CoronAlert

Version 1.0

****

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 09/06/2020 | 1.0 | Plan cycle de vie | Ettahi Yahya |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

[1. Introduction 6](#_Toc42771908)

[1.1 Objectif 7](#_Toc42771909)

[1.2 Portée 7](#_Toc42771910)

[1.3 Référence 7](#_Toc42771911)

[1.4 Contenu du document 7](#_Toc42771912)

[2. Vue d’ensemble du Plan du cycle de vie 8](#_Toc42771913)

[2.1 Modèle de cycle de vie 8](#_Toc42771914)

[2.2 Disciplines 9](#_Toc42771915)

[2.3 Configuration des disciplines 9](#_Toc42771916)

[2.3.1 Discipline 9](#_Toc42771917)

[2.3.2 Produits de travail 10](#_Toc42771918)

[2.3.3 Notes sur les produits de travail 11](#_Toc42771919)

[2.3.4 Rapports 11](#_Toc42771920)

[2.4 Classification des produits de travail 11](#_Toc42771921)

[2.5 Procédures de revue 12](#_Toc42771922)

[2.6 Exemples de plans d’itération 12](#_Toc42771923)

[2.6.1 Phase de lancement 12](#_Toc42771924)

[2.6.2 Phase d’élaboration 12](#_Toc42771925)

[2.6.3 Phase de construction 12](#_Toc42771926)

[2.6.4 Phase de transition 12](#_Toc42771927)

[3. Disciplines 13](#_Toc42771928)

[3.1 Expression des exigences 13](#_Toc42771929)

[3.1.1 Discipline 13](#_Toc42771930)

[3.1.2 Produits de travail 13](#_Toc42771931)

[3.1.3 Notes sur les produits de travail 14](#_Toc42771932)

[3.1.4 Rapport 14](#_Toc42771933)

[3.1.5 Procédures de revue additionnelles 15](#_Toc42771934)

[3.1.6 Autres difficultés 15](#_Toc42771935)

[3.1.7 Configurer la discipline 15](#_Toc42771936)

[3.2 Analyse & conception 15](#_Toc42771937)

[3.2.1 Discipline 15](#_Toc42771938)

[3.2.2 Produits de travail 15](#_Toc42771939)

[3.2.3 Notes sur les produits de travail 16](#_Toc42771940)

[3.2.4 Procédures de revue additionnelles 17](#_Toc42771941)

[3.2.5 Autres difficultés 17](#_Toc42771942)

[3.2.6 Configurer la discipline 17](#_Toc42771943)

[3.3 Gestion de projet 17](#_Toc42771944)

[3.3.1 Discipline 17](#_Toc42771945)

[3.3.2 Produits de travail 17](#_Toc42771946)

[3.3.3 Notes sur les produits de travail 18](#_Toc42771947)

[3.3.4 Procédures de Revue additionnelles 19](#_Toc42771948)

[3.3.5 Autres difficultés 19](#_Toc42771949)

[3.3.6 Configurer la discipline 19](#_Toc42771950)

[3.4 Environnement 19](#_Toc42771951)

[3.4.1 Discipline 19](#_Toc42771952)

[3.4.2 Produits de travail 19](#_Toc42771953)

[3.4.3 Notes sur les produits de travail 19](#_Toc42771954)

[3.4.4 Procédures de revue additionnelles 20](#_Toc42771955)

[3.4.5 Autres difficultés 20](#_Toc42771956)

[3.4.6 Configurer la discipline 20](#_Toc42771957)

[3.5 Implémentation 20](#_Toc42771958)

[3.5.1 La discipline 20](#_Toc42771959)

[3.5.2 Les Produits de travail 20](#_Toc42771960)

[3.5.3 Notes sur les produits de travail 21](#_Toc42771961)

[3.5.4 Procédures de revue additionnelles 21](#_Toc42771962)

[3.5.5 Autres difficultés 21](#_Toc42771963)

[3.5.6 Configurer la discipline 21](#_Toc42771964)

[3.6 Test 21](#_Toc42771965)

[3.6.1 Discipline 21](#_Toc42771966)

[3.6.2 Produits de travail 21](#_Toc42771967)

[3.6.3 Notes sur les produits de travail 22](#_Toc42771968)

[3.6.4 Procédures de revue additionnelles 22](#_Toc42771969)

[3.6.5 Autres difficultés 22](#_Toc42771970)

[3.6.6 Configurer la discipline 22](#_Toc42771971)

[3.7 Déploiement 22](#_Toc42771972)

[3.7.1 Discipline 22](#_Toc42771973)

[3.7.2 Produits de travail 22](#_Toc42771974)

[3.7.3 Notes sur les produits de travail 23](#_Toc42771975)

[3.7.4 Procédures de revue additionnelles 24](#_Toc42771976)

[3.7.5 Autres difficultés 24](#_Toc42771977)

[3.7.6 Configurer la discipline 24](#_Toc42771978)

Plan du Cycle de Vie

# Introduction

Une application est un ensemble de programmes, qui permet à une machine ou à un système informatique d'assurer une tâche ou une fonction en particulier. Notre projet consiste à élaborer une application qui permet le traçage et d’accompagnement des citoyens, Afin de contribuer à l’éradication de l’épidémie Covid’19.

L’équipe qui travaillera sur le développement de cette application est composée de 4 personnes.

L’élaboration de ce projet sera selon le cycle de vie. En effet le cycle de vie d'un logiciel indique les étapes par lesquelles doivent passer un logiciel de sa conception jusqu'à la fin de son développement.

Ce cycle de vie permet de détecter les erreurs tout au long du processus de réalisation et ainsi les corriger pour produire un logiciel de qualité.

Les étapes partent de l’identification des besoins jusqu’à la clôture du projet et sont les suivantes :

* Pré-étude : Cette étape permet de définir les objectifs du projet et de définir le domaine d'activité.
* En entrée on a les besoins et en sortie on a un cahier de charges.
* Analyse : Cette étape consiste à recueillir et à formaliser les besoins du client, de définir les contraintes et d'estimer la faisabilité de ces besoins.
* En entrée on a le cahier de charges et en sortie on a le dossier d'analyse.
* Conception : Cette étape permet d'élaborer la structure générale du système et de définir chaque sous-ensemble du logiciel à produire.
* En entrée, on a le dossier d'analyse et en sortie on a un dossier de conception.
* Codage : Cette étape consiste à coder ou à programmer les fonctionnalités définis dans la phase de conception.
* En entrée, on a le dossier d'analyse et en sortie on a des programmes.
* Tests : Cette étape permet de tester le logiciel conformément aux spécifications (fonctionnelle ou non fonctionnelle).
* Réception : Cette étape permet au client de vérifier la conformité du logiciel avec les spécifications initiales.
* Maintenance : Cette étape permet de prendre en charge les actions collectives du système (maintenance curative et évolutive).

En entrée on a un logiciel et en sortie on a un logiciel modifié.

L’objectif d’un tel découpage est de permettre un développement fluide de l’application, de sa conception à la fin de son développement dans le cadre des exigences posées, des besoins exprimés et des méthodes mises en œuvre.

## Objectif

L'objectif de ce document est de présenter le plan du cycle de vie décrivant la méthode développement mis en place pour réaliser notre projet. L’objectif de ce projet est de mettre en œuvre un développement orienté objet à travers un travail d’équipe en appliquant le processus unifié.

Cela va nous apporter les résultats suivants :

* Etablir un planning de la mission en la découpant en plusieurs phases.
* Afin que les membres de l’équipe projet assimilent les bonnes pratiques du déroulement du projet.

## Portée

La protée de fonctionnement de l'application serait simple :

L'application devra être installée sur les smartphones de tous les citoyens. Une sensibilisation des individus servira à les inciter à télécharger l’application en faisant appel à leur responsabilité et citoyenneté. Lorsque les téléphones de deux personnes qui ont installé l'application vont se croiser pendant une certaine durée, et à une distance rapprochée (ces données doivent être paramétrées), le téléphone portable de l'un enregistre les références de l'autre dans son historique via la connexion Bluetooth, wifi ou toute autre technologie équivalente.

## Référence

-L’application que nous proposons sera inspirée de l’application « TraceTogether » développée au Singapour, et celle du Maroc «Wiqaytna» puisqu’elle respecte le volet confidentiel tout en assurant le but souhaité.

## Contenu du document

Ce document spécifie le cycle de vie de la conception ainsi que le développement de notre projet dans lequel on met en œuvre la méthode UP, et est destiné à contenir :

* Nom du projet ;
* Les membres d’équipe ainsi que leur rôles RUP respectifs ;
* Chef de projet ;
* Des liens : Documentation ;
* Documents : Créés lors de la phase de lancement/ d’élaboration/ de construction et de transition.

# Vue d’ensemble du Plan du cycle de vie

## Modèle de cycle de vie

Le modèle de cycle de vie suivi pour le développement de ce projet suit le modèle de développement UP. Le processus que suit ce modèle est composé de quatre phases:

* Phase de lancement : c’est la phase durant laquelle on définit la portée du produit.
* La phase d’élaboration : L’analyse et la conception architecturale.
* La phase de construction : La conception détaillée.
* La phase de transition : Le déploiement du logiciel vers le client.

Tous les livrables sont produit en incrément y compris le système lui-même.

Cette itération sur les différentes phases nous permet d’améliorer la qualité des livrables.

**Chaque phase est composé d'un certain nombre d'étape répétée às chaque phase :**

* **L’expression des besoins**
* **Analyse**
* **Conception**
* **Implémentation**
* **Test**

## Disciplines

**Les disciplines mises en œuvre sur le projet sont les suivantes :**

* **Expression des exigences**
* **Analyse et conception**
* **Gestion de projet**
* **Gestion de processus (Environnement)**
* **Implémentation**
* **Test**
* **Déploiement**

**Les workflows que nous allons utiliser sont ceux présentés dans le processus RUP.**

* **Modélisation d’affaires (Business modeling)**
* **Elicitation des exigences (Use cases)**
* **Analyse et conception**
* **Implémentation : codage, test unitaire et intégration**
* **Test : Cas de tests, procédures, defect-tracking**
* **Déploiement**
* **Gestion de configuration**
* **Gestion de projet**
* **Environnement : conception de l’infrastructure nécessaire au développement**

## Configuration des disciplines

Ce paragraphe explique comment fonctionnent les disciplines, l’objectif des différents tableaux et sections.

### Discipline

Le présent chapitre explique l’objectif des différents tableaux qui seront présentés dans les parties suivantes du document.

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | | | | **Revoir les détails** | **Outils employés** | **Plans types/ Exemples** |
| **Lancem** | **Elab** | **Const** | **Trans** |
|  |  |  |  |  |  |  |  |

#### Explications sur le tableau précédent

|  |  |  |
| --- | --- | --- |
| **Nom de la colonne** | **Objectif** | **Contenu/Commentaires** |
| **Produit de travail** | Le nom du produit de travail. | Une référence au produit de travail dans le processus. |
| **Comment l’utiliser ?** | Décrire comment le produit est utilisé tout au long du cycle de vie. | Décider pour chacune des phases si on :   * doit impérativement l’avoir * devrait l’avoir * pourrait l’avoir * ne l’utilise pas   Ceci est défini dans le guide [: Classifying Artifacts](D:RationalRationalUnifiedProcess2000processmodguidemd_uclaar.htm). |
| **Revoir les détails** | Définir le niveau de revue, et les procédures à appliquer sue le produit. | Décider le niveau de la revue:   * Formelle-Externe * Formelle-Interne * Informelle * aucun   Pour plus de détails consulter le guide : Review Levels. |
| **Outils employés** | Définition de l’outil utilisé pour créer le produit. | Références aux détails des outils utilisés pour créer et maintenir les produits de travail. |
| **Plans types/ Exemples** | Les plans types à utiliser et les exemples de produits construits à partir de ces plans types. | Références aux plans types et exemples. On peut faire référence à des plans types ou des exemples du processus. Cette colonne peut également contenir des références à des produits de travail courant qui peuvent fournir une aide complémentaire aux membres du projet. |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
|  |  |  |

### Rapports

|  |  |  |  |
| --- | --- | --- | --- |
| **Rapport** | **Comment l’utiliser ?** | **Plans types / Exemples** | **Outils utilisés** |
|  |  |  |  |

## Classification des produits de travail

Un produit de travail est une fourniture du processus. Il est souvent produit dans le cadre d’une discipline, bien qu’il existe des exceptions. Les produits de travail sont organisés dans la discipline où ils sont crées. Pour décrire comment un produit de travail sera utilisé, on emploie le schéma de classification suivant:

* **Doit impérativement**
* **devrait**
* **pourrait**
* **ne sera pas**

Voir le Guide : "Classifying Artifacts” pour plus de détails.

## Procédures de revue

Le projet utilise les niveaux de revue suivants :

* **formelle externe**
* **formelle interne**
* **informelle**
* **aucun niveau**

## Exemples de plans d’itération

### Phase de lancement

* Vision globale du projet,
* Identification des cas d’utilisation,
* Identification des fonctionnalités essentielles du produit,
* Définition du calendrier du projet,
* Définition des objectifs de chaque itération,
* Identification des risques,
* Première étude des outils.

### Phase d’élaboration

* Définition d’une architecture stable du projet
* Maquette,
* Prototype,
* Ajustation des objets,
* Adaptation de la liste des risques.

### Phase de construction

Implémentation de l’application, en s’appuyant sur l’architecture définie précédemment.

### Phase de transition

* Correction des derniers bugs selon les remarques formulées par les intervenants.
* Déploiement de la version de référence de GNRProject.
* Livraison des documents du projet.

# Disciplines

## Expression des exigences

### Discipline

Définir les différentes fonctionnalités du produit exigés par le client.

Bien comprendre les besoins du client et de ses collaborateurs et d’où vient le besoin de cette application.

Définir les différentes interfaces utilisateurs qui répondent aux besoins non fonctionnels de la solution.

* Planification du contenu technique des itérations ;
* Permettre à l’équipe de projet une bonne compréhension des besoins de l’application ;
* Convenir avec l’enseignant encadrant du projet sur les fonctionnalités de l’application ;
* Fournir des estimations, des délais pour le développement de l’application ;
* Définir l’IHM, en se focalisant sur les besoins exprimés ;
* Définir les limites de l’application.

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Acteur | DOIT | DOIT | peut | peut | informel | STAR UML |  |
| Classe d’interface | peut | Doit | Doit | Peut | informel | STAR UML |  |
| Glossaire | Doit | Doit | Doit | Doit | Formel externe | Editeur texte |  |
| Spécifications supplémentaires | peut | Doit imp | Doit | Peut | Formel externe | Editeur texte |  |
| Cas d’utilisation | Doit | Doit imp | Peut | Pas Fait | Formel externe | STAR UML |  |
| Modèle de cas d’utilisation | Doit | peut | Pas fait | Pas fait | Formel externe | Star uml |  |
| Paquetage de cas d’utilisation | Peut | Doit | Pas fait | Pas fait | Formel externe | STAR UML |  |
| Prototype d’interface utilisateur | Pas fait | Doit imp | Doit | Pas fait | Formel externe | Axure |  |
| Vision | Doit | Doit | Doit | Doit | Formel externe | Word |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | **Raison** |
| Spécification supplémentaire | Etudier à chaque fois que l’on veut faire une itération | Ne pas perdre de vue le but du projet. |
| Glossaire | Lire pour chercher un mot qu’on ne comprend pas | Eviter les ambiguïtés lors de la discussion sur le travail à faire. |
| Prototype des IHMs | Montrer aux clients afin d’avoir son avis. | Eviter de perdre du temps à créer des interfaces qui ne conviennent pas aux clients. |
| Paquetage de cas d’utilisation | Trouver dans le produit des cas d’utilisation. | Pouvoir empaqueter les use cas pour pouvoir mieux voir leurs utilisation. |

### Rapport

|  |  |  |  |
| --- | --- | --- | --- |
| **Rapport** | **Comment les utiliser ?** | **Outils nécessaires** | **Plans types / Exemples** |
| Résumé du Modèle des Cas d'Utilisation |  |  |  |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline

## Analyse & conception

### Discipline

Cette étape permet de transformer les exigences trouver dans l’étape précédentes en diagrammes et exigences métier qui nous permettrons de faciliter la création de l’application en lui élaborant une bonne architecture qui devrait éviter les bugs et être en accords avec l’environnement où elle sera implémentée.

* Passer des exigences à la conception d’une maniére correcte,
* Elaborer une architecture compléte,
* Elaborer une conception qui tient compte de l’environnement d’implémentation.

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Classe d’analyse | Peut | Doit | Doit | Doit | Informel | UML |  |
| Modèle d’analyse | Peut | Doit | Doit | Doit | Formel interne | UML |  |
| Modèle de déploiement | Pas | Doit | Peut | Doit | Formel interne | Editeur Texte |  |
| Modèle de données | Peut | Doit | Doit | Doit | Formel interne | Java |  |
| Classe de conception | Peut | Doit | Doit | Peut | Informel | UML |  |
| Modèle de conception | Pas | Doit | Doit | Pas | Formel interne | UML |  |
| Paquetage de conception | Pas | Doit | Doit | Doit | Informel | UML |  |
| Sous-système de conception | Pas | Peut | Peut | Peut | Informel | UML |  |
| Interface | Peut | Doit | Doit | Peut | Formel externe | Axure |  |
| Document d’architecture logicielle | Doit | Doit | Doit | Doit | Formel externe | Editeur de texte |  |
| Réalisation de cas d’utilisation | Pas | Peut | Doit imp | Doit | Informel |  |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
| Modèle d’analyse | Suivi lors de l’analyse et conception de la solution | Guidelines permettant de diriger notre conception et donc de bien conceptualise les différentes exigences du client. |
| Modèle de déploiement | Suivi lors de la finition du projet et de la passation aux clients | Permet de bien planifier le déploiement de la solution chez les utilisateurs et donc d’éviter l’apparition de problème lorsqu’on met la solution a la disposition du client |
| Modèle de donnes | Conception des bases de données utilise donc dans la création de classe et lors de la programmation | Permet de garder une cohérence de données entre les différents modules de notre application et donc d’éviter les bugs et de permettre une maintenance facile par n’importe qui. |
| Document d’architecture | Suivi a chaque fois qu’on commence une nouvelle itération dan l’élaboration | Permet de décrire l’architecture du logiciel de notre projet concernant les différents modules qu’englobe ce dernier. La structure suivante permet de garantir que la conception réalisée répond aux besoins du système et que l’architecture en conséquence est maintenable et extensible. |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline

## Gestion de projet

### Discipline

Planifier l’avancement des taches dans un projet de développement logiciel, avoir une possibilité de contrôle de risque suivant la méthode de gestion de projet utilisée

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Evaluation de l’itération | Doit | Doit | Doit | Doit |  | Editeur de texte |  |
| Plan d’itération | Doit | Doit | Doit | Doit |  | Editeur de texte |  |
| Compte rendu de revue | Doit | Doit | Doit | Doit |  | Editeur de texte |  |
| Liste des risques | Doit | Doit | Peut | pas |  | Editeur de texte |  |
| Plan de Développement Logiciel (PDL) | Peut | Doit imp | Doit | Doit |  | Editeur de texte |  |
| Liste des problèmes | Peut | Peut | Doit | Doit |  | Editeur de texte |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
| Evaluation d’itération | Fait à chaque fin d’itération | Permet d’aider a la planification en suivant les taches qui ont été faite ou pas avec leur commentaire pour avoir une trace écrite de tous ce qui a été fait. |
| Plan d’itération | Fait avant le début de chaque itération | Permet de diviser le travail entre les différents membres de l’équipe et ainsi avoir une trace écrite de ce que doit faire chaque personne. |
| Compte rendu de revue | Fait après chaque réunion | Permet de mettre sous écrit tous ce qui a été dit et fait durant une réunion. |
| Liste des risques | Lister les différents encouru lors dans le projet | Permet d’avoir une idée précise et d’anticiper les différents risques avec leurs dégrats et donc d’avoir une stratégie corrective |
| Plan de développement logiciel | Suivi lors des étapes de programmation | Permet d’avoir une idée unique sur la façon de développer la solution et donc d’avoir un code compréhensible entre les différentes personnes développant la solution. |

### Procédures de Revue additionnelles

### Autres difficultés

### Configurer la discipline

## Environnement

### Discipline

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Plan du cycle de vie | Doit | Doit | Doit | Doit | Formel externe | Editeur de texte |  |
| Plans types spécifiques au projet | Doit | Doit | Doit | Peut | Formel externe | Editeur de texte |  |
| Guides de programmation | pas | Doit | Doit | Doit | Formel interne | Editeur de texte |  |
| Outils | peut | Doit | Doit | Doit | informel |  |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
| Plan de cycle de vie | Etudier autant que possible | Permet de savoir les différentes étapes de la méthodologie de gestion de projet choisie et donc de savoir ce qu’il faut faire a chaque étape et les livrables a rendre |
| Guide de programmation | Etudier a chaque nouvelle étape de programmation | Permet de rendre l’équipe de développement cohérente dans la façon de programmer la solution et que tous le monde puisse comprendre le code de tous le monde et donc le maintenir ou l’améliorer. |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline

## Implémentation

### La discipline

Cette étape permet l’implémentation des différents concepts mis en œuvres dans l’architecture UML de notre application et donc de suivre les différentes étapes d’intégration des différents modules de notre projet.

### Les Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types /**  **Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Build | pas | peut | Doit imp | Doit | Informel |  |  |
| Composant | pas | peut | Doit imp | Doit | Informel |  |  |
| Modèle d’implémentation | pas | peut | Doit | Doit | Formel interne |  |  |
| Sous-système d’implémentation | pas | pas | Doit | Doit | Formel interne |  |  |
| Plan d’Intégration du « build » | peut | peut | Doit | Doit imp | Formel interne |  |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
|  |  |  |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline

## Test

### Discipline

Mettre en place les différentes étapes de tests des différentes fonctionnalités de la solution préparer.

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Cas de test |  |  |  |  |  |  |  |
| Classe de test |  |  |  |  |  |  |  |
| Composants de test |  |  |  |  |  |  |  |
| Résumé de l’Evaluation des tests |  |  |  |  |  |  |  |
| Modèle de test |  |  |  |  |  |  |  |
| Paquetage de Test |  |  |  |  |  |  |  |
| Plan de Test |  |  |  |  |  |  |  |
| Procédure de Test |  |  |  |  |  |  |  |
| Résultats des tests |  |  |  |  |  |  |  |
| Script de test |  |  |  |  |  |  |  |
| Sous-système de test |  |  |  |  |  |  |  |
| Document d’analyse de la charge de travail |  |  |  |  |  |  |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
|  |  |  |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline

## Déploiement

### Discipline

### Produits de travail

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Produits de travail** | **Comment les utiliser ?** | | | | **Détails de revue** | **Outils nécessaires** | **Plans types / Exemples** |
| **Lancemt** | **Elaborat°** | **Construct°** | **Transit°** |
| Bill of Materials |  |  |  |  |  |  |  |
| Deployment Plan |  |  |  |  |  |  |  |
| Deployment Unit |  |  |  |  |  |  |  |
| End-User Support Material |  |  |  |  |  |  |  |
| Installation Artifacts |  |  |  |  |  |  |  |
| Produit |  |  |  |  |  |  |  |
| Product Artwork |  |  |  |  |  |  |  |
| Notes de sortie d’une version (release notes) |  |  |  |  |  |  |  |
| Support de formation |  |  |  |  |  |  |  |

### Notes sur les produits de travail

|  |  |  |
| --- | --- | --- |
| **Produit de travail** | **Comment l’utiliser ?** | **Raison** |
| Photoshop | Réaliser les différents logos et figure utiliser dans notre application | Permet de finir les différents dessins trouver dans notre solution |
| Support de formation | Ecrit a la fin du développement | Permet d’aider l’utilisateur dans son utilisation de l’application |
| Déploiement plan | Ecrit juste avant la fin du projet | Planifier les release et l’installation de la solution dans les différents ordinateurs du clients |

### Procédures de revue additionnelles

### Autres difficultés

### Configurer la discipline