

Stratégie de test

Projet : ETNA CALENDAR

Auteur : Mohamed SLAMANI

Date : 01/06/2021

Version : 1.0

Table des matières

Suivi de la Stratégie de test	3
Gestion des versions du document	3
Documents de référence	3
Périmètre, champ d'application	4
Exigences ou User Stories	4
Identification des fonctions critiques	4
Objectifs de test	5
Nature des tests	6
Tests Unitaires	6
Tests d'intégrations	7
Tests Fonctionnels ou Système	8
Tests de Validation	9
Documentation de test	9
Les documents	9
Non-Régression	11
Stratégie de Non-Régression	11
Automatisation des tests	11
Gestion des anomalies	12
Schéma du workflow	12
Définition des statuts	13
Gestion des risques	13
Identification des risques	13
Acteurs & moyens	15
Les acteurs	15
Les moyens	16
Le planning	17
Enchainement des différentes natures de test	17
Matrice des tests	18
Matrice sur nature de test	18

Suivi de la Stratégie de test

Gestion des versions du document

Suivi du document			
Version	Date	Auteur	Motifs
1.0	01/06/2021	Mohamed SLAMANI	Création

Documents de référence

Nom	Identification	Version
Dossier d'architecture	https://agendaetna.atlassian.net/secure/RapidBoard.jspa?rapidView=1&projectKey=AG&view=planning&selectedIssue=AG-1&issueLimit=100	
Spécifications (fonctionnelle ou technique), description des UC	https://agendaetna.atlassian.net/wiki/home	

Périmètre, champ d'application

La stratégie de test porte sur la totalité de l'application de la création de compte au partage d'événements, l'application va être testé sur Android 10

Exigences ou User Stories

Exigence	Document	Type d'exigence
Client	Cahier des Charges	Exigences / US fonctionnelles et technique
Client	Normes d'exploitabilité	Exigences / US d'exploitabilité du client
Produit	Documents de spécifications (Confluence ou e-Collaborative)	Fonctions et règles de gestion
Produit	Documents de spécifications (Confluence ou e-Collaborative)	Spécifications détaillées des composants
Produit	Documents d'interface (Confluence ou e-Collaborative)	Spécifications des interfaces
Produit	Manuel d'Installation (Confluence ou e-Collaborative)	Exigences d'installation en production

Identification des fonctions critiques

Élément critique	Actions
La fonctionnalité de création d'événement a un impact fort sur le CA du client	Porter une attention particulière aux tests sur cette fonctionnalité
La fonctionnalité d'envoi de notification a un impact fort sur l'utilisation de l'application	Porter une attention particulière aux tests sur cette fonctionnalité Essayer plusieurs scenarios

Objectifs de test

1	Faible
2	Moyenne
3	Forte

Objectif de test	Moyen de test	Effort de test	Criticité métier
Créer un nouveau compte Compte non existant préalablement Erreur dans la création du compte Compte existant déjà	Plateforme	2	3
Creation d'événement Choix de la date Choix de l'heure Choix de libellé Persistance base de données	Plateforme	1	3
Push notification Si le checkbox est coché Si le checkbox est décoché Si le checkbox est décoché après la création Si le checkbox est coché après la création	Plateforme	2	3

Nature des tests

Tests Unitaires

Objectif	<p>Chaque composant créé ou modifié fait l'objet de tests unitaires.</p> <p>La définition et l'exécution des tests unitaires sont de la responsabilité du développeur.</p>
Environnement & Outil	<p>Poste de développement.</p> <p>Jira</p>
Description des tests	<p>Lister les composants logiciels testés : les écrans, boutons, onglets, méthodes, fonctions, classes, IHM, fonctionnalités des spécifications...</p>
Documents et enregistrements produits	<p>Les tests unitaires sont tracés dans :</p> <ol style="list-style-type: none">1. Une fiche de tests unitaire/taches incluant<ul style="list-style-type: none">• La technique de test (test manuel IHM),• Le nom du test,• Le nom du composant testé,• Le rédacteur du test,• La date du test et le résultat de passage du test,• Fichier de données ou modification des données.2. Check-list de tests unitaires
Couverture attendue / Critères d'arrêt	<p>Au minimum tous les cas nominaux d'exécution sont testés (BUILD) il n'existe aucune anomalie résiduelle.</p>
Gestion des anomalies	<ul style="list-style-type: none">• Décrire les moyens mis en place pour la traçabilité des anomalies.• Décrire les règles de nommage.

Tests d'intégrations

Objectif	<p>Assembler tous les composants et s'assurer que les appels des composants et que leurs interfaces sont correctes (passage de données entre composants, navigation entre composants interactifs, interface entre chaînes de traitement, etc.).</p> <p>Ils portent sur les liens physiques et logiques entre composants matériel, logiciel, progiciels, réseaux, conformément aux spécifications techniques.</p>
Environnement & Outil	<p>Plate-forme d'intégration</p> <p>Jira-Xray</p>
Description des tests	Liste des modules à intégrer (une partie de l'application).
Documents et enregistrements produits	
Couverture attendue / Critères d'arrêt	<p>On peut avoir une couverture des appels des fonctions (70%) ou des méthodes à atteindre, une couverture des interfaces internes (par exemple si on a un client et un serveur, couvrir les interfaces d'échange entre le client et le serveur).</p> <p>Aucune anomalie résiduelle.</p>
Gestion des anomalies	<ul style="list-style-type: none">Les anomalies détectées aux cours d'une campagne Jira sont directement saisies à partir du pas de test au cours duquel survient l'anomalie.

Tests Fonctionnels ou Système

Objectif	<p>Ces tests consistent à valider fonctionnellement l'application et sont définis dans le plan de tests, conformément aux spécifications fonctionnelles et techniques.</p> <p>Deux types de campagnes de fonctionnels sont réalisées selon l'impact de la modification à effectuer.</p> <ul style="list-style-type: none">• Toute nouvelle fonctionnalité applicative est testée avant d'être livrée. Les résultats des tests sont enregistrés dans le dossier de tests.• Les tests de fonctionnels ou systèmes des corrections d'anomalies et leurs résultats sont enregistrés dans le dossier de tests. <p>La fourniture et la complexité des jeux de données de tests doivent être identifiés : fournis par le client ou à produire par le projet.</p> <p>La profondeur des tests et la liste des scénarii à rejouer pour chaque campagne sont définies par une analyse d'impact des corrections ou des évolutions.</p> <p>Les résultats des tests fonctionnels sont tracés dans le dossier de tests, lui-même enregistré dans l'espace projet. Les anomalies sont saisies sous Jira.</p>
Environnement & Outil	<p>Une plate-forme dédiée</p> <p>Jira-Xray</p>
Description des tests	<p>Build : s'assurer que les exigences sont bien toutes couvertes par les tests via Jira.</p>
Documents et enregistrements produits	
Couverture attendue / Critères d'arrêt	<p>Au minimum tous les tests sont exécutés et il n'existe aucune anomalie bloquante résiduelle.</p>
Gestion des anomalies	<ul style="list-style-type: none">• Les anomalies détectées au cours d'une campagne sont directement saisies à partir du pas de test au cours duquel survient l'anomalie.

Tests de Validation

Objectif	Tests formels destinés à déterminer si un système répond aux exigences. Ils doivent permettre au client de se prononcer pour l'acceptation ou réception du système.
Environnement & Outil	Une plate-forme client Jira-Xray

Documentation de test

Les documents

Nature	Objet	Règle	Remarque
Tests Unitaires	Test Ficher pilote JUnit Fichier résultat de test	TU-<fonction>-<numéro>-... FIC-JUNIT-... FIC-RES-...	Répertoire sur Confluence
Tests Unitaires	Checklist IHM	TU-<IHM>-	
Tests d'Intégration	Organisation des tests	TI-FONC-CRE-01	
Tests Fonctionnels	Test Fichiers en entrée Fichiers en sortie	TF-<fonction>-<numéro>-... FIC-E-xxx FIC-S-yyy	

Tests de Charge	Nb d'utilisateurs connectés	Charge Min :	
Tests de Performance	Tests d'installation		
Tests de Volumétrie	Nb de contrats traités		
Autres	Bilan de tests		
Autres	Suivi des anomalies		

Non-Régression

Stratégie de Non-Régression

Objectif :

- Réitérations sélectives de tests pour vérifier que les modifications n'ont pas provoqué de dégradation du fonctionnement du système par rapport à la version précédente et que le système modifié satisfait toujours aux exigences spécifiées.
- La profondeur des tests et la liste des scénarii à rejouer pour chaque campagne sont définies par une analyse d'impact des corrections ou des évolutions.
- La dernière campagne de tests de validation avant livraison au client doit inclure des tests de non-régression. Cette campagne est composée des sous parties suivantes :
 - **Test des éléments critiques** de l'application,
 - **Les tests génériques** : ces tests permettent de vérifier qu'il n'y a pas de régression d'autres fonctionnalités sur la version demandée. Il s'agit ici de dérouler tout ou partie des tests techniques et fonctionnels qui ont été capitalisés (catalogue de scénarii). Ce sont plutôt des tests de bout en bout fonctionnels et métiers
 - **Les tests spécifiques aux modifications** : extraction des tests en rapport direct avec la correction ou la modification. Ces tests sont des tests plus unitaires
 - **L'outil** (robot) utilisé pour la mise en œuvre de la NR et permettant de l'exécution de chaîne de chaîne de tests.

Automatisation des tests

Préciser ici les éléments qui permettent de mettre en œuvre l'automatisation :

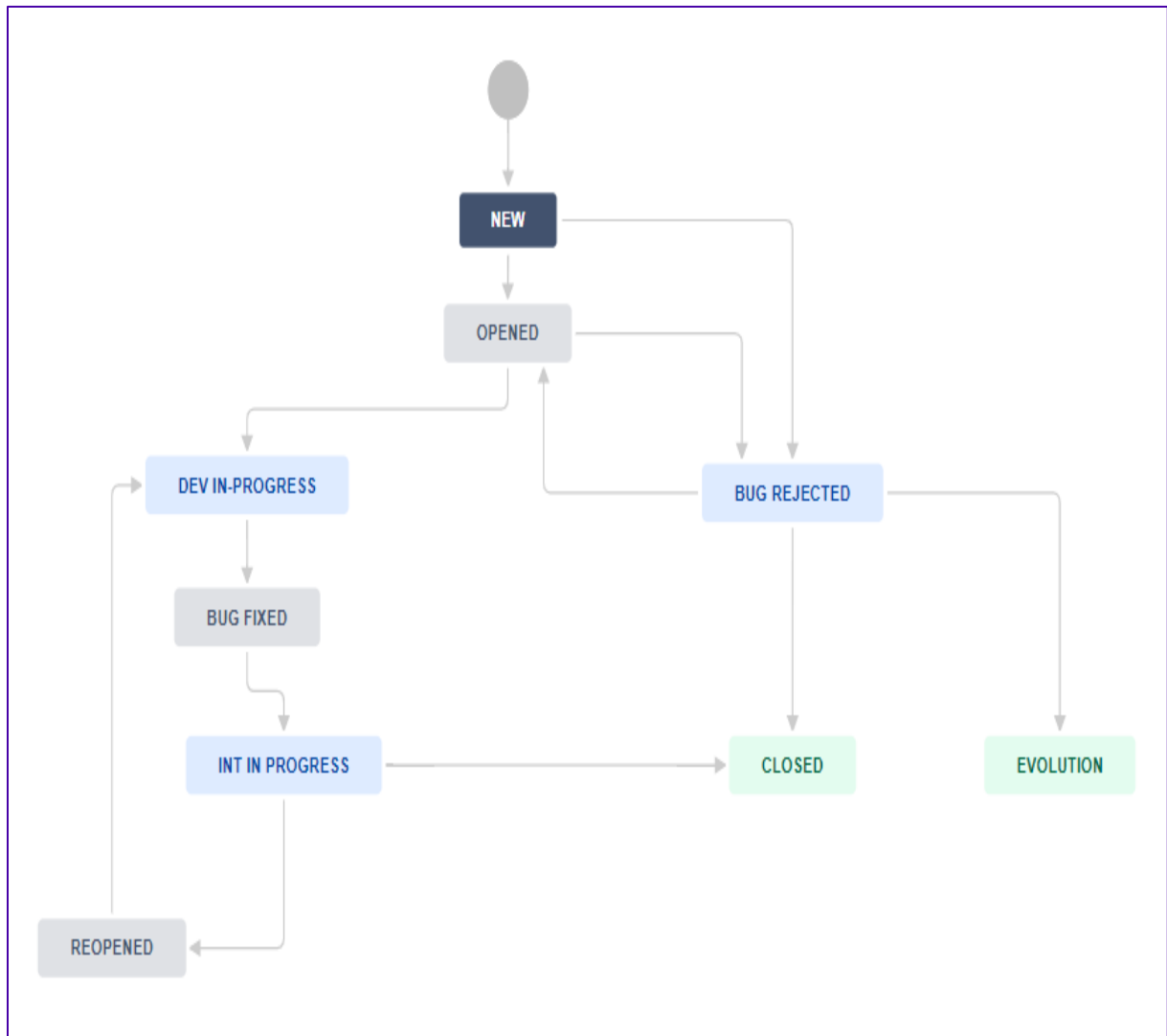
- Choix du périmètre de tests,
- Avantages et inconvénients de cette automatisation,
- Outil mis en œuvre.

Gestion des anomalies

Schéma du workflow

Objectif :

- Workflow utilisé : Jira.



Définition des statuts

Status	Description
New	L'anomalie est créée mais non encore prise en compte
Opened	L'anomalie est en-cours de diagnostic
Dev-in-progress	L'anomalie est prise en charge
Bug-fixed	La correction est validée et prête pour la livraison.
Int in Progress	L'anomalie est en intégration
Reopened	L'anomalie est réouverte
Bug Rejected	Après analyse, le responsable de l'anomalie considère que ce n'est pas une anomalie et demande son abandon (par exemple : conforme aux specs, ...)
Evolution	Anomalie transformée en Evolution
Closed	Fermeture de l'anomalie

Gestion des risques

Identification des risques

- Identifier les facteurs pouvant causer un retard ou une impossibilité et identifier les actions de réduction associées.

R	Retard
I	Inadéquation
S	Satisfaction
C	Coût

Id	Libellé	Criticité (1 à 6)	Impact (RISC)	Actions
R1	Absence de plate-forme de validation	4	R,C	Regarder la possibilité d'utiliser la plate forme d'intégration (solution d'attente)

Acteurs & moyens

Les acteurs

Activités	Qui	Quand	Remarques
Élaboration des jeux de données de test	Mohamed SLAMANI		
Elaboration des outils de tests (simulation...)	Mohamed SLAMANI		
Exécution des tests unitaires	Nicolas BOULEIN		
Exécution des tests d'intégration	Nicolas BOULEIN		
Exécution des tests de fonctionnels	Nicolas BOULEIN		
Organisation des campagnes de test	Mohamed SLAMANI		
Analyse des résultats obtenus	Mohamed SLAMANI		
Enregistrement des résultats obtenus	William BROWAYS		
Ouverture et Enregistrement des FA (defects)	William BROWAYS		
Archivage des résultats	William BROWAYS		
Production des rapports de tests	Mohamed SLAMANI		

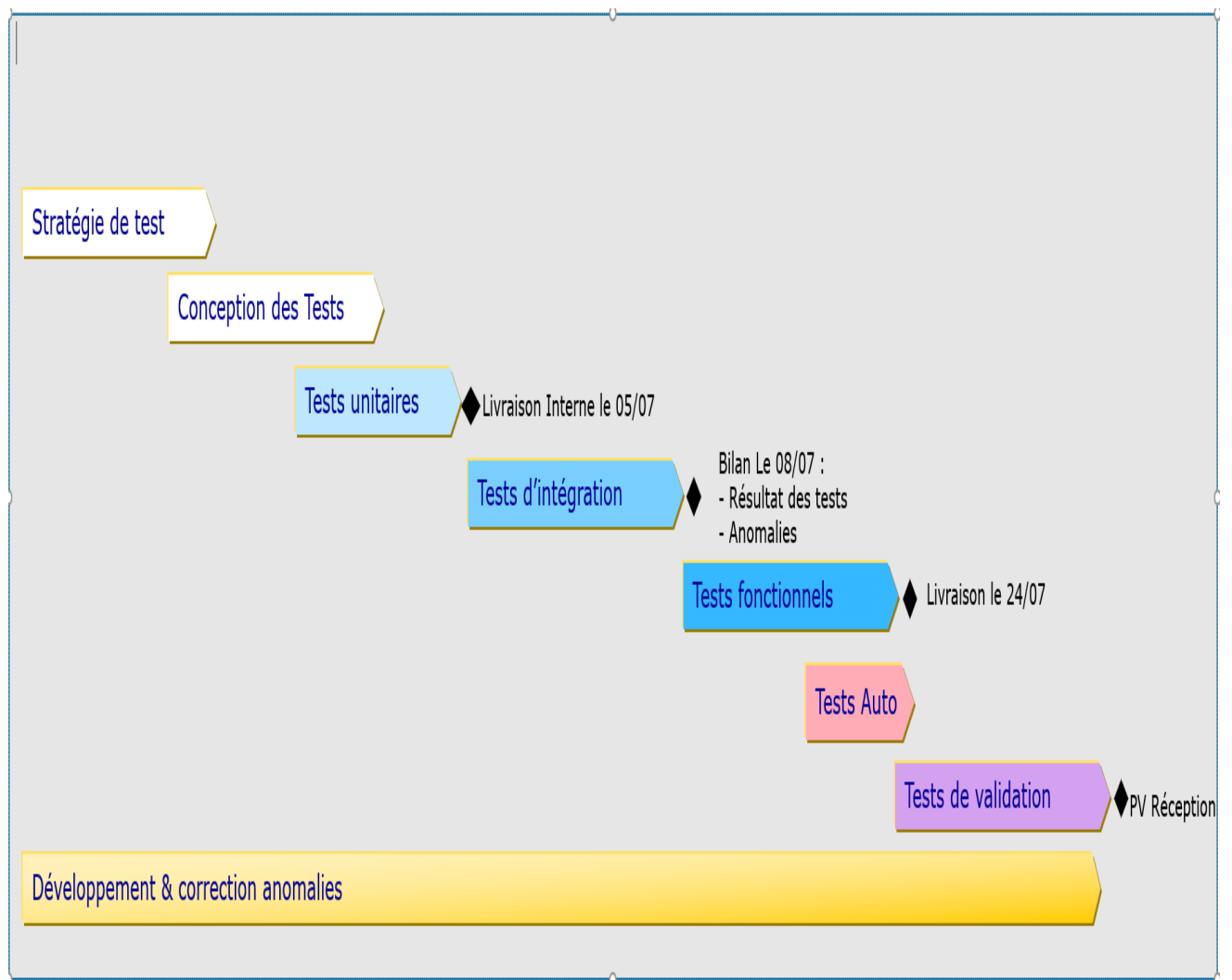
Les moyens

Les besoins	Les moyens	Fournisseur	Priorité	Date
Matériel pour testeurs	- PC , Smartphone Android - Outil de lecture en BDD : Toad, SQL Navigator, Squirrel (Freeware)		2	
Outil de gestion des tests	Jira-Xray, HP ALM		2	
Outil de gestion des anomalies et évolutions	Jira, HP ALM		2	
Outil d'automatisation de test	Selenium, HP UFT			
Création des jeux de test				
Scripts d'initialisation de la bdd: -Effacement des données de test erronées -Chargement des jeux de tests				
Outil de test en charge : -Simulation de la charge en connexion utilisateurs -Simulation de chargement en bdd	NRG Global Apploader			
Outil de suivi des performances				

Outils de profiling mémoire	JProfiler/Jprobe, Valgrind/Purify			
-----------------------------	--------------------------------------	--	--	--

Le planning

Enchaînement des différentes natures de test



Matrice des tests

Matrice sur nature de test

1	Simple
2	Important
3	Indispensable

Nature de test	Unitaire	Intégration	Fonctionnel	Validation
Tests structurels				
Couverture du code : mode nominal	3	2		
Couverture du code : cas d'erreurs	3	1		
Respect des règles de programmation	3			
Mesures de complexité	3			
Portabilité	3		3 (si exigence)	
Tests d'utilisabilité				
Test des IHM : navigation, saisies, libellés	3		1	
Respect de la charte d'ergonomie	3		2	
Pertinence de l'aide en ligne			2	
Tests fonctionnels				
Respect des spécifications fonctionnelles	2	1	3	

Respect des spécifications techniques	2	3		
Autres fonctions à ajouter...				
Tests de sécurité				
Contrôle des accès	1	2	3	
Test de la confidentialité de données	1	2	3	
Tests d'intrusions dans le système	1	2	3	
Coupe-feu			3	
Tests de sûreté				
Intégrité de la base de données	1	2	2	
Reprise à chaud / reprise à froid		3	2	
Sauvegarde / restauration			2	
Robustesse des interfaces		2	2	
Tests d'exploitation				
Installation / génération du système		2	2	
Téléchargement des applications			2	
Remontée d'alarmes	1	2	2	
Configuration / administration de paramètres	1	2	2	
Création et gestion des utilisateurs		2	2	

Nature de test	Unitaire	Intégration	Fonctionnel	Validation
Tests de performance				
Durée d'exécution des traitements	1	2	3	
Temps de réponse moyen	1	2	3	
Temps de réponse maximum dans 90% des cas		2	3	
Montée en charge		3	2	
Occupation des ressources		2	3	
Tests de fiabilité				
Analyse des défaillances (tests difficilement jouables)			2	
Mesure de la disponibilité (difficulté de faire des tests)			1	
Passage en modes dégradés		2	3	
Tests d'interopérabilité				
Interopérabilité entre applications		2	3	
Interopérabilité avec des systèmes externes			3	
Tests administratifs				
Vérification de la complétude de la fourniture contractuelle			2	