

# CCI-Appro

## 1. Contexte

En 2016, sur la base d'un cahier des clauses transmis par les Services Généraux du Pôle Formation de la CCI, trois étudiants SIO 2015/17 ont développé un outil de gestion des demandes en matériel bureautique. L'outil mis effectivement en production en mai 2017 diffère du cahier des clauses fourni, car des évolutions ont été directement intégrées en cours de développement.

Les contraintes de temps n'ont pas permis un développement réellement structuré de l'application, réalisé entre autres sans adopter une approche objet. Bien que mis en production, le code existant est donc plutôt à considérer comme une maquette fonctionnelle. En 2017, les Services Généraux ont rédigé un cahier des charges pour des évolutions et corrections de l'application installée. Ces éléments devront être intégrés dans la nouvelle version de l'application.

## 2. Mission

Durant leur 2<sup>e</sup> année, l'équipe des étudiants SIO 2019-21 doit se comporter pour cette application comme un prestataire de service interne à CCI Campus. Le but du travail est donc de rationaliser le projet, avec une démarche professionnelle :

- organisation du travail en utilisant un outil de gestion de projet,
- production du code dans le cadre strict du framework Laravel,
- rationalisation de la mise en production,
- rédaction des différentes documentations,
- formalisation de la maintenance corrective et évolutive,
- formation des administrateurs.

Ce travail sera réalisé par les étudiants durant leur PPE3, soit durant le premier semestre de leur 2<sup>e</sup> année. Cependant ils pourront être sollicités pour une éventuelle maintenance corrective, si une situation réellement bloquante devait se produire durant le second semestre.

La mise en production effective fait partie du projet. Ainsi un processus de mise en production de l'application doit être prévu, en intégrant la reprise des données de l'application précédente. La nouvelle application est une version 2, l'ancienne application prend alors le numéro 1. L'organisation dans GitLab doit être modifiée pour prendre en compte ce changement de version, pour le code source et pour la documentation, mais aussi pour distinguer les versions de production et de développement.

### 3. Documents

Un ensemble de documents est mis à disposition de l'équipe travaillant sur le projet. Ces documents sont conservés dans le projet CCI\_Appro de GitLab :

- cahier des clauses originel daté du 6 novembre 2016,
- dossier de développement de la première version de l'application,
- annonce du déploiement de la solution,
- demandes complémentaires au 16 février 2017,
- nouveau cahier des charges daté du 4 septembre 2017,
- code source de la première version de l'application.

Dans GitLab, ces documents sont répartis dans trois répertoires : Cahier de charges, CCIAppro et Documentation.

### 4. Compléments

La méthode de travail tend à se placer dans des conditions professionnelles. Un certain nombre de fondamentaux sont donc à respecter : démarche de gestion de projet, règles de développement, attention particulière à la documentation.

#### 4.1. Gestion de projet

Une démarche de gestion de projets implique la définition de tâches, leur attribution à une personne, la définition d'échéances, un plan de charge, des états d'avancement, une gestion du temps, un travail d'équipe... GitLab propose un outil de gestion de projet : les tickets et jalons devront donc être utilisés pour le projet, et renseignés au fur et à mesure de son avancement. Le travail de planification doit évidemment débuter lors de la première séance. De plus, chaque démarche d'autoformation individuelle devra être notée en tant que tâche dans GitLab.

Le but des PPE est d'aboutir sur les épreuves E4 et E6 du BTS, dans les meilleures conditions possible. L'évaluation individuelle des projets sera réalisée selon plusieurs critères :

- l'implication dans le travail d'équipe,
- le partage de connaissances,
- l'organisation personnelle,
- l'effort d'autoformation,
- le respect des délais,
- la qualité du résultat obtenu.

#### 4.2. Règles de développement

Dans le cadre d'une approche d'équipe, un développement raisonnable ne peut se concevoir sans règles de développement. Dans le cadre des PPE :

- le code source doit être versionné dans Gitlab,
- les règles de codage à respecter sont celles du PHP-FIG (Framework Interoperability Group), <https://www.php-fig.org>

### **4.3. Documentation**

La documentation constitue un élément vital d'un projet, trop souvent négligé. Un formalisme doit être défini, avec un modèle de document comportant des données de base : auteur, date de rédaction, date de mise à jour, titre, pagination. Comme pour le code, une gestion des versions est indispensable, car les modifications de la documentation sont à prévoir nécessairement en parallèle avec celles des fonctionnalités et donc du code.

La documentation se subdivise en deux catégories : la documentation technique et la documentation d'utilisation, elle-même subdivisée en fonction du niveau : administrateur, utilisateur... Chacune de ces catégories se destine à un autre public. Une réflexion doit être menée pour la rendre accessible de manière appropriée pour chaque destinataire. Ce moyen n'est pas nécessairement le même pour chacun d'eux.

#### **4.3.1. Documentation technique**

La documentation technique est destinée aux développeurs et mainteneurs de l'application. Dans le cadre des PPE, les rédacteurs des cahiers des charges ont aussi accès au développement. Le but de la documentation technique est de permettre de comprendre le code, de faciliter les corrections, les évolutions... Pour les PPE, les documents seront rédigés avec le formalisme d'UML2. La documentation technique comportera au minimum les documents suivants :

- le résumé des normes de codage,
- les prérequis d'installation (logiciels et technologies utilisées),
- la procédure d'installation et de mise à jour,
- le diagramme de contexte statique,
- les diagrammes de cas d'utilisation,
- les descriptions textuelles des cas d'utilisation,
- le diagrammes de classes,
- le modèle entité-associations pour les bases de données.

Selon leur utilité pour la compréhension du projet, les diagrammes suivants pourront aussi être intégrés :

- un ou plusieurs diagrammes de séquence système,
- un ou plusieurs diagrammes d'activité,
- un ou plusieurs diagrammes de packages.

#### **4.3.2. Documentation utilisateur**

Les utilisateurs de l'application sont à regrouper en fonction de leur responsabilité : administrateur, administrateur réseau et utilisateur final. La documentation utilisateur doit toujours s'adapter à son lecteur et être facilement accessible. Ainsi les administrateurs n'auront pas les mêmes besoins que les utilisateurs finaux. Une réflexion pourra donc être menée avec un choix entre une documentation en ligne (grâce à l'outil de wiki de GitLab) ou un document au format PDF. Dans tous les cas, la documentation utilisateur devra comporter :

- des consignes de configuration, d'installation ou de mise à jour,
- un manuel d'administration,
- un manuel d'utilisation de l'outil de suivi de bugs et de demandes d'évolutions,
- un manuel utilisateur.