer. Certains peuvent être intéressés par ce que fait l'initiative d'Architecture d'Entreprise, d'autres peuvent ne pas s'en soucier.

Cette étape permet aux équipes d'AMOA et de MOE d'identifier facilement quelles parties prenantes sont censées être des bloqueurs ou des critiques, et quelles parties prenantes sont susceptibles d'être des défenseurs et des partisans de l'initiative.

Ainsi, il est primordial de déterminer le pouvoir, l'influence et l'intérêt des parties prenantes, de manière à concentrer l'engagement de l'architecture d'entreprise sur les personnes clés. Ceux-ci peuvent être cartographiés sur une matrice pouvoir/intérêt, qui indique également la stratégie à adopter pour s'engager avec eux.

Ainsi, la matrice représente la catégorisation à réaliser pour chaque partie prenante, en fonction de son pouvoir décisionnel et de son intérêt pour le projet :



A partir de la matrice ci-dessus, nous pouvons l'appliquer à l'identification des parties prenantes réalisée précédemment :

Nom Prénom	Fonction	Niveau d'implication
Alex Z	Directeur technique	D - joueurs clé
Marie M	Responsable ingénierie	C – à garder satisfait
Pierre PARKER	Administrateur du système	C – à garder satisfait
Gérald ATTARD	Architecte logiciel	D - joueurs clé

Tout comme les précédents paragraphes précédents, celui-ci sera à maintenir à jour durant le cycle de vie du projet si de nouvelles parties prenantes étaient identifiées.



V. Livrables

Cette étude comprendra trois livrables principaux, sans exclure la possibilité que de nouveaux documents y soient ajoutés à l'avenir en fonction des besoins. Aussi, les documents rédigés seront :

- un cahier des charges : le présent document recensant, notamment, des critères appréhendant et présentant le projet, dont une liste des caractéristiques d'interactivité préconisées ;
- un document de définition d'architecture : un document décrivant l'architecture cible, dont ses avantages et ses lacunes ;
- un plan de test : élaboré selon la méthodologie BDD, ce document détaillera les processus permettant d'assurer la conformité de l'architecture cible au travers de listes de contrôle comprenant, au niveau des composants et des applications, des indicateurs de suivi d'objectifs et des critères de validation.



VI. Critères et procédures d'acceptation

La réussite de ce projet dépendra de la capacité de l'équipe de développement à répondre aux besoins décrits dans le § *Objectifs et besoins fonctionnels*.

La communication entre les parties prenantes et l'équipe de développement jouera un rôle essentiel dans la livraison d'une solution correspondant aux exigences du produit et du marché.

De façon générale, les difficultés surviennent quand les besoins sont expliqués de façon vague et que l'équipe ne peut obtenir d'exigences claires. Afin d'éviter ce genre de situation, il est primordial de disposer d'une documentation logicielle de haute qualité, incluant des *User Stories* et des critères d'acceptation au sein d'une documentation d'exigences.

Une US (*User Story*) est une description formelle rédigée en langage naturel et décrivant une et une seule fonctionnalité souhaitée. Une US est généralement accompagnée de critères d'acceptation qui sont des conditions de sortie que l'US doit satisfaire pour être acceptée par un l'émetteur de l'US.

VI.A. Critères d'acceptation

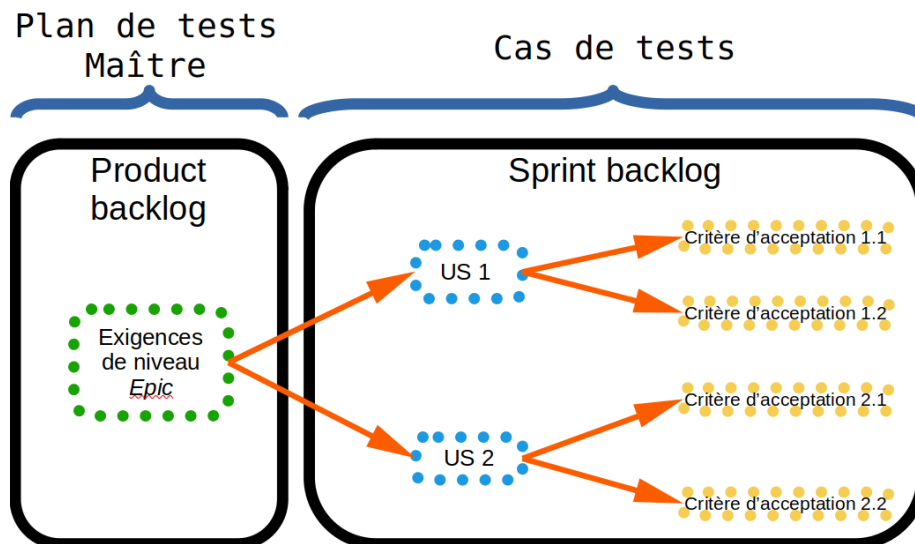
Les critères d'acceptation sont les conditions que le projet devra remplir pour être accepté par les parties prenantes. Chaque critère d'acceptation sera associé à un plan de test, et pourra intervenir à deux niveaux différents :

- le plan de tests Maître, appelé également *plan de tests Projet*, dont l'objectif est d'implémenter la stratégie de test sur le projet en particulier, sera applicable au niveau d'un *Product backlog* ;
- le plan de tests de niveau, appelé aussi communément *cas de tests*, ayant pour but de décrire les activités précises à mettre en œuvre pour chaque niveau de tests (tests unitaires, tests d'intégration, tests système et tests d'acceptance), sera applicable au niveau d'un *Sprint backlog*.

Chaque critère, associé à chaque *epic* ou *US*, sera unique et définira le comportement de la fonctionnalité décrite au sein de l'*epic* ou l'*US*, du point de vue de l'utilisateur final.

Des critères d'acceptation bien rédigés aident à éviter des résultats inattendus à la fin d'une phase de développement, et garantissent que toutes les parties prenantes et tous les utilisateurs sont satisfaits de ce qu'ils obtiennent.

Ces critères d'acceptation peuvent être schématisés selon le diagramme suivant :



Le format d'écriture des critères d'acceptation orienté scénario est connu sous le nom de type *Given / When / Then* (GWT).

- **Étant donné** (*Given*) une condition préalable ;
- **Quand** (*When*) une action est réalisée ;
- **Alors** (*Then*) ce résultat est attendu.

Cette approche, héritée du *Behavior Development Driven*, fournit une structure cohérente aidant les testeurs à définir quand commencer et quand terminer le test d'une fonctionnalité particulière. Cette méthode permet également de réduire le temps consacré à la rédaction des cas de test, puisque le comportement du système est décrit à l'avance.

Chaque critère d'acceptation sera alors rédigé selon le format ci-dessous :

Campagne - Responsable	<i>le nom de la campagne qui sera décrite – le nom du responsable du test</i>
Scénario	<i>le nom du comportement qui sera décrit</i>
Étant donné	<i>définit l'état initial du scénario</i>
Quand	<i>spécifie le moment à partir duquel l'utilisateur effectue l'action</i>
Ensuite	<i>le résultat de l'action dans "Quand"</i>
Et	<i>ajoute une assertion au point précédent</i>
Résultat attendu	<i>présente le résultat attendu à l'issue du test</i>

Lorsqu'elles sont combinées, ces instructions couvrent toutes les actions qu'un utilisateur entreprend pour terminer une tâche et connaître le résultat.

VI.B. Procédures d'acceptation

Pour établir chacun des critères d'acceptation décrits, le plan de tests suivra les étapes suivantes :

- **clarification de la portée de la fonctionnalité** : les critères d'acceptation définiront les limites de l'*epic* ou de l'*US* en fournissant des détails précis sur leurs fonctionnalités nécessaires à l'équipe pour comprendre **QUAND elles seront terminées** et **SI elles fonctionneront comme prévu**.
- **description des scenarii négatifs** (non passants) : ces scenarii représentent l'ensemble des cas de tests non passants, c'est à dire ne devant jamais aboutir. Par exemple, les critères d'acceptation pourront exiger que le système reconnaisse les entrées de mot de passe dangereuses et ainsi empêcher un utilisateur de continuer. Le format de mot de passe invalide est un exemple de scénario dit négatif, lorsqu'un utilisateur effectue des entrées non valides ou se comporte de manière inattendue. Les critères d'acceptation définissent ces scénarios et expliquent comment le système doit y réagir.
- **amélioration de la communication** : les critères d'acceptation synchroniseront les visions des parties prenantes et de l'équipe de développement, en garantissant que tout le monde a une compréhension commune des exigences définies : les développeurs sauront exactement quel type de comportement la fonctionnalité doit démontrer, tandis que les parties prenantes comprendront ce qui est attendu de la fonctionnalité.
- **rationalisation des tests d'acceptation** : les critères d'acceptation seront la base des tests d'acceptation de chaque *epic* et *US* définies, et chacun d'entre eux devra pouvoir être testé indépendamment et décrire des scenarii de réussite ou d'échec clairs et précis.
- **estimation de fonctionnalité** : les critères d'acceptation préciseront ce qui doit être développé exactement par l'équipe technique. Une fois que l'équipe de développement aura des exigences claires, précises et sans ambiguïté, elle divisera les *epics* fonctionnelles en *US* pouvant être correctement estimées.



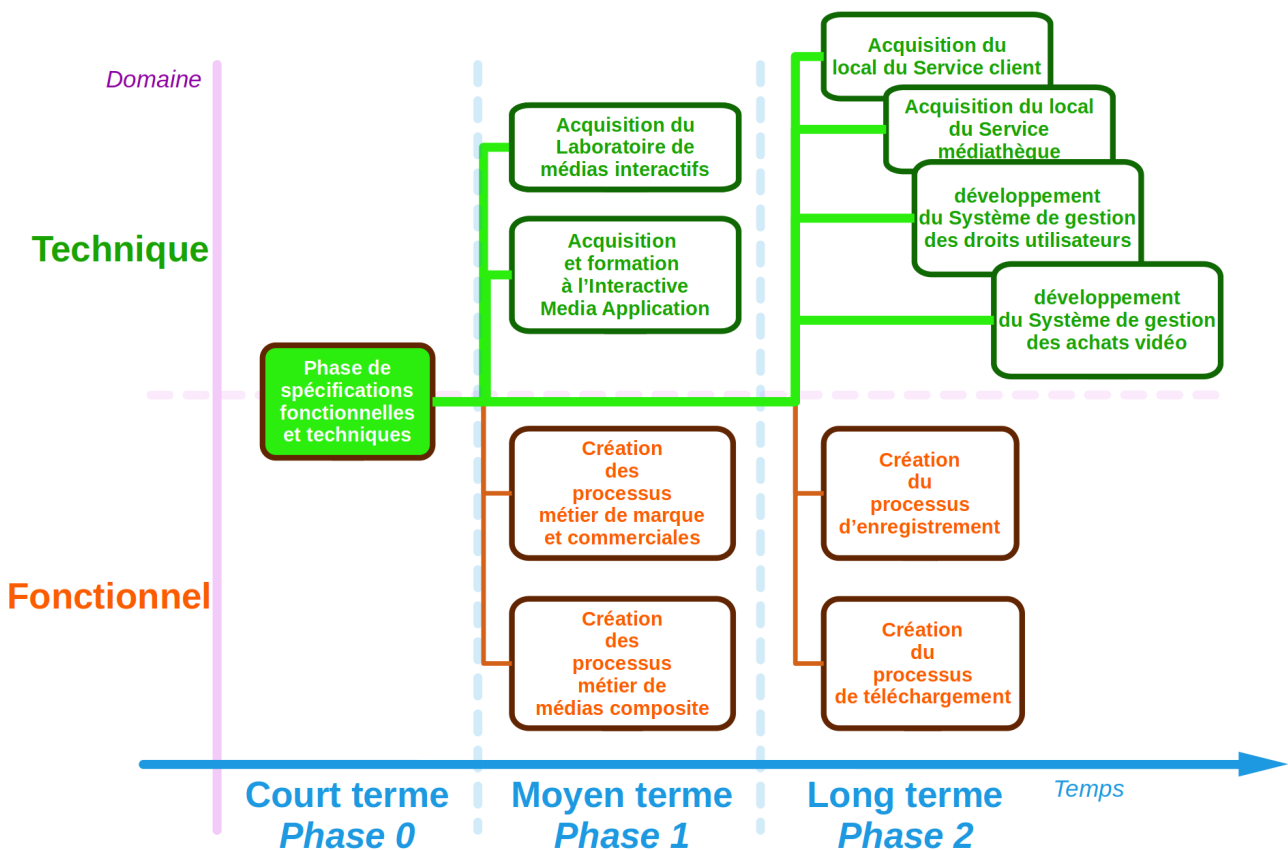
VII. Feuille de route

VII.A. Plan du projet

Enfin, relativement aux sections précédentes, cette étude préconise d'adopter une *roadmap* organisée en deux axes différents :

- un axe de domaine d'approche :
 - domaine fonctionnelle,
 - domaine technique.
- Un axe temporel selon trois périodes différentes :
 - **court terme – Phase 0** : période de spécifications fonctionnelles et techniques de la Phase 1 ;
 - **moyen terme – Phase 1** : période de développement des processus de création de vidéos interactives ;
 - **long terme – Phase 2** : période de développement et de mise en place des processus de vente et d'utilisation des vidéos interactives.

Les différentes phases présentées si-dessus seront détaillées dans le DDA et le plan de tests associés et peuvent être représentées selon le diagramme suivant :



GIBBERISH