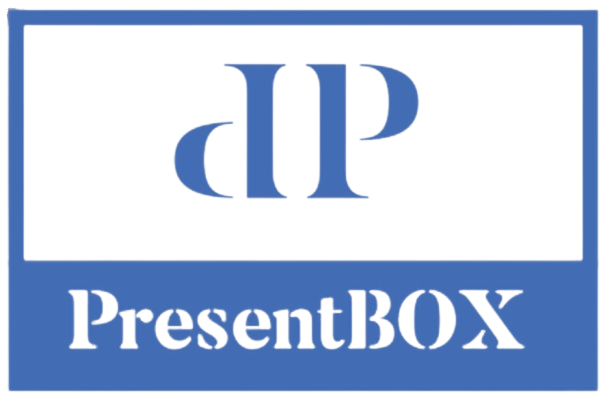
PresentBox

 : phase de lancement

Version <1.0>



Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| <17/05/2022> | <1.0> | <le plan d’itération de la phase de lancement> | <DEROUICH Amal> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

1.1 Objectif 4

1.2 Portée 4

1.3 Références 4

1.4 Contenu du document 4

2. Plan 4

3. Ressources 6

4. Cas d’utilisation 6

5. Critères d’évaluation 6

: phase de lancement

# Introduction

## Objectif

Ce plan d’itération a pour objectif de détailler le planning de la première itération avec lequel l’équipe du projet va se baser pour terminer la phase de lancement.

## Portée

Ce plan d’itération portera sur la durée de la phase de lancement ainsi que les différentes tâches mis en évidence pendant cette phase. Il mettra en évidence l’estimation générale de la phase en question. Ce document nous permettra d’avoir une visibilité globale sur la répartition des tâches de façon équilibrée du coup il permettra d’influencer tous les autres documents planifiés pour les itérations à venir.

## Références

Site web de documentation UP > aterfacts.

## Contenu du document

Ce document contient la subdivision des tâches de la phase de lancement en terme de jours pour pouvoir livrer le produit au client à tems ainsi que toutes les ressources nécessaires à la réalisation du projet.

Ce document contient aussi le diagramme de PERT propre à cette phase.

# Plan

Pendant cette première itération, la phase de lancement est mise en place. Les tâches liées à cette itération sont principalement le remplissage de tous les documents nécessaires à l’équipe avant de se lancer dans la phase d’élaboration et ensuite la réalisation de l’application.

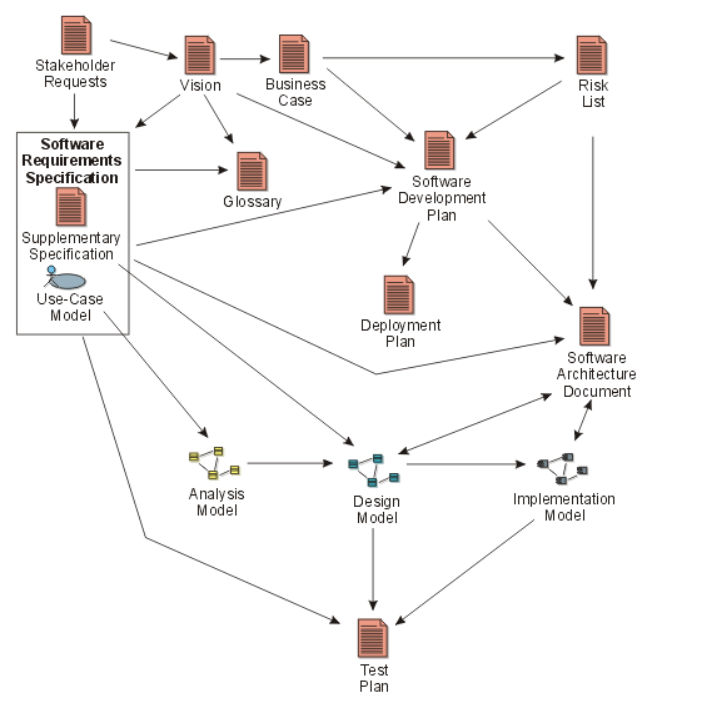
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symbole | Tâche | Durée | Date au plus tôt | Date au plus tard |
| A | Identification des risques et développement du business model | 3 jrs | 3 | 4 |
| B | Elaboration des objectifs du projet | 3 jrs | 3 | 4 |
| C | Définition du problème et développement de la vision du projet | 3 jrs | 3 | 4 |
| D | Analyse de l’architecture du projet | 3 jrs | 3 | 4 |

Détaillons un peu plus ces tâches.

* Pour la tâche A, elle est spécifique au chef de projet. Pour accomplir ces tâches, il est nécessaire de réaliser quelques documents à savoir : la liste des risques, les plans d’itérations, le planning général du projet, etc.
* Pour la tâche B, elle est spécifique au busines analyst. Pour accomplir ces tâches, il est nécessaire de réaliser quelques documents à savoir : les spécifications supplémentaires, le business glossary, etc.
* Pour la tâche C, elle est spécifique au system analyst. Pour accomplir ces tâches, il est nécessaire de réaliser quelques documents à savoir : le document vision, glossary, etc.
* Pour la tâche D, elle est spécifique au software architect. Pour accomplir ces tâches, il est nécessaire de réaliser quelques documents à savoir : l’architecture logicielle, le modèle de déploiement, etc.
* Diagramme de PERT :

Pour réaliser le diagramme de PERT en relation avec les documents cités ci-dessus, nous allons utiliser les artefacts et repérer la succession des tâches.

Référence (capture d’écran du site web référence pour UP) :



Le diagramme de PERT est alors comme suit :

A

C

D

D

B

B

D

# Ressources

Pour aboutir à notre objectif pendant cette itération, nous aurons besoin de plusieurs ressources :

* Humaines : 4 rôles sont indispensables (project manager, software architect, system analyst, business analyst)
* Financières : un budget de 20.000 DH pour pouvoir faire les études nécessaires dans les plus brefs délais.

# Cas d’utilisation

Pendant cette phase, on ne résout pas des cas d’utilisation du système à développer mais nous essayons d’élaborer une idée générale concernant le projet afin de bien se lancer.

# Critères d’évaluation

Les critères d’évaluation que notre équipe a fixée restent toujours celles sont :

* Les fonctionnalités : elles doivent être bien claires et visibles à l’utilisateur pour lui permettre une compréhension facile de notre application
* Les performances : le système doit être performant
* La réactivité : l’interaction de l’utilisateur avec le système développée doit être fluide et sans obstacle
* L’adaptabilité : notre système doit fonctionner dans n’importe quel système d’exploitation
* La capacité : notre application doit supporter un bon nombre d’étudiants et de professeurs