

PLAN D'ASSURANCE QUALITE LOGICIELLE

VERSION 4.0 PROSE Équipe A1

Responsable du document : Hugo BOUY, Romain TROVALLET

État du document : Version finale



AVERTISSEMENT : Le présent document est un document à but pédagogique. Il a été réalisé sous la direction de Jérôme DELATOUR, en collaboration avec des enseignants et les étudiants de l'option SE, groupe A1 (Hugo BOUY, Bastien CASSAR, Paul CHIRON, Paul JURET, Laurent LETENNEUR, Mathis MOULIN, Romain TROVALLET) du groupe ESEO. Ce document est la propriété de Jérôme DELATOUR du groupe ESEO. En dehors des activités pédagogiques de l'ESEO, ce document ne peut être diffusé ou recopié sans l'autorisation écrite de ses propriétaires.

Date	Actions	Auteur	Version	Révision
14/02/2023	Création du document	FORMATO	0.00	0
21/02/2023	Création des bases et rédaction du chapitre 2.17	TROVALLET Romain	0.1	0
27/02/2023	Rédaction du chapitre 2.3	TROVALLET Romain	0.2	0
04/03/2023	Rédaction des règles de codage en C et en Java	TROVALLET Romain	0.3	0
05/03/2023	Rédaction de différents chapitres	TROVALLET Romain	0.4	0
07/03/2023	Rédaction des jalons	TROVALLET Romain	0.5	0
09/03/2023	Relecture + rédaction partie 10.2	Hugo BOUY	0.6	0
09/03/2023	Rédaction de la partie 8.2.3	Hugo BOUY	0.7	0
09/03/2023	Relecture finale version 1	Hugo BOUY	1.0	0
17/03/2023	Correction mineure	TROVALLET Romain	1.1	0
27/03/2023	Changement règles de versionnage	Hugo BOUY	1.2	0
27/03/2023	Ajout de liens cliquables pour le RDP	TROVALLET Romain	2.0	0
05/05/2023	Ajout du versionnage du GIT	TROVALLET Romain	3.0	0
15/05/2023	Ajout détails sur le drive	TROVALLET Romain	4.0	0

Figure 1 : Table des évolutions et validations internes du document

Table des matières

1	BUT	4
2	GESTION	7
3	DOCUMENTATION	25
4	STANDARDS, PRATIQUES, CONVENTIONS ET METRIQUES	34
5	REVUES ET AUDITS	36
6	TEST	38
7	NOTIFICATION DES PROBLEMES ET CORRECTIONS	38
8	OUTILS, TECHNIQUES ET METHODOLOGIE	39
9	CONTROLE DES MEDIAS	44
10	CONTROLE DES FOURNISSEUR	45
11	COLLECTE, MAINTENANCE ET CONSERVATION DES ARCHIVES.....	46
12	FORMATION.....	48
13	GESTION DU RISQUE	48
14	OUTILS ET CONFIGURATIONS	51
15	GLOSSAIRE : DEFINITIONS, ACRONYMES ET ABREVIATIONS	53
16	VALIDATION DU DOCUMENT	55

1 BUT

1.1 OBJECTIFS DU DOCUMENT

Ce document est un Plan d'Assurance Qualité Logicielle (PAQL) visant à définir toutes les règles, les méthodes et les outils utilisés dans le projet Prototype Sonnette Connectée (PSC) afin de définir et contrôler la qualité du projet.

Ce document poursuit les objectifs suivants :

- Définir le niveau de qualité attendu par l'équipe projet pour le projet ProSE.
- Définir les outils utilisés, les processus et procédures à suivre par l'équipe projet tant au niveau organisationnel que technique lors du projet ProSE.

Ce document est disponible sur le Référentiel Documentaire Projet (RDP) dans le répertoire [[Qualité/Livrable](#)] sous le nom PAQL_A1.

1.2 PORTEE

Ce document est destiné :

- À l'équipe projet
- Aux consultants de la société FORMATO.

Il pourra être consulté par le client sur sa demande.

1.3 COPYRIGHT

Ce document est la propriété de Jérôme DELATOUR du groupe ESEO. En dehors des activités pédagogiques de l'ESEO, ce document ne peut être diffusé ou recopié sans l'autorisation écrite de son propriétaire.

1.4 VUE D'ENSEMBLE

Ce PAQL est structuré suivant les grandes parties proposées par la norme [IEEE-730_1998].

Il est donc décomposé en **24** parties :

La norme IEEE 730 décrit le contenu d'un plan d'AQL pour un logiciel :

- Intention et portée
- Définitions et abréviations
- Documents de références
- Survol du plan d'assurance qualité logicielle
 - Organisation
 - Niveau de criticité du logiciel
 - Outils, techniques et méthodologies
 - Ressources
 - Normes, pratiques et conventions
 - Calendriers
- Activités et tâches de cycle de vie de l'AQL
 - Rôle de l'assurance de produit
 - Rôle de l'assurance du processus
 - Assurance sur les activités et les tâches du système de management de la qualité
 - Activités et tâches additionnelles
- Processus et politiques additionnelles
 - Processus de revue de contrat
 - Processus de mesures de la qualité
 - Politiques de tests
 - Politique de dérogation et de déviation
 - Politique d'itération des tâches
- Enregistrements et rapports de l'AQL
 - Enregistrements
 - Rapports

1.5 REFERENCES

[PAQL_Exemple_2023] : Exemple, exemple_plan_qualite_ProSE.odt, campus ESEO, révision finale, 2023.

[IEEE-730_1998] : IEEE, std 730-1998 « IEEE Standard for Software Quality Assurance Plans », <http://standards.ieee.org/findstds/standard/730-2002.html>, 1998.

[IEEE-830_1998] : IEEE, std 830-1998 « Recommended Practice for Software Requirements Specifications », <http://standards.ieee.org/findstds/standard/830-1998.html>, 1998.

[UML_2.5.1_2017] : OMG, Unified Modeling Language, version 2.5.1, <http://www.omg.org/spec/UML/>, 2017.

2 GESTION

Cette partie décrit l'organisation, les tâches et les responsabilités en rapport avec les activités d'Assurance Qualité (AQ) du projet PSC.

2.1 ORGANISATION

2.1.1 PROJET CONCERNE

Le projet dénommé Prototype Sonnette Connectée (PSC), consiste à mettre en place les logiciels AOP (application Android), SoftSonnette et SoftPorte (applications C). L'objectif du projet est de concevoir un prototype pour une démonstration des capacités de la STM32MP15. Le prototype est capable de reconnaître le visage de testeurs à l'aide de reconnaissance faciale, de simuler l'ouverture de la porte en fonction de leurs horaires et de permettre au démonstrateur de lire le flux vidéo sur l'application Android.

2.1.2 RESSOURCES HUMAINES

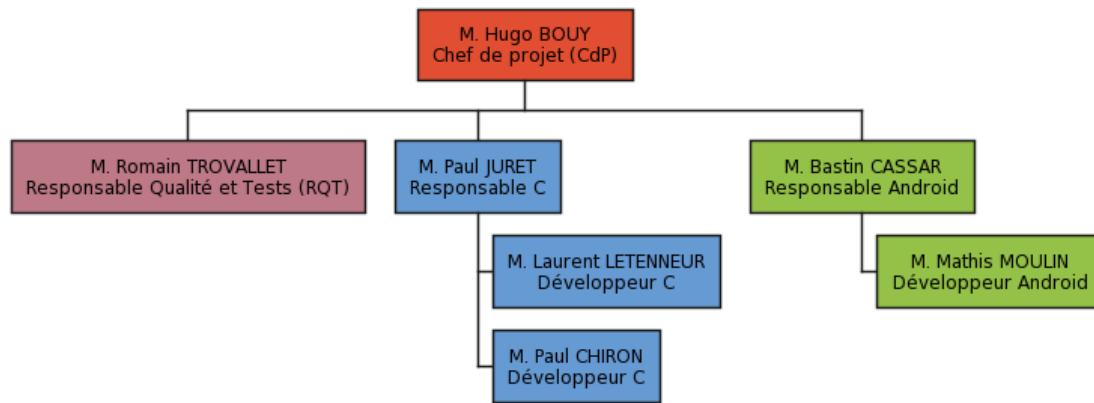
2.1.2.1 ÉQUIPE PROJET

L'équipe projet se compose de :

Rôle	Nom	Prénom	Mail
Chef de projet	BOUY	Hugo	hugo.bouy@reseau.eseo.fr
RQT	TROVALLET	Romain	romain.trovallet@reseau.eseo.fr
Responsable développement C	JURET	Paul	paul.juret@reseau.eseo.fr
Développeur C	CHIRON	Paul	paul.chiron@reseau.eseo.fr
Développeur C	LETENNEUR	Laurent	laurent.letenneur@reseau.eseo.fr
Responsable développement Android	CASSAR	Bastien	bastien.cassar@reseau.eseo.fr
Développeur Android	MOULIN	Mathis	mathis.moulin@reseau.eseo.fr

Figure 2 : Tableau des membres de l'équipe projet

L'organigramme ci-dessous reflète le tableau ci-dessous :



2.1.2.2 CLIENT

Le client unique du projet PSC est la société STMicroelectronics. Hugo BOUY est l'unique intermédiaire entre l'équipe projet et le client. Le chef de projet s'engage à faire un point hebdomadaire d'avancement au client (hors périodes de vacances scolaires) par mail aux représentants du client.

Il est à préciser que toute prise de contact du client par un des membres de l'équipe projet sera sanctionnée par le chef de projet. Le matériel mis à disposition par STMicroelectronics ne doit être utilisé que dans le cadre du projet et doit être rendu dans un parfait état au client en fin de projet.

Les clients sont représentés par M. Erwan SZYMANSKI et Gatien CHEVALLIER, dont les adresses e-mail figurent ci-dessous. M. Erwan SZYMANSKI est le contact privilégié avec le chef de projet.

- Erwan SZYMANSKI – Ingénieur en développement Linux embarqué : erwan.szymanski@st.com
- Gatien CHEVALLIER – Ingénieur en sécurité logicielle : gatien.chevallier@st.com

2.1.2.3 CONSULTANTS ET AUDITEURS

Si nécessaire, l'équipe projet pourra faire appel à la société FORMATO en tant que support technique. Les consultants et leurs compétences privilégiées sont :

- Matthias BRUN (Java Android, automatisation des tests - Android et C) : matthias.brun@eseo.fr
- Camille CONSTANT (activités de test et qualité) : camille.constant@eseo.fr
- Jérôme DELATOUR (gestion de projet, PAQL, spécification, conception) : jerome.delatour@eseo.fr
- Frédéric JOUAULT (spécification, conception, C) : frederic.jouault@eseo.fr

Des activités d'audits externes (cf. [5.3 Audits](#)) seront menées par les auditeurs de la société FORMATO ou missionnées par elle.

2.2 TACHES DU PROJET

2.2.1 TACHES TRANSVERSALES

Les tâches transversales de l'assurance qualité incluent les activités suivantes :

- Documentation (cf. [3 Documentation](#))
- Revues et audits (cf. [5 Revues et Audits](#))
- Inspections internes
- Validation et tests
- Activités d'amélioration du processus d'AQ

2.2.1.1 INSPECTIONS INTERNES

- **Le CdP** - Vérification des consignes de l'ENTP (gestion des heures) : 1 fois / semaine
 - Validation des tâches effectuées sur Redmine
 - Bonne organisation des tâches et des sous-tâches
 - Présence d'un descriptif pertinent pour chaque tâche et avancement
- **Le RQT** – Mise à jour cahier non-respect de la qualité : 1 fois / semaine
 - Des conventions de nommage et de respect de la convention de nommage Git
 - Lisibilité du code (indentation, ...)
 - Analyse sur le respect des règles PAQL
 - Présence de commentaires pertinents
- **Responsable Android et C**
 - Vérification de la bonne forme des commits git
 - Validation des merges de branche sur la branche « master »
 - Vérification entre document de spécifications, conception et code produit

2.2.1.2 VALIDATION ET TEST

Le plan de test a pour objectif d'identifier les informations existantes du projet et les composants qui doivent être testés. Il énumère les exigences d'évaluation à différents niveaux, décrit les stratégies de test qui seront employées, identifie les ressources nécessaires et met en évidence les biens livrables pour les tests.

- Portée du document, termes et abréviations
- Références
- Périmètre de test (composant concernés ou non par les tests, fonctionnalités testées ou non, critères d'acceptation des tests)
- Processus et stratégies de test (activités, techniques, outils, procédures de test et gestions des anomalies)
- Infrastructure de test
- Documents de tests et livrables
- Responsabilités
- Équipe de tests
- Planning prévisionnel

Cf : [PDT TEST A1](#)

2.2.1.3 ÉVOLUTION ET AMELIORATION DU PAQL

Le PAQL est susceptible d'évoluer au cours du projet, en particulier pour les raisons suivantes :

- Toutes les informations nécessaires à la rédaction d'un chapitre ou d'un paragraphe ne sont pas connues ou suffisamment stabilisées lors de la rédaction.
- Il s'agit d'une phase du cycle de développement qui sera engagée ultérieurement (cas de la mention « Rédaction réservée »).
- Des événements techniques ou organisationnels, nécessitant une prise en compte dans le PAQL, peuvent apparaître lors du déroulement du projet (modification d'organisation, mise en place de nouvelles normes ou de procédures ou modification de normes ou procédures existantes, ...).

Le PAQL est rédigé par le Responsable Qualité et Test (RQT) de l'équipe projet. Le Chef de Projet (CdP) et le RQT participent aux décisions de modifications. Il incombe au RQT d'effectuer les modifications jugées nécessaires du PAQL. En cas de modifications du PAQL, celui-ci devra être signé à nouveau par les membres de l'équipe projet.

2.2.2 TACHES PAR RAPPORT AU CYCLE DE DEVELOPPEMENT

L'équipe projet suit un cycle de développement en V en deux incréments. Les activités d'AQ sont décrites par rapport à ce cycle. Le planning et les échéances associées sont disponibles sur l'Espace Numérique de Travail du Projet (ENTP).

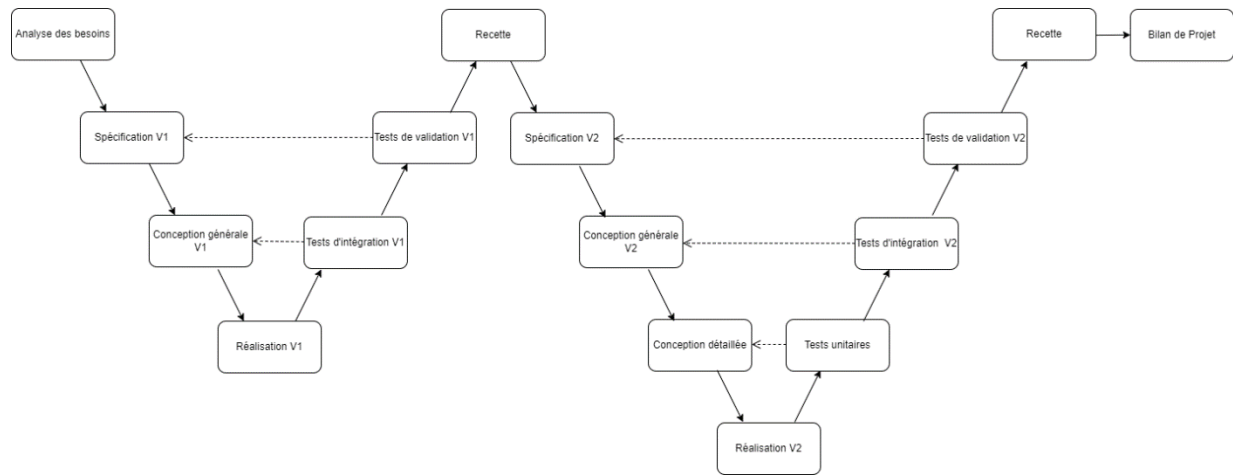


Figure 4 : Diagramme du cycle de développement du projet

2.2.2.1 PHASE D'INITIALISATION DU PROJET

PHASE : INITIALISATION		
Objectifs : Prendre en charge le projet, l'organiser, le planifier et en valider les bases. Évaluer les actions nécessaires pour mettre en place le projet. Échanger avec l'équipe sur les règles à définir.		
Remarque :		
Acteurs et responsabilités : <ul style="list-style-type: none"> CdP et RQT Équipe projet pour la signature du PAQL 	Méthodes & Règles : <ul style="list-style-type: none"> Règles pour l'utilisation de l'ENTP Anticipation et organisation des deadlines personnelles 	Moyens & Outils : <ul style="list-style-type: none"> Initialisation du projet sous ENTP
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organisation de la réunion de lancement Organisation de la phase en aval 	Activités de production/soutien : <ul style="list-style-type: none"> Élaboration du PAQL Mise en place de l'ENTP Définition de la démarche du projet Initialisation du planning et du suivi du projet 	Activités de vérification/contrôle : <ul style="list-style-type: none"> Réunion de lancement Suivi hebdomadaire du CdP des entrées (Redmine)
Produits/données en entrée : <ul style="list-style-type: none"> Wiki Prose et RedMine (ENTP) Documents pédagogiques 	Produits/données en sortie : <ul style="list-style-type: none"> Planning des tâches sur l'ENTP 	Produits révisés : <ul style="list-style-type: none"> PAQL
Jalons de la phase : <ul style="list-style-type: none"> J1 : Jeudi 9 février 2023 : Rendez-vous client et analyse de la demande J2 : Vendredi 3 mars 2023 : Fin de l'installation des logiciels et outils : Sublime Merge, Latex J3 : Jeudi 9 mars 2023 : PAQL rédigé J4 : Vendredi 10 mars 2023 : PAQL signé par toute l'équipe 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Nomination des CdP et RQT 	<ul style="list-style-type: none"> Validation des futurs livrables Validation de l'équipe du planning prévisionnel 	<ul style="list-style-type: none"> ENTP opérationnel Attribution des rôles

2.2.2.2 PHASE DE SPECIFICATION

Spécifications : Le dossier de spécification devra respecter le plan défini par la norme [IEEE-830_1998] et s'appuyer sur la notation UML [UML_2.5.1_2017]. Deux audits (un consultatif et un normatif) porteront sur le dossier de spécification. Le plan de test ainsi que le cahier de test de validation seront établis durant cette étape de spécification. Deux audits (un consultatif et un normatif) porteront sur cette activité. Une revue de mi-avancement aura lieu pour présenter au client le dossier de spécification et les éléments contractuels.

PHASE : SPECIFICATION V1 & V2		
Objectifs : Mener des activités d'exploration techniques afin d'évaluer la complexité et le temps nécessaire à la réalisation du futur produit. Présenter les principales fonctions, les performances requises, les exigences de qualité et les contraintes de réalisation. Faire la description complète de toutes les fonctionnalités des sous-ensembles du projet. Présenter le dossier final de spécification. Présenter le plan de test. Lancer les explorations techniques nécessaires au projet.		
Remarque :		
Acteurs et responsabilités : <ul style="list-style-type: none"> CdP et équipe Client 	Méthodes & Règles : <ul style="list-style-type: none"> PAQL 	Moyens & Outils : <ul style="list-style-type: none"> ENTP
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organiser les échanges d'informations avec le client 	Activités de production/soutien : <ul style="list-style-type: none"> Rédaction du dossier de spécifications Rédaction du plan de test Élaboration des maquettes des écrans Explorations techniques Rédaction de documentations explicatives du code produit 	Activités de vérification/contrôle : <ul style="list-style-type: none"> Consulting sur le document de Spécification Consulting sur l'élaboration des maquettes des écrans AC Spécification Incrément 1 le 17/03/2023 AC Spécification Incrément 2 le 24/05/2023
Produits/données en entrée : <ul style="list-style-type: none"> Cahier des charges Client PAQL 	Produits/données en sortie : <ul style="list-style-type: none"> Dossier de spécifications Plan de test Contrat client 	Produits révisés : <ul style="list-style-type: none"> ENTP Planning prévisionnel PAQL
Jalons de la phase : <ul style="list-style-type: none"> J1 : Vendredi 10 mars 2023 : Spécification version éclatée J2 : Mercredi 14 mars 2023 : Spécification non relue J3 : Jeudi 16 mars 2023 : Plan de Test J4 : Jeudi 16 mars 2023 : Spécification relue J5 : Vendredi 17 mars 2023 : AC Spécification J6 : Vendredi 17 mars 2023 : AC Plan de test J7 : Samedi 18 mars 2023 : Scénario de Validation J8 : Mardi 21 mars 2023 : AC Scénario de Validation J9 : Mercredi 22 mars 2023 : Contrat et devis signés J10 : Lundi 27 mars 2023 : Livraison spécification incrément 1 		

PHASE : SPECIFICATION V1 & V2		
<ul style="list-style-type: none"> J11 : Mercredi 24 mai 2023 : AN Spécification J12 : Vendredi 26 mai 2023 : AN Plan de Test & Scénario de Validation 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Initialisation effectuée 	<ul style="list-style-type: none"> Validation des produits en sortie Signature du client 	<ul style="list-style-type: none"> Dossier de spécifications signé par le client Plan de test validé

2.2.2.3 PHASE DE CONCEPTION

Conception générale (ou système) : Un audit consultatif portera sur la conception générale.

Conception détaillée : Un audit consultatif portera sur la conception détaillée.

Un audit normatif portera sur la conception générale et détaillée.

PHASE : CONCEPTION V1 & V2		
Objectifs : Organiser et optimiser les temps de conception Définir l'architecture logicielle des applications Finaliser les ultimes explorations techniques et les intégrer au dossier Décrire explicitement le dossier de conception		
Remarque : La conception générale peut commencer en parallèle des spécifications.		
Acteurs et responsabilités : <ul style="list-style-type: none"> CdP et équipe Client 	Méthodes & Règles : <ul style="list-style-type: none"> PAQL Dossier de spécification 	Moyens & Outils : <ul style="list-style-type: none"> ENTP
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organiser les échanges d'informations avec le client 	Activités de production/soutien : <ul style="list-style-type: none"> Définition de l'architecture technique Conception logicielle Normalisation des fonctions Rédaction du dossier de conception Rédaction des tests de validation 	Activités de vérification/contrôle : <ul style="list-style-type: none"> Consulting sur la conception AC Conception Incrément 1 le 19/04/2023 AN Conception Incrément 2 le 31/05/2023
Produits/données en entrée : <ul style="list-style-type: none"> Dossiers de spécifications PAQL 	Produits/données en sortie : <ul style="list-style-type: none"> Dossier de conception 	Produits révisés : <ul style="list-style-type: none"> Normes de développement Dossier de spécification PAQL
Jalons de la phase : <ul style="list-style-type: none"> J1 : Vendredi 7 avril 2023 : Dossier de Conception V1 (non relu) 		

PHASE : CONCEPTION V1 & V2		
<ul style="list-style-type: none"> J2 : Mercredi 19 avril 2023 : AC Conception Générale J3 : Vendredi 28 avril 2023 : Dossier de Conception V2 (non relu) J4 : Lundi 8 mai 2023 : Dossier de Conception V2 (relu) J6 : Mercredi 31 mai 2023 : AN Conception Générale & Détaillée 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Document de spécification signé par le client 	<ul style="list-style-type: none"> Validation des produits en sortie 	<ul style="list-style-type: none"> Dossier de conception validé Normes de développement Tests de validation rédigés

2.2.2.4 PHASE DE REALISATION

Codage : Quatre audits porteront sur le code source produit afin notamment de s'assurer du bon respect du PAQL et des normes de programmation. Deux porteront sur le code écrit en langage C sur cible embarquée (consultatif et normatif) et deux autres sur le code fonctionnant sur la plate-forme Android.

PHASE : REALISATION V1 & V2		
Objectifs : Développer des applications logicielles Tester les applications logicielles définies lors de la phase de conception Codifier les tests d'intégration et unitaires et adopter les outils de test		
Remarque : L'étape de réalisation peut commencer en parallèle de la conception.		
Acteurs et responsabilités : <ul style="list-style-type: none"> CdP et équipe 	Méthodes & Règles : <ul style="list-style-type: none"> PAQL Spécification Conception 	Moyens & Outils : <ul style="list-style-type: none"> ENTP Moyen de test (cf. PDT_TEST_A1) Android Studio CubeIDE et autres outils de l'environnement ST Visual Studio Code
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organiser les fonctions prioritaires Organiser les échanges d'informations avec le client 	Activités de production/soutien : <ul style="list-style-type: none"> Écriture du code Android Écriture du code C Conception des tests unitaires Conception des tests d'intégration 	Activités de vérification/contrôle : <ul style="list-style-type: none"> Interface Android testée et validée Code C testé et validé Écriture des tests de validations système AC Code C et Android incrément 1 le 03/05/2023 AN Code C et Android incrément 2 le 08/06/2023
Produits/données en entrée : <ul style="list-style-type: none"> Dossiers de conception PAQL 	Produits/données en sortie : <ul style="list-style-type: none"> Code Moyen de test opérationnel Tests unitaires 	Produits révisés : <ul style="list-style-type: none"> Normes de développement Document de conception Document de spécification PAQL

PHASE : REALISATION V1 & V2		
Jalons de la phase : <ul style="list-style-type: none"> J1 : Mercredi 19 avril 2023 : Fin écriture code incrément 1 avant les tests J2 : Mercredi 26 avril 2023 : Livraison 1 – Reconnaissance faciale et flux vidéo J3 : Mercredi 3 mai 2023 : AC Code J4 : Mercredi 10 mai 2023 : Code incrément 1 corrigé J5 : Vendredi 26 mai 2023 : Fin écriture code incrément 2 avant les tests J6 : Mercredi 31 mai 2023 : Livraison 2 - Gérer les testeurs et la simulation de la porte J7 : Jeudi 8 juin 2023 : AN Code J8 : Dimanche 11 juin 2023 : Code incrément 2 corrigé 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Document de conception validé 	<ul style="list-style-type: none"> Validation des produits en sortie 	<ul style="list-style-type: none"> Application logicielle réalisée Tests réalisables

2.2.2.5 PHASE DE TEST

Tests unitaires, tests d'intégration : Un audit consultatif portera sur chacune de ces activités.

Tests de validation : Un audit normatif portera sur l'application des tests (validation, intégration et unitaire).

PHASE : TEST V1 & V2		
Objectifs : Développer des tests fiables et efficaces. Contrôler la fiabilité du logiciel. Identifier les erreurs logiques. Vérifier interactions des interfaces. Valider l'adéquation aux spécifications du logiciel.		
Remarque : La phase de programmation des tests peut commencer en parallèle de la réalisation		
Acteurs et responsabilités :	Méthodes & Règles :	Moyens & Outils :
<ul style="list-style-type: none"> CDP & RQT Équipe projet 	<ul style="list-style-type: none"> PAQL Conception Spécification Plan de test Cahier de tests 	<ul style="list-style-type: none"> Moyen de test (cf. PDT_TEST_A1) ENTP SquashTM
Activités d'organisation/pilotage :	Activités de production/soutien :	Activités de vérification/contrôle :
<ul style="list-style-type: none"> Vérification croisée Organiser les tests principaux 	<ul style="list-style-type: none"> Exécution tests de communication Compléter cahier de test Exécution tests unitaires Exécution tests validation 	<ul style="list-style-type: none"> Vérifications de la présence d'une couverture maximale. Prise en considération des cas spécifiques. AC Cahier de test incrément 1 le 15/05/2023 AN Cahier de Test incrément 2 le 12/06/2023

PHASE : TEST V1 & V2		
Produits/données en entrée : <ul style="list-style-type: none"> • Application logicielle • Dossier de conception • PAQL • Moyen de test • Cahier de test 	Produits/données en sortie : <ul style="list-style-type: none"> • Document d'analyse des tests • Application logicielle testée • Cahier de test 	Produits révisés : <ul style="list-style-type: none"> • Normes de développement • Document de conception • Document de spécifications • Code • PAQL
Jalons de la phase : <ul style="list-style-type: none"> • J1 : Samedi 29 avril 2023 : TU incrément 1 exécutés • J2 : Mardi 2 mai 2023 : TI incrément 1 exécutés • J3 : Lundi 15 mai 2023 : AC Cahier de test • J4 : Samedi 3 juin 2023 : Fin des tests avec le code de production - Début de la vérification de tous les TV, TI et TU et correction du code • J5 : Dimanche 11 juin 2023 : Tests incrément 2 exécutés • J6 : Lundi 12 juin 2023 : AN Cahier de test 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> • Plan de test partiellement validé 	<ul style="list-style-type: none"> • Validation des produits en sortie 	<ul style="list-style-type: none"> • Tests réalisés • Cahier de test complet

2.2.2.6 PHASE DE RECETTE

Une revue de recette aura lieu pour remettre au client le produit demandé, ainsi que fournir les livrables suivants : les dossiers de spécification et conception, l'ensemble des tests réalisés (cahier et plan de test), les codes sources, les manuels d'utilisation et d'installation, le procès-verbal de recette définitive ainsi que l'application logicielle fonctionnelle.

PHASE : RECETTE		
Objectifs : Réceptionner les applications de manière définitive		
Remarque : Se fera en même temps que la phase de bilan de fin de projet		
Acteurs et responsabilités : <ul style="list-style-type: none"> • CdP et équipe • Client 	Méthodes & Règles : <ul style="list-style-type: none"> • PAQL • Spécification • Conception • Intégration 	Moyens & Outils : <ul style="list-style-type: none"> • ENTP

PHASE : RECETTE		
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organiser les échanges d'informations avec le client 	Activités de production/soutien : <ul style="list-style-type: none"> Mettre en forme les documents suivants : <ul style="list-style-type: none"> Dossiers de spécification Dossier de conception Cahier et plan de test Codes source Android Code source C Procès-verbal de recette définitive Manuel d'utilisation 	Activités de vérification/contrôle : <ul style="list-style-type: none"> AC Gestion de Projet incrément 1 le 15/05/2023 AN Gestion de Projet incrément 2 le 12/06/2023
Produits/données en entrée : <ul style="list-style-type: none"> PAQL Dossier de spécification Dossier de conception Cahier de test Code 	Produits/données en sortie : <ul style="list-style-type: none"> Application opérationnelle livrée Manuel d'utilisation et manuel d'installation validés Dernière version à jour des documents, codes source des applications (documenté sous Doxygen) et exécutables Procès-verbal de recette définitive 	Produits révisés :
Jalons de la phase : <ul style="list-style-type: none"> J1 : Lundi 15 mai 2023 : AC Gestion de Projet J2 : Mardi 16 mai 2023 : Soutenance de mi-avancement J3 : Mardi 30 mai 2023 : Écrire le manuel d'utilisation (non relu) et écrire le procès-verbal de recette définitive J4 : Mardi 6 juin 2023 : Manuel d'utilisation (relu) J5 : Mercredi 7 juin 2023 : Support et entraînement de l'audit terminés J6 : Jeudi 8 juin 2023 : Vérification des dossiers à remettre au client J7 : Lundi 12 juin 2023 : AN Gestion de Projet J8 : Mercredi 14 juin 2023 : Facture finale au client 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Phase d'intégration terminée 	<ul style="list-style-type: none"> Validation des produits en sortie 	<ul style="list-style-type: none"> Fin du projet

2.2.2.7 PHASE DE BILAN DE FIN DE PROJET

Le bilan a lieu à la fin du projet et est présenté lors de l'Incrément 2.

PHASE : BILAN FIN DE PROJET		
Objectifs : Clôturer le projet, archiver les productions, effectuez les statistiques du projet (temps passé), tirer les enseignements du projet.		
Remarque : Se fera en même temps que la phase de recette.		
Acteurs et responsabilités : <ul style="list-style-type: none"> CdP 	Méthodes & Règles : <ul style="list-style-type: none"> PAQL 	Moyens & Outils : <ul style="list-style-type: none"> ENTP (Redmine)
Activités d'organisation/pilotage : <ul style="list-style-type: none"> Organiser les échanges d'informations avec le référent ESEO et M. DELATOUR 	Activités de production/soutien : <ul style="list-style-type: none"> Rédaction des documents de bilan Diaporama de présentation de fin de projet 	Activités de vérification/contrôle : <ul style="list-style-type: none"> Entraînement en groupe entier dans le but de préparer la partie orale Consulting
Produits/données en entrée : <ul style="list-style-type: none"> ENTP (RedMine) 	Produits/données en sortie : <ul style="list-style-type: none"> Statistiques projet Bilan de projet Évaluation équipe 	Produits révisés : <ul style="list-style-type: none"> ENTP
Jalons de la phase : <ul style="list-style-type: none"> J1 : Lundi 5 juin 2023 : Diaporama d'audit J2 : Mercredi 7 juin 2023 : Entraînement de passage à l'oral du groupe entier J3 : Samedi 10 juin 2023 : Statistiques su projet et compte rendu de réunion de bilan de projet J4 : Mercredi 14 juin 2023 : Présentation du projet final au client et remise de l'ensemble du matériel 		
Conditions de début de phase :	Conditions de fin de phase :	Conditions de passage à la phase suivante :
<ul style="list-style-type: none"> Phase d'intégration terminée 	<ul style="list-style-type: none"> Validation des produits en sortie 	<ul style="list-style-type: none"> Fin du projet

2.3 RESPONSABILITE

2.3.1 DEFINITION GENERALE DES ROLES

Chaque membre de l'équipe projet est tenu de respecter et d'appliquer les normes du PAQL dans son travail. Chaque membre pourra avoir l'un des rôles suivants :

- **Chef de projet (CdP)** : Responsable global du projet et assure la communication entre le client et l'équipe projet. Il veille au bon déroulement du développement et prend les décisions techniques en tenant compte de différents critères (coûts, délais, complexité technique...)
- **Responsable Qualité et test (RQT)** : Son rôle est de vérifier que les normes et règles établies sont respectées, afin d'assurer la bonne qualité du code source développé et sa conformité à certaines règles. Il est aussi le responsable du cahier de test.
- **Responsable développement** : Il y a un responsable C et un responsable Android. Leurs rôles sont de proposer au CdP les décisions techniques qui guideront le développement du projet et d'avoir une vision globale sur le code afin notamment de valider les merges de branche.
- **Développeur** : Il a pour responsabilité de réaliser les activités/tâches techniques définies par le CdP pour faire progresser le projet tout en évaluant la qualité du travail accompli.

2.3.2 RECAPITULATIF DES RESPONSABILITES CLIENT SUR LES PHASES

PHASE	IMPLICATION CLIENT	IMPLICATION ÉQUIPE PROJET
Initialisation	<ul style="list-style-type: none">• Fournir le cahier des charges• Arrivée à l'ESEO pour la première réunion client• Fournir le matériel	<ul style="list-style-type: none">• Rédiger le contrat et le planning• Former les équipes et définir les responsabilités• Rédiger le PAQL
Spécification	<ul style="list-style-type: none">• Répondre aux questions concernant l'étude des besoins• Valider l'IHM• Fournir l'existant système si nécessaire• Valider les dossiers	<ul style="list-style-type: none">• Réaliser l'étude des besoins• Rédiger le dossier de spécifications• Rédiger le cahier de recette• Rédiger le plan de test
Conception	<ul style="list-style-type: none">• Apporter des réponses sur les divers aspects techniques à élaborer• Apporter des réponses sur les divers tests à réaliser• Valider la conformité des dossiers	<ul style="list-style-type: none">• Rédiger le dossier de conception• Rédiger le cahier de test

PHASE	IMPLICATION CLIENT	IMPLICATION ÉQUIPE PROJET
Réalisation	<ul style="list-style-type: none"> • S'assurer de l'avancement du projet aux différents jalons • S'assurer de la conformité des modules logiciels vis-à-vis du cahier des charges 	<ul style="list-style-type: none"> • Développer les modules logiciels • Tests unitaires de l'application
Test	<ul style="list-style-type: none"> • Discute puis approuve le plan de test • Vérifie la matrice de conformité entre fonctionnalités demandées et tests • Vérifie les résultats des tests de validation système 	<ul style="list-style-type: none"> • Rédige le plan de test • Rédige et implémente les tests de validation système • Implémente les tests d'intégration • Implémente les tests unitaires • Exécute tous les tests • Produit le cahier de test • Produit la matrice de conformité
Recette	<ul style="list-style-type: none"> • Valider le taux de fiabilité des tests • Valider le fonctionnement global de l'application • Réceptionner les applications définitives du projet • Valider l'ensemble des livrables reçus • Signature du procès-verbal de recette définitive • Paiement 	<ul style="list-style-type: none"> • Rédaction du procès-verbal de recette • Livrer l'ensemble des livrables
Bilan de fin de projet	<ul style="list-style-type: none"> • Recueillir le rapport de fin de projet ainsi que le rapport d'avancement du projet • Discuter des potentiels points à améliorer et recueillir une liste non exhaustive d'améliorations 	<ul style="list-style-type: none"> • Proposer des améliorations • Rédiger un rapport de fin de projet

Figure 5 : Tableau des responsabilités client

2.3.3 RECAPITULATIF DES RESPONSABILITES CDP SUR LES PHASES

PHASE	IMPLICATION CDP
INITIALISATION	<ul style="list-style-type: none"> Organiser l'environnement de gestion de projet Redmine. Définir, en accord avec le RQT, et écrire des règles de bonnes pratiques que devront respecter les membres de l'équipe afin que tout se coordonne correctement. Gestion du contrat client / devis. Gestion du planning prévisionnel en relation avec le RQT
SPECIFICATION	<ul style="list-style-type: none"> Encadrement des équipes attachées à la rédaction des spécifications. Participation à la rédaction du dossier de spécifications Validation régulière du travail. Relecture des livrables
CONCEPTION	<ul style="list-style-type: none"> Encadrement des équipes attachées à la rédaction des livrables de conception. Validation régulière du travail. Relecture des livrables
REALISATION	<ul style="list-style-type: none"> Encadrement et vérification de la bonne avancée du développement logiciel. Aide occasionnelle sur le code (exceptionnellement)
TEST	<ul style="list-style-type: none"> Vue globale du travail du RQT et comptes-rendus réguliers. Relecture des livrables "cahier de tests" et "PAQL".
RECETTE	<ul style="list-style-type: none"> Gestion des comptes de l'équipe Projet Prose A1 (factures, paiement des salaires...).
BILAN DE FIN DE PROJET	<ul style="list-style-type: none"> Faire une analyse et les statistiques du projet Organisation des audits Établissement du bilan de fin de projet (ce qu'on a fait par rapport à ce sur quoi on s'était engagé)

Figure 6 : Tableau des responsabilités Cdp

2.3.4 RECAPITULATIF DES RESPONSABILITES RQ SUR LES PHASES

PHASE	IMPLICATION RQT
INITIALISATION	<ul style="list-style-type: none"> Rédiger le PAQL. Présenter le PAQL aux membres de l'équipe de projet. Organiser le versionnage sur le dépôt en appliquant des règles précises.
SPECIFICATION	<ul style="list-style-type: none"> Mise à jour du PAQL et signature par les membres de l'équipe projet Mise en œuvre du plan qualité en concertation avec l'équipe Rédaction du plan de test
CONCEPTION	<ul style="list-style-type: none"> Développement des différents documents de conception
REALISATION	<ul style="list-style-type: none"> Conformité du code source vis-à-vis des règles et normes préétablies (code homogène, lisible et maintenable) Traçabilité du code effectué (git, réunions...) Développement d'une portion du code source
TEST	<ul style="list-style-type: none"> Suivi de l'architecture de tests en conformité avec le plan de test Suivi et validation du cahier de tests, du plan de test
RECETTE	<ul style="list-style-type: none"> Dernière vérification des documents tests et codes. (Conformité et respect des règles établies dans le PAQL)
BILAN DE FIN DE PROJET	<ul style="list-style-type: none"> Aide à la conception des documents de synthèse Réunion de fin de projet pour faire le bilan du travail établi par chacun et des améliorations futures à apporter Prise de parole sur la pertinence et la qualité du suivi établi par les RQT.

Figure 7 : Tableau des responsabilités du RQT

2.3.5 RECAPITULATIF DES RESPONSABILITES DES DEVELOPPEURS SUR LES PHASES

Les responsables de développement ont les mêmes responsabilités que les développeurs. Leurs rôles spécifiques sont indiqués en *vert italique*.

PHASE	IMPLICATION ÉQUIPE DEVELOPPEMENT
INITIALISATION	<ul style="list-style-type: none">• Prise en main des différents outils de développement et de rédaction de documents.• Première prise en main des différents outils techniques• Lecture du PAQL
SPECIFICATION	<ul style="list-style-type: none">• Rédaction du document de spécification• Explorations techniques ciblées sur les différents points à approfondir
CONCEPTION	<ul style="list-style-type: none">• Rédaction du document de conception• Continuité des explorations techniques• <i>Proposition des solutions techniques explorées au CdP</i>
REALISATION	<ul style="list-style-type: none">• Développement de l'architecture du code• Premières ébauches de l'architecture du code de tests• <i>Vision globale sur le code et gestion des merges de branches vers la branche master</i>
TEST	<ul style="list-style-type: none">• Développements des tests en accord avec le cahier de tests et le plan de test
RECETTE	<ul style="list-style-type: none">• Vérification des livrables en accord avec le RQT
BILAN DE FIN DE PROJET	<ul style="list-style-type: none">• Rédaction des documents de synthèse

Figure 8 : Tableau des responsabilités des développeurs

3 DOCUMENTATION

3.1 BUT

Ce chapitre décrit les règles de gestion de la documentation du projet. En effet, un certain nombre d'artefacts du projet concerne des documents.

3.2 TYPE DE DOCUMENTS

Les documents suivants sont distingués suivant leur nature, qu'ils soient livrés par le client ou non, consultables par les auditeurs ou réservés à l'équipe projet, et ce suivant la phase où ils sont produits.

Les artefacts de documentation, nommés documents "livrables client" sont :

- Dossier de spécification
- Plan d'Assurance Qualité Logicielle (PAQL)
- Contrat et devis
- Dossier de conception
- Plan de test
- Artefacts de code
- Cahier de test
- Manuel d'utilisation et manuel d'installation
- Présentation revue
- Compte-rendu(s) réunion client
- Point(s) d'avancement météo

Les artefacts consultables par les consultants, nommés documents "consultables auditeur", sont :

- Tous les documents "livrables client"
- Ordre(s) du jour réunion projet
- Compte-rendu réunion projet
- Correspondance échangée avec le client
- Les codes source des explorations techniques

Les artefacts de code sont :

- Les codes sources
- Les codes de tests

Les autres documents du projet sont considérés comme internes au projet.

3.3 REFERENCE DES DOCUMENTS

La référence d'un document est de la forme suivante : « SIGLE_A1 », où A1 désigne l'identifiant de l'équipe (une lettre et un numéro) et le SIGLE correspond à l'une des combinaisons de lettres citées ci-dessous.

Ce système de référencement ne sera appliqué que pour les livrables "consultables auditeur". Les ébauches et documents internes à l'équipe échappent donc à cette règle de nommage.

REFERENCE	LIBELLE DOCUMENT
PAQL_A1	Plan d'Assurance Qualité Logicielle
SPEC_A1	Document de spécification
CONC_A1	Document de conception
TEST_A1	Document de test
BILAN_A1	Bilan de projet
SV_A1	Scénario de validation
MANUEL_A1	Manuel du projet remis au client
PLANNING_A1	Document comportant les contributions de chaque membre
PLANNING_TEST_A1	Planning de test avec les jalons importants
LIVRAISON_A1	Document final regroupant les différents éléments de conception, vérification, validation et l'ensemble de la solution logicielle
CR_R[N°]_[JOUR_MOIS_ANNEE]	Compte-rendu de réunion formelle
SLIDE_[THEME]_[JOUR_MOIS_ANNEE]	Supports de présentation
SANS_A1	Pour tous les autres documents

Figure 9 : Tableau des références des documents

Par exemple, pour le dossier de spécification, produit par l'équipe A1, sa référence sera la suivante : « SPEC_A1 ».

La référence d'un document permet d'identifier la catégorie à laquelle appartient ce document, cette référence figurera dans le document, mais aussi dans le nom de son fichier.

Ainsi, dans l'exemple précédent, le fichier pourra se nommer « dossier_de_specification_SPEC_A1 ».

3.4 ÉTAT D'UN DOCUMENT

Un document possède un état, parmi quatre possible, indiqué sur la page de garde de ce dernier:

- **En rédaction** : Document en cours de rédaction
- **En relecture** : Document en relecture globale
- **En attente de validation** : Document en attente de validation pour le responsable du document
- **Version finale** : Document validé et livrable

En plus des quatre états disponibles, un document dispose d'un numéro de version sous la forme X.Y, rev. Z. L'état d'un document est mis à jour suivant la pseudo-machine à état suivante :

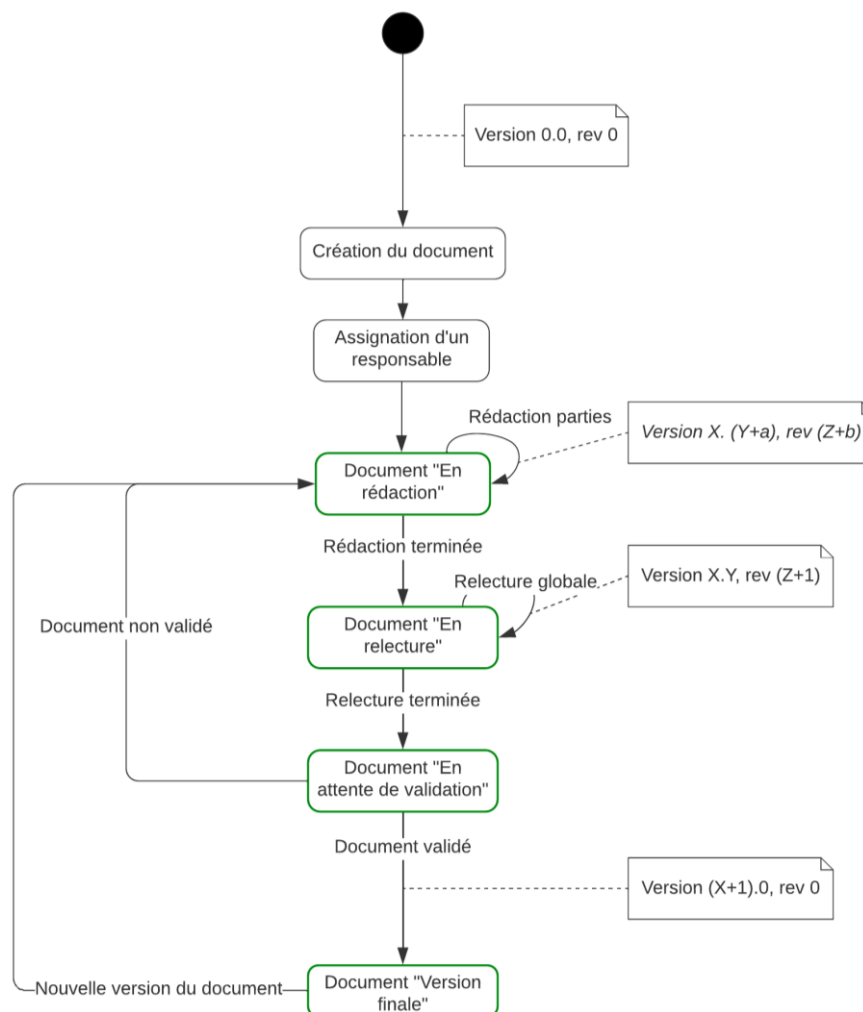


Figure 10 : Processus de versionnage d'un document

Toute création d'un document doit être suivie de la nomination d'un responsable du document. C'est lui qui aura la charge de faire évoluer l'état du document et donc de valider sa version finale. La table des versions en début de document retrace les différentes versions et révisions du document.

Lors de la rédaction du document, plusieurs parties sont amenés à être rédigées en parallèle. Ainsi pour suivre plus précisément l'état d'avancement du document, chaque partie et sous-partie dispose de sa tâche correspondante sur le Redmine apportant une visibilité accrue sur l'état d'avancement de la partie :

- **En cours** : La rédaction de la partie est débutée.
- **En relecture** : La rédaction est terminée, un relecteur a été assigné
- **En demande d'approbation** : La relecture est terminée, en attente de validation du responsable du document
- **Approuvée** : La rédaction de la partie a été validée par le responsable du document

L'état d'avancement d'une partie suit la pseudo machine à état suivant :

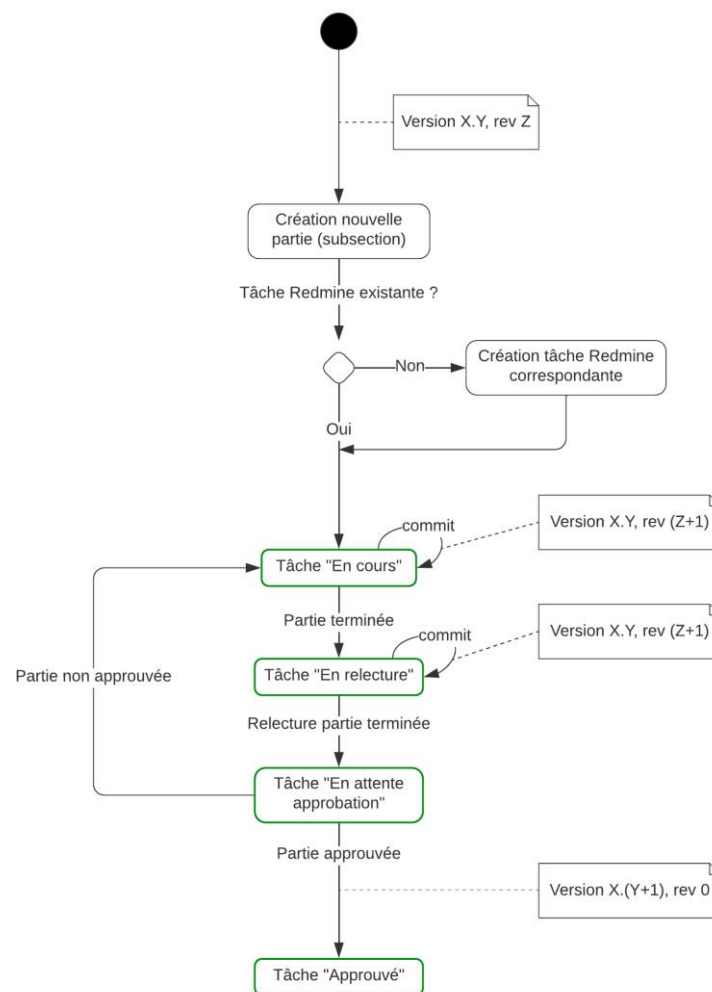


Figure 11 : Différents états d'une partie

Lorsque la partie est en cours de rédaction et en relecture, chaque commit incrémente la révision Z. Lorsque la partie a été approuvée par le responsable du document, la version Y est incrémentée. Le responsable du document peut décider d'approuver une partie sans relecture. Celle-ci sera alors relue lors de la relecture globale.

Une fois toutes les parties rédigées, le document passe en relecture. Il s'agit d'une relecture globale de l'ensemble du document, en plus des relectures de parties. Une fois la relecture globale terminée, il incombe au responsable du document de valider sa rédaction. A noter que chaque commit lors de la relecture d'un document incrémente la révision Z dans la mesure où cette relecture n'induit que des modifications mineures.

Un document en « version finale » ne signifie pas qu'aucune modification ne sera plus jamais apportée sur ce document. Cela signifie que le document est dans une version permettant sa diffusion externe et sa livraison au client. Chaque « version finale » d'un document incrémente le X.

A NOTER : Le processus présenté ci-dessus décrit la construction d'un document et de parties très complètes nécessitant un versionnage accrue. Pour certains documents et parties plus simples ou nécessitant peu de modifications, l'incrément des versions Y peut se faire sans relecture à la demande du responsable du document.

3.5 FORMAT DES DOCUMENTS

Pour tous les documents "livrables client" et "consultables auditeur", il faut respecter un modèle de documents. Ces modèles sont disponibles sur le RDP, dans le dossier [\[Qualite\]](#).

3.5.1 MODELE DE DOCUMENT

Des modèles de documents sont proposés suivant le sigle de ces documents. Ces modèles doivent être impérativement utilisés. Ils sont disponibles dans l'ENTP, dans le dossier [Qualite/Modeles].

3.5.2 ARTEFACT DE CODE

Dans le RDP, dossier [Qualite/Modeles], les modèles d'artefact de codes suivants sont disponibles :

- example.c
- example.h
- example.java

Il est impératif que tous les codes source produits dans le projet utilisent ces modèles et respectent les conventions de nommage qu'ils proposent (détaillées ci-après). Il est impératif que les codes source soient parfaitement commentés et ce en utilisant les balises au format Doxygen. Ce point est crucial, car il permet la génération automatique de la documentation du code source de tout le projet.

Ces documents reprennent les règles de codage suivants :

3.5.3 REGLES DE CODAGE EN LANGAGE C :

Éditeurs imposés :

- CubeIDE pour les briques logicielles à destination du Cortex M
- Visual Studio Code pour les briques logicielles à destination du Cortex A (linux)

Programmation sur Linux imposé :

- Version Ubuntu 22.04
- Kernel Linux version : 5.15

Convention sur le fichier .c :

- Le nom du fichier.c est uniquement en minuscule et ne comporte pas de caractères spéciaux ou d'accents, seul l'underscore est autorisé pour améliorer la lisibilité du nom de fichier
- Appelle uniquement des fichiers .h et des librairies standards
- N'appelle pas de fichier.c
- Appelle nécessairement son .h attribué

Convention sur le fichier .h :

- Le nommage du fichier.h suit les mêmes règles que le fichier.c.
- Si le fichier.h est lié à un fichier.c, les deux fichiers doivent porter le même nom.
- Appelle uniquement des fichiers.h et des librairies standards
- Déclare les fonctions publiques mais ne les définit pas
- Les fonctions déclarées doivent être expliquées selon le format Doxygen : quelles sont les entrées, les sorties et le sens du résultat.
- Peut contenir la définition de types (structures, énumérations, types) et des macros utilisables par les autres fichiers du projet

Organisation du code :

- Les .h déclarent les fonctions publiques et peuvent contenir des macros et des type utilisables dans tout le projet
- Les .c définissent ces fonctions
- Utilisation des tabulations
- Enlever les problèmes d'inclusion multiples dans les .h
- L'inclusion d'un fichier se fera à l'aide de guillemets pour les fichiers internes au projet et avec des chevrons pour les fichiers externes :

```
#ifndef IDENTIFICATEUR
#define IDENTIFICATEUR
#include "lampe.h"
#include <stdio.h>
#endif /* guard */
```

Écriture des variables :

- Les variables ne commencent pas avec un underscore “_”
- Variables ASCII autorisés - pas de caractère étranger ni accent.
- Codage en UTF-8
- La taille d'un tableau doit être définie dans une macro (pas d'expression constante)
#define TAILLE_TABLEAU 12
- Les constantes “vraies” telles que les macros et les énumérations sont en majuscules
- Les pointeurs commencent par un p : int * pEntier;
- La convention de nommage camelCase est gardée pour tout le projet :

Autorisé	Interdit
➤ uneVariableExemple	➤ une_variable_exemple ➤ une_Variable_Exemple ➤ _une_variable_exemple ➤ Une_variable_exemple ➤ une-variable-exemple

Écriture des fonctions :

- Toutes les fonctions déclarées doivent posséder un prototype
- Le nom est composé du nom du fichier suivi d'un underscore et du nom de la fonction en suivant le camelCase : void app_uneFonctionExemple(void)
- La déclaration d'une fonction avec le mot-clé extern est interdit
- Le retour d'une fonction doit être mentionnée
- Pas plus d'un return par fonction
- Interdit d'introduire une instruction return sans expression
- Les accolades qui suivent la définition d'une fonction seront placées une ligne en-dessous :

```
void lampe_allumeLampe(int etat)
{
    if (etat>0)
    {
        //code
    }
}
```

Écriture des structures / énumérations / type :

- Les structures, les énumérations et les types suivent le PascalCase : struct UnStructExemple{...}
- Les énumérations finissent par _e : enum uneEnumExemple_e{...}
- Les types finissent par _t : typedef struct UnStructExemple UnStructExemple_t;
- Les structures et les énumérations doivent être expliquée : @brief

Commentaires :

- Pour un commentaire sur une ligne la convention “//” est autorisée
- Pour un commentaire sur plusieurs lignes, la convention “/* */” est obligatoire

Pour isoler une portion de code :

- Ne pas utiliser de commentaires, mais une forme :

```
#if 0
    //Compteur
    int cpt;
#endif
```

Convention sur la compilation :

- Une compilation est correcte seulement si elle ne génère ni erreur ni warning.

3.5.4 REGLES DE TRADUCTION DE LA CONCEPTION VERS DU CODE EN LANGAGE C

Les règles de traduction de la conception vers du code C devront impérativement suivre celles vues en cours avec les auditeurs ProSE.

Cf : [Cours_Conception_2023]

3.5.5 REGLES DE CODAGE EN LANGAGE JAVA

Éditeur imposé : Android Studio

Package

- Écrit en minuscule : com.prose.aop

Fonction

- Commence par un verbe : public void transformValue(...) {...}
- Accessibilité à une valeur => get : public int getValue(...) {...}
- Modification d'une valeur => set : public void setValue(...) {...}
- Créateur d'objet => create : public Object createObject(...) {...}
- Les accolades qui suivent la définition d'une fonction seront placées une ligne en-dessous :

```
void lampe_allumeLampe(int etat)
{
if (etat>0)
{
//code
}
}
```

Classe

- Suivent la convention d'écriture PascalCase : class UneClasseExemple {...}

Variable

- Suffisamment compréhensible à la lecture
- Noms de variable autorisés pour les entiers : i,j,k,l,m et n
- Noms de variable autorisés pour les caractères : c,d et e

Exemple : age_du capitaine = 45; est AUTORISÉ

variable = 3; est INTERDIT

Constante

- Toujours initialisée sur la même ligne
- Ecrite en majuscule et underscore si plusieurs mots

Exemple : `static final CONSTANCE = 3;`

Commentaires :

- Le commentaire de type "documentation" décrivant une méthode : `/** Documentation de la méthode */`
- Les commentaires sur une ligne : `// un commentaire`
- Le commentaire doit mettre en avant :

o Une description

o Tags : `@param`, `@exception`

Code :

- Utilisation des tabulations pour l'indentation

Convention sur la compilation :

- Une compilation est correcte seulement si elle ne génère ni erreur ni warning.
- L'utilisation de `@SuppressWarnings` est interdite

3.5.6 REGLES DE TRADUCTION DE LA CONCEPTION VERS DU CODE EN JAVA

Les règles de traduction de la conception vers du code Java devront impérativement suivre celles vues en cours avec les auditeurs ProSE.

Cf : [Cours_Conception_2023]

3.6 DOCUMENTS INTERNES

Tout document interne, n'étant pas considéré comme document livrable, n'est pas soumis aux conventions de nommage ni de mise en forme. Ces documents internes ne passeront pas par le processus d'édition de document cité plus haut (pas de relecture ni de validation). Il est entièrement de la responsabilité du responsable du document de vérifier son contenu et expliciter synthétiquement son objectif dans son nommage et sur l'ENTP.

4 STANDARDS, PRATIQUES, CONVENTIONS ET METRIQUES

4.1 BUT

Cette section décrit les standards, pratiques, conventions et métriques utilisés pour le projet PSC. Ceux-ci ont pour but d'assurer la qualité du logiciel tout en fournissant des données quantitatives sur le processus d'AQ.

4.2 EXIGENCES QUALITES GENERALES

DESCRIPTION DES EXIGENCES QUALITE	
LIEES AU PRODUIT (par ordre décroissant de priorité)	<ul style="list-style-type: none">• Conformité : le produit livré devra être conforme au dossier de spécification livré et livré dans les délais promis.• Maintenabilité (Aptitude du produit à permettre une maintenance facile, rapide et peu coûteuse).• Adaptabilité (Aptitude de la partie logicielle à supprimer ou modifier les fonctionnalités existantes, ou ajouter de nouvelles fonctionnalités).• Maniabilité (Aptitude du produit à être convivial et facile d'emploi pour l'utilisateur).
LIEES AU PROCESSUS (par ordre décroissant de priorité)	<ul style="list-style-type: none">• Traçabilité.• Conformité (au présent PAQL et normes indiquées).• Simplicité.

Figure 11 : Tableau de description des exigences qualité

4.3 EXIGENCES QUALITES SUR LES ARTEFACTS

Pour tous les artefacts remis au client ou consultés par les auditeurs, une gestion de version et un suivi des modifications devront être activés afin de permettre la fourniture de n'importe quelle version d'un artefact de ce type et de pouvoir identifier clairement les modifications apportées entre 2 versions.

4.3.1 EXIGENCES SUR LES DOCUMENTS CONSULTABLES PAR LES AUDITEURS

Concernant les documents "consultables auditeur", tous devront respecter la même présentation et respecteront les mêmes modèles de document disponibles dans le dossier [Qualite/Modeles] du RDP.

4.3.2 EXIGENCES SUR LES DOCUMENTS LIVRABLES

Tous les documents livrables, avant d'être remis au client devront avoir été relus et corrigés. Hormis pour les artefacts de code, une version PDF doit être disponible pour les livrables lors de leur livraison.

Critère de qualité sur les documents livrables :

- Respect du modèle de document et ses champs.
- Pas plus de deux fautes d'orthographe par page du document.

Le document de spécification devra se baser sur la norme IEEE 830 [IEEE-830_1998] et utiliser la notation UML [UML_2.5.1_2017]. Le document de conception devra utiliser la notation UML [UML_2.5.1_2017].

4.3.3 SUR LE CODE SOURCE

Concernant les codes sources du projet, ils devront respecter les règles de programmation et les conventions de nommage associées. Afin de permettre une meilleure maintenabilité et lisibilité, ils seront documentés en utilisant des balises Doxygen.

Le code source livré devra être compilable sans warning et pouvoir produire un exécutable fonctionnel.

Il y aura au minimum les balises suivantes dans les artefacts de code :

@file : le nom du fichier

@brief : le résumé du contenu du fichier @date : la date de création du fichier

@author : le créateur du fichier et développeurs impliqués

@licence : Licence MIT

@copyright : Texte de la licence MIT

5 REVUES ET AUDITS

5.1 BUT

Cette section présente les actions d'audit interne, externe et de revue qui pourront être menées afin d'évaluer la qualité du projet, et ce sur différentes activités.

5.2 REVUES

5.2.1 REVUE DE MI-AVANCEMENT

La revue de mi-avancement permet de présenter l'ensemble des actions menées sur le premier incrément du cycle en V aux acteurs externes au projet. Un ensemble de détails est présenté sur le Wiki ProSE à l'adresse : [\[Wiki ProSE - Exigences pédagogiques pour SE\]](#). Tous les membres de l'équipe doivent être présents et intervenir durant la présentation. L'équipe disposera de 20 minutes de présentation et de 5 minutes de démonstration, suivies de 25 minutes de questions.

5.2.2 REVUE DE RECETTE

De même que pour la revue de mi-avancement, les précisions ont été faites sur le site du Wiki ProSE à l'adresse [\[Wiki ProSE - Exigences pédagogiques pour SE\]](#). Tous les membres de l'équipe doivent être présents et intervenir durant la présentation. L'équipe disposera de 20 minutes de présentation, dont 5 minutes de démonstrations, suivies de 30 minutes de questions.

5.3 AUDITS

Des audits externes seront menés par les consultants FORMATO durant toute la vie du projet. Un planning prévisionnel des audits est donné sur l'ENTP. Toutefois, les dates d'audit ne sont données qu'à titre indicatif, les auditeurs pouvant décaler leur audit d'une ou deux séances suivant leurs disponibilités. L'ordre de passage des équipes des audits au cours d'une séance n'est d'ailleurs jamais connu et reste à la discrétion de l'auditeur. L'équipe est tenue de mettre à disposition un membre compétent lors des audits externes.

Il y a 2 types d'audits externes :

- Les audits consultatifs,
- Les audits normatifs.

Lors de chacun de ces audits, l'équipe devra choisir un membre ayant le rôle de secrétaire qui prendra en notes les remarques et les changements à effectuer qui seront cités lors de l'audit. Le compte-rendu de cet audit sera déposé ensuite sur le RDP dans le dossier [Gestion_projet/Audit] sous le nom : « A(C ou N) JJ-MM-AAAA ».

5.3.1 AUDIT CONSULTATIF

Les consultants donneront une indication sur le travail effectué sous forme de code couleur (grade) sur l'avancement du projet. La signification des grades est la suivante :

- Le grade OR est attribué à des travaux de qualité exemplaire ;
- Le grade VERT est attribué à des travaux satisfaisants (de corrects à très bons) ;
- Le grade ORANGE est attribué à des travaux présentant quelques lacunes mais ne portant pas de préjudice grave pour la suite du projet ;
- Le grade ROUGE est attribué à des travaux déficients, présentant des lacunes importantes et dommageables pour le projet.

L'audit consultatif n'est pas noté et n'est pas transmis au client.

5.3.2 AUDIT NORMATIF

Cet audit nécessite une présentation soignée des documents audités. Il faut à *minima* que les remarques faites lors de l'audit consultatif correspondant aient été prises en compte et corrigées. Un code couleur (le même que pour un audit consultatif) est attribué.

Le rapport d'audit normatif est transmis au client et sera pris en compte pour l'évaluation finale du projet.

5.3.3 INSPECTION ET REVUE CROISEE

Pour les inspections internes, il convient de respecter les règles énoncées dans la partie associée (cf. [2.2.1.1 Inspections internes](#)).

6 TEST

L'ensemble des documents liés aux tests sont consultables dans le répertoire suivant :

[\[Test\]](#)

7 NOTIFICATION DES PROBLEMES ET CORRECTIONS

La notification des problèmes, à toute étape du processus de développement, se fait directement sur l'ENTP et ce par chacun des membres de l'équipe. Cela se fait par l'émission d'une demande de type « Bug » sur l'outil Redmine de l'ENTP.

Les états possibles d'une demande Bug sont :

- Nouveau
- En cours
- Fermé
- Réouvert
- Résolu

L'émission d'une telle demande se fera selon le formalisme suivant qui permet de tracer les origines des dysfonctionnements :

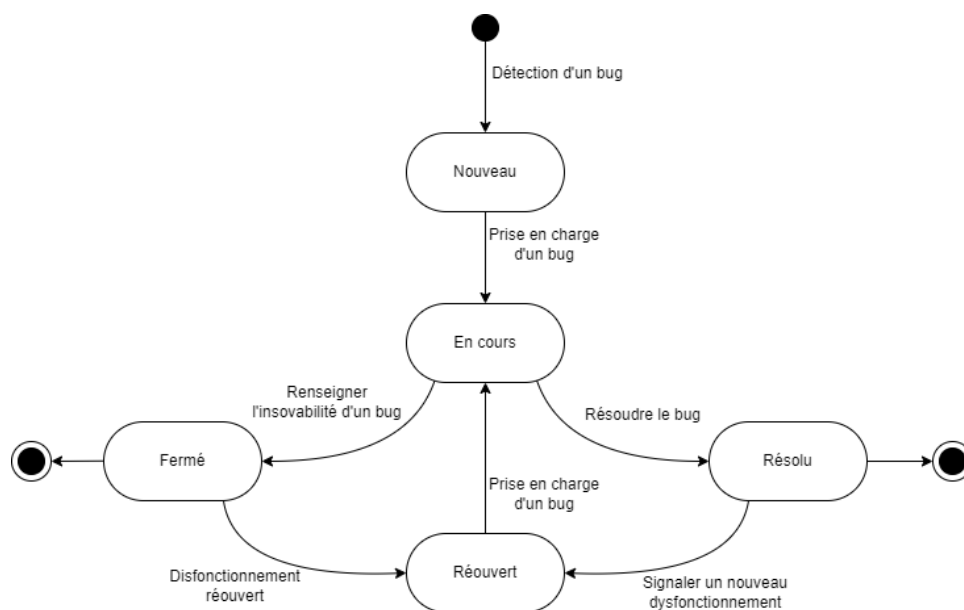


Figure 12 : Processus traitement d'un bug

Lorsqu'un membre de l'équipe signale un bug, l'état du bug devient **[Nouveau]**, si le bug a déjà été déclaré, son état sera **[Réouvert]**, puis, si le Bug est pris en charge par un membre, son état passe à **[En cours]**. Si la correction arrive à terme, l'état suivant sera **[Résolu]**. Dans le cas où la correction est impossible, due aux coûts, aux délais ou à des moyens techniques insuffisants, l'état deviendra **[Fermé]**.

Ce sont les responsables de développement ou le chef de projet qui décideront de l'assignation des membres aux bugs qui seront détectés.

8 OUTILS, TECHNIQUES ET METHODOLOGIE

8.1 L'ESPACE NUMERIQUE DE TRAVAIL DU PROJET (ENTP)

L'ENTP est un espace offrant les outils suivants :

- Redmine : un gestionnaire de suivi de projet.
- WikiProse : Wiki contenant un ensemble d'informations sur le projet ProSE, Wiki proposé par l'équipe pédagogique.
- Le RDP (Référentiel Document Projet) : C'est le dépôt git.

L'ENTP est accessible de la même façon à l'intérieur et à l'extérieur de l'ESEO. Les identifiants (login et mot de passe), lorsqu'ils sont réclamés, correspondent à ceux de l'ESEO.

Résumé des URL pour les différents composants de l'ENTP :

<u>Composant</u>	<u>URL</u>
WikiProSE	https://172.24.2.6/projects/documentation-prose/wiki
Redmine	https://172.24.2.6/
RDP	https://172.24.2.6/projects/se2024-a1/repository https://172.24.2.6/projects/se2024-a1/repository/c https://172.24.2.6/projects/se2024-a1/repository/android

Figure 13 : Tableau résumant les différents composants de l'ENTP

8.1.1 REDMINE

Redmine est un gestionnaire de projet ayant comme fonctionnalités :

- La gestion de plusieurs projets paramétrables
- La gestion des utilisateurs
- La gestion de documents
- La gestion de demandes
- Les priorités paramétrables d'une demande
- Un historique
- La modulation fine des statuts et la gestion des transitions de statuts par rôle.
- L'ajout de champs personnalisés
- La gestion du temps

Les consultants peuvent laisser des informations dans la rubrique « Dernières Annonces ».

8.1.2 PLANNING PREVISIONNEL

La version détaillée du planning prévisionnel est saisie sur Redmine. Chaque phase du projet, pour les deux incréments, dispose de sa propre « Version » et apparaît ainsi dans l'onglet « Roadmap ».

Les livrables des différentes phases du projet disposent de leur propre « Demandes » appelés « **Parties** » découpées en Sous-Parties. Pour chaque livrables les Parties et Sous-Parties décrivent la structure du document. Exemple pour le dossier de spécification :

- Spécification
 - 1. Introduction
 - 2. Description générales
 - 2.1 Caractéristiques des acteurs
 - 2.1.1. Acteurs directs
 - etc.

Les Parties sont communes pour les deux incréments du projet car la structure des livrables ne varient pas entre les incréments. Comme présenté dans la partie associée (cf. [3.4. État d'un document](#)) :

- Les Parties font l'objet d'un suivi de statut (« En cours », « En relecture », « En attente d'approbation », « Validé »)
- Les commits associés à ces demandes doivent leur faire références avec la convention #ID_PARTIE

Sous ces demandes, figurent des demandes dites « **Tâches** » permettant de répartir le travail entre les membres du projet. Les Tâches doivent être liés à la « Version » qu'elles traitent. Une Tâche comporte généralement les informations suivantes :

- Titre et description
- Date de début et échéance dit « Début » et « Échéance »
- Membre de l'équipe projet assigné à la tâche dit « Assigné à »
- Relation avec la version associée dit « Version cible »
- Saisie de temps

Ainsi, par exemple, dans la Sous-Partie 2.1.1. Acteurs directs, il peut y avoir une Tâche associée à l'Incément 1 et une autre associée à l'Incément 2. Les tâches apparaissent ainsi dans l'onglet « Roadmap »

Une tâche peut être effectuée par plusieurs membres de l'équipe projet dans lequel ils saisissent leur temps. Le membre de l'équipe à qui a été assigné la tâche est responsable de sa bonne conduite.

Pour le bon fonctionnement et une bonne gestion du projet, seul le chef de projet est autorisé à modifier le planning prévisionnel. Le CdP est donc le seul responsable de la concordance dans le temps du planning prévisionnel. Néanmoins, les autres membres de l'équipe sont autorisés à modifier, ajouter/supprimer ou modifier une Tâche tout en respectant la structuration de Tâches imposée par le CdP, et ceci pour garder une cohérence dans l'ensemble des tâches réalisées au cours du projet.

Un planning général a été réalisé sous PlantUML:

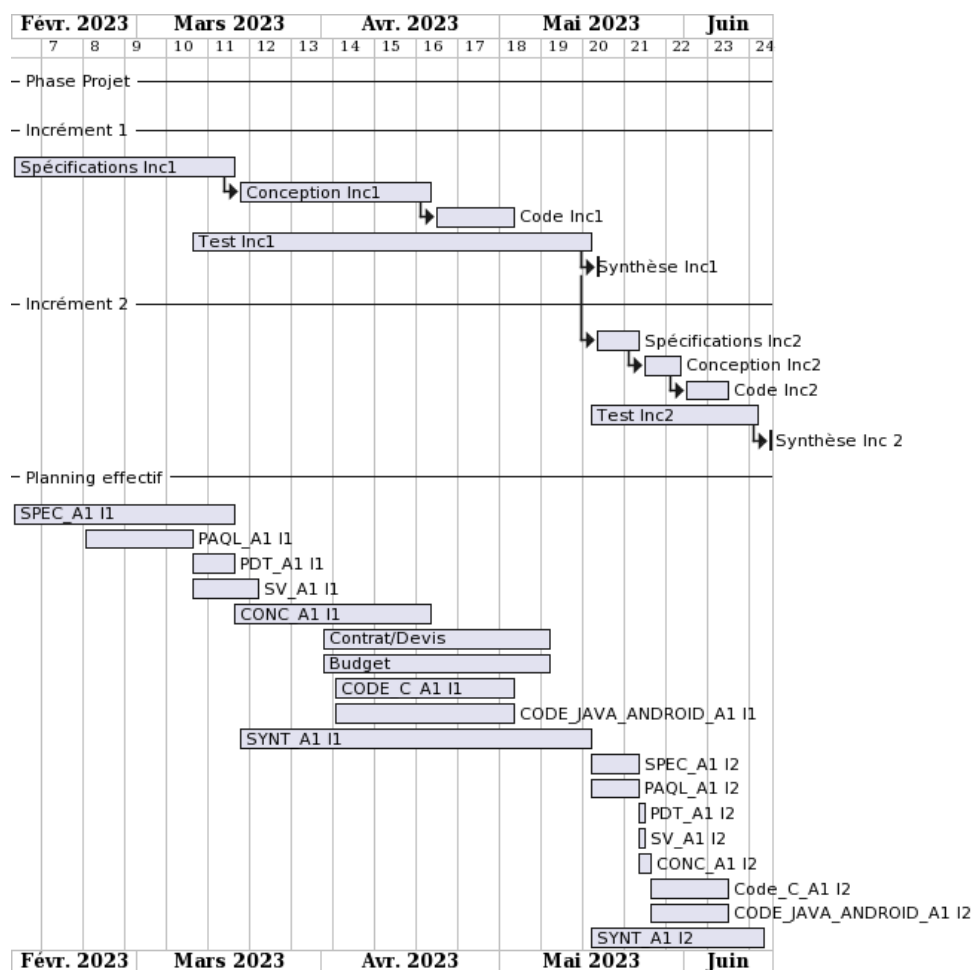


Figure 14 : Diagramme GANT du planning prévisionnel

8.1.3 SUIVI DU TRAVAIL

8.1.3.1 UNE MISE A JOUR IMMEDIATE SUR LE RDP

À chaque fois qu'un membre effectue un travail (en séance ou hors séance), il doit saisir le temps qu'il y a consacré sur Redmine par le biais de ses demandes de type « Tâche » et compléter l'état d'avancement de la demande. De plus, il devra alors déposer la nouvelle version de ces artefacts projet sur le RDP.

Remarque : La seule preuve du temps consacré à une tâche est le dépôt des artefacts. Un temps saisi sur Redmine sans dépôt associé peut ne pas être pris en compte par le CdP ou les consultants lors des audits « projet ».

De plus, une réunion d'équipe projet est organisée de façon hebdomadaire. Cette réunion a pour but de faire un point sur le travail de la semaine effectuée et de tenir à jour l'avancée du projet et ainsi établir les objectifs de la semaine à venir. Lors de chacune de ces réunions, un secrétaire est désigné pour rapporter les éléments importants abordés et les publier sous forme de compte-rendu sur le RDP dans le dossier gestion de projet.

8.1.3.2 SYNTHÈSE PERSONNELLE

Une synthèse du temps passé (dite synthèse personnelle) doit être régulièrement mis à jour sur le wiki de l'ENTP. Il est demandé de remplir définitivement et impérativement la synthèse personnelle avant les audits individuels.

Voici un exemple de synthèse personnelle :

NOM Prénom :

Nombre total d'heures travaillées :

Spécifications => 20h

Tâches	Travail effectué	Temps passé	Part du Travail	Révision(s)
#4496	IHM	12h	100%	doc 85f456fd doc 857aeb5f
#4512	Introduction	2h	100%	N/A
#4568	CU	6h	80%	doc 888856fd

Figure 15 : Tableau présentant un exemple de synthèse personnelle

8.2 LISTE DES OUTILS AUTORISÉS

Voici la liste de tous les outils que l'équipe projet est autorisée à utiliser lors du projet.

Usage	Outil utilisé
Communication spécialisée et appels groupés	Discord
Communication générale	Messenger
Messagerie de l'équipe de projet	prose2024a1@gmail.com
Stockage partagé	Google Drive
Document projet – planification- attribution des tâches	Redmine – version 4.1.0
OS Linux	Ubuntu - version 22.04
VPN ESEO	Stormshield SSL VPN Client – version 3.1.1 pour les clients Windows ou Tunnelblick – version 3.8.7 pour les clients Mac
Traitement de texte pour travail en commun (pour les documents n'étant pas des livrables clients)	Word (Online ou Desktop) ou Google Docs
Traitement de texte pour le livrables clients	LaTeX avec packages précisés dans le répertoire git de chaque livrable
Tableur	Excel (Online ou Desktop) ou Google Sheets

Diagramme UML	Plant UML Version : 1.2023.1
Gestion de test	SquashTM Version : 2.1.1
Création de figures illustratives	Google Drawings, Canva (https://www.canva.com/), LucidChart (https://lucid.app/) ou Draw.io (https://draw.io)
Présentation/soutenance de projet	PowerPoint (Online ou Desktop), Google Slides ou Canva (https://www.canva.com/)
Création des IHMs	LucidChart (https://lucid.app/) ou Figma (https://www.figma.com/)
Gestion de versions	Git
Développement sous Android	Android Studio Version : 2022.1.1
Développement C/Linux-Embarqué	OpenSTLinux (OSTL) : Linux déployé sur le Microprocesseur version 4.1 / kernel linux v5.15 Cube FW Package : ensemble des bibliothèques à destination du Microprocesseur (version 1.6) CubeIDE : IDE pour le développement des applications C à destination du Microcontrôleur (version 1.12)
Documentation du code C et Java	Doxygen version 1.9.1

Figure 16 : Tableau des logiciels autorisés

9 CONTROLE DES MEDIAS

9.1 COMMUNIQUER ENTRE MEMBRES INTERNES DU PROJET

Pour toute communication spécialisée ou appels de groupes : **Discord – Projet ProSE**

- L'envoi de documents et de liens est autorisé

Pour toute communication générale : **Messenger – ProSE – ST**

- Tout envoi de document est prohibé, seul l'envoi de photos ou de liens est autorisé

Pour tout document texte partagés : **Google Drive – S8_PROSE_SE** [\[lien\]](#)

- Contient la version modifiable des comptes-rendus rédigés lors des réunions et des audits.
- La version non-modifiable (PDF) est disponible sur le Git dans [\[gestion_projet/reunions\]](#) ou [\[gestion_projet/audit\]](#)

9.2 GESTION DES MEDIAS, SOURCES, REFERENCES, COPYRIGHT

Le Titulaire des droits d'auteur (le prestataire) concède au client par la présente, une licence non exclusive, libre de redevance et mondiale pour l'utilisation, la copie, la modification, la fusion, la publication, la distribution, la sous-licence et/ou la vente du Logiciel et/ou des copies du Logiciel sans restriction conformément à la licence MIT.

La présente autorisation est assortie des conditions suivantes :

- La mention du droit d'auteur ci-dessus doit être incluse dans toutes les copies ou parties substantielles du Logiciel.
- Le Logiciel est fourni « en l'état », sans garantie d'aucune sorte, expresse ou implicite, y compris mais sans s'y limiter les garanties de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon. En aucun cas les auteurs ou les détenteurs des droits d'auteur ne pourront être tenus responsables de tout dommage résultant de l'utilisation du Logiciel.

9.3 COMMUNIQUER AVEC DES MEMBRES EXTERNES AU PROJET

9.3.1 CONTACTER LE CLIENT

Toute la correspondance mail avec le client doit être également archivée dans le compte mail du projet : prose2024a1@gmail.com

Pour garantir une communication optimum entre le client et l'équipe projet, seul le chef de projet dispose des droits pour communiquer avec le client. Les différents membres de l'équipe devront regrouper les questions pour le client lors des réunions d'équipe hebdomadaires pour que le chef de projet puisse les transmettre au client (en cas d'urgence, contacter le chef de projet directement).

10 CONTROLE DES FOURNISSEURS

Le cabinet de conseil FORMATO forme et conseille les équipes des projets ProSE. Ainsi, certaines formations sont obligatoires et gratuites, tandis que d'autres sont payantes et facultatives. De plus, FORMATO propose aussi un service consulting sur la plupart des domaines abordés durant le projet. La liste des consultants et de leurs spécialités est disponible dans le chapitre associé (cf. [2.1.2.3 Consultants et Auditeurs](#)) tandis que les procédures d'inscription aux formations et de demandes de consulting sont détaillées ci-après.

10.1 S'INSCRIRE A UNE FORMATION

Toute demande de formation doit être décidée en accord avec le CdP. Ce dernier transmettra la demande d'inscription au plus tard 48 heures avant le début de la formation.

Toutes les demandes effectuées par le chef de projet sont enregistrées et archivées dans un document spécifique. Les éléments suivants seront à indiquer : la date, le sujet de formation, nom du formateur, durée, nombre de membres inscrits, présence des membres inscrits.

10.2 DEMANDER UN CONSULTING

Pour demander un consulting, il est important de prendre un rendez-vous même si le consultant est planifié sur la séance ProSE. En dehors des séances, les consultants ne sont pas nécessairement disponibles et il faut impérativement envoyer une demande par email avec un préavis de 48h.

Tout membre de l'équipe peut demander à faire un consulting. Pour se faire, il doit préalablement obtenir l'accord du CdP. Il envoie ensuite un mail au consultant selon les modalités précisées ci-dessus.

Au minimum deux membres de l'équipe doivent être présent pour un consulting. Le consulting peut se dérouler dans une salle de réunion de l'ESEO réservée à cet effet, ou dans la salle de cours de SE.

Les consultings sont notés et suivis dans le fichier [SUIVI_BUDGET.xlsx] trouvable sur le RDP dans [\[Gestion Projet - Budget\]](#).

Pour plus de détails voir : [\[Wiki ProSE - Règles pour contacter clients et consultants\]](#)

11 COLLECTE, MAINTENANCE ET CONSERVATION DES ARCHIVES

11.1 LE REFERENTIEL DOCUMENTAIRE PROJET (RDP)

Le RDP est le dépôt de tous les artefacts numériques du projet. Il est impératif que tous les artefacts numériques produits lors du projet y soient stockés sans délais. Ces artefacts seront pris en charge par un système de gestion de version, de sorte qu'il sera possible de revenir à n'importe quelle version de ces artefacts.

11.1.1 STRUCTURATION DU RDP

Le RDP est composé de trois référentiel structurés en différents dossiers comme suit :

11.1.1.1 REFERENTIEL ANDROID

Les fichiers compilés ou tout autre exécutable ne doit pas apparaître dans le référentiel.

- Explorations : Tous les fichiers utilisés en phase d'exploration. Ces fichiers ne sont pas soumis au respect des conventions de qualités du présent PAQL.
- Production : Tous les fichiers permettant la réalisation de la solution technique pour le client

11.1.1.2 REFERENTIEL C

- Explorations : Tous les fichiers utilisés en phase d'exploration. Ces fichiers ne sont pas soumis au respect des conventions de qualités du présent PAQL.
- Production : Tous les fichiers permettant la réalisation de la solution technique pour le client

11.1.1.3 REFERENTIEL DOC

Lien vers le référentiel : [\[Référentiel\]](#)

- conception
 - ebauches : Ce sont tous les fichiers utilisés au cours de la constitution des dossiers de conception.
 - livrables : Documents dès qu'ils ont atteint une première fois l'état « diffusable » ainsi que toutes leurs révisions ultérieures.
 - schémas : Toutes les sources des dessins ou fichiers UML produits lors de la phase de conception.
- gestion_projet
 - audit : Répertoire regroupant les comptes-rendus des différents audits ainsi que les présentations.
 - budget : Répertoire regroupant les documents relatifs à la gestion du budget.
 - client : Répertoire regroupant les documents en rapport avec le client.

- planning : Répertoire regroupant les plannings du projet.
 - réunions : Répertoire regroupant les ordres du jour et comptes-rendus des réunions.
- qualite : Cette partie regroupe tous les documents liés à la gestion de la qualité du projet.
 - modeles : Répertoire regroupant les modèles des documents à utiliser pendant le projet.
- specification
 - ebauches : Répertoire contenant les fichiers utilisés au cours de la constitution du dossier de spécification.
 - figures : Répertoire contenant les images utilisées au cours de la création du dossier de spécification.
 - schemas : Répertoire contenant les sources ou dessins produits lors de la phase de spécification.
- test : Regroupe tous les artefacts liés à l'activité de test (Cahier de test, Plan Test, Planning).
 - cahier_test : Répertoire regroupant les fichiers utilisés au cours de la constitution du cahier de test.
 - plan_test : Répertoire regroupant les fichiers utilisés au cours de la constitution du plan de test.
 - planning : Répertoire regroupant les fichiers utilisés pour le planning de test.

11.1.2 OBTENIR UN ACCES AU RDP

Les accès au RDP sont gérés par la société FORMATO. Pour obtenir un accès, veiller à les contacter.

11.1.3 NOM DES FICHIERS DES ARTEFACTS

Seules des lettres non accentuées de l'alphabet latin (haut de casse, bas de casse et tirets) sont autorisées pour le nom de fichier. Les espaces ne sont pas autorisés.

11.1.4 DEPOSER UN ARTEFACT SUR LE RDP

Pour assurer une traçabilité de qualité, certaines règles sont à prendre en compte lors du dépôt d'un quelconque document sur le RDP. Le dépôt de document important se fait de façon séparée, et en aucun cas le commit simultané de plusieurs fichiers n'est autorisé si les modifications ne sont pas identiques ou en lien entre elles. De cette façon il sera aisé d'assigner un commentaire de « commit » au document déposé, ce commentaire sera constitué d'une description des modifications réalisées ainsi que la référence de la tâche associée aux modifications.

11.2 GESTION DES DOCUMENTS PAPIER

Aucun document papier ne sera conservé dans un souci de respect de l'environnement. Les documents éventuellement produits seront scannés et il incombera au créateur du document de veiller à sa dématérialisation sur le RDP. Les livrables ne seront évidemment pas imprimés toujours dans un souci de respect de l'environnement.

12 FORMATION

Si des compétences requises par le projet ne sont pas maîtrisées par suffisamment de membres dans l'équipe, les membres veilleront à acquérir ces compétences par le biais de lectures, de formations entre les membres, par des consultings auprès des consultants FORMATO ou par des participations à des formations proposées par la société FORMATO.

Lors de chacune des formations, un membre préalablement choisi sera chargé de la prise de note. Il n'oubliera pas les éléments suivants :

- La date, le(s) professeur(s), thème (Android, C, Spécifications, ...)
- Les noms et prénoms de personnes présentes avec le secrétaire en italique
- Pour chaque question :
 - Problème ou question (en gras souligné)
 - Nouvelles connaissances acquises et/ou sources recommandées
 - Solutions, pistes et conseils pour résoudre le problème

13 GESTION DU RISQUE

La gestion du risque au sein du projet permet d'anticiper au maximum les risques techniques et humains susceptibles de mettre le projet en péril. La fréquence et la gravité de chaque risque est noté sur une échelle de 1 à 5, 1 étant la valeur la plus faible et 5 la plus élevée dans chacune de ces catégories.

Risque	Fréquence (1-5)	Gravité (1-5)	Risque (F*G)	Mesure à prendre
Indisponibilité d'un membre de l'équipe sur plusieurs séances	3	2	6	<p>La création d'un planning permet d'appréhender les tâches qui ont été assigné au membre indisponible et ainsi les réassigner aux autres membres du projet suivant leur importance.</p> <p>De plus, la sur facturation des heures clients, permet d'anticiper une charge de travail plus importante liée à l'indisponibilité d'un membre de l'équipe.</p>
Pandémie / confinement	2	4	8	Le CdP doit s'assurer de pouvoir continuer à travailler à distance notamment via les outils de collaboration en ligne.
Perte/corruption de données	3	5	15	<p>L'utilisation du logiciel de gestion de versions Git permet d'éviter les pertes. Son utilisation est complétée d'une formation de l'équipe de projet permettant de faire de sauvegarde fréquentes.</p> <p>De plus, le dépôt git fait l'objet d'une redondance permettant de minimiser le risque</p>
Problème de disponibilité cible	3	5	15	La cible est régulièrement sauvegardée et l'équipe est formée à sa bonne utilisation afin de minimiser le risque.
Spécification non à jour	4	3	12	Si la spécification n'est pas approuvée par les clients, le CdP doit les contacter pour se mettre d'accord sur le travail à revoir.
Retard dans le projet ou planification trop optimiste	5	4	20	<p>Le suivi du projet doit se faire à l'aide d'un planning, de jalons et de points météo reportés aux clients toutes les 2 semaines.</p> <p>Si un retard est inévitable, le CdP doit prévenir les clients le plus tôt possible.</p>
Inefficacité récurrente équipe	3	4	12	Le CdP se doit de toujours "prendre la température" sur l'ambiance générale. Si l'ambiance générale se désagrège, il est libre d'organiser un Team building.
Problème de compétence techniques	5	4	20	Les membres en question en informent le CdP qui autorise la prise d'un consulting.

				Si quelqu'un est mieux qualifié dans l'équipe, une rotation des tâches attribuées est possible.
Paiement tardif de l'équipe - RedMine est fermé	2	5	10	Le CdP doit contacter les syndicats représentés par M. DELATOUR pour entamer une discussion. Dès que le problème est résolu, il en tient informé l'équipe projet à l'oral et/ou sur Messenger .

Figure 17 : Tableau de gestion des risques

14 OUTILS ET CONFIGURATIONS

14.1 RESEAU INFORMATIQUE

Le réseau informatique de l'ESEO permet d'accéder à internet ainsi qu'à tous les outils cités précédemment. Par conséquent, les membres projet sont tenus de respecter la charte informatique ESEO (déjà signé lors de l'entrée à l'ESEO).

14.2 GIT

Afin de réduire le risque « **Perte / Corruption de données** », est mis en place l'outil de gestion de version et de sauvegarde « git ». Git permet efficacement de gérer les modifications apportées à des fichiers texte, de créer plusieurs branches de travail, de fusionner des changements apportés par plusieurs membres du projet ou encore de collaborer de façon efficace tout en gardant des sauvegardes via l'historique des versions

Le projet PSC utilise le git mis à disposition sur le Redmine suivant les règles suivantes :

- Tout fichier du projet doit être importé sur le git dans le dossier de sa catégorie ;
- Les livrables clients se basent sur des projets LaTeX converti en documents PDF ;
- Aucun document Word n'est admis sur le git. Ils doivent être préalablement converti au format PDF ;
- Toute modification de fichier doit faire l'objet d'un commit au format suivant :
« DESCRIPTION_DU_COMMIT #ID_DE_LA_TACHE_CORRESPONDANTE_SUR_LE_REDMINE » ;

Le git du projet s'organise de la manière suivante :

- se2024-a1.doc
 - conception
 - ebauches
 - livrables
 - schemas
 - gestion_projet
 - audit
 - budget
 - client
 - planning
 - reunions
 - qualite
 - specification
 - ebauches
 - figures
 - schemas
 - test
 - cahier_test
 - plan_test
 - planning
- se2024-a1.c
 - exploration
 - production
- se2024-a1.android
 - exploration
 - production

Il est interdit de créer de nouveaux dossiers de niveau supérieur ou de même niveau que les dossiers présentés ci-dessus. Les sous-dossiers pour organiser l'espace de travail sont eux vivement conseillés.

La méthode ci-dessous décrit comment versionner de manière efficace en utilisant l'outil GIT. Cette pratique ne sera mise en place qu'à partir de la phase de conception (Non pris en compte pour la phase de spécification de l'Inc1) et ne s'intéresse qu'aux codes sources soit les dépôts c et android de l'ENTP.

Le dépôt distant regroupe 2 branches principales :

- Une branche de déploiement (ou branche Master) accessible à tous les membres et permettant de déployer une version saine et testée du code et/ou des différents documents livrables, et conforme à la dernière version du PAQL. Attention la branche master devant être considérée comme une branche saine et cloisonnée, par conséquent seuls le Responsable C/Android, le CdP et le RQT sont autorisés à déployer des documents dessus.
- Une branche de production (ou develop) accessible à tous les membres de l'équipe et permettant le développement et pouvant être subdivisée en plusieurs sous-branches. Il s'agit d'une branche sur laquelle seront déposées uniquement les modifications validées par les différents responsables (Responsable C/Android, RQT ou CdP).

Pour implémenter une nouvelle fonctionnalités/partie du code, il faut créer une nouvelle branche de type `feature_branches` nommées `feature_[Nom_Feature]` (avec `Nom_Feature` le nom de la nouvelle fonctionnalité) dédiée pour ne pas polluer la branche `develop` par des versions non finalisées ou modifications intempestives.

Les `feature_branches` sont assignées à un développeur qui est le seul autorisé à modifier. Il intègre les modifications à une nouvelle branche de type `release_branches` nommée `release_[Nom_Feature]` (avec `Nom_Feature` le nom de la fonctionnalité modifiée) pour qu'elle soit partagée.

Si des modifications majeures sont nécessaires après vérification, le document peut être reporté dans la branche `feature` correspondante.

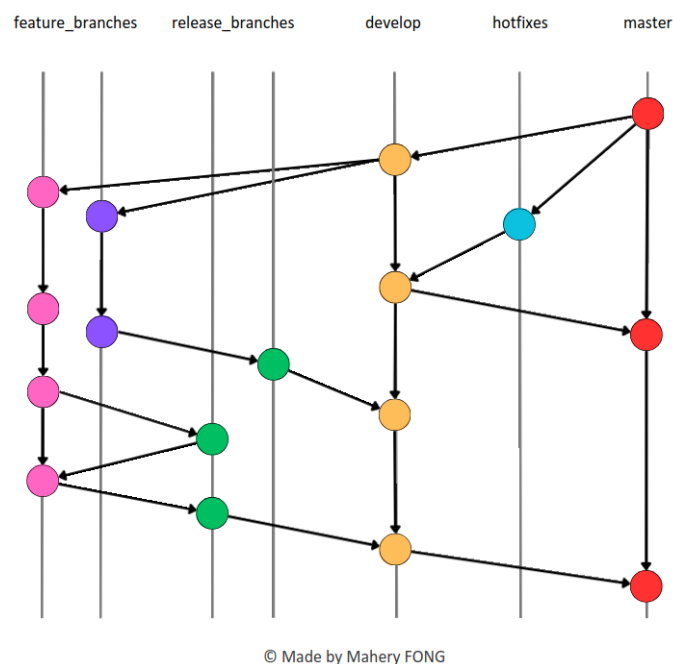


Figure 18 : Versionnage des branches

Quand porter le code sur la branche master ?

Lorsque le code en cours a été validé, testé et est conforme dans la branche develop, on peut alors demander au Responsable C/Android, au CdP ou au RQT de le porter sur la branche master.

Que faire si un bug important apparaît sur la branche master ?

Il faut réaliser un correctif pour pallier ce problème qui sera directement appliqué à la branche develop, puis ensuite à la branche master. Ce patch sera réalisé sur une nouvelle branche de type hotfixes nommée hotfix_[Nom_Correctif] (avec Nom_Correctif le nom du correctif à corriger).

15 GLOSSAIRE : DEFINITIONS, ACRONYMES ET ABREVIATIONS

Les définitions, acronymes et abréviations utilisées dans le présent document sont répertorié dans le tableau ci-dessous :

<u>Acronyme</u>	<u>Définition</u>
AP (Artefact Projet)	Désigne tous les éléments numériques ou analogiques produits lors du développement du projet (documents numérique ou papier, code sources, jeux de test...).
AQ (Assurance Qualité)	Fourniture de produits et services tels que biens de consommation, produits manufacturés, projets industriels, prestations ou logiciels.
AC (Audit Consultatif)	Chaque membre de l'équipe devra intervenir au cours de la présentation, ce temps de parole devant être équitablement partagé. Il est toléré que le chef de projet ait un temps de parole légèrement supérieur aux autres. Durant cette présentation, une démonstration de votre application est demandée. Cette présentation n'est pas évaluée.
AN (Audit Normatif)	L'audit normatif est similaire à l'audit consultatif. Cette présentation est évaluée.
CdP (Chef de Projet)	Personnage clef du projet, il planifie, dirige, prend en charge la relation client, le suivi global du projet, l'aspect budgétaire et financier ainsi que les formations nécessaires à l'équipe.
ENTP (Espace Numérique de Travail du Projet).	Espace regroupant les différents outils permettant l'organisation du projet. [ENTP]

IEEE (Institute of Electrical and Electronics Engineers)	Association professionnelle internationale définissant entre autres des normes dans le domaine informatique et électronique.
OMG (Object Management Group)	Association professionnelle internationale définissant entre autres des normes dans le domaine informatique.
RQT (Responsable Qualité et Test)	Il/elle devra s'assurer de la bonne qualité du travail effectué par l'équipe et ce à tous les niveaux du développement. Il/elle aura notamment en charge la gestion des documents, des versions et la définition des conventions de nommage (en particulier le paramétrage du logiciel Doxygen).
RDP (Référentiel Document Projet)	Dépôt de tous les artefacts numériques du projet. Ce dépôt est mis à disposition de l'équipe projet ainsi qu'à l'équipe des consultants FORMATO.
ProSE (Projet Système Embarqué)	Le but de ce projet est de vous familiariser avec les différentes facettes du développement d'un système embarqué telles que vous pourrez les rencontrer, plus tard, lors de votre vie professionnelle.
PAQL (Plan d'Assurance Qualité Logicielle)	Document qui définit les normes et règles fixées pour toute l'équipe projet. Ce document est signé par tous les membres.
RDP (Référentiel Documentaire Projet)	Dépôt de tous les artefacts numériques du projet. Ce dépôt est mis à la disposition de l'équipe projet, ainsi qu'à l'équipe des consultants FORMATO.
UML (Unified Modeling Language)	Notation graphique normalisée définie par l'OMG et utilisé en génie logiciel.
PSC (Prototype Sonnette Connecté)	Nom du projet.

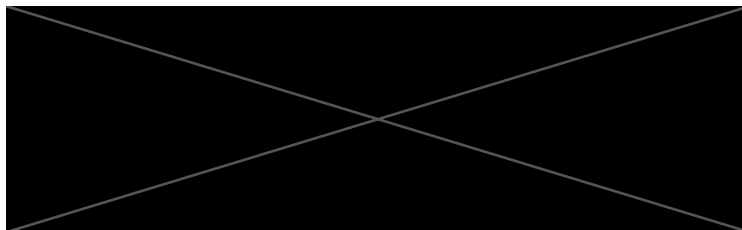
Figure 19 : Tableau de glossaire

16 VALIDATION DU DOCUMENT

Signature de tous les membres projets. La signature vaut avoir lu le présent document et engage son signataire à respecter le présent PAQL pendant toute la durée du projet ProSE.

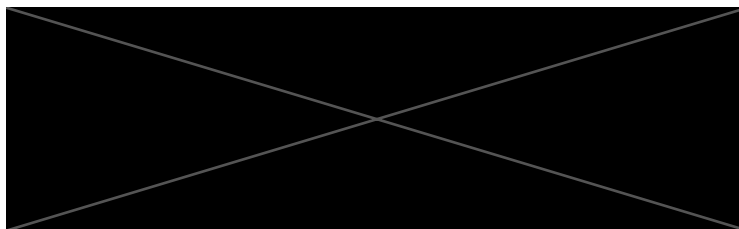
CdP - BOUY Hugo

RQT - TROVALLET Romain



MOULIN Mathis

CASSAR Bastien



CHIRON Paul

JURET Paul

LETENNEUR Laurent

