Plan d’assurance Qualité Logicielle

Version 0.3.2  
PROSE Équipe B1 2024

Responsable du document : Paul TRÉMOUREUX

État du document : En attente de relecture

**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 des étudiants de l'option SE 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 son propriétaire.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Actions** | **Auteur** | **Version** | **Révision** |
|  | Création du document | FORMATO | 0.0 | 1 |
| 24/02/2023 | Début adaptation du document au projet CANvegers, CANdroid, CANnect | Paul TRÉMOUREUX | 0.1 | 0 |
| 03/03/2023 | Première révision du document | Paul TRÉMOUREUX | 0.1 | 1 |
| 06/03/2023 | Responsabilités client et équipe ProSE | Paul TRÉMOUREUX | 0.1 | 2 |
| 07/03/2023 | Travail sur les jalons | Paul TRÉMOUREUX | 0.1 | 3 |
| 08/03/2023 | Travail sur 9.3, 10.2, 11.1, 13 | Théo BENARD | 0.1 | 4 |
| 09/03/2023 | Finition des jalons, ajout de schémas | Paul TRÉMOUREUX | 0.1 | 5 |
| 10/03/2023 | Finalisation de la version | Paul TRÉMOUREUX | 0.1 | 6 |
| 14/03/2023 | Relecture du PAQL | Théo BÉNARD | 0.2 | 0 |
| 27/03/2023 | Correction du PAQL jusqu’à 8.1.3 | Paul TRÉMOUREUX | 0.2 | 1 |
| 28/03/2023 | Correction du PAQL jusqu’à la fin (exception pour 10.2 et 14.2 | Paul TRÉMOUREUX | 0.2 | 2 |
| 29/03/2023 | Travail de mise en page | Paul TRÉMOUREUX | 0.2 | 3 |
| 30/03/2023 | Ajout de nouveaux types de fichiers et travail sur la mise en page | Paul TRÉMOUREUX | 0.2 | 4 |
| 06/04/2023 | Relecture et mise en commentaire des corrections à apporter | Théo BÉNARD | 0.2 | 4 |
| 07/04/2023 | Mise en place des corrections | Paul TRÉMOUREUX | 0.3 | 0 |
| 28/04/2023 | Correction fautes et ajout précisions dans 14.2 | Paul TRÉMOUREUX | 0.3 | 1 |
| 03/05/2023 | Précisions dans 14.2 | Paul TRÉMOUREUX | 0.3 | 2 |

***Tableau 1 : Table des évolutions et validations internes du document***

Contenu

[1. But 4](#_Toc130894001)

[2. Gestion 6](#_Toc130894002)

[3. Documentation 20](#_Toc130894003)

[4. Standards, pratiques, conventions et métriques 27](#_Toc130894004)

[5. Revues et Audits 29](#_Toc130894005)

[6. Test 31](#_Toc130894006)

[7. Notification des problèmes et corrections 32](#_Toc130894007)

[8. Outils, Techniques et Méthodologie 34](#_Toc130894008)

[9. Contrôle des médias 38](#_Toc130894009)

[10. Contrôle des fournisseurs 40](#_Toc130894010)

[11. Collecte, maintenance et conservation des archives 41](#_Toc130894011)

[12. Formation 44](#_Toc130894012)

[13. Gestion du risque 45](#_Toc130894013)

[14. Outils et configurations 46](#_Toc130894014)

[15. Glossaire : Définitions, acronymes et abréviations 46](#_Toc130894015)

[16. Validation du document 49](#_Toc130894016)



# But

## 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 « Passerelle Android-CAN vers banc CAN réel ou simulé » du groupe B1 2024 « CANvengers » 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 [[qualite/PAQL/version](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/qualite/PAQL/version)] sous le nom [[PAQL\_B1\_2024](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/qualite/PAQL/version/PAQL_B1_2024.docx)].

## Portée

Ce document est destiné :

* À l’équipe projet ;
* Aux consultants de la société FORMATO.

Il pourra être consulté par le client sur sa demande.

## Copyright

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 des étudiants de l'option SE 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 son propriétaire.

## Vue d’ensemble

Ce PAQL est structuré suivant les grandes parties proposées par la norme [[IEEE-730\_1998](https://standards.ieee.org/ieee/730/971/)].

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.

## Références

[[PAQL\_Exemple\_2023](https://campus.eseo.fr/mod/resource/view.php?id=39395)] : Exemple, exemple\_plan\_qualite\_ProSE.odt, campus ESEO, révision finale, 2023.

[[IEEE-730\_1998](https://standards.ieee.org/ieee/730/971/)] : IEEE, std 730-1998 « IEEE Standard for Software Quality Assurance Plans», http://standards.ieee.org/findstds/standard/730-2002.html, 1998.

[[IEEE-830\_1998](https://standards.ieee.org/ieee/830/1222/)] : IEEE, std 830-1998 « Recommended Practice for Software Requirements Specifications », <http://standards.ieee.org/findstds/standard/830-1998.html>, 1998.

[[UML\_2.4\_2011](http://www.omg.org/spec/UML/)] : OMG, Unified Modeling Language, version 2.4.1, <http://www.omg.org/spec/UML/>, 2011.

[[GNU-GPL\_3.0\_2007](https://www.gnu.org/licenses/gpl-3.0.html)] : GNU General Public License, version 3.0, <http://www.gnu.org/licenses/gpl-3.0.html>, 2007.

# Gestion

Cette partie décrit l’organisation, les tâches et les responsabilités en rapport avec les activités d’Assurance Qualité (AQ) du projet de l’équipe CANvengers.

## Organisation

### Projet concerné

Le projet « Passerelle Android-CAN vers banc CAN réel ou simulé » concerne les deux entités suivantes :

* L’entreprise FORMATO avec :
  + L’équipe ProSE B1 2024 de l’ESEO, réalisatrice du projet ;
  + Une équipe de consultant.
* L’entreprise KEREVAL, commanditaire du projet.

### Ressources humaines

#### Équipe projet

Les différents membres de l’équipe projet CANvengers et leurs informations importantes sont listées dans le tableau ci-dessous :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rôle** | **Nom** | **Prénom** | **Courriel** | **Téléphone** |
| CdP | BÉNARD | Théo | [theo.benard@reseau.eseo.fr](mailto:theo.benard@reseau.eseo.fr) | +33 6 44 82 30 64 |
| RQT | TRÉMOUREUX | Paul | [paul.tremoureux@reseau.eseo.fr](mailto:paul.tremoureux@reseau.eseo.fr) | +33 6 37 23 77 07 |
| Référent Android | MARQUETTE | Gabriel | [gabriel.marquette@reseau.eseo.fr](mailto:gabriel.marquette@reseau.eseo.fr) | +33 6 75 71 83 45 |
| Développeuse Android | LENNE | Camille | [camille.lenne@reseau.eseo.fr](mailto:camille.lenne@reseau.eseo.fr) | +33 7 69 69 45 42 |
| Référente C | DECLERCK | Elisa | [elisa.declerck@reseau.eseo.fr](mailto:elisa.declerck@reseau.eseo.fr) | +33 6 84 27 38 56 |
| Développeur C | ROCHER | Thomas | [Thomas.rocher@reseau.eseo.fr](mailto:Thomas.rocher@reseau.eseo.fr) | +33 6 15 22 35 15 |

***Tableau 2 : Membres de l'équipe projet***

Pour la réalisation du projet, les différents membres de l’équipe projet CANvengers s’organisent de la manière décrite sur l’organigramme suivant :

Une image contenant diagramme

Description générée automatiquement

***Figure 1 : Organigramme de l’équipe projet***

#### Client

Le projet de l’équipe CANvengers est réalisé pour l’entreprise KEREVAL.

KEREVAL, une entreprise spécialisée dans les tests de systèmes embarqués véhicule, 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 novices dans le métier, tels que les nouveaux salariés, lors de leur arrivés chez KEREVAL, ou des étudiants lors des forums.

Pour répondre à ce besoin, KEREVAL souhaite utiliser un simulateur de tableau de bord (type ICSim) sur Linux et une carte électronique de type Raspberry Pi pour permettre la gestion des capteurs et actionneurs connectés 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 sur téléphone. Ce dernier sera connecté au système (via la Raspberry Pi) par un réseau TCP/IP. Sur l’application, il sera possible d’envoyer et recevoir des trames.

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

KEREVAL a fait appel à l’équipe B1 2024 pour réaliser le prototype de ce système.

Le client est représenté par MM. Alain RIBAULT, Yannick GUYOMARCH et Killian HENRY. Les adresses courriels de ces contacts sont disposées ci-dessous :

* M. Alain RIBAULT (directeur technique – CTO) : [alain.ribault@kereval.com](mailto:alain.ribault@kereval.com) ;
* M. Yannick GUYOMARCH (ingénieur projet) : [yannick.guyomarch@kereval.com](mailto:yannick.guyomarch@kereval.com) ;
* M. Killian HENRY (technicien système) : [kilian.henry@kereval.com](mailto:kilian.henry@kereval.com).

#### 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](mailto:matthias.brun@eseo.fr) ;
* Camille CONSTANT (activités de test et qualité) : [camille.constant@eseo.fr](mailto:camille.constant@eseo.fr) ;
* Jérôme DELATOUR (gestion de projet, PAQL, spécification, conception) : [jerome.delatour@eseo.fr](mailto:jerome.delatour@eseo.fr) ;
* Frédéric JOUAULT (spécification, conception, C) : [frederic.jouault@eseo.fr](mailto:frederic.jouault@eseo.fr).

Des activités d’audits externes (cf. chapitre , page) seront menées par les auditeurs de la société FORMATO ou missionnées par elle.

## Tâches du projet

### Tâches transversales

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

* Documentation (cf. chapitre , page ) ;
* Revues et audits (cf. chapitre , page ) ;
* Inspections internes ;
* Validation et tests ;
* Activités d’amélioration du processus d’AQ.

#### Inspections internes

Les inspections internes seront réalisées toutes les semaines par tous les membres de l’équipe en fonction de leur rôle :

* Le CdP :
  + Vérification des demandes sur l’ENTP ;
  + Vérification du RDP ;
  + Bonne organisation des tâches et sous-tâches ;
  + Présence d’un descriptif pertinent pour chaque tâche et avancement.
* Le RQT :
  + Mise à jour cahier non-respect de la qualité ;
  + Vérification des saisies d’heure sur l’ENTP ;
  + Respect de la convention de nommage des dépôts sur le RDP ;
  + Lisibilité du code ;
  + Analyse sur le respect des règles du PAQL ;
  + Vérification de la pertinence des commentaires.
* Les référents langage :
  + Vérification entre le document de spécification, conception et code produit ;
  + Vérification de la présence de commentaires.

#### 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.

Cf : [[plan\_de\_test\_TEST\_B1.pdf](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/test/plan_test/livrables/plan_de_test_TEST_B1.pdf)]

#### Évolution et amélioration 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.

### Tâches par rapport au cycle de développement

L’équipe projet suivra 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).

Une image contenant diagramme

Description générée automatiquement

***Figure 2 : Diagramme du cycle de développement du projet***

#### 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 :**   * CdP et RQT | **Méthodes & Règles :**   * Règles pour l'utilisation de l'ENTP * Anticipation et organisation des deadlines personnelles | **Moyens & Outils :**   * Initialisation du projet sous ENTP |
| **Activités d’organisation/pilotage :**   * Organisation de la réunion de lancement * Organisation de la phase en aval | **Activités de production/soutien :**   * Élaboration PAQL * Mise en place de l’ENTP * Définition de la démarche du projet * Initialisation du planning et du suivi du projet * Création des modèles d’ODJ et de CR | **Activités de vérification/contrôle :**   * Réunion de lancement * Première réunion client |
| **Produits/données en entrée :**   * Wiki ProSE et ENTP * Documents pédagogiques | **Produits/données en sortie :**   * Planning des tâches sur l’ENTP * Modèles ODJ et CR * Points météo | **Produits révisés :**   * PAQL |
| **Jalons de la phase :**  J1 : 07/02/2023 – Initialisation de l’ENTP  J2 : 08/02/2023 – Formation ENTP pour l’équipe projet  J3 : 08/02/2023 – Validation des livrables et du planning prévisionnel avec l’équipe projet  J4 : 07/02/2023 – Lecture du cahier des charges  J5 : 08/02/2023 – Réunion de compréhension du cahier des charges et préparation de la rencontre avec le client  J6 : 09/02/2023 – Rencontre avec le client  J7 : 08/03/2023 – Installation des différents logiciels par l’ensemble des membres de l’équipe projet  J8 : 09/03/2023 – PAQL rédigé  J9 : 10/03/2023 – PAQL signé par tout l’équipe | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Nomination des CdP et RQT * Répartition des langages entre les développeurs | * Validation des futurs livrables * Validation de l'équipe du planning prévisionnel * Validation et signature du PAQL par l’équipe | * ENTP opérationnel * Attribution des rôles |

***Tableau 3 : Phase Initialisation du projet***

#### Phase de spécification

Spécifications : Le dossier de spécification devra respecter le plan défini par la norme [[IEEE-830\_1998](https://standards.ieee.org/ieee/830/1222/)] et s’appuyer sur la notation UML [[UML\_2.4\_2011](https://www.omg.org/spec/UML/)]. 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 : Spécification 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 :**   * CdP et équipe * Client | **Méthodes & Règles :**   * PAQL | **Moyens & Outils :**   * ENTP * SquashTM |
| **Activités d’organisation/pilotage :**   * Organiser les échanges d’informations avec le client | **Activités de production/soutien :**   * Rédaction du dossier de spécifications * Rédaction du plan de test * Élaboration des maquettes des écrans * Explorations techniques * Rédaction des tests de validation | **Activités de vérification/contrôle :**   * AC Spécification, Plan de test et Scénarios de validation * Organiser le suivi des tâches et production de l’équipe projet * Consultings axés sur le document de spécification * Réunions avec le client * Réunions d’équipe |
| **Produits/données en entrée :**   * Cahier des charges Client * PAQL | **Produits/données en sortie :**   * Dossier de spécifications * Plan de test * Contrat client * Scénarios de validation * Points météo | **Produits révisés :**   * ENTP * Planning prévisionnel * PAQL |
| **Jalons de la phase :**   * J1 : 16/03/2023 – Spécification V1.0.0 * J2 : 16/03/2023 – Plan de test V1.0.0 * J3 : 17/03/2023 – AC Spécification * J4 : 17/03/2023 – AC Plan de test * J5 : 17/03/2023 – Scénarios de validation V1.0.0 * J6 : 21/03/2023 – AC Scénarios de validation * J7 : 24/03/2023 – Signature contrat client * J8 : 23/05/2023 – Spécification V2.0.0 * J9 : 24/05/2023 – AN Spécification * J10 : 25/05/2023 – Plan de test V2.0.0 * J11 : 25/05/2023 – Scénarios de validation V2.0.0 * J12 : 26/05/2023 – AN Plan de test * J13 : 26/05/2023 – AN Scénarios de validation | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Initialisation effectuée | * Validation des produits en sortie * Signature du client | * Dossier de spécifications signé par le client |

***Tableau 4 : Phase Spécification V1 & V2 du projet***

#### Phase de conception

La phase de conception se divise en deux parties :

* Une conception générale (ou système) avec un audit consultatif ;
* Une conception détaillée avec un audit consultatif et un audit normatif.

|  |  |  |
| --- | --- | --- |
| **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 :**   * CdP et équipe * Client | **Méthodes & Règles :**   * PAQL * Dossier de spécification | **Moyens & Outils :**   * ENTP * Plan de test * Scénarios de validation * Dossiers de fonctionnement de tableaux de bord |
| **Activités d’organisation/pilotage :**   * Organiser les échanges d’informations avec le client | **Activités de production/soutien :**   * Définition de l’architecture technique * Conception logicielle * Normalisation des fonctions * Rédaction du dossier de conception | **Activités de vérification/contrôle :**   * Contrôles hebdomadaires de membres de l’équipe projet * Consultings axés sur la conception et le test * Réunions avec le client * Réunions d’équipe |
| **Produits/données en entrée :**   * Dossiers de spécifications * PAQL * Plan de test * Scénarios de validation | **Produits/données en sortie :**   * Dossier de conception * Points météo | **Produits révisés :**   * Normes de développement * Dossier de spécification * PAQL |
| **Jalons de la phase :**   * J1 : 18/04/2023 – Conception V1.0.0 * J2 : 19/04/2023 – AC Conception * J3 : 30/05/2023 – Conception V2.0.0 * J4 : 31/05/2023 – AN Conception | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Document de spécification signé par le client | * Validation des produits en sortie | * Dossier de conception validé * Normes de développement * Tests de validation rédigés |

***Tableau 5 : Phase Conception V1 & V2 du projet***

#### Phase de réalisation

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 : Réalisation V1 & V2** | | |
| **Objectifs :**   * Développer des applications logicielles * Tester les applications logicielles définies lors de la phase de conception * Coder 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 :**   * CdP et équipe | **Méthodes & Règles :**   * PAQL * Spécification * Conception | **Moyens & Outils :**   * ENTP * Moyen de test * Visual Studio Code |
| **Activités d’organisation/pilotage :**   * Organiser les fonctions prioritaires * Organiser les échanges d'informations avec le client | **Activités de production/soutien :**   * Rédaction du code de test * Rédaction du code Android * Rédaction du code C | **Activités de vérification/contrôle :**   * Tests unitaires validés * Contrôles hebdomadaires de membres de l’équipe projet * Consultings axés sur le développement (Android et C) et le test * Réunions avec le client * Réunions d’équipe |
| **Produits/données en entrée :**   * Dossiers de conception * PAQL * Plan de test * Dossier de spécification | **Produits/données en sortie :**   * Code source (Android et C) * Code de test * Moyen de test opérationnel | **Produits révisés :**   * Normes de développement * Document de conception * Document de spécification * PAQL |
| **Jalons de la phase :**   * J1 : 10/05/2023 – Code source V1.0.0 * J2 : 10/05/2023 – Code de test V1.0.0 * J3 : 12/05/2023 – AC Code source * J4 : 12/05/2023 – AC Code de test * J5 : 07/06/2023 – Code source V2.0.0 * J6 : 07/06/2023 – Code de test V2.0.0 * J7 : 08/06/2023 – AN Code source * J8 : 08/06/2023 – AN Code de test | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Validation du dossier de conception | * Validation des produits en sortie | * Application logicielle réalisée * Tests réalisables |

***Tableau 6 : Phase Réalisation V1 & V2 du projet***

#### 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 :**   * CDP & RQT * Équipe de test | **Méthodes & Règles :**   * PAQL * Conception * Spécification * Plan de test * Cahier de tests | **Moyens & Outils :**   * Moyen de test * ENTP * SquashTM |
| **Activités d’organisation/pilotage :**   * Vérification croisée * Organiser les tests principaux | **Activités de production/soutien :**   * Exécution tests de communication * Compléter cahier de test * Exécution tests intégration * Exécution tests unitaires * Exécution tests validation | **Activités de vérification/contrôle :**   * Vérification de la pertinence des tests * Vérification de la couverture des tests * Vérification des cas spécifiques * Contrôles hebdomadaires de membres de l’équipe projet * Consultings axés sur le test * Réunions avec le client * Réunions d’équipe |
| **Produits/données en entrée :**   * Application logicielle * Dossier de spécification * Dossier de conception * PAQL * Moyen de test * Cahier de test * Plan de test * Scénarios de validation | **Produits/données en sortie :**   * Document d'analyse des tests * Application logicielle testée * Cahier de test | **Produits révisés :**   * Normes de développement * Document de conception * Document de spécifications * Code * PAQL |
| **Jalons de la phase :**   * J1 : 12/05/2023 – Cahier de test V1.0.0 * J2 : 15/05/2023 – AC Cahier de test * J3 : 10/06/2023 – Cahier de test V2.0.0 * J4 : 12/06/2023 – AN Cahier de test | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Plan de test partiellement validé | * Validation des produits en sortie | * Tests réalisés * Cahier de test complet |

***Tableau 7 : Phase Test V1 & V2 du projet***

#### 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 :**   * CdP et équipe * Client | **Méthodes & Règles :**   * PAQL * Spécification * Conception * Intégration | **Moyens & Outils :**   * ENTP |
| **Activités d’organisation/pilotage :**   * Organiser les échanges d’informations avec le client | **Activités de production/soutien :**   * Mise en forme des documents livrables | **Activités de vérification/contrôle :**   * Relecture et correction des documents livrables * Relecture du procès-verbal de recette définitive |
| **Produits/données en entrée :**   * PAQL * Dossier de spécification * Dossier de conception * Cahier de test * Code | **Produits/données en sortie :**   * 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 :**   * Cahier de test * Manuel d’utilisation * Manuel d’installation * Code source * Recette |
| **Jalons de la phase :**   * J1 : 12/05/2023 – Gestion de projet V1.0.0 * J2 : 15/05/2023 – AC Gestion de projet * J1 : 10/06/2023 – Cahier de test V2.0.0 * J2 : 12/06/2023 – AN Cahier de test | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Phase d’intégration terminée | * Validation des produits en sortie | * Fin du projet |

***Tableau 8 : Phase Recette du projet***

#### Phase de bilan de fin de projet

|  |  |  |
| --- | --- | --- |
| **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 :**   * CdP | **Méthodes & Règles :**   * PAQL | **Moyens & Outils :**   * ENTP (RedMine) |
| **Activités d’organisation/pilotage :**   * Organiser les échanges d’informations avec le référent ESEO et M. Delatour | **Activités de production/soutien :**   * Rédaction des documents de bilan * Diaporama de présentation de fin de projet | **Activités de vérification/contrôle :**   * Audit interne * Réunion de fin de projet |
| **Produits/données en entrée :**   * ENTP (Redmine) * Livrables * RDP | **Produits/données en sortie :**   * Statistiques projet * Bilan de projet * Évaluation équipe * Compte-rendu de réunion de fin de projet | **Produits révisés :**   * ENTP * Livrables * RDP |
| **Jalons de la phase :**   * J1 : 12/06/2023 – Dossiers de synthèse * J2 : 13/06/2023 – Entrainement soutenance de fin de projet * J3 : 14/06/2023 – Soutenance de fin de projet * J4 : 14/06/2023 – Remise de l’ensemble du matériel emprunté pour le projet * J5 : 15/06/2023 – Bilan ProSE | | |
| **Conditions de début de phase :** | **Conditions de fin de phase :** | **Conditions de passage à la phase suivante :** |
| * Phase d’intégration terminée | * Validation des produits en sortie | * Fin du projet |

***Tableau 9 : Phase Bilan Fin de Projet du projet***

## Responsabilité

### Définition Générale des rôles

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 de l’équipe, le CdP réparti les tâches entre les différents membres de l’équipe. Il doit aussi s’assurer du respect des délais de la part de tous les membres de l’équipe. Représentant l’équipe, c’est lui qui sera l’interlocuteur privilégié avec le client, les consultants et les responsables pédagogiques ;
* **Responsable Qualité et Test (RQT) :** Le RQT a deux tâches principales. Tout d’abord, il est responsable du respect du PAQL de la part des différents membres de l’équipe. Il devra par ailleurs rédiger ce PAQL en début de projet. Ensuite, le RQT sera chargé d’imaginer les tests qui seront effectuer par les programmeurs afin de garantir au client que ce qui est livré correspond aux attentes listées dans le cahier des charges ;
* **Référent langage :** Le référent est un développeur qui sera l’interlocuteur privilégié avec le CdP et le RQT. Son rôle sera de faciliter le passage d’information entre les développeurs et les managers ;
* **Développeur :** Les développeurs réaliseront la grande majorité de la programmation. Chaque développeur est affecté à un langage (C ou Android).

### Récapitulatif des responsabilités client sur les phases

|  |  |  |
| --- | --- | --- |
| **Phase** | **Implication Client** | **Implication Équipe Projet** |
| * Initialisation | * Mettre à disposition le cahier des charges | * Initialiser l’ENTP * Répartir le travail entre les membres de l’équipe projet * Rédiger le PAQL * Répartir les rôles de référents langage |
| * Spécification | * Répondre aux questions concernant l’étude des besoins * Valider l'IHM * Fournir l’existant système si nécessaire * Valider les dossiers | * Réaliser l’étude des besoins * Rédiger le dossier de spécifications * Rédiger le cahier de test |
| * Conception | * Discuter puis approuver le dossier de conception | * Rédiger le dossier de conception * Implémenter les tests d’intégration |
| * Réalisation |  | * Développer les modules logiciels * Implémente les tests unitaires |
| * Test | * Discuter puis approuver le plan de test * Vérifier la matrice de conformité entre fonctionnalités demandées et tests * Vérifier les résultats des tests de validation système | * Rédiger le plan de test * Rédiger et implémente les tests de validation système * Implémenter les tests d’intégration * Implémenter les tests unitaires * Exécuter tous les tests * Produire le cahier de test * Produire la matrice de conformité |
| * Recette | * 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 * Signer le procès-verbal de recette définitive * S’acquitter des obligations contractuelles | * Rédiger le procès-verbal de recette * Livrer l'ensemble des livrables |
| * Bilan de fin de projet | * Faire une analyse du projet | * Faire une analyse et les statistiques du projet |

***Tableau 10 : Responsabilités Client/Équipe projet***

### Récapitulatif des responsabilités CdP sur les phases

|  |  |
| --- | --- |
| **Phase** | **Implication CdP** |
| Initialisation | * Initialiser l’ENTP * Répartir le travail entre les membres de l’équipe projet * Rédiger le PAQL |
| Spécification | * Réaliser l’études ou les études des besoins * Rédiger le dossier de spécifications * Rédiger le cahier de test |
| Conception | * Rédiger le dossier de conception |
| Réalisation | * Développer les modules logiciels |
| Test |  |
| Recette | * Rédiger le procès-verbal de recette * Livrer l'ensemble des livrables |
| Bilan de fin de projet | * Faire une analyse et les statistiques du projet |

***Tableau 11 : Responsabilités Chef de Projet***

### Récapitulatif des responsabilités RQ sur les phases

|  |  |
| --- | --- |
| **Phase** | **Implication RQT** |
| Initialisation | * Rédiger le PAQL |
| Spécification | * Réaliser l’études ou les études des besoins * Rédiger le cahier de test |
| Conception | * Implémenter les tests d’intégration |
| Réalisation | * Développer les modules logiciels |
| Test | * Rédiger le plan de test * Produire la matrice de conformité |
| Recette |  |
| Bilan de fin de projet |  |

***Tableau 12 : Responsabilités Responsable Qualité et Tests***

### Récapitulatif des responsabilités des développeurs sur les phases

|  |  |
| --- | --- |
| **Phase** | **Implication Équipe Développement** |
| Initialisation | * Répartir les rôles de référents langage |
| Spécification | * Réaliser l’études ou les études des besoins * Rédiger le dossier de spécifications * Rédiger le cahier de test |
| Conception | * Rédiger le dossier de conception * Implémente les tests d’intégration |
| Réalisation | * Implémente les tests unitaires |
| Test | * Rédiger et implémente les tests de validation système * Implémenter les tests d’intégration * Implémenter les tests unitaires * Exécuter tous les tests * Produire le cahier de test |
| Recette |  |
| Bilan de fin de projet |  |

***Tableau 13 : Responsabilités Équipe de Développement***

# Documentation

## 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.

## 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 ;
* 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) 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(s) réunion projet ;
* Correspondance échangée avec le client ;
* Les codes source des explorations techniques ;
* Plan Assurance Qualité Logicielle (PAQL).

Les artefacts de code sont :

* Les codes source ;
* Les codes de test.

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

## Référence des documents

La référence d’un document est de la forme suivante : « SIGLE\_XX », où XX 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.

Les documents faisant références aux réunions (équipe, client, consulting) sont de la forme « SIGLE\_XX\_JJ\_MM ». Dans ce cas, JJ correspond au jour de la réunion et MM au mois de la réunion.

|  |  |
| --- | --- |
| **Référence** | **Libellé document** |
| PAQL\_XX\_AAAA | Plan d’Assurance Qualité Logicielle |
| SPEC\_XX\_AAAA | Document de spécification |
| CONC\_XX\_AAAA | Document de conception |
| TEST\_XX\_AAAA | Document de test |
| BILAN\_XX\_AAAA | Bilan de projet |
| TRESO\_XX\_AAAA | Documents de trésorerie |
| CONT\_XX\_AAAA | Contrat client |
| CR\_XX\_JJ\_MM | Compte rendu de réunion |
| ODJ\_XX\_JJ\_MM | Ordre du jour d’une réunion |
| PM\_XX\_JJ\_MM | Point météo |
| SANS\_XX\_AAAA | Pour tous les autres documents |

***Tableau 14 : Sigles pour la documentat******ion***

Par exemple, pour le dossier de spécification, produit par l’équipe CANvengers, sa référence sera la suivante : « SPEC\_B1 ».

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\_B1\_2024 ».

## État d’un document

Les différents états d’un document sont :

* **En création :** Document en cours de création ;
* **En construction**: Document en cours de développement :
  + **En révision :** Document en cours de modification ;
  + **En attente de relecture** : Document en attente de relecture ;
  + **En attente de validation** : Document en attente de validation.
* **Version finale En attente de livraison :** Document validé et en attente de livraison ;
* **Document livré au client :** Document livré au client.

Une image contenant diagramme

Description générée automatiquement

***Figure 3 : Machine à états du processus de suivi d'un document***

## Responsable du document

Un seul membre de l’équipe est responsable d’un document. Tout document doit avoir un responsable désigné. C’est ce responsable qui suit l’évolution du document et ses différents états.

C’est aussi le responsable qui gérera les cas où différents membres d’une équipe désirent travailler en même temps sur la même version d’un document dont il est le responsable. Il lui appartient alors de mettre en œuvre la stratégie qui lui semble la mieux adaptée, soit en :

* Séquençant les mises à jour ;
* Demandant que les membres lui envoient leur version afin qu’il intègre lui-même les modifications dans le document et qu’il le dépose alors dans le RDP. Dans ce dernier cas, chacun des contributeurs pose sa contribution sur le RDP avant intégration.

Étant responsable du document, il sera également responsable du contenu de ce dernier. De plus le responsable se chargera de la création du document.

## Processus d’édition d’un document

Le processus d'édition de document nous permet d'assurer un suivi en continu sur les différents documents du projet et ainsi être en mesure de réagir rapidement en cas de pertes d'informations ou pour retrouver l'origine d'erreurs commises.

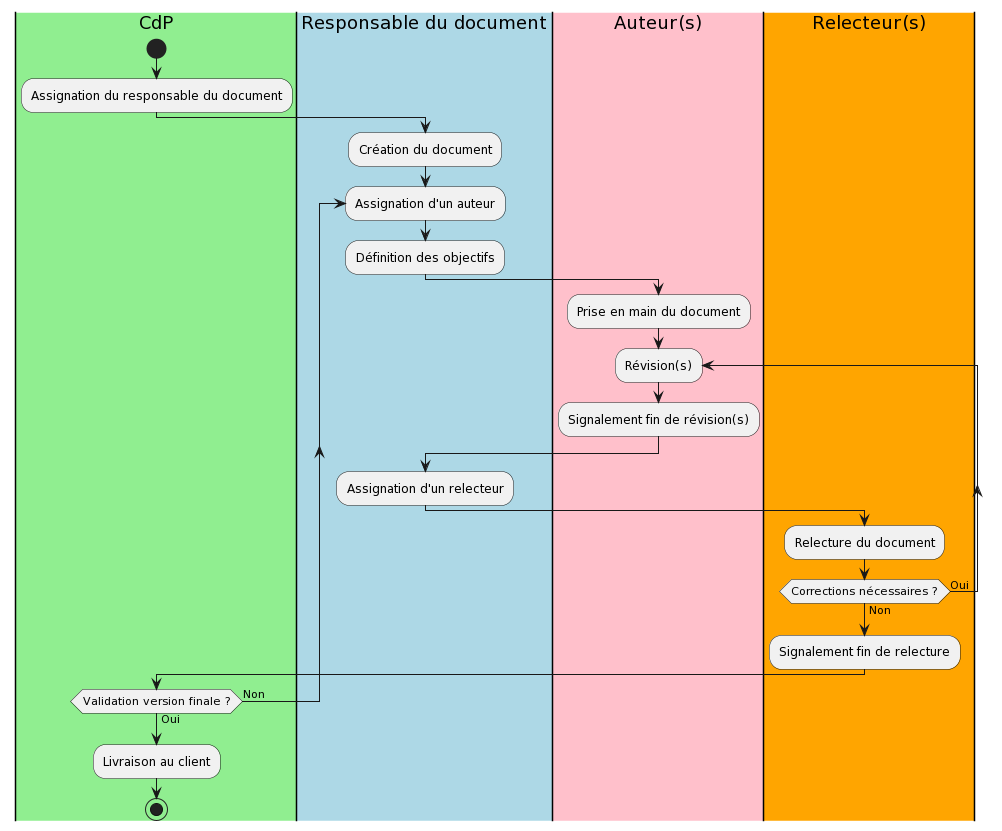
Les décisions concernant un document sont toujours faites à deux. Lorsque le responsable du document est le CdP, le RQT viendra aider à la prise de décisions.

Lors de ce processus, des personnes seront chargées de rédiger le document. Ces personnes seront désignées par le responsable du document en accord avec le chef de projet. La sélection de la personne en charge de la rédaction sera faite au début de la phase « En construction » du document. De plus, chaque fois que le document passe en phase « En construction », le responsable de la rédaction peut être amené à changer. Enfin, le CdP ou le responsable du document peuvent avoir la charge de la rédaction.

Lors de ce processus, des personnes seront chargées de relire le document. Ces personnes seront désignées par le responsable du document en accord avec le chef de projet. La sélection de la personne en charge de la relecture sera faite au début de la phase « En attente de relecture » du document. De plus, chaque fois que le document passe en phase « En attente de relecture », le responsable de la relecture peut être amené à changer.

La personne en charge de la relecture sera :

* Si possible : un membre de l’équipe n’ayant pas travaillé sur ce projet ;
* Sinon : un membre de l’équipe n’ayant pas travaillé sur le document depuis la dernière relecture ;
* Sinon : un membre de l’équipe.



***Figure 4 : Diagramme d'activité du processus d'édition d'un document***

## version d’un document

Dans le PAQL ici présenté, il s’agit d’une version, sur 3 nombres X.Y.Z. La gestion des versions des documents doit respecter les règles suivantes :

* X s’incrémente à chaque validation d’un document (X s'incrémentera donc au maximum jusqu'à 2) ;
* Y s'incrémente à chaque relecture complète du document (chaque fois qu'une personne termine une phase de relecture et publie le document sur le RDP) ;
* Z s’incrémente à chaque révision du document (chaque fois qu’une personne apporte une modification au document ;
* Chaque incrément de X remet Y à 0 ;
* Chaque incrément de Y remet Z à 0 ;
* Il n’est pas possible de valider un document s’il s’agit d’une version avec Z non nul ;
* Il n’est pas possible de livrer un document s’il s’agit d’une version avec Y et/ou Z non nul.

Pensez à mettre à jour le numéro de version dans le tableau en tête de document, ainsi que sur la page de couverture et sur les pieds de page.

## 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](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/qualite)].

### Modèle de document

Des modèles de documents sont proposés suivant le sigle de ces documents. Ces modèles doivent être impérativement utilisés. Ils sont disponibles dans le RDP, dans le dossier [[qualite/PAQL/modeles](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/qualite/PAQL/modeles)]. Tous les fichiers seront au format OpenDocument (modifiables et lisibles avec LibreOffice). Veillez à activer le suivi des modifications pour tous ces fichiers.

### Artefact de code

Dans le RDP, dossier [[qualite/PAQL/modeles](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/qualite/PAQL/modeles)], les modèles d'artefact de code suivants sont disponibles :

* [[example.h](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/qualite/PAQL/modeles/example.h)] ;
* [[example.c](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/qualite/PAQL/modeles/example.c)] ;
* [[example.java](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/qualite/PAQL/modeles/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.

Avec ces fichiers modèles, vous trouverez également des fichiers utilitaires dont l’utilisation est vivement recommandée.

### Règles de codage en langage C

Les règles de codage en C sont disponibles dans les modèles correspondants disponibles sur le RDP.

### Règles 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](https://campus.eseo.fr/course/view.php?id=8387)]

### Règles de codage en langage Java

Les règles de codage en Android sont disponibles dans les modèles correspondants disponibles sur le RDP.

### Règles 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](https://campus.eseo.fr/course/view.php?id=8387)]

## 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.

Exception est faite pour ce document.

# Standards, pratiques, conventions et métriques

## But

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

## Exigences qualités générales

|  |  |
| --- | --- |
| **Description des Exigences qualité** | |
| Liées au produit (par ordre décroissant de priorité) | * 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). |
| Liées au processus (par ordre décroissant de priorité) | * Traçabilité. * Conformité (au présent PAQL et normes indiquées). * Simplicité. |

***Tableau 15 : Liste des exigences qualité du projet***

## Exigences qualités 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.

### Exigences sur les documents consultables par les auditeurs

Concernant les documents "consultables auditeurs", tous devront respecter la même présentation et respecteront les mêmes modèles de document disponibles dans le dossier [[/qualite/PAQL/modeles](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/qualite/PAQL/modeles)] du RDP.

### 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](https://standards.ieee.org/ieee/830/1222/)] et utiliser la notation UML [[UML\_2.5\_2017](https://www.omg.org/spec/UML/)]. Le document de conception devra utiliser la notation UML [[UML\_2.5\_2017](https://www.omg.org/spec/UML/)].

### 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 ;
* @version : la version du fichier ;
* @date : la date de création du fichier ;
* @author : le créateur du fichier et développeurs impliqués ;
* @todo : à la fin de la dernière révision, l’auteur indique ce qu’il reste à faire ;
* @copyright : © Copyright Équipe ProSE B1 2024 CANvengers.

# Revues et Audits

## 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.

## Revues

### 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 dans la partie [[Exigences pédagogiques](https://172.24.2.6/projects/documentation-prose/wiki/Exigences_p%C3%A9dagogiques_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.

### Revue de recette

De même que pour la revue de mi-avancement, les précisions ont été faites sur le site du Wiki ProSE dans la partie [[Exigences pédagogiques](https://172.24.2.6/projects/documentation-prose/wiki/Exigences_p%C3%A9dagogiques_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.

## 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 fournir 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](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/gestion_projet/audit)] sous le nom : « A(C/N)\_JJ\_MM\_AAAA\_SANS\_XX ».

### Audit consultatif

Les consultants vous donneront une indication sur le travail que vous avez fait sous forme de code couleur (grade) sur l’avancement de votre 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.

### 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.

### Inspection et revue croisée

Pour les inspections internes, il convient de respecter les règles énoncées dans le chapitre « 2.2.1.1 Inspections internes ».

# Test

Tous les éléments relatifs aux tests sont répertoriés dans le plan de test déposé sur le RDP, dans le dossier [[/test/plan\_test/livrables](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/test/plan_test/livrables)], sous le nom [[plan\_de\_test\_TEST\_B1\_2024.pdf](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/test/plan_test/livrables/plan_de_test_TEST_B1.pdf)].

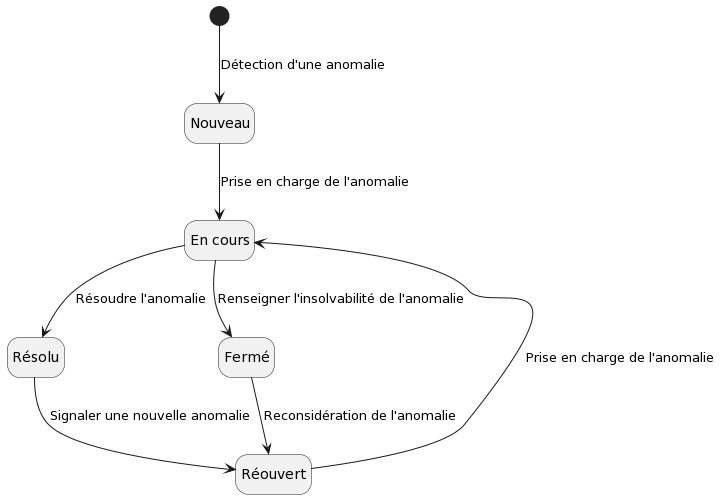
# Notification des problèmes et corrections

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.

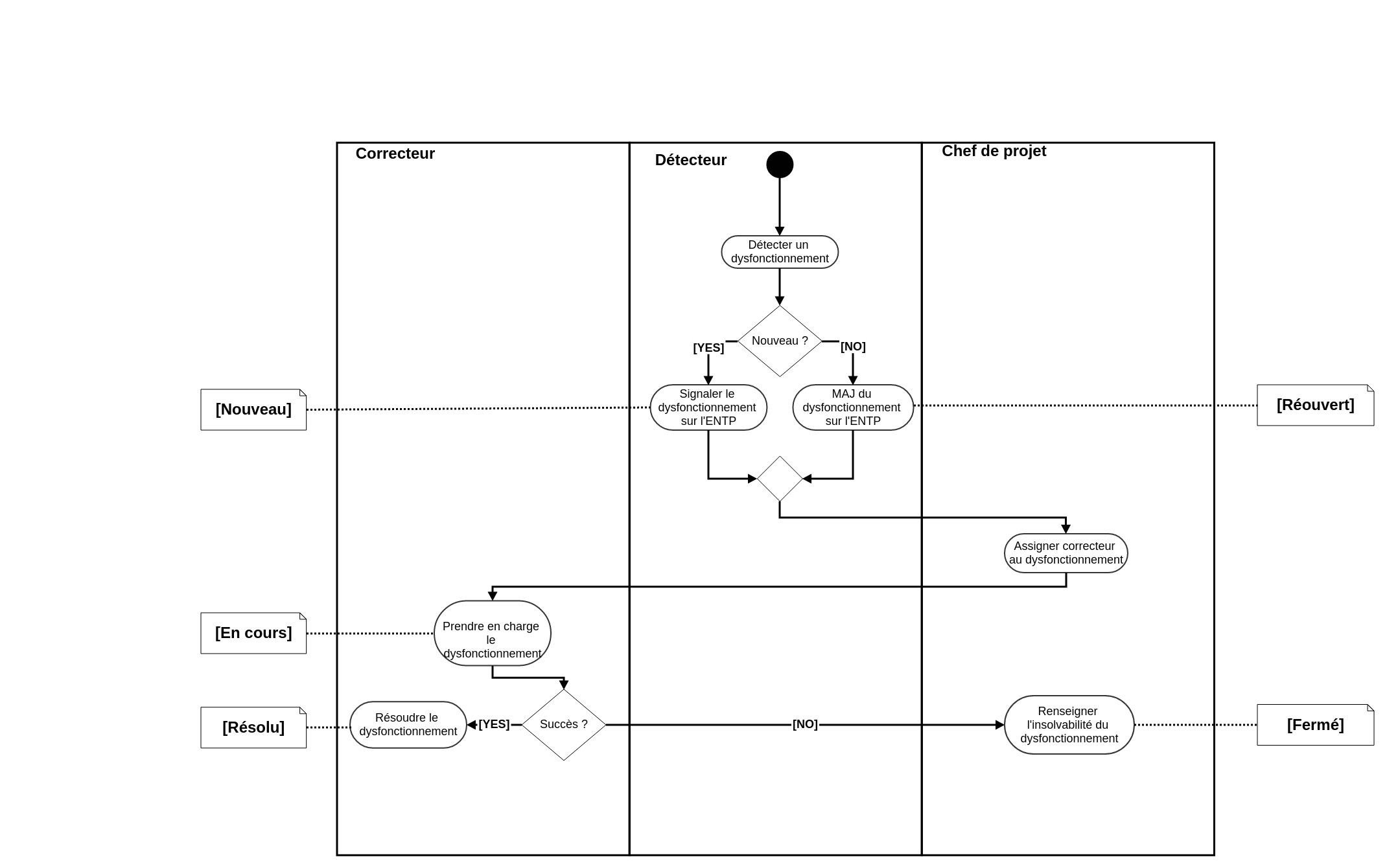
L'émission d'une telle demande se fera selon le formalisme suivant qui permet de tracer les origines des dysfonctionnements :



***Figure 5 : Machine à état du processus de traitement d'un dysfonctionnement***

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, l'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**].

C'est le chef de projet qui décidera de l'assignation des membres aux bugs qui seront détectés. Le diagramme suivant permet d'illustrer l'activité des membres impliqués lors d'une tâche de Debug :

**

***Figure 6 : Diagramme d'activité expliquant le processus de traitement d'un dysfonctionnement***

# Outils, Techniques et Méthodologie

## L’espace Numérique de Travail du Projet (ENTP)

L’ENTP est un espace vous 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. Ce wiki est disponible dans la partie [[Wiki](https://172.24.2.6/projects/documentation-prose/wiki)] de la [[documentation ProSE](https://172.24.2.6/projects/documentation-prose/)] de l’ENTP ;
* Le RDP (Référentiel Document Projet) : c'est le [[dépôt git](https://172.24.2.6/projects/se2024-b1/repository)].

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 :

|  |  |
| --- | --- |
| **Composant** | **URL** |
| WikiProSE | * <https://172.24.2.6/projects/documentation-prose/wiki> |
| Redmine | * <https://172.24.2.6/> |
| RDP | * [https:// 172.24.2.6/projects/se2024-b1/repository/revisions/master/show](https://172.24.2.6:447/) * <https://172.24.2.6/projects/se2024-b1/repository/android> * <https://172.24.2.6/projects/se2024-b1/repository/c> |

***Tableau 16 : Liens des espaces numériques de travail***

### 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.

N’oubliez pas de vérifier tous les jours la page d’accueil, dans la rubrique [[Annonces](https://172.24.2.6/news)]. Vous pourrez y trouver des informations venant des consultants (ex : salle de formation, horaires, etc.).

### Planning prévisionnel

La version détaillée du planning prévisionnel est à saisir sur Redmine, par le biais de demandes de type « Version ». Ces demandes sont à hiérarchiser pour offrir une lecture du plus global au plus détaillé. Par exemple :

* Spécifications :
  + Cas d'utilisation ;
  + Cas d'utilisation stratégique ;
  + IHM ;
  + Rédaction du dossier ;
  + …

Ces phases servent à structurer le planning prévisionnel du projet. À ce titre, les informations à saisir sont généralement (à adapter selon les situations) :

* Titre et description ;
* Date de début et échéance ;
* Relation de dépendances avec d'autres phases projet (précède, suit, …) ;
* Saisie de temps.

Les activités proprement dites sont suivies par le biais de demandes de type « tâche ». Ces demandes sont nécessairement des sous-demandes de demandes de type « Version ». Les demandes de type « tâche » :

* Ne doivent pas être hiérarchisées par rapport aux membres ;
* Doivent être affectées à une et une seule personne pour laquelle elle représente le travail à faire ;
* Sont dupliquées autant que nécessaire si plusieurs personnes effectuent le même travail.

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.

Le planning prévisionnel est disponible dans la partie [[Roadmap](https://172.24.2.6/projects/se2024-b1/roadmap)] de l’ENTP.

### Suivi du travail

#### Une mise à jour immédiate 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. Avant chacune de ces réunions, un ordre du jour est envoyé à chacun des membres de l'équipe à minima 48h avant le début de la réunion et chacun doit le lire et en prendre connaissance avant la réunion. 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.

#### Synthèse personnelle

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

Voici un exemple de synthèse personnelle :

Prénom NOM

Temps total passé :

Contenu

Rôle dans l’équipe

Spécification : 29h

Conception : 40h

…

Spécifications : 29 h

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tâche** | **Travail effectué** | **Temps passé** | **Part du travail** | **Révisions** |
| #17502 | IHM : définition des écrans Android et MAE | 12h | 30% | 44a584ee |
| #17562 | Relecture des CU | 2h | 70% | 5521fzze |
| #17522 | Mise en cohérence des CU | 6h | 45% | 5zef115z |
| … | … | … | … | … |
| … | … | … | … | … |

***Figure 7 : Exemple de rédaction d’une synthèse personnelle***

Et ainsi de suite pour les activités de conception, de test, de gestion de la qualité, de codage, de gestion de projet (pas à détailler) et de consulting (pas à détailler) ...

## 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** |
| Développement sous Android | * Android studio Electric Eel   + Version : 2022.1.1 |
| Développement C/Linux-Embarqué | * SDK armadeus ESEO + Eclipse * Visual Studio Code   + Version : 1.76.0 |
| Diagramme UML | * PlantUML * Version : 1.2020.02 |
| Documentation du code C et Java | * Doxygen   + Version : 1.9.6 |
| Gestion de test | * SquashTM   + Version : 2.1.1 |
| Gestion de versions | * Git   + Version : 1.8.3.1 |
| Présentation/soutenance de projet | * Microsoft PowerPoint   + Version : 2108 |
| RDP | * RedMine   + Version : 3.14.13.stable |
| Rédaction des documents | * LaTeX   + Version : PI-2.6-1.40.24 * Microsoft Word   + Version : 10374.20040 |
| Simulateur de tableau de bord | * ICSim   + Version : 12/06/2020 |
| Tableur | * Microsoft Excel   + Version : Excel 2021 |
| Visioconférences | * Microsoft Teams   + Version : 77.3.14.99 |

***Tableau 17 : Liste des outils numériques autorisés***

# Contrôle des médias

## Communiquer entre membres internes du projet

Toute communication interne se fera sur les canaux dédiés suivants :

* Courriel :
  + Pour toute communication interne par courriel, les messages devront être envoyés en copie à l'adresse suivante : [proSE\_kereval\_2023@hotmail.com](mailto:proSE_kereval_2023@hotmail.com).
* Facebook Messenger :
  + Sur le groupe de discussion « ProSE » ;
  + Par messages privés entre deux membres de l’équipe projet.
* Microsoft Teams :
  + Dans les channels de l’équipe [[[ProSE]2024\_B1\_KEREVAL](https://teams.microsoft.com/l/team/19%3autfg4MFjpj-s0eXRLF-khfV9_4XqPk52UBFiQl48h9c1%40thread.tacv2/conversations?groupId=d69f44be-7b5a-4831-aa05-ab9eefbb7634&tenantId=4d7ad159-1265-437a-b9f6-2946247d5bf9)] ;
  + Par messages privés entre deux membres de l’équipe projet.
* Discord :
  + Par messages privés entre deux membres de l’équipe projet.
* SMS :
  + Par messages privés entre deux membres de l’équipe projet.

## Gestion des médias, sources, références, copyright

Pour tous les documents utilisés et créés par l'équipe CANvengers, l'utilisation de sources externes au projet est autorisée à la seule condition que chaque membre utilisant ces sources ait pris le soin de vérifier les droits d'utilisation et de diffusion de cette source ainsi que les normes de copyright associées. Dans le cas contraire, la source devra être supprimée et remplacée.

Pour l'utilisation d'image, schéma ou tout autre document non textuel, une légende descriptive est à rajouter en dessous du document inséré.

## Communiquer avec des membres externes au projet

### Contacter les consultants

Les consultants devront être contactés en priorité par courriel. Ce courriel devra être envoyé au moins deux jours (48h, heures ouvrées) avant le rendez-vous prévu.

Les règles relatives au contact des consultants sont disponibles sur le wiki ProSE. Chaque sujet/objet devra être préfixé par la chaîne de caractère suivante "[ProSE] **ID\_EQUIPE** - **TYPE** : Texte supplémentaire si besoin" avec :

* **ID\_EQUIPE** : correspond à l'année de diplôme et le numéro d'équipe soit 2024\_B1.
* **TYPE** peut prendre les valeurs suivantes :
  + **RdV** : Signale une demande de rendez-vous
  + **DR** : Demande de présence à une réunion hebdomadaire
  + **CO** : Demande de consulting
  + **AUTRE** : pour toute autre demande

Ainsi, lorsque le chef d’équipe envoie la convocation à la réunion hebdomadaire du 12 mai (prévue pour une durée d'une heure en salle B116), le sujet/objet de son courriel devra être :

* [ProSE] 2024\_B2 - RdV : Réunion hebdo du 12 mai (1h), 18h B116

Toute prise de rendez-vous de consulting se fait par une demande auprès du CdP, avec autorisation du CdP et reporting auprès du CdP pour garder une trace des consultings effectués (pour permettre *a minima* au CdP de tenir à jour la liste des dépenses de consulting. Tous ces consultings seront reportés et mis à jour dans sur le Redmine avec son compte rendu.

Toute la correspondance avec les consultants doit être archivée à l’adresse suivante : [proSE\_kereval\_2023@hotmail.com](mailto:proSE_kereval_2023@hotmail.com)

### Contacter le client

Le client souhaite être contacté par courrier électronique. Il faut prévenir au moins 48 heures (heures ouvrées) à l'avance le client pour une demande de rendez-vous. Veillez à mettre systématiquement en copie l'enseignant référent de votre équipe, en l’occurrence Camille CONSTANT.

Pour les règles d'écriture des courriels (notamment pour l'objet), veuillez respecter les mêmes conventions que pour les consultants. Ce qui donnerait comme courriel (avec Camille CONSTANT en copie) :

* [ProSE]2024\_B1 : Point d'avancement en distanciel sur Teams le 07/03

Toute la correspondance courriel avec le client doit être archivée à l’adresse suivante :

* [proSE\_kereval\_2023@hotmail.com](mailto:proSE_kereval_2023@hotmail.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).

De plus, pour chaque écriture d'un courriel, le chef de projet demandera à l'un de ses RQT une relecture de ce courriel avant l'envoi au client.

Une règle spécifique de gestion de courriels a été mise en place sur ce compte courriel pour permettre d'archiver simplement et efficacement les échanges avec le client et ainsi garder une traçabilité des informations.

# Contrôle 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 au chapitre tandis que les procédures d’inscription aux formations et de demandes de consulting sont détaillées ci-après.

## S'inscrire à 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.

## 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 courriel avec un préavis de 48h. Il est important de respecter les consignes de rédaction des courriels (le sujet doit respecter un format particulier). Les règles relatives au contact des consultants sont disponibles sur le wiki ProSE. [[Règles pour contacter client et consultants](https://172.24.2.6/projects/documentation-prose/wiki/R%C3%A8gles_pour_contacter_client_et_consultants)].

Pour une demande de consulting (de 30 minutes) de notre équipe sur de la spécification pour le 12 mai à 13h en salle Neumann. Il faudra mettre en objet du courriel :

* [ProSE] 2024\_B1 - CO : Spécification (30 minutes), 13H 12/05 Neuman

Tout consulting effectué par le chef de projet ou l'équipe est enregistré et archivé dans un document spécifique par le chef de projet. Les éléments suivants seront à indiquer : la date, le sujet de formation, nom du formateur, durée ainsi que les coûts de consulting.

# Collecte, maintenance et conservation des archives

## Le Référentiel 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.

### Structuration du RDP

Le RDP est structuré en différents dossiers. Il est impératif de respecter la décomposition en premier niveau du RDP.

* **Premier Dépôt : DOC** (le dépôt « doc » contient l’ensemble des différents dossiers à réaliser pour ProSE, avec notamment le dossier de spécification, le dossier de conception, le PAQL, les dossiers pour les tests mais également un dossier pour la gestion de projet).
  + **aide :**
    - aide\_git : Tout document permettant d’aider à l’utilisation de GIT.
    - aide\_latex : Tout document permettant d’aider à l’utilisation de LaTeX
  + **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.
    - **schemas** : Toutes les sources des dessins ou fichiers UML produits lors de la phase de conception.
  + **gestion\_projet :**
    - **audit** : Tous les comptes-rendus des différents audits.
    - **budget** : Tout ce qui correspond au budget du projet (salaire de chaque mois) **TODO**
    - **client** : Tous les documents en rapport avec le client. **TODO**
    - **planning** : Les plannings du projet (en particulier ceux des audits et des formations)
    - **reunions** : Comptes-rendus des réunions.
  + **qualite :**
    - **PAQL** : Contient tous les documents qualité jugés nécessaires à la production/rédaction du PAQL.
      * **modeles** : Modèle de document et modèles de codes à utiliser pendant le projet.
      * **schema :** Contient les schémas à utiliser pour la production du PAQL
      * **version :** Contient l’historique des versions du PAQL, y compris la dernière version signée
  + **specification :** 
    - **dossier\_latex :** Ce sont les différents fichiers qui constituent le dossier de spécification dont le fichier « dossier\_spec\_inc1.tex » qui correspond au « main » du dossier.
      * **sections :** Contient les différents fichiers qui sont ensuite inclus dans le fichier « dossier\_spec\_inc1.tex ».
    - **ebauches** : Ce sont tous les fichiers de tests utilisés au cours de la constitution des dossiers de spécifications.
    - **figures**: Toutes les figures utilisées dans le dossier de spécifications (logo ESEO).
    - **livrables** : Documents dès qu’ils ont atteint une première fois l'état « diffusable » ainsi que toutes leurs révisions ultérieures.
    - **schemas** : Toutes les sources des dessins ou fichiers UML produits lors de la phase de spécification (contexte physique/logique).
  + **test :** Regroupe tous les artefacts liés à l’activité de test (Cahier de test, Plan Test, Planning).
    - **cahier\_test :** Tous les documents liés au Cahier de test
    - **plan\_test :** Tous les documents liés au Plan de test
    - **planning :** Tous les documents liés au Planning
* **Deuxième Dépôt : C** (le dépôt « c » contient l’ensemble des fichiers qui vont être développés en C et qui seront déployé sur la Raspberry).
  + **exploration :** Va contenir toutes les explorations réaliser lors du développement du programme.
  + **production :** Contiendra tous les fichiers permettant la communication avec l’application Android et le simulateur ICSim ainsi que le banc de test.
* **Troisième Dépôt : Android** (le dépôt « Android » contient l’ensemble des fichiers qui vont être développés en Android et qui seront déployés sur un smartphone pour constituer l’application qui enverra des trames. Il constituera l’IHM que l’on a élaboré).
  + **exploration :** Va contenir toutes les explorations réaliser lors du développement de l’application.
  + **production :** Contiendra tous les fichiers permettant le bon fonctionnement de l’IHM.

### Obtenir un accès au RDP

Pour se connecter au RDP (<https://172.24.2.6/>), les identifiants ESEO doivent être utilisés :

Identifiant : [prenom].[nom]@reseau.eseo.fr

Mot de passe : [mot de passe personnel]

Il faut ensuite suivre et effectuer les imports des trois dépôts Git en suivant la méthodologie présente dans le document : <https://172.24.2.6/attachments/download/4/guide_exercice_git.pdf>

### Nom des fichiers des artefacts

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

Exemple :

* GUI\_buttonConnexionRaspberry (nom artefact, objet, etc.)
* PAQL\_B1\_2024 (nom livrable)

### Déposer 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 annoté d'un sigle de la façon suivante : [SIGLE] : <Commentaire> :

|  |  |
| --- | --- |
| **Sigle** | **Libellé document** |
| CREAT | Lors de la création du document et de l'ajout sur le RDP |
| REV | Apport d’une nouvelle version d’un document déjà présent sur le RDP |
| ATT-REL | Document en attente de relecture |
| ATT-VAL | Document en attente de validation |
| VALIDE | Document validé |
| DEL | Document supprimé |

***Tableau 18 : Liste des sigles du dépôt***

## Gestion des documents papier

Afin de garder toutes traces du travail effectué, les documents papier seront stockés dans une chemise intitulée : Projet ProSE – XX – 2024, chemise conservée par le CdP.

À l’intérieur, il sera retrouvé la même classification que le premier niveau du RDP. Chaque document papier produit devra contenir le nom et prénom de l’auteur du document ainsi que la date de création et date de modification.

Si cela est matériellement possible, l'auteur du document papier pourra faire une copie numérisée du document et la déposer sur le RDP pour en garder une trace supplémentaire.

# 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.

Les auto-formations ou recherchent de connaissances sur des thèmes très spécifiques (fonctionnement interne d’un composant) doivent faire l’objet de la création d’un rapport à poster dans la section Aide de l’ENTP.

# Gestion du risque

La gestion du risque au sein de notre projet va nous permettre de pouvoir anticiper au maximum les risques techniques et humains susceptibles de mettre le projet en péril.

Fréquence de 1 à 5 (1 peu de chance que cela survienne, 5 grandes chances que cela survienne).

Gravité de 1 à 5 (1 faible impact sur le projet, 5 gros impact sur le projet).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Échelle de 1 à 5** | **Échelle de 1 à 5** | **R =**  **F \* G** | **Correction** | | **Mesures correctives** |
| **Fréquence (F)** | **Gravité (G)** | **Risque** | **F revu** | **G revu** |
| **Indisponibilité de l’équipe sur plusieurs séances** | 3 | 2 | 6 | 3 | 1 | * Planning ; * Surfacturation. |
| **Pandémie/**  **confinement** | 2 | 4 | 8 |  |  | * Travaille à distance. |
| **Perte/corruption de données** | 3 | 5 | 15 | 2 | 1 | * Gestion de version ; * Sauvegarde ; * Formation. |
| **Disponibilité/**  **problèmes des cibles** | 3 | 5 | 15 | 2 | 2 | * Cible redondante ; * Sauvegarde/backup ; * Formation de l’équipe projet. |
| **Spécification non à jour (avant écriture tests/conception)** | 4 | 3 | 12 |  |  | * Point régulier avec le client. |
| **Retard dans le projet, planification prévisionnelle trop optimiste** | 5 | 4 | 20 | 3 | 3 | * Suivi de planning ; * Négociation des spécifications (priorisation des fonctionnalités) ; * Point météo ; * Formation procrastination. |
| **Inefficience récurrente de l’équipe** | 3 | 4 | 12 | 2 | 4 | * Team building ; * Accompagnement/ binômage ; * Suivi personnalisé. |
| **Problème technique** | 5 | 4 | 20 | 4 | 3 | * Consulting préventif/anticipé ; * Stand up meeting ; * Switch en fonction des capacités de chacun ; * Collaboration entre équipes ProSE. |

***Tableau 19 : Gestion des risques***

# Outils et configurations

## Réseau informatique

Le réseau informatique de l’ESEO vous permettra 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).

## Git

“Le logiciel git est utilisé notamment par les développeurs du noyau Linux. Il s'agit d'un logiciel de contrôle de version, comme Subversion, CVS, Arch ou encore Darcs, qui utilise des dépôts décentralisés. Il permet ainsi de travailler tout en gardant une trace des modifications apportées successivement, et de retrouver ainsi l'état antérieur de ses données. À titre d'exemple, git est le logiciel utilisé par les Tuteurs pour gérer cette documentation.

Il permet également, comme beaucoup de ses congénères, de travailler à plusieurs sur un même projet. Mais contrairement à CVS ou Subversion, par exemple, Git ne fait pas de différence entre un dépôt principal et les copies locales chez les différents contributeurs.

Ce système permet également une organisation hiérarchisée de gros projets, comme c'est le cas du noyau Linux, en rendant complètement naturelle l'existence de dépôts alternatifs pour chaque sous-projet, avec une granularité de modifications très fine, et à plus grande échelle, une faible granularité du dépôt principal qui reçoit périodiquement les modifications par paquets. “

C.f. [Definition\_Git]

Pour plus de simplicités, une aide GIT a été créée par le RQT placé dans le répertoire [[doc/aide](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/show/aide)] nommé [[guide\_git.txt](https://172.24.2.6/projects/se2024-b1/repository/revisions/master/entry/aide/aide_git.txt)]. Ce guide regroupe les commandes de base à utiliser pour savoir utiliser GIT de manière autonome. Il regroupe plusieurs explications sur la méthode de travail à utiliser pour obtenir un versionnage efficace, ainsi qu’une version rapide des autorisations des documents.

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 de l’incrément 1 (non pris en compte pour la phase de spécification de l’incrément 1) et ne s’intéresse qu’aux codes source, 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 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 (Référents C/Android, RQT ou CdP).

Pour implémenter une nouvelle fonctionnalité/partie du code, il faut créer une nouvelle branche de type feature\_branches nommées feature\_[Nom\_Feature]\_[Nom\_Propriétaire] (avec [Nom\_Feature] le nom de la nouvelle fonctionnalité, et [Nom\_Propriétaire] le nom de la personne responsable de cette branche) 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]\_[Nom\_Propriétaire] (avec [Nom\_Feature] le nom de la fonctionnalité modifiée, et [Nom\_Propriétaire] le nom de la personne responsable de cette branche) pour qu’elle soit partagée.

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

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 CdP ou 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 à la branche master. Ce patch sera réalisé sur une nouvelle branche de type hotfixes nommée hotfix\_[Nom\_Correctif]\_[Nom\_Propriétaire] (avec [Nom\_Correctif] le nom du bug à corriger, et [Nom\_Propriétaire] le nom de la personne responsable de cette branche)).

Ci-dessous se trouve la liste des [Nom\_Propriétaire] utilisables dans le projet :

|  |  |  |
| --- | --- | --- |
| **Nom** | **Prénom** | **[Nom\_Propriétaire]** |
| BÉNARD | Théo | TBEN |
| TRÉMOUREUX | Paul | PTRE |
| MARQUETTE | Gabriel | GMAR |
| LENNE | Camille | CLEN |
| DECLERCK | Elisa | EDEC |
| ROCHER | Thomas | TROC |

***Tableau 20 : Liste des [Nom\_Propriétaire]***

Une image contenant graphique

Description générée automatiquement

***Figure 8 : Architecture des branches des dépôts GIT c et android***

# Glossaire : Définitions, acronymes et abréviations

|  |  |
| --- | --- |
| **Acronyme** | **Définition** |
| AP (Artefact Projet) | Désigne tous les éléments numériques ou analogiques produits lors du développement du projet (documents numériques ou papier, code sources, jeux de test…). |
| AQ (Assurance Qualité) |  |
| AC (Audit Consultatif) | Audits de l’incrément 1 |
| AN (Audit Normatif) | Audits de l’incrément 2 |
| 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 |
| CR (Compte rendu) | Document rédigé par le CdP et envoyé au client après chaque réunion avec ce dernier. Ce document répertorie l’ensemble des éléments qui ont été abordés avec le client lors de la réunion. |
| ENTP (Espace Numérique de Travail du Projet). | <https://172.24.2.6/projects/se2024-b1> |
| ESEO (Ecole Supérieure de l’Electronique de l’Ouest) | Établissement de formation des membres de l’équipe projet |
| ICSim (Instrument Cluster Simulator) | Logiciel de simulation d’un tableau de bord |
| IEEE (Institute of Electrical and Electronics Engineers) | Association professionnelle internationale définissant entre autres des normes dans le domaine informatique et électronique. |
| ODJ (Ordre du Jour) | Document rédigé par le CdP et envoyé au client avant chaque réunion avec ce dernier. Ce document répertorie l’ensemble des éléments qui seront abordés avec le client lors de la réunion. |
| OMG (Object Management Group) | Association professionnelle internationale définissant entre autres des normes dans le domaine informatique. |
| RQT (Responsable Qualité et Test) | Rôle d’un membre de l’équipe responsable du respect des règles dictées par ce PAQL et de l’élaboration des tests validant le cahier des charges |
| ProSE (Projet Système Embarqué) | Cours dans lequel s’inscrit ce projet |
| PAQL (Plan d’Assurance Qualité Logicielle) | Ce document |
| 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. |

***Tableau 21 : Glossaire***

# Validation du document

Signature de tous les membres projets, précédée de la mention « J’ai lu et je m’engage à respecter le présent PAQL pendant toute la durée du projet ProSE » :

|  |  |
| --- | --- |
| **CdP – Théo BÉNARD** | **RQT – Paul TRÉMOUREUX** |
|  |  |
|  |  |
| **Référent Android – Gabriel MARQUETTE** | **Référente C – Élisa DECLERCK** |
|  |  |
|  |  |
| **Développeuse Android – Camille LENNE** | **Développeur C – Thomas ROCHER** |
|  |  |
|  |  |

***Tableau 22 : Signature des membres de l’équipe CANvengers***