Planification - Projet ITI 4

POK’HEIR

|  |  |
| --- | --- |
| **Etudiants :** | Douvrin Nicolas  Bourree François |
| **Nom du superviseur :** | LEFEVERE Vincent |

**Résumé du projet (8 lignes max) :**

|  |
| --- |
| Le projet consiste en la réalisation d’un site WEB à destination de l’association « POK’HEIR ». Le site comprendra une page d’accueil, une page d’inscription, une page de connexion, une page de statistiques et une page de plan de table/temps de jeu. Sur la page d’accueil, l’association pourra mettre en avant ses différents évènements, les différents lots à gagner, des photos des derniers tournois ainsi que les derniers vainqueurs des tournois. Une base de données sera créée afin que les étudiants puissent s’inscrire et voir leur évolution/niveau. Sur la page plan de table, les tables de jeu seront établies automatiquement en fonction d’un système de classement des joueurs, le temps de jeu sera bien sûr affiché sur la table. |

**Références du document :**

|  |  |  |
| --- | --- | --- |
| Référence | Planification\_Pok’Heir | |
| Nom du projet | Pok’Heir | |
| Emetteur | DOUVRIN Nicolas  BOURRE François  59000 LILLE | Tél. : 0638410119 / 0675434103  Fax. :  Mail : [nicolas.douvrin@hei.yncrea.fr](mailto:nicolas.douvrin@hei.yncrea.fr)  [francois.bourree@hei.yncrea.fr](mailto:francois.bourree@hei.yncrea.fr) |
| Date d’émission | 06/12/2017 | |

**Validation :**

|  |  |  |  |
| --- | --- | --- | --- |
| Nom(s) validateur | Date | Validation (O/N) | Commentaires |
| LEFEVERE Vincent | 05/12/17 |  |  |
|  |  |  |  |

**Versions :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numéro de version | Date | Etat | Auteur(s) | Remarque(s) / modification(s) |
| 1.0 | 30/11/17 | docx | Nicolas Douvrin / François Bourree |  |
|  |  |  |  |  |
|  |  |  |  |  |

Sommaire

[Sommaire 2](#_Toc435783263)

[1. Planification globale de la phase de mise en oeuvre 3](#_Toc435783264)

[2. Planification détaillée des différentes fonctionnalités 4](#_Toc435783265)

# Planification globale de la phase de mise en oeuvre

*Faire apparaître dans le planning :*

* *Les jalons du projet intégrant entre autres (en plus des jalons spécifiques liés à votre projet) les dates :*
  + *Du 12,13 et 15 décembre (soutenance de la phase d’études)*
  + *Du 23 février (validation à mi-parcours)*
  + *Du 17 et 18 avril (soutenance finale de la phase de mise en œuvre)*
* *Les phases globales de développement*
* *Les phases de test*
* *Les phases de mise en production (hébergement…) des différents lots*
* *La phase de formation utilisateur*
* *Les phases de rédaction du rapport final*

*Nb : Plan proposé pour le rapport final à rendre en avril :*

* *Partie I - Situation d’avancement du projet (technologies employées, outils utilisés, réalisé par rapport aux éléments du cahier des charges, plan de test effectué, mise en production, planning réel / prévisionnel)*
* *Partie II – Description technique de l’application, à l’intention d’éventuels binômes susceptibles de poursuivre le travail à l’avenir (insistez sur les aspects techniques complémentaires que vous avez dû mettre en œuvre, et que vous n’aviez pas eu l’occasion de mettre en œuvre au sein des enseignements)*
* *Annexes :* 
  1. *Documentation utilisateur*
  2. *Code commenté*

# Planification détaillée des différentes fonctionnalités

*Compléter le tableau ci-dessous, en cochant, pour chacune des fonctionnalités de votre application, le lot dans lequel elle sera intégrée.*

*Pour rappel :*

* *Lot n°1 : fonctionnalités principales, deadline au 23 février*
* *Lot n°2 : fonctionnalités annexes, deadline au 17 avril*

*Ce tableau devra être validé par votre enseignant superviseur*

|  |  |  |
| --- | --- | --- |
| **Fonctionnalité** | **Lot**  **n°1** | **Lot**  **n°2** |
|  |  |  |
|  |  |  |

# Plan de test du projet

*Rédiger le plan de test que vous vous proposez de suivre au second semestre, pour valider les développements réalisés sur votre projet. Tous les différents tests devront être listés. Veillez à être suffisamment précis dans cette partie (ne pas se limiter à des généralités sur les tests logiciels, il faut décliner ce que vous avez appris en cours de Qualité Logicielle pour les décliner effectivement à votre projet)*

PLAN TEST

Site Web – Pok’Heir (association de poker)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Prénom** | **Mail** | **Classe** |
| Douvrin | Nicolas | nicolas.douvrin@hei.yncrea.fr | H44 |
| Bourree | François | francois.bourre@hei.yncrea.fr | H44 |

[**1. Test Plan identifier** 3](#_Toc435448493)

[**2. Références** 3](#_Toc435448494)

[**3. Introduction** 4](#_Toc435448495)

[**4. Test Items** 4](#_Toc435448496)

[4.1. Objectif 4](#_Toc435448497)

[**5. Software Risk Issues** 4](#_Toc435448498)

[**6. Features to be tested** 6](#_Toc435448499)

[**7. Features not to be tested** 7](#_Toc435448500)

[**8. Approach** 7](#_Toc435448501)

[8.1 Objectif 7](#_Toc435448502)

[8.2 Techniques de test 9](#_Toc435448505)

[8.2.1. Test par des «Tables de Décision » : 9](#_Toc435448506)

[8.2.2. Test de “Transition d’état” : 9](#_Toc435448507)

[8.2.3. Autres techniques : 9](#_Toc435448508)

[8.3. Outillage de Recette 10](#_Toc435448509)

[8.3.1. Outil de gestion d’anomalies 10](#_Toc435448510)

[8.3.2. Qualification des anomalies 10](#_Toc435448511)

[8.3.3. Workflow des anomalies 11](#_Toc435448512)

[8.3.4. Processus de gestion d'anomalie 12](#_Toc435448513)

[8.4. Outil de gestion des plans de test 12](#_Toc435448514)

[**9. Item Pass/Fail Criteria** 12](#_Toc435448515)

[**10. Suspension Criteria and Resumption Requirements** 13](#_Toc435448516)

[**11. Test Deliverables** 14](#_Toc435448517)

[**12. Remaining Test Tasks** 15](#_Toc435448518)

[**13. Environmental Needs** 15](#_Toc435448519)

[**14. Staffing and Training Needs** 16](#_Toc435448520)

[**15. Responsibilities** 16](#_Toc435448521)

[**16. Schedule** 18](#_Toc435448522)

[16.1. Actions des différentes phases : 18](#_Toc435448523)

[**17. Planning Risks and Contingencies** 19](#_Toc435448524)

[**18. Approvals** 20](#_Toc435448525)

[**19. Glossary** 21](#_Toc435448526)

**1. Test Plan identifier**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Nom** | **Versions** | **Auteurs** | **Emplacement** |
| 02/11/17 | PLANTEST\_Douvrin\_Bourree\_v1.0 | 1.0 | Douvrin, Bourre | GitHub/projet |

**2. Références**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Nom** | **Versions** | **Auteurs** | **Emplacement** |
| 21/09/17 | Cahier des charges Pok’Heir | 1.0 | Douvrin, Bourree | GitHub/projet |

Tous les documents relatifs au projet se trouvent sur un dossier partagé sur GitHub par les acteurs du projet : Nicolas Douvrin, François Bourree.

Quand ce plan de test sera validé, il sera également enregistré à cet emplacement.

Nombre de documents ont servi de référence à l’écriture de ce plan de test :

1. Cahier des charges Pok’Heir.

2. Les comptes rendus de réunion avec le client.

**3. Introduction**

Ce plan de test va nous permettre de découvrir l’organisation et la planification de la démarche de test grâce aux comptes rendus des différentes réunions avec l’association ainsi qu’avec l’appui du cahier des charges qui sera établi ultérieurement.

Nous définirons également les acteurs (leurs rôles respectifs dans les étapes de test) qui les mettrons en place ainsi que leurs conditions d’exécution.

Ce plan de test sera sujet à des modifications et/ou améliorations au fur et à mesure des réunions et des découvertes de nouvelles contraintes lors de la réalisation de certaines fonctionnalités.

**4. Test Items**

## 4.1. Objectif

Valider les points suivants :

* Inscription sur le site pour un compte personnalisé
* Connexion au compte utilisateur créé
* Connexion du site web à la base de données
* Ergonomie
* Gestion des statistiques utilisateur
* Création de tables de tournoi automatisées

Les objectifs seront complétés ou modifiés en fonction des prochaines réunions et selon le cahier des charges qui sera défini ultérieurement.

**5. Software Risk Issues**

L’analyse des risques se fera selon la méthode suivante :

* Identifier
* Analyser
* Planifier
* Suivre
* Contrôler

|  |  |  |
| --- | --- | --- |
|  | **Justification impact** | **Justification probabilité** |
| **Ergonomie** | Navigation simple et épurée de l’outil et bonne présentation globale pour une bonne utilisation | Dépend principalement de la programmation |
| **Interface utilisateur/ administrateur** | Vérifier la mise à disposition de certaines fonctionnalités de l’outil s’il s’agit d’un administrateur, sécurité du site vis-à-vis d’un visiteur | Dépend de la programmation et du système de connexion à l’outil |
| **Système d’informations** | Mise en ligne simplifiée des informations relatives au fonctionnement de l’association | Dépend de l’utilisateur et de la programmation |
| **Identification utilisateur** | Problèmes de confidentialité des informations non publiques et synchronisation des statistiques | Dépend de la programmation et de la bonne gestion des accès |
| **Système de tables automatisées** | Fonctionnalités responsive | Dépend de l’outil utilisé et de la programmation. |
| **Interface statistiques** | Affichage des données de chaque utilisateur liées à la base de données | Dépend de la programmation. |

**6. Features to be tested**

|  |  |
| --- | --- |
| **Elément concerné** | **Objectifs du test** |
| **Ergonomie** | * Vérifier la bonne lisibilité sur écran d’ordinateur * Bon agencement des éléments à l’écran * Traitement des erreurs * Navigation claire et guidage général * Homogénéité |
| **Interface utilisateur/ administrateur** | * Compatibilité suivant les différents navigateurs * Vérification d’authentification |
| **Système d’informations** | * Mise en ligne sécurisée sur les réseaux sociaux * Conformité de l’article posé et de celui partagé |
| **Identification utilisateur** | * Accès à certaines fonctionnalités après identification * Test concordance aux données utilisateurs * Impossibilité d’accès aux personnes non autorisées |
| **Système de tables automatisées** | * Responsivité de la fonctionnalité * Précision du système |
| **Interface Statistiques** | * Gestion de statistiques * Intégrité des informations * Bon agencement des informations |

**7. Features not to be tested**

|  |  |
| --- | --- |
| **Elément concerné** | **Justification** |
| **Récupération de données de Facebook pour intégration sur la page d’accueil** | Fonctionnalité bonus non nécessaire au bon fonctionnement du site web |
| **Gestion de la charge** | La gestion de la charge revient à l’hébergeur du site sélectionné par l’association. |

**8. Approach**

## 8.1 Objectif

L’objectif direct des tests en eux-mêmes est de « détecter un maximum d’erreurs », car c’est la correction de ces erreurs qui améliore la fiabilité du système. Ils doivent fournir l’assurance que le produit livré est bien celui qui a été commandé (cahier des charges lié à l’expression de besoin) et confirmer que l'application répond correctement aux spécifications définies en amont du projet (spécifications fonctionnelles et détaillées).

L’approche de test se base sur un Modèle d’Implémentation de Test détaillé ci-dessous :

* Planification :
* Périmètre projet
* Stratégie de test
* Ressources
* Jalons
* Organisation du projet
* Mode de communication à l’intérieur du projet.

Livrables : ce document même (Plan de Test), le TRH (Test Requirement Hierarchy)

* Préparation :
* Enchainement de deux traductions des exigences de test du TRH :
* Cas de Test Logiques
* Cas de Test Physiques (Procédure de test).
* Paramétrage de l'environnement de test

Livrables : l’ensemble des cas de test physiques intégré dans la feuille Excel destinée à la gestion des tests.

* Exécution :
* Sélection des cas de test du cahier de recette qui seront exécutés dans leur campagne propre
* Activité de Test de Confirmation (re-test des anomalies livrées corrigées)
* Test de non régression fonctionnelle

Livrables : résultats du test enregistrés dans la feuille Excel destinée à la gestion des tests, anomalies rencontrées sur le site web.

* Suivi :
* Mesurer l’avancement du processus de recette de chaque phase du modèle d’implémentation
* Support décisionnel permettant de prendre des actions correctives si nécessaire

## 8.2 Techniques de test

Sur ce projet, au regard du périmètre fonctionnel défini, nous utiliserons entre autres les techniques suivantes :

8.2.1. Test par des « Tables de Décision » :

Cette technique consiste à définir l’ensemble des combinaisons d’entrée/stimuli (causes) et de leurs sorties et/ou actions (effets) associées. Ces combinaisons sont regroupées au sein d’une table. Il s’agira ensuite de concevoir des tests couvrant toutes les combinaisons identifiées.

Cette technique concernera les envois de formulaire et de messages de l'application.

8.2.2. Test de “Transition d’état” :

Il s'agit de représenter graphiquement les différents états possibles.

Cette technique servira donc pour tester l'identification de l'utilisateur et gérer les diverses autorisations.

8.2.3. Autres techniques :

La technique des « Tests exploratoires » (pas de scénario rédigé précisément) sera utilisée dans le test de l'ergonomie.

8.3. Outillage de Recette

8.3.1. Outil de gestion d’anomalies

Une simple feuille Excel accessible par Google Tables et à modifier en ligne uniquement sera utilisée comme outil de gestion d'anomalies. Chaque ligne référencera une anomalie, un identifiant et sa qualification.

8.3.2. Qualification des anomalies

* **Catégorie de l’anomalie paramétrée** :
* Flux chargement : concernera une problématique liée au chargement initial et régulier.
* Paramétrage : concernera une problématique liée au paramétrage
* Moteur d’offre : concernera une problématique liée au calcul commande/panier (erreur quant à la réponse et au résultat du calcul retourné).
* Technique : concernera une problématique technique (temps de réponse, instabilité, mémoire…)
* **Sévérité de l’anomalie** :
* Bloquant : très critique, l’utilisateur est bloqué dans son processus métier (l’anomalie a un impact direct sur l’activité de l’entreprise, remet en cause l’intégrité des données financières et/ou commerciales manipulées)
* Majeur : ne fonctionne pas comme prévu / décrit, mais il existe un moyen de contournement, l’utilisateur peut continuer à travailler.
* Mineur : peu critique, cosmétique (pas d’impact sur les données manipulées, n’empêche pas l’utilisateur de continuer les actions liées à son processus métier)
* Evolutions : ne correspond pas à un incident mais à une demande d’évolution des spécifications et donc de l’application.
* **Priorité (de correction) de l’anomalie** :
* Haute : dans le cas où cette anomalie constitue un point de blocage important pour le déroulement de la Recette
* Normale : dans le cas où la campagne peut se poursuivre sans blocage
* Faible : généralement lié aux problématiques ergonomiques de faible criticité
* **Reproductibilité** :
* Toujours : anomalie récurrente, rencontrée à chaque reproduction du scénario.
* Aléatoire : anomalie non systématiquement reproductible.
* Impossible à reproduire : anomalie rencontrée une seule fois dont la reproduction est impossible.
* **Version/Build** : Chaque version/build doit être identifiée lors de la saisie d’une fiche anomalie.

8.3.3. Workflow des anomalies

Etats possibles d’une anomalie et responsabilités associées.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Acteur** | **Action** | **Etat** | **Etat Facultatif** | **Résolution Possible** |
| Testeur | Enregistrement de l’anomalie | NOUVEAU |  | Ouvert ou Ré-ouvert |
| Développeur / Gestionnaire | Affectation de l’anomalie à un développeur | PLANIFIE ou AFFECTE |  | Ouvert ou Ré-ouvert |
| Développeur / Gestionnaire | Confirmation qu’il s’agit d’une anomalie | ACCEPTE | X | Ouvert ou Ré-ouvert |
| Développeur | Correction de l’anomalie | RESOLU |  | Résolu, Impossible à reproduire et à corriger, Doublon, Pas de bug, Suspendu, Ne sera pas résolu |
| Testeur | Retest => correction Non OK | NOUVEAU | X | Ouvert ou Ré-ouvert |
| Testeur | Retest => correction OK | FERME |  | Résolu, Impossible à reproduire et à corriger, Doublon, Pas de bug, Suspendu, Ne sera pas résolu |
| Tous | Point dont la correction est différée | Ajouter une note et passer le statut à PANIFIE | X | Ouvert, Ré-ouvert, Suspendu |
| Tous | Point en attente d’informations complémentaires | Ajouter une note et conserver le statut actuel |  |  |

8.3.4. Processus de gestion d'anomalie

Le statut doit être passé à "Résolu" une fois que l'anomalie est résolue et livrée pour éviter toute ambiguïté sur la date de re-test possible (une anomalie pouvant être résolue en attente de livraison dans un futur package et donc non encore re-testable).

8.4. Outil de gestion des plans de test

Une simple feuille Excel accessible par OneDrive et à modifier dans Word Online uniquement sera utilisée comme outil de management de Test.

Chaque ligne permettra de référencer les mesures à prendre pour le traitement des anomalies identifiées dans la feuille Excel de l'outil de gestion d'anomalies.

**9. Item Pass/Fail Criteria**

Le maximum de fonctionnalités seront testées et il sera fait en sorte de laisser le moins de fonctionnalités non testées.

Les fonctionnalités principales seront testées les premières avec rigueur, les reste des fonctionnalités seront traitées avec une sévérité moindre étant donné leur importance.

**10. Suspension Criteria and Resumption Requirements**

Tout ou parties des activités de test devront être suspendues selon les critères définis ci-après. Selon les critères listés, deux conséquences apparaissent : la couverture des tests sera réduite (conséquence Type 1) et les délais de réalisation des tests en sera prolongée (conséquence Type 2).

* L'apparition d'une défaillance d'une fonctionnalité dont la gravité ne permet de trouver aucune solution satisfaisante autorisant la poursuite des tests ou en limite la progression (conséquence Type 1)
* Le nombre de défaillances rencontrées est un obstacle à la poursuite des tests dans la mesure où les résultats obtenus par la suite ne seraient pas significatifs. Ce taux de défaillances acceptables pour la poursuite des tests est spécifié dans la partie précédente du présent document (conséquence Type 2)
* La non-diminution des défaillances détectées persiste malgré des solutions mises en place (conséquence Type 2)

L'indisponibilité des ressources nécessaires précisées dans la section Environmental Needs n'autorise pas l'exécution des tests (conséquence Type 2)

* La mise en place d'une fonctionnalité principale intégrée au système, exigée par le client, n'a pas été réalisée (conséquence B).
* Lors d'une période de congé, le personnel chargé des tests n'est pas suffisant ou n'est pas disponible pour les effectuer (conséquence B).
* La date limite de remise du livrable accordée par le client est dépassée, ce dernier n'acceptant pas de délais supplémentaires pour la poursuite des tests (conséquence A).

La reprise de tout ou partie des tests sera effective selon les critères définis ci-après :

* Une ou plusieurs solutions déployées ont permis de réduire le taux trop élevé de défaillances, relativement au taux jugé acceptable pour l'obtention de résultats significatifs.
* Les ressources nécessaires à l'environnement de tests défini dans ce présent document sont de nouveau disponibles et en quantité suffisante.
* Des délais supplémentaires pour la remise du livrable ont été accordés par client et permettent un allongement de la durée de test.
* Les membres de l'équipe de test constituent un effectif suffisant pour la réalisation de ceux-ci.

**11. Test Deliverables**

Selon les différentes phases de tests, les documents listés ci-dessous constitueront les livrables de tests remis au client :

En phase de planification :

* Le plan de test ci-présent

En phase de préparation :

* La présentation des différents scénarios de cas de test dans un Cahier de recette

En phase d'exécution :

* Des comptes rendus hebdomadaires des tests présentant :
* L’avancement des tests présentant le nombre de cas de test effectués et ceux restant à réaliser.
* Les conséquences des anomalies sur l'utilisation des fonctionnalités par les utilisateurs.
* Les résultats livrés par les feuilles Excel pour la gestion des anomalies et pour la gestion des tests présentant :
* La liste des anomalies détectées
* La liste des résultats des tests
* Les actions correctives mises en place
* Les maquettes du design des fonctionnalités produites à la suite des résultats de tests ergonomiques
* Un rapport relatif aux différentes anomalies rencontrées accompagnées de leur degré de sévérité.
* Le rapport final des tests contenant :
* La version des logiciels et outils utilisés à l'exécution des tests
* Les résultats des actions mises en place
* La conclusion relative aux tests
* Les ressources engagées
* Les anomalies rencontrées et corrigées
* Les anomalies restantes

**12. Remaining Test Tasks**

Ce projet n'est pas divisé en plusieurs phases. En effet l'intégralité du site web définit dans le cahier des charges sera développé par François Bourrée et Nicolas Douvrin.

Les fonctions qui restent donc à tester sont les fonctions qui sont explicitées dans la partie 7 "Features not to be tested". Il y a :

* **Récupération de données de Facebook pour intégration sur la page d’accueil :** Fonctionnalité bonus non nécessaire au bon fonctionnement du site web
* **Gestion de la charge :** La gestion de la charge revient à l’hébergeur du site sélectionné par l’association.

**13. Environmental Needs**

Pour effectuer les tests, nous devons avoir à disposition des ordinateurs pour chaque personne effectuant les tests, ainsi que les l’Android SDK couplé avec IDE Arduino.

On testera aussi l’intégrité Hardware du prototype afin de s’assurer du bon fonctionnement de l’objet. Une fois les tests Hardware et les simulations par émulations faites, les fonctionnalités seront testées sur l’appareil.

Pour tester l’application dans son intégralité, nous avons besoin de l'héberger sur un serveur (Heroku, OpenShift ou Amazon) : Heroku étant gratuit et la possibilité d’obtention de 100Go utilisable chez Amazon à l’année ne nécessite pas de budget. Ainsi, tous les testeurs pourront y avoir accès moyennant la connaissance du nom de l’app hébergée.

**14. Staffing and Training Needs**

Les premiers tests ne nécessiteront l'implication que des deux concepteurs et développeurs du projet. Chacun aura un rôle de responsable et d'acteur dans chacune des tâches du plan de tests.

L’utilisation de l’objet sera testée par le Club Persévérante pont sur Yonne.

Concernant la formation de ce personnel, elle devra couvrir les utilisations des feuilles Excel relatives à gestion et au référencement des anomalies et cas de test associés, ainsi que des différentes techniques de tests citées dans le paragraphe 8 (Approach). Il pourra également être nécessaire de former le personnel aux différentes problématiques web et réseaux en général, afin d'effectuer des cas de tests efficaces et de trouver des solutions faces aux diverses anomalies (Le personnel étant à la fois testeur et développeur).

**15. Responsibilities**

R A (S) C I

Légende -> R : Responsable

A : Acteur

S : Superviseur

C : Contributeur (ex : architecte)

I : informé

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RACI Tests App BTW** | **Réalisateur du projet Nicolas Douvrin** | **Réalisateur du projet François Bourree** | **Superviseur du projet Mr LEFEVERE** | **Président du club Lucas** | **Membres du Club** |
| **Analyse des risques de l’activité de tests** | R | A | I | I | I |
| **Définition de la couverture des tests** | R | A | I | I | I |
| **Etablissement de la stratégie des tests** | A | R | I | I | I |
| **Définition des exigences quant aux fonctionnalités de l’application** | R | A | I | C | C |
| **Définition des critères d’arrêts des tests** | A | R | I | I | I |
| **Détermination des ressources humaines et matérielles** | R | A | I | I | I |
| **Estimation de la charge et du délai des phases de tests** | A | R | I | C | I |
| **Détermination des fonctionnalités testées et non testées** | R | A | I | C | I |
| **Définition des livrables du processus de tests** | A | R | I | C | I |
| **Mise en place des actions correctives** | R | A | I | I | I |
| **Conception des cas de tests** | A | R | I | I | I |
| **Définir procédures de tests (manuelles ou automatiques)** | R | A | I | I | I |
| **Récupération des résultats de tests** | A | R | I | I | I |
| **Rapports aux parties prenantes** | R | A | I | C | C |
| **Installer un environnement de tests** | R | R | I | C | I |

**16. Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **22/10/17** | **06/12/17** | **17/04/2018** |
| **Cahier des charges** | **X** |  |  |
| **Ergonomie IHM** |  | **X** |  |
| **Plan de test** |  | **X** |  |
| **DSL et DAT** |  | **X** |  |
| **Développement** |  |  | **X** |
| **AVANCEMENT** | **FINI** | **En cours** | **Pas commencé** |

**17. Planning Risks and Contingencies**

Nous avons les délais du projet 100h à respecter.

Ainsi, les événements tels que la possibilité que l’un des membres ne pourra pas remplir son rôle durant la phase de test, il est donc important que les tâches soient réparties afin de pouvoir palier à ce problème.

On pourra aussi noter qu’étant étudiants, divers facteurs peuvent survenir dus à notre vie étudiante (examens, autres projets …).

On peut de plus ajouter qu’une panne de matériel ralentira considérablement voir obligera le groupe de travail à redémarrer le projet et donc l’exécution du plan de tests. Des backups seront donc mis en place afin d’éviter ce risque.

**18. Approvals**

L'approbation du processus sera faite par le client Sylvain LEVASSEUR, le professeur supervisant le projet Mme HASSAN-OUARI, le professeur en charge de l'enseignement du module de Qualité Logiciel M. EL KHALOUI et les réalisateurs du projet Quentin LEVASSEUR et Karl-Milann LOBIR.

Le client Sylvain LEVASSEUR évalue la satisfaction aux exigences du produit livré en fonction de celles spécifiées lors de l'élaboration du cahier des charges et s'assure ainsi que les tests ont permis l'obtention de fonctionnalités satisfaisant aux contraintes imposées. Son approbation conditionne la remise du livrable dans la mesure où il en est le destinataire.

Mr NASSAR Jad estimera la démarche des tests afin d'évaluer l'aspect technique de ceux-ci, leur cohérence ainsi que leur pertinence. Son approbation portera donc sur la stratégie du plan de tests. En raison des compétences et de l'expérience de Mr NASSAR Jad, son approbation apportera la certitude que le plan de tests élaboré conduira à l'obtention d'un projet répondant aux exigences du client.

Des conseils quant à la démarche de tests exposée seront fournis par M. EL KHALOUI lors d'une éventuelle sollicitation de la part de l'équipe de tests.

**19. Glossary**

- **Cahier des charges :** Le cahier des charges a pour objectif de définir les besoins du client et d'y apporter des solutions sans les détailler techniquement.

- **Plan de test :** Le plan de test est un document qui permet de définir les tests qui seront effectués afin d'identifier les comportements problématiques qui peuvent apparaître lors de la conception. Il permet d'augmenter la qualité d'un logiciel et d'apporter des précisions sur la qualité finale du logiciel.

- **Ergonomie IHM :** Établir l'ergonomie de l'application va nous permettre de définir l'aspect général de notre ergonomie. Nous allons créer des scénarios de navigation dans l'application et les présenter à notre client afin de s'assurer que notre application corresponde à ses exigences. Il est important que notre client se sente à l'aise avec l’IHM de l'application car une de nos exigences principales est que l'application soit intégrée au fonctionnement de l'association.

- **DSL (Document de Spécifications Logiciel) :** C'est un dossier de spécification qui va permettre de décrire le fonctionnement technique de l'application. On y trouvera les solutions choisies afin de concilier la conception du code avec les exigences clients.

- **Développement :** Développement de l'application à proprement parler.

- **DAT (Dossier d'architecture technique) :** Ce document va nous permettre de réfléchir sur l'architecture de l'application. On pourra par exemple y trouver l'identification des systèmes externes avec lesquels nous allons communiquer, ainsi que des présentations de Framework utilisés.