

PLAN D'ASSURANCE QUALITÉ LOGICIELLE

VERSION 3.0.0 PFE ÉQUIPE 16 2023/2024

Responsable du document : Paul TRÉMOUREUX

État du document : Validé

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
15/05/2023	Création du document	CANvengers (Paul TRÉMOUREUX)	0.0	1
02/10/2023	Début adaptation du document au projet MNJS	Paul TRÉMOUREUX	0.1	0
04/10/2023	Adaptation du document en Google Doc	Paul TRÉMOUREUX	0.1	1
05/10/2023	Modification du document suite réunion de lancement & réunion équipe	Paul TRÉMOUREUX	0.1	2
06/10/2023	Modification du document suite réunion avec le client	Paul TRÉMOUREUX	0.1	3
06/10/2023	Intégration du travail effectué par les autres membres de l'équipe concernant leurs parties	Paul TRÉMOUREUX	0.1	4
06/10/2023	Gestion style graphique du PAQL	Paul TRÉMOUREUX	0.1	5
13/10/2023	Relecture du PAQL	Marie BORDET	0.2	0
13/10/2023	Correction des erreurs relevées lors de la relecture	Paul TRÉMOUREUX	0.2	1
13/10/2023	Relecture du PAQL	Marie BORDET	0.3	0
13/10/2023	Validation du PAQL	Paul TRÉMOUREUX	1.0	0

08/11/2023	Modification du PAQL suite retours sprint 1	Paul TRÉMOUREUX	1.0	1
09/11/2023	Relecture du PAQL	Marie BORDET	1.1	0
09/11/2023	Validation du document	Paul TRÉMOUREUX	2.0	0
09/11/2023	Modification du PAQL suite retours version 2.0.0	Paul TRÉMOUREUX	2.0	1
10/11/2023	Modification prise en compte JUnit	Paul TRÉMOUREUX	2.0	2
13/11/2023	Modification suite obsolescence de l'utilisation de Goland dans le projet	Paul TRÉMOUREUX	2.0	3
18/11/2023	Ajout commit à partir duquel la dépréciation de Goland est prise en compte & gestion des branche bis et ter	Paul TRÉMOUREUX	2.0	4
19/11/2023	Ajout de la nouvelle structure du Git	Baptiste BELLIER	2.0	5
21/11/2023	Correction de la structure de GitLab	Paul TRÉMOUREUX	2.0	6
22/11/2023	Correction liste des livrables client	Paul TRÉMOUREUX	2.0	7
24/11/2023	Proofread of the document	Auréliе CHAMOULEAU	2.1	0
24/11/2023	Validation du document	Paul TRÉMOUREUX	3.0	0

Tableau 1 : Table des évolutions et validations internes du document

CONTENU

1. But.....	4
2. Gestion.....	6
3. Documentation.....	11
4. Standards, pratiques, conventions et métriques.....	18
5. Revues et Audits.....	21
6. Test.....	22
7. Notification des problèmes et corrections.....	23
8. Outils, Techniques et Méthodologie.....	25
9. Contrôle des médias.....	28
10. Collecte, maintenance et conservation des archives.....	30
11. Gestion du risque.....	34
12. Outils et configurations.....	36
13. Glossaire : Définitions, acronymes et abréviations.....	39
14. Validation du document.....	41

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 « Metro Network Journey Simulator v3 » du groupe PFE_2023_16 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 PFE;
- Définir les outils utilisés, les processus et procédures à suivre par l'équipe projet tant au niveau organisationnel que technique lors du PFE.

Ce document est disponible sur le Référentiel Documentaire Projet (RDP) dans le répertoire [[Qualite/PAQL](#)] sous le nom [[plan_d_assurance_qualite_logicielle_MNJS_QLT_PFE_2023_16](#)].

1.2. PORTÉE

Ce document est destiné :

- À l'équipe projet ;
- Aux enseignants encadrant le projet 2023_16.

Il pourra être consulté par le client sur sa demande.

1.3. 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.

1.4. VUE D'ENSEMBLE

Le contenu de ce PAQL est le suivant :

- 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 ;

- o Ressources ;
 - o Normes, pratiques et conventions ;
 - o Calendriers.
- Activités et tâches de cycle de vie de l'AQL :
 - o Rôle de l'assurance de produit ;
 - o Rôle de l'assurance du processus ;
 - o Assurance sur les activités et les tâches du système de management de la qualité ;
 - o Activités et tâches additionnelles.
- Processus et politiques additionnelles :
 - o Processus de mesures de la qualité ;
 - o Politiques de tests ;
 - o Politique de dérogation et de déviation ;
 - o Politique d'itération des tâches.
- Enregistrements et rapports de l'AQL :
 - o Enregistrements ;
 - o Rapports.

1.5. RÉFÉRENCES

- [[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_2.4_2011](#)] : OMG, Unified Modeling Language, version 2.4.1, <http://www.omg.org/spec/UML/>, 2011.
- [[GNU-GPL_3.0_2007](#)] : GNU General Public License, version 3.0, <http://www.gnu.org/licenses/gpl-3.0.html>, 2007.

2. 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 PFE_2023_16.

2.1. ORGANISATION

2.1.1. PROJET CONCERNÉ

Le projet « Metro Network Journey Simulator v3 » concerne les deux entités suivantes :

- L'ESEO avec :
 - L'équipe PFE_2023_16 de l'ESEO, réalisatrice du projet ;
 - Une équipe d'enseignants encadrants.
- L'entreprise Thales - Ground Transportation Systems (Hong Kong) Limited, commanditaire du projet.

2.1.2. RESSOURCES HUMAINES

2.1.2.1. ÉQUIPE PROJET

Les différents membres de l'équipe projet PFE_2023_16 et leurs informations importantes sont listées dans le tableau ci-dessous :

Rôle	Option	Nom	Prénom	Courriel	Téléphone
Git Maintainer	LD	BELLIER	Baptiste	baptiste.bellier@reseau.eseo.fr	+33 6 41 29 19 16
External Relationship Supervisor	LD	BONAMY	Alexis	alexis.bonamy@reseau.eseo.fr	+33 7 81 52 78 80
Scrum Master	LD	BORDET	Marie	marie.bordet@reseau.eseo.fr	+33 6 83 05 23 61
Product Owner	LD	CHAMOULEAU	Aurélie	aurelie.chamouleau@reseau.eseo.fr	+33 6 74 15 44 20
Quality Supervisor	SE	TRÉMOUREUX	Paul	paul.tremoureux@reseau.eseo.fr	+33 6 37 23 77 07
Tests Supervisor	LD	VAVASSEUR	Benoît	benoit.vasseur@reseau.eseo.fr	+33 7 82 93 88 63

Tableau 2 : Membres de l'équipe projet

En cas d'indisponibilité d'un membre, une autre personne est désignée pour prendre ses responsabilités. La répartition se fait comme suit :

Rôle	Remplaçant
External Relationship Supervisor	Quality Supervisor
Git Maintainer	Product Owner

Product Owner	Scrum Master
Quality Supervisor	Tests Supervisor
Scrum Master	External Relationship Supervisor
Tests Supervisor	Git Maintainer

Tableau 3 : Liste des remplaçant pour chaque rôle

Cas particulier, si une personne et son remplaçant sont absents en même temps, c'est le remplaçant du remplaçant qui prend les responsabilités des deux absents.

2.1.2.2. CLIENT

Le projet de l'équipe PFE_2023_16 est réalisé pour l'entreprise Thales - Ground Transportation Systems (Hong Kong) Limited.

Thales - Ground Transportation Systems (Hong Kong) Limited est un leader mondial dans le domaine des innovations pour la mobilité et le transport. Créée dans le but d'améliorer la sécurité, l'efficacité et la durabilité des transports publics, Thales - Ground Transportation Systems (Hong Kong) Limited s'est spécialisée dans les technologies avant-gardistes et les systèmes à destination des métros, réseaux urbains et autoroutes.

Stéphane DUGUET, directeur de la recherche technologique et de l'innovation à Thales - Ground Transportation Systems (Hong Kong) Limited, a fait appel à l'ESEO pour réaliser la troisième version de ce simulateur lors d'un projet de fin d'étude (PFE). L'équipe PFE_2023_16 s'est proposée pour exécuter cette commande.

Le client est représenté par M. Stéphane DUGUET. L'adresse courriel de ce contact est disposée ci-dessous :

- M. Stéphane DUGUET (Head of Research Technology and Innovation) : stephane.duguet@urbanandmainlines.com.

2.1.2.3. ENSEIGNANTS ENCADRANTS

Si nécessaire, l'équipe projet pourra faire appel aux enseignants encadrants en tant que support technique. Les enseignants encadrants sont :

- M. Olivier CAMP (ENSEIGNANT-CHERCHEUR - DIS - Dept Informatique & Systèmes) : olivier.camp@eseo.fr ;
- Camille CONSTANT (ENSEIGNANT-CHERCHEUR - DIS - Dept Informatique & Systèmes) : camille.constant@eseo.fr.

2.2. TÂCHES DU PROJET

2.2.1. TÂCHES TRANSVERSALES

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

- Documentation [3. Documentation] ;
- Revues et audits [5. Revues et audits] ;
- Inspections internes ;
- Validation et tests ;
- Activités d'amélioration du processus d'AQ.

2.2.1.1. INSPECTIONS INTERNES

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

- Product Owner
 - o Vérification des demandes sur l'ENTP : s'assurer que toutes les demandes du client sont prises en compte.
 - o Vérification du RDP : confirmer que tous les documents sont à jour, complets et conformes aux normes établies.
 - o Organisation des tâches : les user stories et tech stories doivent être bien organisés et priorisés, avec des descriptifs pertinents pour chaque item.
 - o Suivi de l'avancement : surveillance continue de l'avancement des tâches et ajustement si nécessaire.
- Scrum Master
 - o Facilitation : aider à résoudre les blocages et faciliter les discussions entre les membres de l'équipe.
 - o Amélioration continue : travailler constamment à l'amélioration des processus Scrum dans l'équipe.
- Quality Supervisor
 - o Vérification de la convention de nommage : les dépôts sur le RDP et sur le Git doivent suivre une convention de nommage cohérente.
 - o Vérification de la lisibilité du code : le code doit être propre, bien organisé et facile à comprendre.
 - o Analyse du respect du PAQL : revue continue du respect des règles et normes du PAQL.
 - o Vérification de la pertinence des commentaires : les commentaires dans le code doivent être pertinents, utiles et permettre la génération des documentations.
- Tests Supervisor
 - o Exécution des tests : s'assurer que les tests sont effectués de manière rigoureuse et documentés (notamment la rigueur des critères de validation)

- o Rapport de bugs : rapport et suivi des bugs découverts pendant les tests.
 - o Tenue du cahier de tests
- Git Maintainer
 - o Gestion des branches : s'assurer que les branches sont correctement créées, fusionnées et supprimées lorsque nécessaire.
 - o Revue des pull-requests : vérification et approbation des demandes de tirage, en veillant à la qualité du code.
- External Relationship Supervisor
 - o Communication avec le client : tenir le client informé de l'avancement du projet et collecter ses retours et exigences.
 - o Communication avec les EC : tenir les EC informés de l'avancement du projet et collecter leurs retours et exigences.
 - o Gestion des attentes : aider à aligner les attentes du client avec les livrables du projet.
- Tous les membres
 - o Revue collective : participation active aux revues de sprint et aux rétrospectives, en apportant des feedbacks constructifs et en s'engageant à l'amélioration continue.

2.2.2. 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.

Cf : [[plan_de_test_TEST_B1.pdf](#)]

2.2.2.1. 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 Quality Supervisor (QS) de l'équipe projet. Le Scrum Master (SM) et le QS participent aux décisions de modifications. Il incombe au QS 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.

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 (AD), nommés documents "livrables client" sont :

- Cahier des charges ;
- Artefacts de code ;
- Cahier de test ;
- Manuel d'utilisation et manuel d'installation ;
- Guide d'installation des outils utilisés pour le développement
- Plan Assurance Qualité Logicielle (PAQL) ;
- Supports de présentations ;
- Poster ;
- Ordre du jour réunion client ;
- Compte-rendu(s) réunion client.

Les artefacts consultables par les enseignants encadrants, nommés documents "consultables EC", 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.

Les artefacts de code (AC) sont :

- Les codes source ;
- Les codes de test.

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

3.3. RÉFÉRENCE DES DOCUMENTS

La référence d'un document est de la forme suivante : « SIGLE_XX », où SIGLE correspond à l'une des combinaisons de lettres citées ci-dessous et XX correspond au nom de l'équipe projet à savoir PFE_2023_16.

Ce système de référencement ne sera appliqué que pour les livrables "consultables EC". 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_PFE_2023_16_JJ_MM ». Dans ce cas, JJ correspond au jour de la réunion et MM au mois de la réunion.

Référence	Libellé document
QLT_PFE_2023_16	Documents liés à la qualité
SPC_PFE_2023_16	Cahier des Charges
USS_PFE_2023_16	User Stories
DOD_PFE_2023_16	Definition of Done
TST_PFE_2023_16	Document de test
BGT_PFE_2023_16	Bug Tracker
CMR_PFE_2023_16_JJ_MM	Compte rendu d'une réunion client
CMA_PFE_2023_16_JJ_MM	Ordre du jour d'une réunion client
SRR_PFE_2023_16_JJ_MM	Compte rendu d'un Sprint Retro
SRA_PFE_2023_16_JJ_MM	Ordre du jour d'un Sprint Retro
PRS_PFE_2023_16	Présentations
PST_PFE_2023_16	Poster
WOO_PFE_2023_16	Pour tous les autres documents

Tableau 4 : Références pour les artefacts de documentation

Par exemple, pour le cahier des charges produit par l'équipe PFE_2023_16, sa référence sera la suivante : « CDC_PFE_2023_16 ».

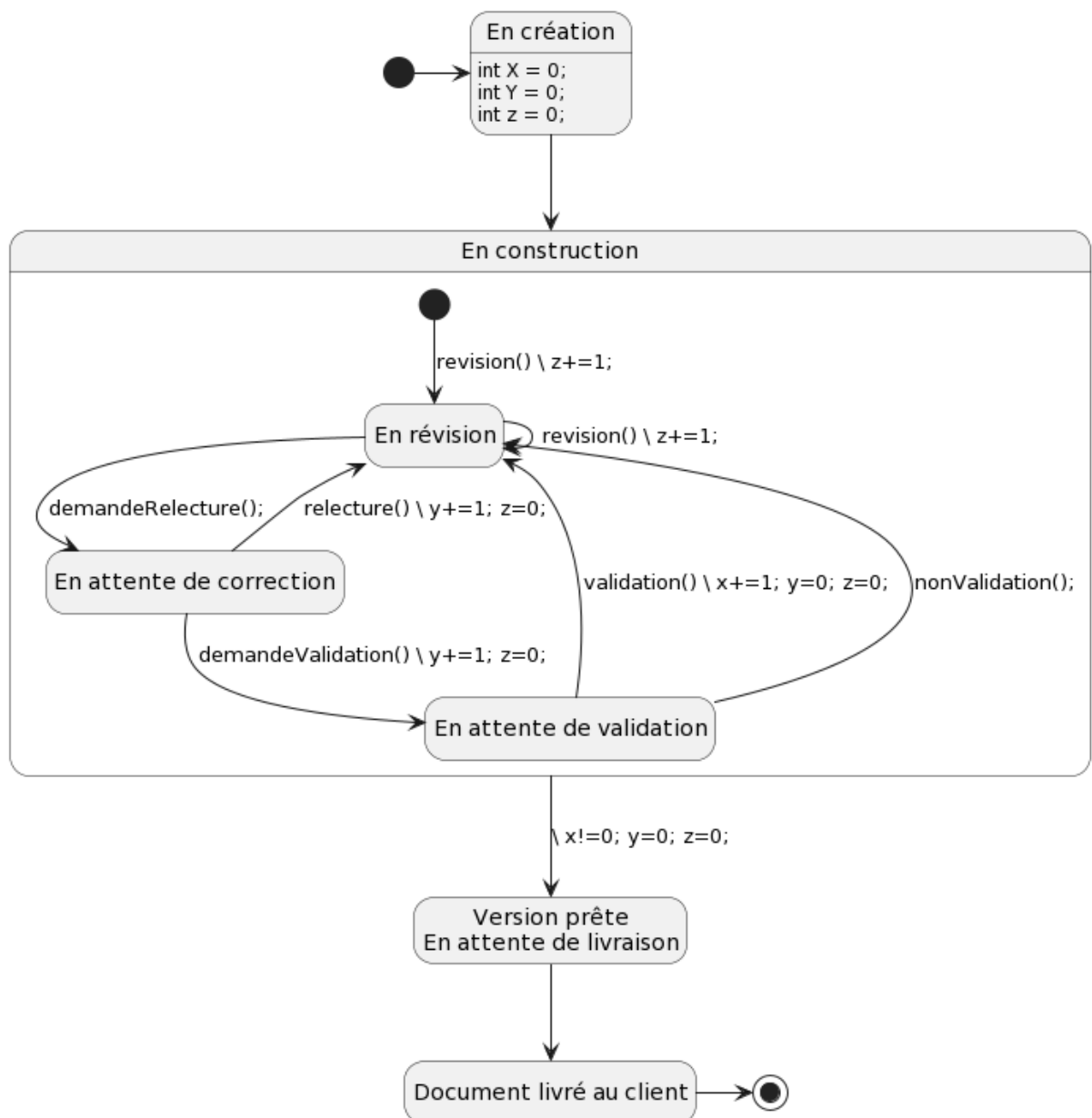
La référence d'un document permet d'identifier la catégorie à laquelle appartient ce document, cette référence figurera dans le document, mais aussi dans le nom de son fichier.

Ainsi, dans l'exemple précédent, le fichier pourra se nommer « cahier_des_charges_CDC_PFE_2023_16 ».

3.4. 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.



© Made by Paul TRÉMOUREUX

Figure 1 : Machine à états du processus de suivi d'un document

3.5. 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 gèrera 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.

3.6. 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.

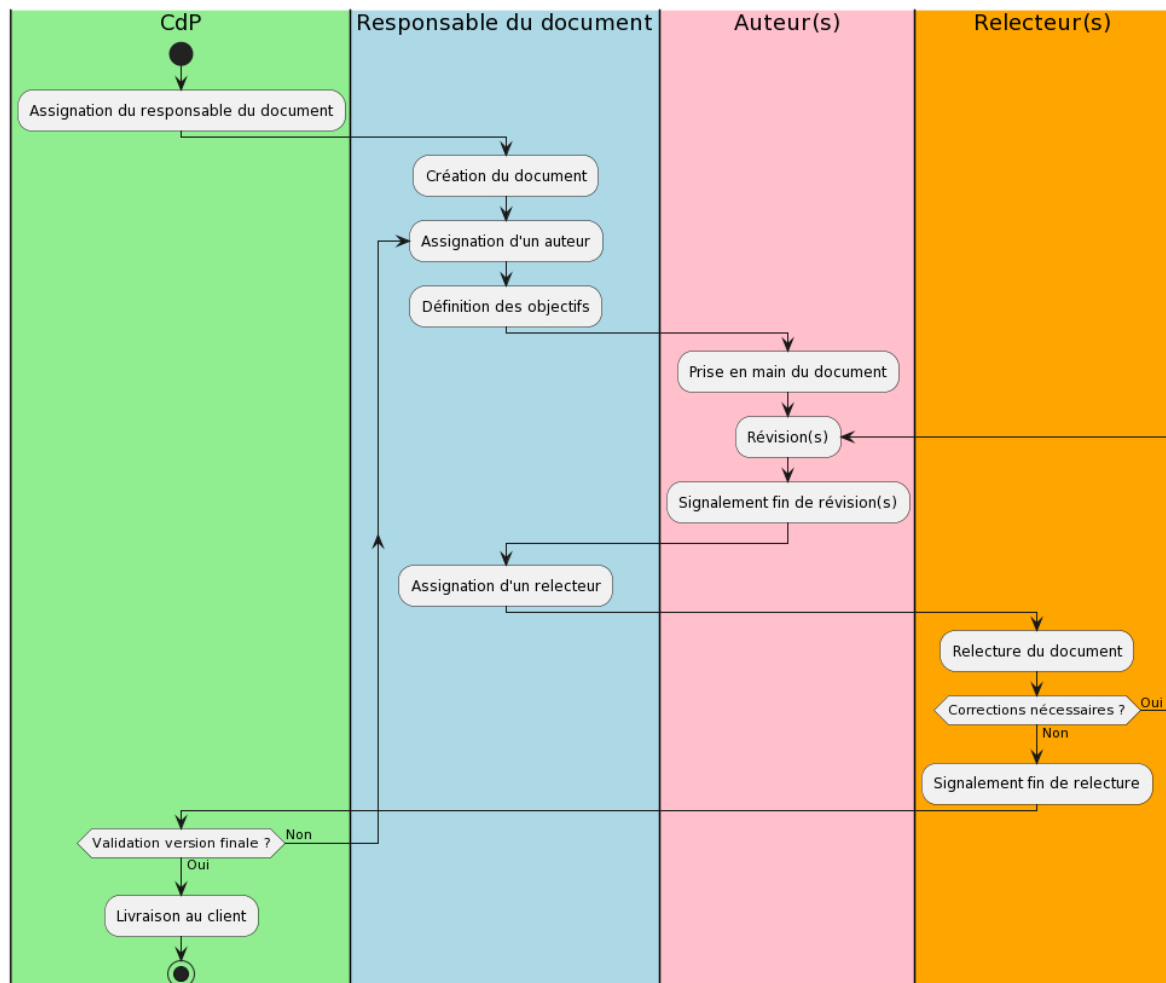
Les décisions concernant un document sont toujours prises à deux. Lorsque le responsable du document est le SM, le QS viendra aider à la prise de décisions.

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 SM. 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 PO 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 SM. 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.



© Made by Paul TRÉMOUREUX

Figure 2 : Diagramme d'activité du processus d'édition d'un document**3.7. 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 ;
- 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.

Les artefacts code ne sont pas soumis à du versionnage.

3.8. FORMAT DES DOCUMENTS

Pour tous les documents "livrables client" et "consultables EC", il faut respecter le modèle de document correspondant lorsque celui-ci existe. Ces modèles sont disponibles sur le RDP, dans le dossier [Qualite](#).

3.8.1. 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/modeles](#). Tous les AD seront créés sous un format de la suite bureautique de Google. Veillez à activer le suivi des modifications pour tous ces fichiers.

3.8.2. ARTEFACT DE CODE

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

- [example.go] ;
- [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) et ce dans la limite des modifications apportées par la version 3 de MNJS. En effet, des parties de codes déjà existantes non modifiées durant le projet pourront ne pas respecter les modèles d'artefact de code. Il est impératif que les codes source soient parfaitement commentés et ce en utilisant le format JavaDoc ou GoDoc pour chaque langage respectif. 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.

3.8.3. RÈGLES DE CODAGE EN LANGAGE GO

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

3.8.4. RÈGLES DE CODAGE EN LANGAGE JAVA

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

3.9. 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.

4. STANDARDS, PRATIQUES, CONVENTIONS ET MÉTRIQUES

4.1. BUT

Cette section décrit les standards, pratiques, conventions et métriques utilisés pour le PFE. 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 QUALITÉS GÉNÉRALES

Description des Exigences qualité	
Liées au produit (par ordre décroissant de priorité)	<ul style="list-style-type: none">• Conformité : le produit livré devra être conforme au cahier des charges 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)	<ul style="list-style-type: none">• Traçabilité.• Conformité (au présent PAQL et normes indiquées).• Simplicité.

Tableau 5 : Liste des exigences qualité du projet

4.3. EXIGENCES QUALITÉS SUR LES ARTEFACTS

Pour tous les artefacts remis au client ou consultés par les enseignants encadrants, 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 EC", 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.

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 commentaires JavaDoc ou GoDoc.

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

Il y aura au minimum les éléments suivants dans les artefacts de code :

Élément	Description
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
License	<p>MIT License</p> <p>Copyright (c) 2023 Équipe PFE_2023_16</p> <p>Permission is hereby granted, free of charge, to any person obtaining a copy</p> <p>of this software and associated documentation files (the "Software"), to deal</p> <p>in the Software without restriction, including without limitation the rights</p> <p>to use, copy, modify, merge, publish, distribute, sublicense, and/or sell</p> <p>copies of the Software, and to permit persons to whom the Software is</p> <p>furnished to do so, subject to the following conditions:</p>

	<p>The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.</p> <p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 6 : Liste des balises d'en-tête des artefacts code

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 la première moitié du projet aux acteurs externes au projet. Un ensemble de détails est présenté sur le campus dans le cours [[E5e-Sog-PFE - Projet de fin d'études](#)].

5.2.2. POSTER

De même que pour la revue de mi-avancement, les précisions ont été faites sur le campus dans le cours [[E5e-Sog-PFE - Projet de fin d'études](#)].

5.3. AUDITS

5.3.1. INSPECTION ET REVUE CROISÉE

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

6. TEST

Un cahier de tests doit être rédigé pour renseigner l'avancement des tests implémentés. L'outil Squash est utilisé afin d'implémenter plus facilement les tests d'intégration et de générer le cahier de tests sous format Word pour être ajouté au GitLab.

La rigueur d'un test est définie par le respect des critères de validation associés. Ces derniers sont déterminés à partir des US.

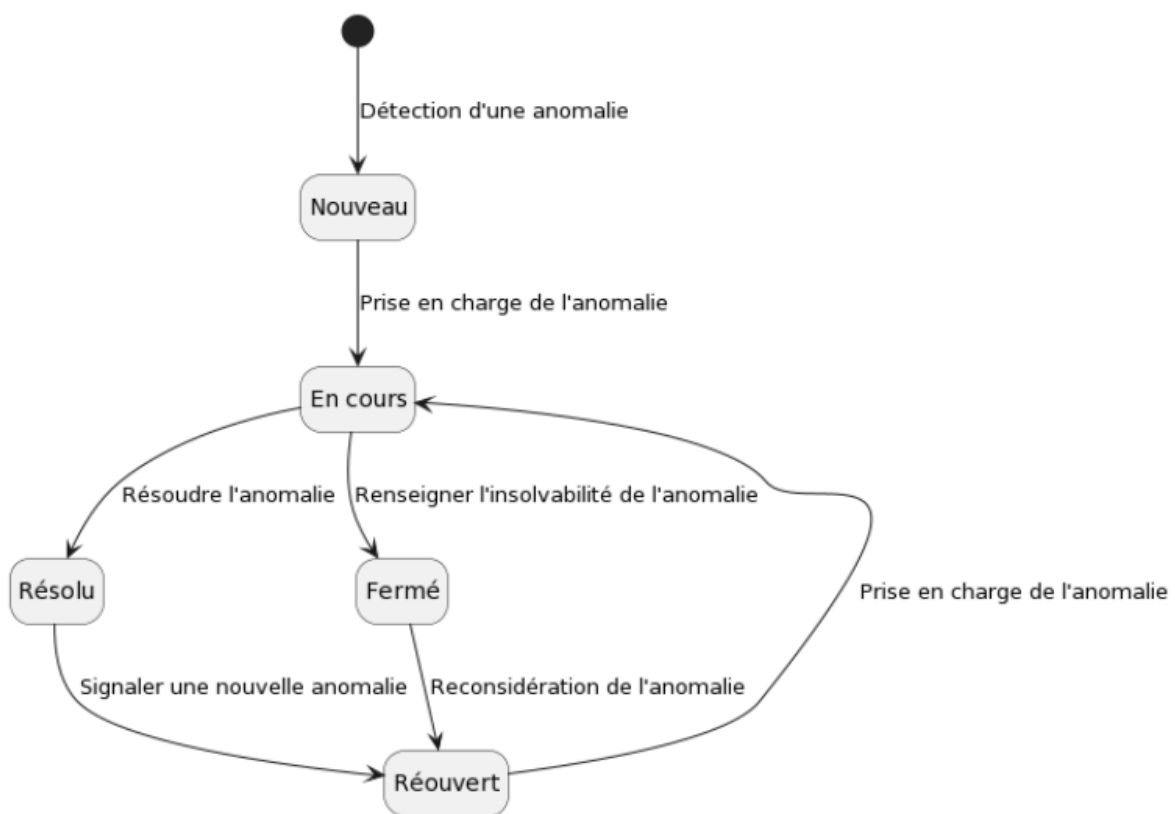
7. NOTIFICATION DES PROBLÈMES ET CORRECTIONS

La notification des problèmes, à toute étape du processus de développement, se fait directement sur le RDP et ce par chacun des membres de l'équipe. Cela se fait par l'ajout du bug dans le document [BT_PFE_2023_16] disponible dans le répertoire [[Tests](#)] du RDP.

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 :



© Made by Paul TRÉMOUREUX

Figure 3 : 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 Scrum Master 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 :

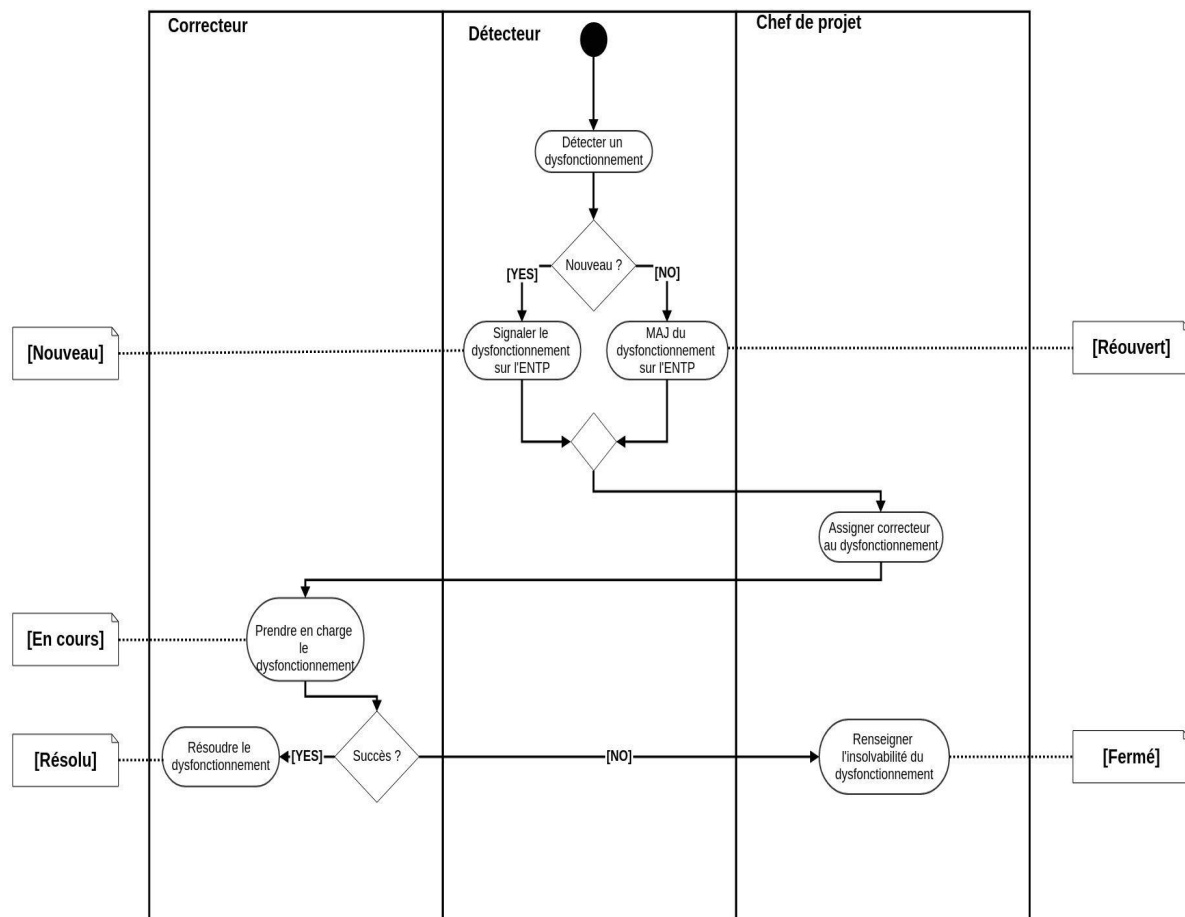


Figure 4 : Diagramme d'activité expliquant le processus de traitement d'un dysfonctionnement

8. OUTILS, TECHNIQUES ET MÉTHODOLOGIE

8.1. L'ESPACE NUMÉRIQUE DE TRAVAIL DU PROJET (ENTP)

L'ENTP correspond à l'ensemble des outils suivants :

- GitLab: un gestionnaire de suivi de projet et également [[dépôt GitLab](#)] appartenant au client ;
- Le RDP (Référentiel Document Projet) : c'est le [[dépôt Google Drive](#)].

L'ENTP est accessible de la même façon à l'intérieur et à l'extérieur de l'ESEO.

Pour GitLab, les identifiants (login et mot de passe), lorsqu'ils sont réclamés , correspondent à ceux donnés par le client.

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

Composant	URL
GitLab	http://hkgit.thales.ai:8081/
RDP	https://drive.google.com/drive/folders/1ZfqX3bhm8v7UAJKCwsTEi6ofCj2Oja4S

Tableau 7 : Liens des espaces numériques de travail

8.1.1. GITLAB

GitLab est un gestionnaire de projet ayant comme fonctionnalités :

- Un contrôle de version ;
- La gestion des utilisateurs ;
- La gestion des différents branches du projet ;
- La gestion de documents ;
- La gestion de demandes ;
- Un historique ;
- L'ajout d'étiquettes personnalisés ;
- L'ajout d'étapes clés ;
- La gestion du temps.

8.1.2. PLANNING PRÉVISIONNEL

Le planning prévisionnel est à saisir sur GitLab par le biais de milestone dans les tickets GitLab.

La méthodologie Scrum est utilisée pour ce projet. Le planning prévisionnel se fait via des User Stories (US) réparties sur quatre Sprints.

Chaque US possède :

- Un nom sous la forme [xxx] - [yy] suivi du nom de l'US
- Un numéro
- Une liste des tâches définies par l'équipe
- La Definition of Done
- Une personne assignée pour réaliser la tâche
- Un milestone correspondant au sprint sur lequel sera réalisé cette US
- Un label correspondant à l'état en cours de cette US

[xxx] correspond à la cible de la tâche réalisée et [yy] correspond au nombre de points attribués à cette US par l'équipe lors du Poker Planning.

[xxx]	Cible
Railway-Editor	Railway Editor
Simulator	Simulateur Go
Dashboard	PowerBi

Tableau 8 : Liste des cibles pour les noms d'US

Dans le cas de Tech Stories, la même méthodologie est appliquée à l'exception du nom, celui-ci sera de la forme "TS_[XXX]_[YY]_<nom de la tech story>" avec [XXX] correspondant à la cible de la tâche réalisée et [yy] correspondant au nombre de points attribués à cette US par l'équipe lors du Poker Planning.

[xxx]	Cible
RED	Railway Editor
SIM	Simulateur Go

Tableau 9 : Liste des cibles pour les noms de Tech Stories

8.1.3. SUIVI DU TRAVAIL

8.1.3.1. UNE MISE À JOUR IMMÉDIATE SUR L'ENTP

À chaque fois qu'un membre effectue un travail, il doit déposer la nouvelle version de ces artefacts projet sur l'ENTP.

De plus, une réunion d'équipe projet de type "Stand-up Meeting" est organisée de façon bi-hebdomadaire.

Ces réunions ont pour but de faire un point sur le travail effectué depuis la dernière réunion d'équipe projet et de tenir à jour l'avancée du projet et ainsi établir les objectifs jusqu'à la prochaine réunion d'équipe projet.

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
Développement Java	<ul style="list-style-type: none"> IntelliJ IDEA <ul style="list-style-type: none"> Version : 2023.2.2
Développement Go	<ul style="list-style-type: none"> GoLand <ul style="list-style-type: none"> Version : 2023.2.2 "Déprécié commit : 2961bf1b4629b4b58ddfa2964e1dbfe3e197f269
Contrôle qualité du code Java	<ul style="list-style-type: none"> CheckStyle-IDEA <ul style="list-style-type: none"> Version : 5.83.0 SonarLint <ul style="list-style-type: none"> Version : 9.0.0.75308
Contrôle qualité du code Go	<ul style="list-style-type: none"> SonarLint <ul style="list-style-type: none"> Version : 9.0.0.75308
Documentation du code Java	<ul style="list-style-type: none"> JavaDoc <ul style="list-style-type: none"> Version : N/A
Documentation du code Go	<ul style="list-style-type: none"> GoDoc <ul style="list-style-type: none"> Version : N/A
Diagramme UML	<ul style="list-style-type: none"> PlantUML <ul style="list-style-type: none"> Version : 1.2020.02
Gestion de test	<ul style="list-style-type: none"> SquashTM <ul style="list-style-type: none"> Version : 2.1.1
Gestion de versions	<ul style="list-style-type: none"> Git <ul style="list-style-type: none"> Version : 1.8.3.1
Présentation/soutenance de projet	<ul style="list-style-type: none"> Google Slides <ul style="list-style-type: none"> Version : N/A
RDP	<ul style="list-style-type: none"> Google Drive <ul style="list-style-type: none"> Version : N/A
Rédaction des documents	<ul style="list-style-type: none"> Google Doc <ul style="list-style-type: none"> Version : N/A
Tableur	<ul style="list-style-type: none"> Google Sheets <ul style="list-style-type: none"> Version : N/A
Visioconférences	<ul style="list-style-type: none"> Microsoft Teams <ul style="list-style-type: none"> Version : 77.3.14.99

Tableau 10 : Liste des outils numériques autorisés

9. CONTRÔLE DES MÉDIAS

9.1. COMMUNIQUER ENTRE MEMBRES INTERNES DE L'ÉQUIPE PROJET

Toute communication interne se fera sur les canaux dédiés suivants :

- Courriel :
 - Pour toute communication interne par courriel, les messages devront être envoyés via les adresses mail ESEO des membres de l'équipe projet (partie 2.1.2.1).
- Facebook Messenger :
 - Sur le groupe de discussion « PFE Metro ».
 - Par messages privés entre deux membres de l'équipe projet.
- Microsoft Teams :
 - Sur l'équipe « PFE Metro Journey Network Simulator  ».
 - 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.

9.2. GESTION DES MÉDIAS, SOURCES, RÉFÉRENCES, COPYRIGHT

Pour tous les documents utilisés et créés par l'équipe PFE_2023_16, 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'images, schéma ou tout autre document non textuel, une légende descriptive est à rajouter en dessous du document inséré.

9.3. COMMUNIQUER AVEC DES PERSONNES EXTERNES À L'ÉQUIPE PROJET

9.3.1. CONTACTER LES ENSEIGNANTS ENCADRANTS

Les enseignants encadrants devront être contactés en priorité par courriel.

Chaque courriel devra avoir comme sujet/objet : "[PFE ESEO] - TITRE - Metro Network Journey Simulator" avec TITRE le sujet/objet du courriel.

Par exemple, lorsqu'un courriel concernant la réunion de lancement est envoyé par un des membres de l'équipe projet, le sujet/objet de son courriel devra être :

- [PFE ESEO] - Réunion de lancement - Metro Network Journey Simulator

Lorsqu'un membre de l'équipe projet contacte un enseignant encadrant par courriel, il doit mettre l'autre enseignant encadrant ainsi que les autres membres de l'équipe projet en copie cachée.

9.3.2. CONTACTER LE CLIENT

Le moyen privilégié pour contacter le client est le courriel. 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 les enseignants encadrants de l'équipe.

Pour les règles d'écriture des courriels (notamment pour l'objet), veuillez respecter les mêmes conventions que pour les enseignants encadrants. Ce qui donnerait comme courriel :

- [PFE ESEO] - CR réunion 29/09 - Metro Network Journey Simulator

Lorsqu'un membre de l'équipe projet contacte le client par courriel, il doit mettre les enseignants encadrants ainsi que les autres membres de l'équipe projet en copie cachée.

Pour garantir une communication optimum entre le client et l'équipe projet, seul le ERS 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 ERS puisse les transmettre au client (en cas d'urgence, contacter le ERS directement).

De plus, pour chaque écriture d'un courriel, le ERS demandera à son SM une relecture de ce courriel avant l'envoi au client.

Le deuxième moyen pour contacter le client est le groupe WhatsApp créé par ce dernier. Celui-ci s'appelle "ESEO PFE Metro Network Journey Simulator v3". Tous les membres de l'équipe ont l'autorisation de discuter avec le client sur ce groupe après concertation avec le ERS.

10. COLLECTE, MAINTENANCE ET CONSERVATION DES ARCHIVES

10.1. L'ENVIRONNEMENT NUMÉRIQUE DE TRAVAIL DU PROJET (ENTP)

Le RDP est le dépôt de tous les artefacts documents du projet. Il est impératif que tous les artefacts documents produits lors du projet y soient stockés sans délai. 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.

10.1.1. STRUCTURATION DE L'ENTP

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

- RDP :
 - o Aides :
 - L'ensemble des artefacts d'aides créés lors du projet.
 - o Client :
 - L'ensemble des artefacts transmis par le client lors du projet.
 - o Reunions :
 - L'ensemble des artefacts de réunion créés lors du projet (compte rendus, ordre du jour, ...).
 - o Presentations :
 - L'ensemble des artefacts de présentation créés lors du projet (réunion de lancement, mi-avancement, poster, ...).
 - o Qualite :
 - L'ensemble des artefacts de qualité créés lors du projet (PAQL, modèles d'AC, ...).
 - o Specifications :
 - L'ensemble des artefacts liés à la description technique du projet créés lors du projet (CDC, DOD, ...).
 - o Tests :
 - L'ensemble des artefacts de test créés lors du projet (cahier de test, ...).
- GitLab :
 - o doc :
 - L'ensemble des artefacts documents du projet.
 - o network-journey-simulator :

- L'ensemble des artefacts de code concernant le simulateur Go.
- o railway-editor :
 - L'ensemble des artefacts de code concernant l'IHM Java.
- o .gitignore :
 - Le fichier qui est utilisé pour spécifier les artefacts et dossiers que Git doit ignorer lors de la gestion de version.
- o .gitlab-ci.yml :
 - Le fichier de configuration GitLab utilisé pour définir les étapes du pipeline d'intégration continue, spécifiant comment les projets GitLab doivent être construits, testés et déployés.
- o pom.xml :
 - Le fichier de configuration essentiel dans un projet Maven, décrivant les dépendances, les plugins, et d'autres éléments nécessaires à la gestion du cycle de vie de construction et à la gestion des dépendances dans un projet Java.

10.1.2. OBTENIR UN ACCÈS AU RDP

Pour se connecter au RDP dédié au projet de l'équipe PFE_2023_16 (<https://drive.google.com/drive/folders/1ZfqX3bhm8v7UAJKCwsTEi6ofCj2Oja4S>), un compte Google doit être utilisé.

10.1.3. NOM DES FICHIERS DES ARTEFACTS DOCUMENTS

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.

Exemples :

- cahier_des_charges_CDC_PFE_2023_16
- plan_d_assurance_qualite_logicielle_MNJS_QLT_PFE_2023_16

10.1.4. NOM DES BRANCHES

L'ensemble des branches créées sur le GitLab doivent respecter la dénomination suivante : "[xxx]_[ii]_[nnnn]".

[xxx] correspond à la cible de la tâche qui sera effectuée sur cette branche suivant le tableau suivant :

[xxx]	Cible
red	Railway Editor

sim	Simulateur Go
pbi	PowerBi
dck	Docker
bug	Bug
mgt	Management

Tableau 11 : Liste des cibles pour les noms de branches

[ii] correspond à l'identifiant de la tâche sur GitLab. Une exception est faite pour les branches bug : dans ce cas [ii] correspond à un identifiant de bug dans le bug tracker.

[nnnn] correspond à la personne étant responsable de la branche (et donc de la tâche) suivant le tableau suivant :

[nnnn]	Nom	Prénom
bbel	BELLIER	Baptiste
abon	BONAMY	Alexis
mbor	BORDET	Marie
acha	CHAMOULEAU	Aurélie
ptre	TRÉMOUREUX	Paul
bvav	VAVASSEUR	Benoît

Tableau 12 : Liste des abréviations de noms des membres pour les noms de branches

10.1.5. DÉPOSER UN ARTEFACT SUR LE GIT

Pour assurer une traçabilité de qualité, certaines règles sont à prendre en compte lors du dépôt d'un quelconque document sur le Git. 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.

Les dépôts seront effectués sur les logiciels/outils de développement Java et Go respectifs. Le commentaire de commit sera de la forme suivante :

"[ii] [SIGLE] [X.Y.Z]

<Commentaire>"

[ii] correspond à l'identifiant de la tâche sur GitLab. Une exception est faite pour les branches bug : dans ce cas [ii] correspond à un identifiant de bug dans le bug tracker.

[X.Y.Z] correspond à la version de la branche. L'incrémentation de cette version répond au processus de versionning décrit dans la partie 3.4.

[SIGLE] correspond à un des sigles de libellé de tâche décrit dans le tableau suivant.

[SIGLE]	Libellé tâche
DEV	Lors d'un travail de code effectué sur un artefact déjà existant ou non.
DOC	Lors d'un travail uniquement porté sur la documentation d'un artefact.
TST	Lors d'un travail de code de test effectué sur un artefact déjà existant ou non.
QLT	Lors d'un travail de contrôle qualité effectué sur un artefact.
VAL	Lors d'un travail de validation effectué sur un artefact.
DEL	Lors d'un travail de suppression effectué sur un artefact.

Tableau 13 : Liste des sigles lors d'un commit

Plusieurs libellés peuvent être utilisés en même temps. Par exemple, pour un commit d'un travail de développement et de test le nom du commit pourra être le suivant : [42] [DEV] [TST] [1.2.5]

Suite à un changement de structure ne permettant pas de retravailler sur une branche existante, la création d'une branche se basant sur la nouvelle structure est possible. Dans ce cas, la nouvelle branche sera nommée comme l'ancienne avec l'ajout de la mention " [bis]" (voir " [ter]") à la fin du nom.

Lors d'un commit de relecture ou de validation, comme aucun artefact n'est modifié sur la branche, un commit de type "AMEND" sera fait. On modifiera alors le nom de l' "AMEND" pour qu'il respecte les règles énoncées ci-dessus.

10.1.6. NOM DES FICHIERS DES ARTEFACTS CODES

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.

Exemples :

- ActionManager.java
- stationModel.go
- stationModel_test.go

10.2. GESTION DES DOCUMENTS PAPIER

Afin de garder toutes traces du travail effectué, les documents papier seront stockés dans une chemise intitulée : PFE_2023_16, chemise conservée par le SM.

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, le nom du document ainsi que les dates de création et des différentes modifications.

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.

11. 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	<ul style="list-style-type: none"> Planning ; Désignation des remplaçants.
Pandémie/ confinement	2	4	8			<ul style="list-style-type: none"> Travail à distance.
Perte/corruption de données	3	5	15	2	1	<ul style="list-style-type: none"> Gestion de version ; Sauvegarde ; Formation.
Disponibilité/ problèmes des cibles	3	5	15	2	2	<ul style="list-style-type: none"> Cible redondante ; Sauvegarde/back up ; Formation de l'équipe projet.
Cahier des charges non à jour (avant écriture tests/conception)	4	3	12			<ul style="list-style-type: none"> Point régulier avec le client.
Retard dans le projet, planification prévisionnelle trop optimiste	5	4	20	3	3	<ul style="list-style-type: none"> Suivi de planning ; Négociation des spécifications (priorisation des fonctionnalités) ; Formation procrastination.
Inefficacité récurrente de l'équipe	3	4	12	2	4	<ul style="list-style-type: none"> Team building ; Accompagnement / binôme ; Suivi personnalisé.

Problème technique	5	4	20	4	3	<ul style="list-style-type: none"> • Consulting préventif/anticipé ; • Stand up meeting ; • Switch en fonction des capacités de chacun ; • Collaboration entre équipes PFE.
Indisponibilité de GitLab	1	5	5	1	3	<ul style="list-style-type: none"> • Sauvegardes hebdomadaires.

Tableau 14 : Gestion des risques

12. OUTILS ET CONFIGURATIONS

12.1. 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 du projet sont tenus de respecter la charte informatique ESEO (déjà signée lors de l'entrée à l'ESEO).

12.2. 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 simplicité, une aide Git a été créée par le GM placé dans le répertoire [\[Aides\]](#) nommé [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 s'intéresse qu'aux codes source, soit les dépôts avec les extensions .java et .go 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 doit être considérée comme une branche principale et sécurisée. Le Git Maintainer (GM) est la seule personne autorisée à y déployer des documents ;
- Une branche de production (ou branche "dev") est accessible à tous les membres de l'équipe. Cette branche permet le développement et peut être subdivisée en plusieurs sous-branches. Il s'agit d'une branche sur laquelle seront déposées uniquement les modifications validées par les différents responsables (GM et/ou PO).

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 dev ?

Une fois que le code en cours a été validé, testé et jugé conforme dans la branche (version en X.Y.Z avec X != 0, Y = 0 et Z = 0), il est possible de demander au GM de le transférer sur la branche "dev".

Comment porter le code sur la branche dev ?

Lors d'un merge request, le titre donné sera automatiquement généré par gitlab utilisant le nom de la branche. Si la branche à merge a déjà été merge sur dev dans une version précédente, alors il faut ajouter " [X]" à la fin du nom généré, avec "X" la Xème request affectuée pour cette branche. Par exemple, si l'on souhaite merge pour la quatrième fois la branche sim_82_ptre, le nom du merge request sera : Sim 82 ptre [4].

Quand porter le code sur la branche master ?

Une fois que le code en cours a été validé, testé et jugé conforme dans la branche "dev", il est possible de demander au GM de le transférer sur la branche "master".

Comment porter le code sur la branche master ?

Lors d'un merge request, le titre donné doit être le suivant : "master_[X]"

[X] correspond au sprint correspondant au merge request (1 à 4). Pour le tout premier merge, "master_0" sera utilisé.

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_[iii]_[nnnn]".

[iii] correspond à l'identifiant de la tâche sur GitLab.

[nnnn] correspond à la personne étant responsable de la branche (et donc de la tâche) suivant le tableau suivant :

[nnnn]	Nom	Prénom
bbel	BELLIER	Baptiste
abon	BONAMY	Alexis
mbor	BORDET	Marie
acha	CHAMOULEAU	Aurélie
ptre	TRÉMOUREUX	Paul
bvav	VAVASSEUR	Benoît

Tableau 15 : Liste des abréviations de noms des membres pour les noms de hotfix

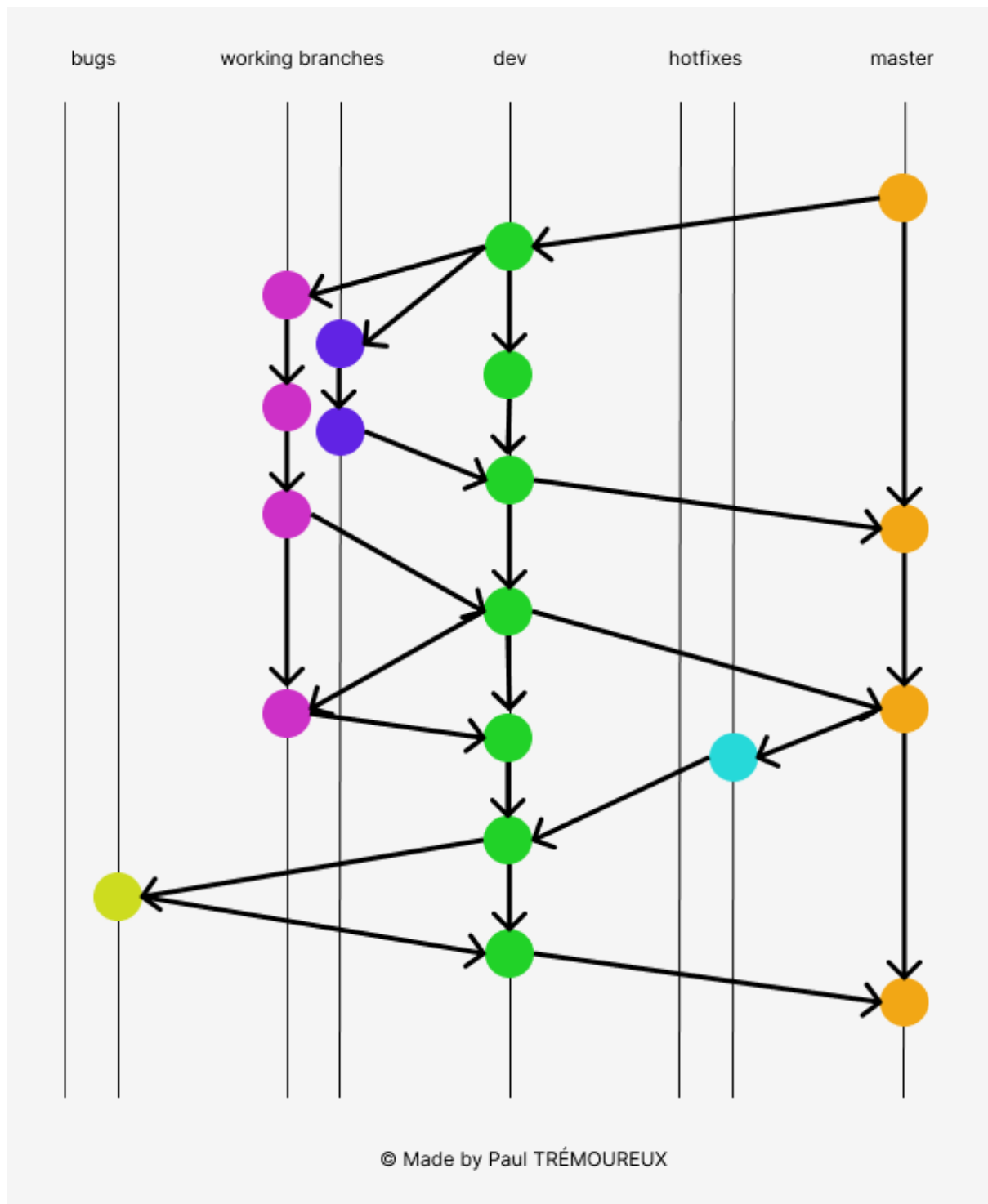


Figure 5 : Architecture des branches des dépôts Git .c et .android

13. GLOSSAIRE : DÉFINITIONS, ACRONYMES ET ABRÉVIATIONS

Acronyme	Définition
AC (Artefact de code)	Désigne tous les éléments de code produits lors du développement. Ces artefacts seront déposés sur le Git.
AD (Artefact de documentation)	Désigne tous les éléments de documentation produits lors du développement. Ces artefacts seront déposés sur le RDP.
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...).
CR (Compte rendu)	Document rédigé par le ERS 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.
CDC (Cahier des Charges)	Document rédigé par l'équipe projet définissant l'ensemble des objectifs du projet ainsi que l'ensemble des fonctionnalités à développer ou corriger lors du projet.
DOD (Definition of Done)	Document rédigé par le PO en accord avec le SM définissant l'ensemble des étapes à réaliser pour compléter l'US/la TS.
EC (Enseignants encadrants)	Désigne les enseignants étant en charge du suivi du projet.
ENTP (Espace Numérique de Travail du Projet).	Ensemble des outils/dépôts qui seront utilisés lors du projet.
ERS (External Relationship Supervisor)	Rôle d'un membre de l'équipe responsable de la relation client.
ESEO (Ecole Supérieure de l'Electronique de l'Ouest)	Établissement de formation des membres de l'équipe projet
GM (Git Maintainer)	Rôle d'un membre de l'équipe responsable du dépôt GitLab.
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 ERS 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.
PFE (Projet de Fin d'Étude)	Cours dans lequel s'inscrit ce projet
QS (Quality Supervisor)	Rôle d'un membre de l'équipe responsable du respect des règles dictées par ce PAQL.
PAQL (Plan d'Assurance Qualité Logicielle)	Ce document
PO (Product Owner)	Rôle d'un membre de l'équipe responsable de la partie technique du projet.
RDP (Référentiel Documentaire Projet)	Dépôt de tous les artefacts document du projet.
SM (Scrum Master)	Rôle d'un membre de l'équipe responsable de la partie management du projet.
TS (Tests Supervisor)	Rôle d'un membre de l'équipe responsable de l'élaboration des tests validant le cahier des charges.
UML (Unified Modeling Language)	Notation graphique normalisée définie par l'OMG et utilisée en génie logiciel.
US (User Story)	Description simple d'un besoin ou d'une attente exprimée par un utilisateur, utilisée pour déterminer les fonctionnalités à développer.

Tableau 16 : Glossaire

14. 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 PFE » :

Baptiste BELLIER	Alexis BONAMY
Marie BORDET	Aurélie CHAMOULEAU
Paul TRÉMOUREUX	Benoît VAVASSEUR

Tableau 17 : Signature des membres de l'équipe PFE_2023_16