
Plan de Test Système

CANvengers

Responsable du document : Paul TRÉMOUREUX

État du document : Validé

Version : 2.0.0

Date	Actions	Auteur	Version	Révision
05/03/2019	Création de la trame du Plan de Test.	Camille CONSTANT	0.1	1
22/02/2021	Modification des critères d'acceptation, ajout des exemples de diagramme d'activité, ajout de la section Validation du document	Camille CONSTANT	0.2	1
03/03/2022	Mise à jour des logiciels utilisés	Camille CONSTANT	0.2	2
29/03/2022	Modification des critères d'acceptation et mention de la couverture fonctionnelle	Camille CONSTANT	0.2	3
06/03/2023	Modifications pour mieux coller à la norme ISO 829-2008	Camille CONSTANT	0.3	1
10/03/2023	Mise à jour du document	Théo BÉNARD	0.4	0
15/03/2023	Prise en main du document	Paul TRÉMOUREUX	0.4	1
16/03/2023	Corrections suite au consulting test du 16/03/2023	Paul TRÉMOUREUX	0.4	2
16/03/2023	Relecture avant consulting	Paul TRÉMOUREUX	0.5	0
17/03/2023	Corrections suite à l'audit consultatif du 17/03/2023	Paul TRÉMOUREUX	0.5	1
21/03/2023	Suite et fin corrections suite à l'audit consultatif du 17/03/2023	Paul TRÉMOUREUX	0.5	2
22/03/2023	Modification de l'utilisation de symbole	Paul TRÉMOUREUX	0.5	3
01/05/2023	Début de correction suite aux remarques du client	Paul TRÉMOUREUX	0.5	4
13/05/2023	Relecture finale de l'incrément 1	Paul TRÉMOUREUX	0.6	0
15/05/2023	Relecture globale du document	Théo BÉNARD	1.0	0

24/05/2023	Modification du document pour l'incrément 2	Paul TRÉMOUREUX	1.0	1
25/05/2023	Dernières modifications du document pour l'incrément 2	Paul TRÉMOUREUX	1.0	2
25/05/2023	Relecture avant l'audit normatif	Paul TRÉMOUREUX	1.1	0
30/05/2023	Début correction suite AN	Paul TRÉMOUREUX	1.1	1
31/05/2023	Fin correction suite AN	Paul TRÉMOUREUX	1.1	2
06/06/2023	Corrections suite relecture de Théo BÉNARD	Paul TRÉMOUREUX	1.2	0
12/06/2023	Validation du document	Paul TRÉMOUREUX	2.0	0

TABLE 2 – Table des évolutions et validations internes du document

Table des matières

1	Introduction	5
1.1	Contexte	5
1.2	Objet	5
1.3	Portée	5
1.4	Copyright	5
1.5	Présentation du système	5
1.6	Références	6
1.6.1	Documents de référence internes	6
1.6.2	Documents de référence externes	6
1.7	Glossaire et abréviations	6
1.7.1	Abréviations	6
1.7.2	Glossaire	6
1.8	Conformité	7
2	Périmètre de test	8
2.1	Caractéristiques du projet	8
2.2	Éléments à tester	9
2.2.1	Éléments concernés par les tests	9
2.2.2	Éléments non concernés par les tests	9
2.3	Spécifications fonctionnelles ou techniques à tester	10
2.3.1	Fonctionnalités à tester	10
2.3.2	Caractéristiques techniques à tester	12
2.4	Spécifications fonctionnelles ou techniques non testées	13
2.5	Criticité	14
2.6	Risques	14
2.6.1	Risques projet	14
2.6.2	Risques produit	15
2.7	Effort de test	16
3	Processus et Stratégie de test	17
3.1	Objectifs (actions de test)	17
3.2	Organisation	17
3.2.1	Découpage en phase de tests/campagnes	17
3.2.2	Gestion des rapports d'anomalie	18
3.3	Critères d'acceptation des tests	18
3.4	Critères d'arrêt	18
3.5	Activités de test	18
3.5.1	Planification	18
3.5.2	Conception	18
3.5.3	Exécution	19
3.5.4	Bilan	19
3.6	Documents de test et livrables	19

4	Environnement de test	20
4.1	Environnement matériel et logiciel de test	20
4.2	Outils de test	20
5	Rôles et responsabilités	21
6	Équipe de test	22
7	Planning prévisionnel	23
8	Validation du document	24

1 Introduction

1.1 Contexte

Produit à tester :	ProSE - version 2.0.0
Type de produit :	Passerelle Android-CAN vers banc CAN réel ou simulé.
Commanditaire :	KEREVAL
Développeur :	CANvengers
Testeur :	CANvengers

1.2 Objet

Ce document décrit l'activité de test système qui sera menée par CANvengers durant le projet ProSE dans le but de valider le produit suivant : Passerelle Android-CAN vers banc CAN réel ou simulé. Il est rédigé sous la responsabilité du Responsable Qualité-Test (RQT), sous la direction du Chef de Projet (CdP), conformément au Plan d'Assurance Qualité Logicielle (PAQL) élaboré sous la responsabilité du RQT (cf. section 6, Équipe de test).

1.3 Portée

Sont concernés par ce document :

- les testeurs : afin que ceux-ci connaissent le périmètre des tests (ce qu'ils vont tester), l'environnement de test (comment les tests seront mis en œuvre) et le processus de test (comment s'y prendre et rendre compte des résultats lors de l'exécution des tests) ;
- les développeurs : à titre informatif, afin que ceux-ci sachent comment va être validée leur production ; à titre indicatif afin qu'ils sachent, par la description de la gestion des anomalies, comment ils s'interfaceront avec l'équipe de test ;
- le client : ce plan de test fait l'objet d'une contractualisation avec le client pour déterminer le périmètre des tests menés pour valider le produit livré et les niveaux d'acceptation de cette validation ;
- les auditeurs : ce plan de test, ainsi que son implication, feront l'objet d'audits par la société Formato.

1.4 Copyright

Cf. [PAQL_B1_2024] (section 1.3. Copyright).

1.5 Présentation du système

Le système développé est un ensemble de 2 programmes : une application Android nommé CANdroid déployé sur un smartphone Android et un programme C nommé CANgateway déployé sur une Raspberry Pi. L'application CANdroid sera relié au programme CANgateway par un réseau TCP/IP. La Raspberry Pi sera connectée à un réseau CAN pour dialoguer avec Tableau de Bord (soit banc de test physique soit simulateur sur un PC). Ce projet permettra

d'envoyer des trames depuis l'application CANdroid vers Tableau de Bord afin de piloter ce dernier à distance.

1.6 Références

1.6.1 Documents de référence internes

Ref.	Nom et auteur	Version	Source
[SPEC_B1]	Dossier de spécifications - CANvengers	2.0.0	se2024-b1.doc/specification/livrables
[PAQL_B1_2024]	Plan d'Assurance Qualité Logiciel - Paul TRÉMOUREUX	1.0.0	pdf sur le dépôt

1.6.2 Documents de référence externes

Ref.	Nom
[ISO-829-2008]	Documentation de test logiciel
[ISO/IEC/IEEE 29119-1 :2022]	Ingénierie du logiciel et des systèmes - Essais du logiciel - Partie 1 : Concepts généraux

1.7 Glossaire et abréviations

Ce sont les termes et abréviations nécessaires à la compréhension de l'activité de test (les termes techniques propres au projet seront indiqués dans le dossier de spécification).

1.7.1 Abréviations

CdP	Chef de Projet
IHM	Interface Homme-Machine
PAQL	Plan Assurance Qualité Logicielle
RQT	Responsable Qualité et Test

1.7.2 Glossaire

Campagne de test	Activité qui consiste à dérouler un ensemble de jeux de test. Un dossier de test est produit à l'issue d'une campagne.
Cas de test	Déclinaison d'un test précisant les valeurs utilisées pour les variables du test ainsi que les résultats attendus.

Dossier de test	Ensemble documentaire qui contient la description des scénarios et des cas de tests, ainsi que l'exécution des jeux de test. Le dossier de test est le reflet d'une campagne de test.
Jeux de test	Ensemble des scénarios et cas de tests permettant de tester un produit logiciel. L'enchaînement des cas et scénarios de tests est relatif à une stratégie de test précisée dans le plan de test.
Plan de test	Document décrivant le déroulement d'un jeu de test : stratégie de test, critères d'arrêt, planification.
Scénarios de test	Ensemble de cas de tests cohérents permettant de traiter un objectif fonctionnel.
Test d'intégration	Vérification effectuée pour montrer des défauts dans les interfaces et interactions de composants ou systèmes intégrés.
Test de non régression	Vérification qu'une nouvelle version du produit fonctionne sans dégradation (technique, fonctionnelle, performance) par rapport à la version précédente.
Test de validation	Vérification que le produit est cohérent et complet par rapport aux spécifications fonctionnelles.
Test fonctionnel	Test (vu de l'utilisateur) du bon fonctionnement d'un produit logiciel, d'une fonctionnalité ou d'une fonction de base. Vérification par rapport aux spécifications.
Test système	Vérification que le système dans son ensemble est cohérent et complet par rapport aux spécifications fonctionnelles et techniques.
Test unitaire	Vérification que les composants logiciels individuels sont cohérents et complets par rapport aux spécifications.

Voir également le [Glossaire CFTL/ISTQB des termes utilisés en tests de logiciels].

1.8 Conformité

Ce plan de test est conforme aux normes :

- IEEE Std. 1012-1986
- ISO Std. 829-2008
- IEEE Std. 1008-1987
- ISO/IEC/IEEE 29119-1 :2022

2 Périmètre de test

Cette section a pour objet l'élaboration d'un tableau des fonctionnalités et/ou composants/traitements/données du système mentionnant pour chacun l'effort de test à mener. Cet effort est fonction de la pondération des exigences, risques et criticité retenus.

2.1 Caractéristiques du projet

KEREVAL, une entreprise spécialisée dans les tests de systèmes embarqués dans les véhicules, souhaite développer un démonstrateur CAN « simulator in the loop » pour permettre à ses équipes de monter en compétences sur le réseau CAN.

L'objectif est également de fournir une visualisation concrète de l'architecture du véhicule, du réseau CAN et du démonstrateur CAN pour les personnes qui débutent dans le métier, tels que les nouveaux salariés lors de leur arrivées chez KEREVAL, ou des étudiants lors de forums.

Pour répondre à ce besoin, KEREVAL souhaite pouvoir utiliser un simulateur de tableau de bord sur Linux (type ICSim) et/ou un banc de test physique connecté(s) à une carte électronique (type Raspberry Pi) pour permettre la connection au réseau CAN, à la fois réels (sur le banc de test fourni par KEREVAL) et simulés (par le simulateur de tableau de bord).

De plus, KEREVAL souhaite contrôler le système à distance via une application Android nommée CANDroid déployée sur un smartphone. Ce dernier sera connecté au système (via la Raspberry Pi) par un réseau TCP/IP. Sur l'application, il sera possible d'envoyer et d'enregistrer des trames mais également d'observer toutes les trames diffusées sur le bus CAN. Il sera possible d'enregistrer les trames diffusées sur le bus dans un fichier présent sur la Raspberry Pi.

En outre, l'entreprise souhaite être en mesure d'injecter des fautes via les trames pour s'assurer du bon fonctionnement du démonstrateur.

Le projet ProSE se décompose en 2 lots/incréments :

- Lot 1 :
 - Communication TCP/IP (entre le programme CANGateway et l'application CANDroid) :
 - Envoi de trame depuis l'application CANDroid
 - Reception de trame sur l'application CANDroid
 - Communication CAN (entre le programme CANGateway et Tableau de Bord) :
 - Sniffer fonctionnel sur le programme CANGateway
 - Communication avec Simulateur ICSim
 - Gestion du SàE :
 - Démarrer le SàE
- Lot 2 :
 - Communication CAN (entre le programme CANGateway et Tableau de Bord) :
 - Communication avec Banc de Test
 - Gestion des objets sur l'application CANDroid :

- Création d'objets sur l'application CANdroid
- Création de trames dans un objet sur l'application CANdroid
- Mode d'envoi
- Suppression d'objets sur l'application CANdroid
- Suppression de trames dans un objet sur l'application CANdroid
- Gestion du sniffer sur l'application CANdroid :
 - Pause du sniffer sur l'application CANdroid
 - Nettoyage du sniffer sur l'application CANdroid
 - Retour en haut du sniffer sur l'application CANdroid
- Gestion de la communication sur l'application CANdroid :
 - Reconnexion
- Gestion du SàE :
 - Stopper le SàE

KEREVAL souhaite connaître la qualité globale de « Passerelle Android-CAN vers banc CAN réel ou simulé » après chaque lot afin d'éventuellement redéfinir chacun des lots.

Ce plan de test concerne le niveau de test système. Pour information, des tests d'intégration (lots 1 et 2) et des tests unitaires (lot 2) auront été réalisés par CANvengers.

2.2 Éléments à tester

Cette partie s'appuie sur la section « 2.3 Fonctions principales développées » du document de spécifications [SPEC_B1].

Les éléments à tester se décomposent en plusieurs catégories :

- Logiciels développés :
 - L'application CANdroid
 - Le programme CANgateway
- Support de communication :
 - Protocole de communication

2.2.1 Éléments concernés par les tests

Cette section désigne ce qui est testé (composant, logiciel, sous-système). Seront concernés par l'activité de test les composants logiciels développés durant le projet ProSE : le programme CANgateway et l'application CANdroid, ainsi que la couche application de la communication entre le programme CANgateway et l'application CANdroid (+ drivers si développés).

2.2.2 Éléments non concernés par les tests

Cette section désigne ce qui ne va pas être testé (composant, logiciel, sous-système).

Ne seront pas concernés par les tests les supports d'exécution logiciels (Linux, Android, librairies, etc.) et matériels (E_Smartphone, E_Raspberry, E_BancTest et E_ICSim) ainsi que les supports de communication matériels (E_PC, E_PEAK, E_Shield) et logiciels (pile TCP/IP).

2.3 Spécifications fonctionnelles ou techniques à tester

Cette section s'appuie sur la section 2.3 du document de spécifications [SPEC_B1]. Cette section s'appuie également sur la matrice Fonctionnalités-CUs [MatriceFonctionnalitesCU-Tests.pdf].

2.3.1 Fonctionnalités à tester

Les fonctionnalités suivantes sont à tester (avec les paramètres et jeux de données prédéfinis) :

Fonctionnalité	Priorité (P0 : priorité max)	Commentaire
Envoi de trames depuis l'application CANdroid	P0	L'utilisateur doit pouvoir demander l'envoi de trames depuis l'application CANdroid. L'application CANdroid doit être capable d'envoyer des trames vers le programme CANgateway.
Réception de trames sur l'application CANdroid	P0	L'application CANdroid doit être capable de recevoir des trames depuis le programme CANgateway.
Sniffer fonctionnel sur le programme CANgateway	P0	Le programme CANgateway doit être capable de lire sur le réseau CAN.
Communication avec Banc de Test	P5	Le programme CANgateway doit être capable de communiquer avec Banc de Test via le réseau CAN.
Communication avec Simulateur IC-Sim	P1	Le programme CANgateway doit être capable de communiquer avec Simulateur ICSim via le réseau CAN.
Création d'objets sur l'application CANdroid	P2	L'utilisateur doit pouvoir créer un objet sur l'application CANdroid. L'application CANdroid doit être capable d'afficher cet objet.
Création de trames dans un objet sur l'application CANdroid	P2	L'utilisateur doit pouvoir créer une trame dans un objet sur l'application CANdroid. L'application CANdroid doit être capable d'afficher cette trame dans l'objet.

Fonctionnalité	Priorité (P0 : priorité max)	Commentaire
Mode d'envoi	P3	L'utilisateur doit pouvoir choisir le mode d'envoi de la trame créée sur l'application CANdroid. L'application CANdroid doit être capable d'envoyer les trames selon leurs modes d'envoi.
Suppression d'objets sur l'application CANdroid	P4	L'utilisateur doit pouvoir supprimer un objet sur l'application CANdroid. L'application CANdroid doit être capable d'enlever cet objet de l'affichage.
Suppression de trames dans un objet sur l'application CANdroid	P4	L'utilisateur doit pouvoir supprimer une trame dans un objet sur l'application CANdroid. L'application CANdroid doit être capable d'enlever cette trame de l'affichage de l'objet.
Pause du sniffer sur l'application CANdroid	P2	L'utilisateur doit pouvoir demander la pause du sniffer sur l'application CANdroid. L'application CANdroid doit pouvoir mettre en pause la réception des trames dans le sniffer.
Nettoyage du sniffer sur l'application CANdroid	P3	L'utilisateur doit pouvoir demander le nettoyage du sniffer sur l'application CANdroid. L'application CANdroid doit pouvoir effacer les trames affichées dans la partie sniffer de l'application CANdroid.
Enregistrement du sniffer dans un fichier de log	P2	L'utilisateur doit pouvoir demander sur l'application CANdroid l'enregistrement des trames reçues dans un fichier. L'application CANdroid doit pouvoir enregistrer les trames reçues dans un fichier.
Retour en haut du sniffer sur l'application CANdroid	P4	L'utilisateur doit pouvoir demander sur l'application CANdroid de revenir en haut de l'affichage des trames du sniffer. L'application CANdroid doit pouvoir revenir en haut de l'affichage des trames du sniffer.

Fonctionnalité	Priorité (P0 : priorité max)	Commentaire
Reconnexion	P2	L'utilisateur doit pouvoir demander sur la reconnexion de l'application CANdroid avec le programme CANgateway. L'application CANdroid doit pouvoir se reconnecter avec le programme CANgateway.
Démarrer le SàE	P0	L'utilisateur doit pouvoir démarrer le SàE.
Stopper le SàE	P2	L'utilisateur doit pouvoir stopper le SàE.
Redémarrer le SàE	P2	L'utilisateur doit pouvoir redémarrer le SàE. Le SàE doit permettre une persistance des données.

2.3.2 Caractéristiques techniques à tester

Cette section s'appuie sur la section « 2.4 Contraintes » du document de spécifications [SPEC_B1].

Les caractéristiques techniques suivantes sont à tester :

Caractéristique	Priorité (P0 : priorité max)	Commentaire
Fiabilité	P1	Le système doit être robuste aux arrêts/redémarrages/coupures électriques.
Ergonomie	P1	Le sous-système application Android doit être ergonomique. Les règles d'ergonomie ont été échangées avec KEREVAL.
Facilité d'utilisation	P0	Le sous-système application Android doit être intuitif, i.e. prise en main et utilisation sans documentation.
Facilité d'utilisation	P1	Le sous-système application Android doit permettre à l'utilisateur de savoir dans quel état est le système complet (état de marche, en erreur, en mode dégradé).

Caractéristique	Priorité (P0 : priorité max)	Commentaire
Lisibilité message	P0	Les messages d'erreur doivent être compréhensibles, doivent permettre de diagnostiquer le système et le remettre dans un état opérationnel.
Performance en sortie	P1	Les actionneurs doivent répondre en moins de 1 seconde suite à une sollicitation via le sous-système application Android (temps de prise en compte d'une commande).
Performance en entrée	P1	Une information capteur est affichée en moins de 1 seconde sur le sous-système application Android (temps de remontée d'une information capteur).
Maintenabilité	P1	L'architecture du système doit être modulaire et permettre de remplacer/d'ajouter des capteurs/actionneurs commandables via l'application Android.

2.4 Spécifications fonctionnelles ou techniques non testées

L'ergonomie ainsi que la conformité de l'emplacement des éléments de l'IHM aux spécifications ne sera pas testée. L'IHM ne sera validée qu'au travers des tests fonctionnels. Le fonctionnement du tableau de bord type banc de test, de ses actionneurs et de ses voyants ne sera pas testé. L'ensemble des éléments listés dans la partie « 2.2.2 Éléments non concernés par les tests » de ce document ne seront pas testés.

Les caractéristiques techniques suivantes ne seront pas testées :

Caractéristique	Commentaire
Sécurité	Le système doit implémenter les bonnes pratiques de sécurité imposées par les développements sécurisés.
Portabilité système	Le sous-système application Android peut être portable sur les différentes versions Android, à partir de version 9.

Caractéristique	Commentaire
Portabilité matériel	Le sous-système application Android peut être portable sur les smartphones supportant une version 9 ou supérieure.
Eco-conception	Le système ne doit pas consommer plus que la valeur vue avec le client.

2.5 Criticité

Les éléments suivants revêtent une importance critique :

- La communication entre le programme CANgateway et l'application CANdroid ;
- La communication entre le programme CANgateway et la partie tableau de bord.

2.6 Risques

Id : identifiant du risque

Description : description du risque

Effet : effet du risque

P : probabilité (3 - très probable, 2 - probable, 1 - peu probable)

I : impact (3 - impact fort, 2 - impact moyen, 1 - impact faible)

EI (élément impacté) : coût/qualité/délai

Action : description de l'action pour maîtriser le risque

2.6.1 Risques projet

Id	Intitulé	Effet	P	I	EI	Action
RPRJ1	Pas de test unitaire	Instabilité de l'application lors des tests système	2	2	C/Q/D	Faire une phase de smoke tests sur l'application avant de réaliser les tests système.
RPRJ2	Pas de test d'intégration	Instabilité de l'application lors des tests système	1	2	C/Q/D	Faire une phase de smoke tests sur l'application avant de réaliser les tests système.
RPRJ3	Pas de test unitaire ou de test d'intégration	Instabilité de l'application lors de tests d'acceptation	1	2	C/Q/D	Réaliser des tests système sur toutes les fonctionnalités système.

Id	Intitulé	Effet	P	I	EI	Action
RPRJ4	Problème de disponibilité des intervenants	Dérive dans le planning des tests	1	1	D	Planifier au plus tôt les actions des différents intervenants.
RPRJ5	Spécifications du produit « Passerelle Android-CAN vers banc CAN réel ou simulé » non à jour	Déviations entre les spécifications et le système d'où une difficulté pour concevoir des tests pertinents	1	2	Q	Analyse des spécifications pour identifier des écarts. Poser toutes les questions nécessaires à une bonne compréhension des spécifications.
RPRJ6	Non respect des règles inscrites dans le PAQL	Conflits de noms/dépôts sur le RDP. Baisse de la qualité de la production.	2	1	Q	Activité de contrôle régulier des productions de l'équipe.

2.6.2 Risques produit

Id	Intitulé	Effet	P	I	EI	Action
RPRD1	Mauvaise implémentation de la communication	Système non fonctionnel	1	3	C/Q/D	Tester la communication en priorité par des tests d'intégrations (voir unitaires) avant de faire les tests système.
RPRD2	Dysfonctionnement du matériel (Tableaux de Bord (type Banc de Test))	Système non fonctionnel	2	2	C/Q/D	Formation à l'utilisation du matériel. Aide de KEREVAL.

Id	Intitulé	Effet	P	I	EI	Action
RPRD3	Dysfonctionnement du matériel (Smartphone, Raspberry PI, Tableaux de Bord (type Simulateur), Connecteurs CAN)	Système non fonctionnel	1	3	C/Q/D	Formation à l'utilisation du matériel, redondance et bouchonnage.

2.7 Effort de test

L'effort de test sera priorisé de la façon suivante :

- Phase de « Smoke test » pour vérifier la stabilité de l'application avant de réaliser la campagne de tests fonctionnels système ;
- Campagne de tests fonctionnels système par priorité ;
- Campagne de tests non fonctionnels système par priorité.

3 Processus et Stratégie de test

3.1 Objectifs (actions de test)

Exigence : exigence concernée

Risque : risque concerné (cf. section 2.6)

Niveau : niveau de test (S : Système, I : Intégration, U : Unitaire)

Technique : technique de test (AP : Analyse Partitionnelle ou Classes d'équivalence, AL : Analyse aux limites, CU : Cas d'Utilisation, PC : Protocole de Communication)

N°	Énoncé de l'objectif	Exigence	Risque	Niveau	Technique	Conditions de mesure/niveau d'atteinte prévu.
1	Tester la communication	Fonct. P0	RPRD1	I, S	AP, PC, CU	Fonctionnalités P0/100% des fonctionnalités testées en utilisant les classes d'équivalence, le protocole de communication et les cas d'utilisation.
2	Smoke test	Toutes	RPRJ1, RPRJ2, RPRJ4	S	Test par expérience	Nombre d'anomalies bloquantes/pas d'anomalie bloquante.
3	Tester 100% des fonctionnalités P0	Fonct. P0	RPRJ3, RPRD1	S	AP, CU	Fonctionnalités P0/100% des fonctionnalités testées en utilisant les classes d'équivalence et les cas d'utilisation.
4	Tester 100% des fonctionnalités non P0	Fonct. non P0	RPRJ3, RPRD1	S	AP, CU	Nombre de cas d'utilisation/tous les cas d'utilisation testés.
5	Contrôle qualité des productions.	Toutes	RPRJ6	U	Relecture, PAQL	Respect des règles du PAQL.

3.2 Organisation

3.2.1 Découpage en phase de tests/campagnes

Deux campagnes de tests système sont prévues dans le projet pour chaque lot/incrément :

- Campagne de tests système comprenant les tests fonctionnels et non fonctionnels pour atteindre les différents objectifs de tests ci-dessus ;
- Campagne de retest (vérification de la correction des anomalies détectées) et de régression.

3.2.2 Gestion des rapports d'anomalie

Les anomalies sont gérées dans l'ENTP sous forme de tâche. Dès l'observation d'une défaillance dans le produit, un rapport d'anomalie est rédigé dans l'ENTP.

3.3 Critères d'acceptation des tests

Pour le passage en test de validation système, la phase de smoke test ne doit pas détecter d'anomalie bloquante.

Pour la mise en production, aucune anomalie bloquante ni majeure n'est acceptée.

Anomalie bloquante	La fonctionnalité n'est pas utilisable.
Anomalie majeure	La fonctionnalité ne répond pas à ses exigences mais une solution de contournement existe pour utiliser la fonctionnalité, ou la fonctionnalité est utilisable en l'état (par exemple, anomalie dans une règle de calcul).
Anomalie mineure	La fonctionnalité est utilisable mais pas de façon optimale (par exemple, problème d'ergonomie ou de charte graphique).

3.4 Critères d'arrêt

Les tests d'une fonctionnalité s'arrêteront si une anomalie bloquante est découverte ne permettant pas de poursuivre les tests de cette fonctionnalité.

3.5 Activités de test

L'activité de test sera faite par CANvengers tout au long du cycle de développement, via notamment :

- Des tests de validation sur le comportement nominal et aux limites du système ;
- Des tests d'intégration sur le comportement nominal du programme CANgateway ;
- Des tests unitaires nominaux sur certaines classes de l'application CANDroid et sur certains modules du programme CANgateway.

3.5.1 Planification

La planification des tests système est réalisée par CANvengers.

3.5.2 Conception

La conception des tests système est réalisée par CANvengers. La conception des jeux de données de test est réalisée par CANvengers.

3.5.3 Exécution

L'exécution des tests système est réalisée par CANvengers. L'exécution des tests d'acceptation est réalisée par KEREVAL.

3.5.4 Bilan

CANvengers rédige un bilan de test en fin de campagne de test système.

3.6 Documents de test et livrables

Documentation	Livrable à transmettre
Plan de test	X
Dossier de test	X
Rapport de test	X
Matrice de conformité exigences et tests	X
Rapport danomalies	X
Données	
Documents d'analyses partitionnelles et aux limites	
Jeux de données de tests	X
Automatisation des tests	
Code de test	

4 Environnement de test

4.1 Environnement matériel et logiciel de test

Liste des matériels et logiciels nécessaires à l'exécution des tests :

- E_PC : Linux Ubuntu (64-bit) ;
- E_Smartphone : Samsung Galaxy A20e sous système Android 9 ;
- E_Raspberry : pi 3B+ ;
- E_Passerelle : RS485 ;
- E_ICSim : version du 12/06/2020 ;
- E_PEAK : PEAK PCAN.USB IPEH-002021 175459 ;
- E_Banc_De_Test : KEREVAL Banc de Test.

4.2 Outils de test

Les tests seront au maximum automatisés grâce aux outils suivants :

- Tests unitaires : Framework de test Android (basé sur JUnit), bouchonnage Mockito, CMocka ;
- Tests d'intégration : JMeter ;
- Tests de validation : automatisation avec Robot Framework, sinon tests manuels ;
- Dossier de test : Squash TM avec intégration de la gestion d'anomalies via Redmine.

5 Rôles et responsabilités

CANvengers :

- Gestionnaire de tests :
 - Rédaction du plan de tests ;
 - Rédaction du bilan de tests ;
 - Suivi de la réalisation des tests système ;
 - Reporting auprès de KEREVAL.
- Analyste de tests :
 - Conception des tests ;
 - Conception des jeux de données ;
 - Exécution des tests ;
 - Gestion des rapports d'anomalies (création et clôture).

KEREVAL :

- Validation des documents produits.
- Réalisation de tests d'acceptation.

CANvengers ne peut être tenu responsable des répercussions d'une défaillance d'une fonctionnalité non validée.

6 Équipe de test

Le CdP est responsable des moyens mis à disposition pour mener à bien l'activité de test et de l'attribution des tâches.

Le RQT est responsable de l'organisation, du déroulement de l'activité de test et de la rédaction du cahier de test.

Les testeurs sont responsables des résultats de test reportés dans le cahier de test et du développement et implémentation des codes de test.

Les rôles au sein de CANvengers se divisent de la façon suivante :

	CdP	RQT	Testeur
Théo BÉNARD	X		X
Elisa DECLERCK			X
Camille LENNE			X
Gabriel MARQUETTE			X
Thomes ROCHER			X
Paul TRÉMOUREUX		X	X

7 Planning prévisionnel

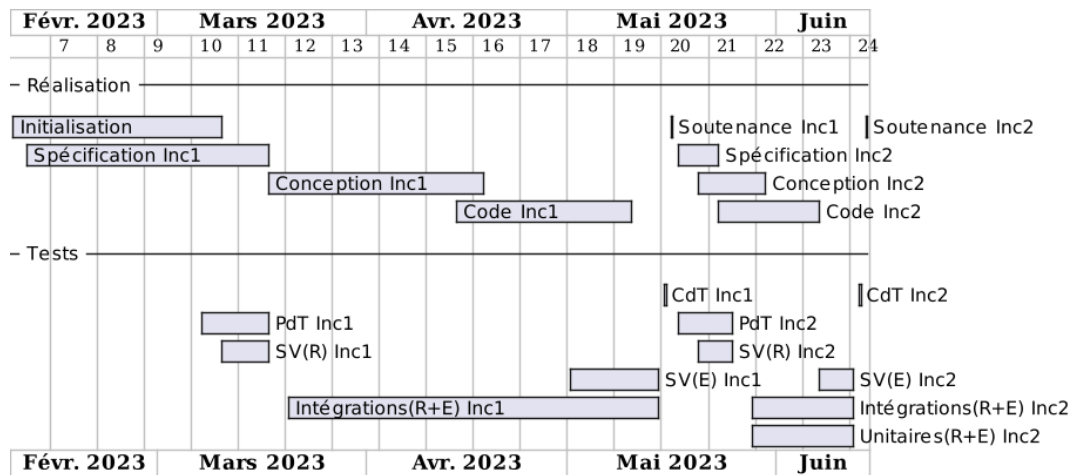


FIGURE 1 – Diagramme de Gantt du projet ProSE

8 Validation du document

Document fait à _____, le _____

Pour la société KEREVAL

Pour la société CANvengers

Mention « Lu et approuvé » :

Mention « Lu et approuvé » :

Signature(s) :

Signature(s) :