

Plan Assurance Qualité

Acteurs :

Auteurs	Approbateurs	Validation
Jérémy VAZ BORGES Quentin CHEYNET	Sébastien TROUSSE	Olivier MICHALON
Rédigé le : 24/10/2013	Approuvé le :	Validé le :

Diffusion :

Diffusion	Externe
À :	Olivier MICHALON
Copie à :	Jérémy VAZ BORGES Quentin CHEYNET Sébastien TROUSSE Jean-Philippe HUGUET Guillaume BROC Clément CARLES

Document de référence :

Libellé	Document

Historique :

N° Version	Auteurs	Approbateurs	Date	Historique des évolutions
V1.0	Jérémy VAZ BORGES Quentin CHEYNET	Sébastien TROUSSE	24/10/2013	Création du document

Plan Assurance Qualité

Objet et caractéristiques du Plan Assurance Qualité

Ce document présente le Plan Assurance Qualité établi pour le Projet 3, dont la responsabilité de son application est assurée par Quentin CHEYNET et Jérémy VAZ BORGES. Il a pour but de définir un ensemble de règles et de dispositions qui sera suivi tout au long de ce projet afin d'en assurer le bon déroulement.

Terminologie

- PAQ: Plan Assurance Qualité
- DS: Diagramme Séquences
- CU: Cas d'Utilisations
- DC : Diagramme de Classes
- DI: Diagramme d'Interactions
- CR: Compte Rendu
- DP: Directeur de Projet
- IT: Itération

Conduite de projet

Planification du projet :

La distribution des tâches se fait via les fiches d'itération, qui sont créées et envoyées au DP avant les réunions avec ce dernier.

Ces fiches d'itération sont faites par un ou deux membres du groupe, et la distribution du travail ne se fait pas sans l'accord des personnes concernées. Il n'y a pas de spécifications particulières au sein du groupe, tous ses membres peuvent travailler sur n'importe quelles parties.

Suivi statique :

Deux réunions ont lieu au début de chaque mercredi, l'une avec le client, l'autre avec le DP. D'autres réunions, entre les membres du groupe C, peuvent être planifiées spontanément.

Démarche de développement du Système d'Information

- Conventions de nommage des fichiers sources : nomdufichier.extension

Convention des extensions

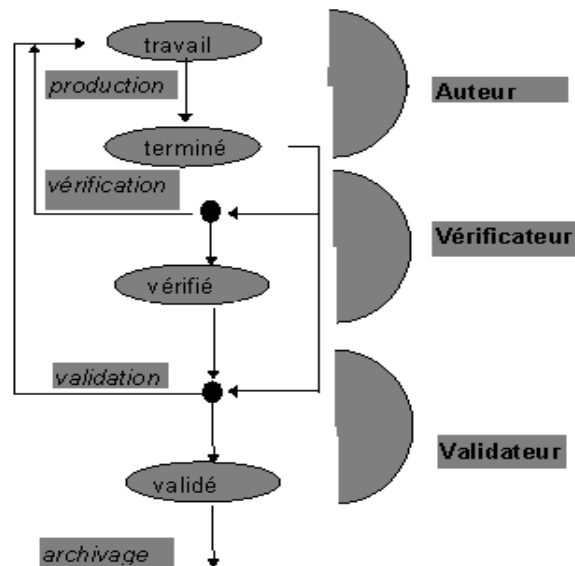
- .cpp et .h:
 - NomDeClasse
 - nomDeFonction() ou nom_de_fonction()
 - nomDeVariable ou nom_de_variable
 - Constructeur (type attribut) {this->attribut = attribut}
 - indentation : 4 caractères
 - l'importation des bibliothèques se fait avant celle des headers
- .h :
 - les attributs sont définis avant les méthodes
 - les méthodes inline sont définies à la fin

Gestion de la documentation

Responsabilités :

- Chargé de la gestion de la documentation :
 - Définition et respect des règles d'identification et de structuration
 - Contrôle de la cohérence et de l'homogénéité dans la gestion et les extensions de la documentation du projet
- Producteurs :
 - Contenu du document
 - Diffusion vers les destinataires identifiés
 - Mise à jour de l'historique du document
 - Sauvegarde du document
- Responsables de la vérification :
 - Relecture pour commentaires et enrichissements éventuels
 - Contrôle du contenu attendu du document, de la présentation et de la structuration du document
- Responsables de la validation :
 - Évaluation du document

Cycle de vie et états des documents :



Identification de la documentation :

Tout document texte doit être identifié de la manière suivante :

[PROJET3][GrpC]nomdocument_numversion

- **nomdocument** : Nom du document ou sous document produit.
- **numversion** : Numéro de la version sous la forme **vn°it.n°mod** où n°it est le numéro d'itération en cours et n°mod est le nombre de modification subi par le document depuis sa création.

Exemple : [PROJET3][GrpC]PAQ_v1.0.pdf pour le sous-document traitant la partie Plan Assurance Qualité de l'itération 1 et ayant subi 0 modification depuis la création du document.

Présentation et structure de la documentation :

Tout document doit comporter la cartouche type composée :

- Nom du groupe
- Logo de l'université
- Titre du document
- Liste des acteurs : Auteurs, Approbateurs, Validateurs
- Liste de diffusion
- Liste des documents de référence
- L'historique

Gestion des versions révisions :

L'auteur de la modification est responsable :

- Du respect du cycle de vérification et validation
- De la modification du numéro de version
- De la mise à jour de l'historique du document
- De la diffusion de la nouvelle version
- De la sauvegarde des modifications

Outils de production de la documentation :

Tout document est produit avec les logiciels MS Office, LibreOffice ou OpenOffice

Sauvegarde et archivage :

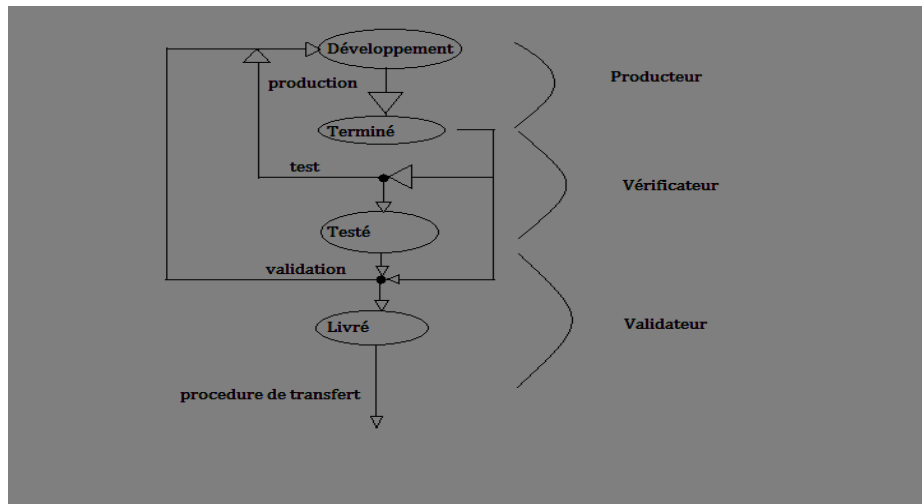
Tout document doit être archivé sur GitHub.

Gestion de la configuration logicielle

Responsabilités :

- Producteurs :
 - Contenu des composants logiciels
 - Sauvegarde des composants
- Responsables de la vérification :
 - Relecture pour enrichissements éventuels
 - Contrôle du bon respect des règles de programmation et de constitution des identifiants
 - Réalisation de tests, afin de contrôler le bon fonctionnement des composants
- Responsables de la validation :
 - Évaluation des composants logiciels

Cycle de vie et états des composants:



Règles de documentation du code :

- En entête de classe ou de méthode :

- Date de création ou modification
- Auteur
- Historique des modifications

Outils de gestion de configuration :

- Langage de programmation : C++
- Bibliothèques : Qt 5.1.1
- Environnement de développement intégré : QtCreator
- Documentation : Doxygen

Sauvegarde et archivage :

Tout composant logiciel doit être archivé sur GitHub.

Gestion des modifications

Utilisation du gestionnaire de versions de GitHub.