**Rapport du Deuxième livrable**

Table des matières

[**Arborescence du deuxième livrable :** 1](#_Toc117295368)

[**Choix du livrable** 2](#_Toc117295369)

[**Premier livrable** 4](#_Toc117295370)

# **Arborescence du deuxième livrable :**

1. Analyse\_des\_besoins :
   * **Cahier\_des\_charges**
     + Un (deuxième) cahier des charges minimal
     + Un cahier des charges cumulé
   * **Organisation :**
     + 1.WBS
     + 2.Les matrices d’antériorité
     + 3.GANTT
   * **Receuil\_des\_besoins**
     + Receuil\_des\_besoins
2. Spécification :
   * **Cas d’utilisation**
     + Cas\_d\_utilisation
     + Scénarii
   * Etude\_technologie\_web
3. Conception :
   * **Conception\_architectural**
     + Conception\_du\_serveur
   * Implementation\_monolithique\_technologies
4. Programmation :
   * **Serveur**
     + Rapport\_installation
5. Test :
   * **Dossier\_de\_test**
     + Document\_test\_validation
     + Document\_test\_intégration

# **Choix du livrable**

**Choix du deuxième livrable**

Nous avons choisi dans ce deuxième livrable de faire un processus de développement concentrer sur le serveur à installer. Nous nous concentrons seulement sur ce point.

**Choix de l’arborescence du deuxième livrable**

Nous avons choisi cette méthode de livrable pour la cause suivante. Afin de mener à bien un projet informatique de cette envergure : la réalisation d’une application web. Nous devons rassembler les activités du génie pour respecter les critères de qualité. Nous avons donc proposé de faire plusieurs livrables (nous allons en dire davantage sur les livrables plus tard) avec une caractéristique : les dossiers on les a divisés en activité du génie logiciel : Analyse des besoins, Spécification, Conception, Programme, Test. Cela permet une meilleure clarté et aussi de suivre un ordre linéaire pour la validation de notre livrable.

**Choix du rendu en plusieurs livrables, Cycle de vie**

Notre équipe suit un processus de développement. Le processus de développement indique la forme dans laquelle les activités sont connectées entre elles. L’ordre dans lequel s’enchaînent les activités s’appelle le cycle de vie du produit logiciel.

Ici dans ce premier livrable, il représente un cycle de vie du produit logiciel. Le produit logiciel est décomposer en plusieurs cycle de vie et sera compléter par de nouveaux livrables.

Cette partie du rapport est rédigé après avoir fait l’activité du génie logiciel : Analyse des besoins. Avec cette activité nous avons pu déterminer la méthode de décomposition de nos livrables qui nous définirons notre cycle de vie.

Nous avons proposé un cycle de vie en Cascade. (Voir figure) C’est un diagramme d’activité, utilisé pour montrer l’enchainement des tâches effectuées par notre équipe. Chaque livrable se basera sur cette méthode. Lorsque ces activités sont terminées on passera à la suivant dans un nouveau livrable.

Cette méthode de cascade nous permet d’avoir un contrôle sur chaque étape. Elle permet dès que possible de pouvoir créer les tests, ainsi lors du développement à chaque étape on réfléchit aux erreurs que l’on cherche. Une fois la programmation terminée nous exécutons les tests de validation qui vont venir valider les besoins fonctionnelles définit dans le cahier des charges et construit à partir des maquettes un plan répondant aux exigences du client. Il y a les tests d’intégrations qui vont venir valider la relation des composants (pages html). On se trouve donc dans un cycle linéaire.

# **Deuxième livrable**

**Analyse des besoins**

Le (premier) cahier des charges minimal :

Nous proposons un deuxième cahier des charges qui n’est pas complet et sera complété au fil du temps. Nous pouvons le retrouver dans le dossier document sur le gitlab du projet.

Le cahier des charges cumulé :

Nous proposons un cahier des charges qui cumule les cahiers des charges des anciens livrables.

Organisation :

Dans le dossier organisation disponible sur le premier livrable de notre dépôt GitLab, nous pouvons retrouver le diagramme WBS des activités à réaliser, les matrices d’antériorité et enfin le diagramme de GANTT pour pouvoir proposer un cycle de vie.

Recueil des besoins :

Nous listons et reformulons les exigences définit par le client de manière claires et concises tout en restant réaliste. Nous pouvons retrouver le recueil des besoins dans le dossier Document du dépôt GitLab.

**Spécification**

Cas d’utilisation :

Nous avons réalisé les cas d’utilisation.

Etudes des technologies utilisées :

On définit toutes les technologies utilisées lors de ce projet.

**Conception**

Conception\_architectural :

La conception architecturale permet de rendre compréhensible le modèle avec les interactions (relations) entre les pages html. Cela permet plus de faciliter au passage à la programmation par la représentation de ce modèle.

Dans la conception architecturale il y a la représentation des interactions entre toutes les pages que nous devons créer.

**Programmation**

Rapport de l’installation du serveur :

C’est un rapport qui contient du code concernant l’installation du serveur pour pouvoir accueillir la base de donnée ainsi que les pages de l’application web.

**Test**

Test d’acceptation :

C’est un type de test qui permet de vérifier si toutes les exigences client, décrites dans le cahier des charges correspondent aux chaque attentes du produit défini dans chaque livrable. Elle est aussi décrite à partir de l’activité de spécification grâce à la réalisation des maquettes.

Nous retrouvons les tests d’acceptation réalisé à partir des attentes du clients et des maquettes, et valider après la programmation si cette dernière correspond bien.

Test d’intégration :

C’est un type de test qui permet de vérifier si les exigences du client sur la navigabilité entre les composants correspondent bien. Les tests sont réalisés à partir du cahier des charges et approfondi dans la conception architecturale dans la partie conception.

Nous retrouvons les tests d’intégration réalisé à partir des attentes du clients et de la conception architecturale. La conception architecturale permet de voir si une page appelle l’autre. Ces tests sont validés après la programmation de toutes les pages html et si toutes les interactions entre les pages correspondent bien.