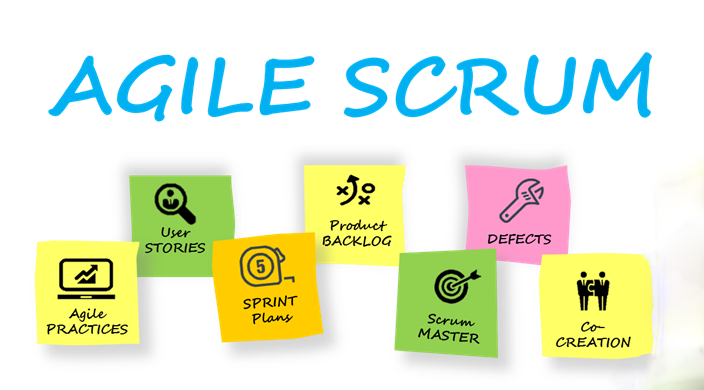
P\_042-Gestion de Projet 2



Stocchetti Jonathan – Dulex Samuel – Rodrigues Alves Dylan- Schaefer Jérémy / Grp2b

Sébeillion 12 - ETML

7 Semaines

Alain Girardet

Table des matières

[1 Spécifications 3](#_Toc478997760)

[1.1 Titre 3](#_Toc478997761)

[1.2 Sujet 3](#_Toc478997762)

[1.3 Description 3](#_Toc478997763)

[1.4 Matériel et logiciels à disposition 3](#_Toc478997764)

[1.5 Prérequis 3](#_Toc478997765)

[2 Planification Initiale 3](#_Toc478997766)

[3 Réalisation 3](#_Toc478997767)

[3.1 Dossier de Réalisation 3](#_Toc478997768)

[3.2 Modifications 3](#_Toc478997769)

[4 Tests 3](#_Toc478997770)

[4.1 Dossier des tests 3](#_Toc478997771)

[5 Conclusion 3](#_Toc478997772)

[5.1 Bilan des fonctionnalités demandées 3](#_Toc478997773)

[5.2 Bilan de la planification 3](#_Toc478997774)

[5.3 Bilan personnel 3](#_Toc478997775)

[6 Divers 3](#_Toc478997776)

[6.1 Journal de travail 3](#_Toc478997777)

[6.2 Webographie 3](#_Toc478997778)

[7 Annexes 3](#_Toc478997779)

# Spécifications

## Titre

Application pour la gestion d'un catalogue de formations, et des formations

## Sujet

Une institution possède un département de formations internes et externes. La gestion se faisait, jusqu'à maintenant, avec des classeurs Excel et des feuilles de calculs, qui permettaient de …

* Gérer la liste des formations (titre, description, date, durée, formateurs, prix, etc.)
* Gérer la liste des formateurs (nom, prénom, adresse, email, tél, titres obtenus, etc.)
* Gérer la liste des participants-étudiants (nom, prénom, adresse, etc.)
* Gérer les inscriptions aux formations (participants, formations, dates, etc.)
* Gérer les résultats des enquêtes de satisfaction pour chacune des formations

De nombreuses erreurs de saisies ont démontré que ce système, simple à utiliser, avait atteint ses limites, et qu'il convient maintenant de lancer un projet informatique qui va permettre de fédérer ces données dans un seul conteneur, et de les gérer avec une application dédiée.

## Description

Ce projet a pour but de remplacer le système actuel de gestion de formations, réalisé à l'aide de classeurs et feuilles Excel, par une application modulaire réalisée en php, mysql et javascript.

Il convient tout d'abord de remplir les fichiers Excel avec des valeurs fictives, qu'il s'agira ensuite de transférer dans la BD.

Le public cible est constitué de personnels de bureau, capables d'utiliser un ordinateur et des applications simples et standards. Ces personnes ne sont pas capables de corriger les incohérences de données (un formateur apparait 2 fois avec des noms différents car il y a un espace dans le nom composé dans un cas, et pas dans l'autre, par exemple).

Le dispositif permettra d'effectuer les saisies des informations, de les modifier, de les supprimer, et de les imprimer sous forme de listes, permettant ainsi de reproduire la gestion faite jusque-là avec des classeurs Excel.

Si le temps le permet, les fonctionnalités suivantes pourront encore être implémentées :

* Une fonctionnalité de statistiques permettant aux gestionnaires d'exploiter les informations provenant des enquêtes de satisfaction.
* Une fonctionnalité de récupération des données des formations pour l'année suivante
* Une fonctionnalité permettant un archivage des formations, des participants, des formateurs, etc.

## Matériel et logiciels à disposition

1 ordinateur PC avec Windows 7 standard de l'ETML

Une architecture WAMP, comme EasyPhp ou Xampp

Des classeurs Excel sans données permettant de comprendre la situation actuelle.

## Prérequis

Avoir suivi les modules ICT.

# Planification Initiale

Pour la planification, le tableau créer sur Trello fera foi. Chaque point est précisé et le nom des personnes affectées à une tâche y sont notées.

Lien vers Trello : <https://trello.com/b/vcPlEByr/042-p-gesproj2-dulexsa>

# Réalisation

## Dossier de Réalisation

### Sprint n°1

Au début du projet, le groupe s’est mobilisé pour créer un tableau Trello et ainsi faire un Product Backlog correcte qui sera utilisé durant le projet. Le tableau a été créé afin de l’utiliser avec une méthodologie Agile se nommant Scrum.

Une fois le tableau Trello correctement terminer, nous avons commencé notre Daily Meeting pour mettre en place le Sprint Planning. Cela a consisté à choisir les différentes étapes et tâches que l’équipe devra effectuer durant la période de sprint.

Après avoir pris ensemble la décision des tâches et avoir affecté les personnes, tout a été déplacé dans le Current Sprint et l’équipe s’est mise au travail.

Jonathan et Jérémy ont commencé à créer une structure de base de donnée (MCD-MLD) avec le logiciel JMerise. Ils ont, par la suite, fait vérifier le fichier par le reste des collègues. Pendant ce temps, Dylan a dessiné un croquis pour la structure du site internet et Samuel a cherché et trouvé un site internet qui servira de patron pour notre site.

Après que Jonathan et Jérémy aient validé la structure que Dylan a fait, aidé par Samuel, ils commencèrent à écrire un script qui créera la base de donnée dans le WAMP nommé EasyPhp. Après quelques minutes, voyant que Jérémy pouvait terminer le script seul, Jonathan entrepris le commencement du rapport du projet.

La mise en forme des informations principales tels que le titre, les auteurs, la description, etc… ont été fait puis, par la suite, le premier chapitre de la Réalisation a été écrite. Celui-ci est le chapitre 3.1.1 qui correspond au Sprint n°1.

Samuel et Dylan se sont penchés sur la modification du patron pour qu’il corresponde au croquis mais n’ont malheureusement pas eu le temps de finir durant le premier sprint.

### Validation du premier sprint

Le Product Owner est venu validé le premier sprint. Il a vérifié les différentes étapes qui ont été potentiellement validé durant le sprint.

Il n’a pas validé deux Users Stories car elles ne correspondaient pas à ses attentes. C’était l’architecture du site internet et le début du rapport.

L’architecture était bonne mais il manquait un lien de redirection. Le rapport devait être modifié afin de séparer les différents chapitres de réalisation pour une meilleure lisibilité des sprints.

Ces quelques « fautes » seront corrigées durant le prochain sprint.

### Sprint n°2

Le groupe a commencé, comme à son habitude, le Daily Meeting. Le sprint planning a été décidé et la durée de chaque étape a été planifié durant le Meeting.

Samuel a rajouté le lien que le Product Owner avait demandé lors de la première validation et Jonathan a mis à jour le rapport afin qu’il convienne aux attentes du client.

Samuel et Dylan doivent, désormais, s’occuper du site internet pour qu’il puisse correctement inclure les données de la base de donnée.

Ils ont d’abord repris les différents croquis qu’ils avaient effectué durant le premier sprint. Puis ils se sont divisé le travail selon les différentes pages qu’il restait à faire.

Samuel s’est concentré sur la page qui permet de créer des nouveau cours grâce à un formulaire. Cette page contient aussi une vue d’ensemble de tous les cours créer sous forme de liste.

Il a aussi retravaillé la page d’accueil afin de mettre des raccourcis qui renvoient sur les pages correspondantes grâce à des boutons munies d’icônes.

Dylan a, quant à lui, fait une page qui servira de détailler chaque cours. Cette page sera atteignable depuis la liste de Samuel et s’adapte selon le cours qui a été choisi par l’utilisateur.

Il a également fait une page qui sert à ajouter des élèves dans la base de donnée à l’aide d’un formulaire similaire à celui utilisé pour la création d’un cours.

Jérémy et Jonathan ont, pendant ce temps, travaillé sur la base de donnée pour écrire les différentes requêtes qui serviront à afficher/modifier/supprimer les données sur le site internet.

Pour cela il a fallu, pour faire des tests, ajouter des données fictives (mais cohérentes) dans la base de donnée.

En attendant qu’un certain nombre de données soient inséré par Jérémy, Jonathan a commencé à faire quelques modifications dans le rapport.

Il a changé l’image du titre de la première page du rapport puis a continué le chapitre sur la réalisation du projet en ajoutant le chapitre « Validation du premier sprint » et « Sprint n°2 ». Le chapitre 3.1.2 est celui de la validation de sprint. Il recense ce que le client a validé ou ce qu’il a trouvé indispensable de changer. Et le 3.1.3 correspond au chapitre sur le second sprint. Celui-ci comprend les informations sur ce que l’équipe de développement a effectué.

Jérémy, ayant terminé d’inséré des données fictives dans la base de donnée, a essayé de créer les requêtes SQL qui serviront pour la page internet.

Il a créé une quinzaine de requêtes SQL. Certaines sont utiles pour inséré des données dans la base de donnée, d’autre pour les afficher selon un certain trie ou selon certains critères.

### Validation du deuxième sprint

Le Product Owner est venu validé le deuxième sprint. Il a vérifié les différentes étapes qui ont été potentiellement validé durant le sprint.

Il a validé la mise en forme du rapport du projet. Ce qui correspond aux chapitres 3.1.1, 3.1.2 ainsi que le chapitre 3.1.3.

Le site internet a été validé que ce soit le croquis ou les fonctionnalités.

Différentes requêtes SQL ont été créé et elles ont, elles-aussi, été validé avec succès.

### Sprint n°3

Les premières minutes ont été utilisées, comme d’habitude, pour le Daily Meeting. Le planning du troisième sprint a été décidé et la durée des étapes a été planifié durant le Meeting.

Celui-ci a dû se dérouler sans l’aide de Jérémy, absent le 26.04.17. Donc le sprint s’est effectué sans lui.

Une fois le Meeting terminé, chacun a commencé son travail. Samuel et Dylan doivent s’occuper de terminer le site internet en le rendant « dynamique ». Par exemple en introduisant du PHP et du SQL pour travailler avec les données qui sont contenus dans la base de donnée.

Samuel s’est concentré sur les formulaire d’ajout du site en utilisant les requêtes créées auparavant. Les pages de « addFormer » et « addStudent » ont été créées.

AddFormer servira pour ajouter des nouveaux enseignants et addStudent pour les étudiants.

Il a eu quelques soucis car certains noms de variables utilisées ne correspondaient pas entre elles. Mais après quelques minutes, cela a été corrigé et Samuel a pu mener à bien ces deux pages et fit de même pour la page « lesson ». C’est-à-dire que cette page pourra créer des leçons et les rajouter dans la base de donnée, comme les deux pages d’avant.

Dylan a, quant à lui, inséré les requêtes qui serviront à la mise en page des informations sur le site internet. Elles vont chercher les informations dans la base de donnée et le PHP les affiches correctement. Grâce à lui, la barre de navigation sur la gauche fonctionne correctement et tous les liens fonctionnent. La page « showLessonInformations » qu’il a effectué peut afficher toutes les informations de la leçon que l’utilisateur aura sélectionné. L’utilisateur aura donc une vue d’ensemble de la leçon, la liste des élèves qui y participent ainsi que la liste des élèves n’y participant pas. L’utilisateur peut également rajouter ou supprimer des élèves du cours.

Jonathan a commencé à créer une présentation pour le projet qui devra comprendre comment le groupe à travailler durant le projet ainsi que le projet en lui-même. Cette présentation sera assez conséquentes car il faudra parler, en détaille, de tout ce qui a été effectué durant le projet.

Mais n’ayant pas d’inspiration pour le moment, Jonathan a décidé de se mettre à continuer le rapport et de reprendre la présentation un peu plus tard durant le sprint.

Dans la documentation, les chapitres 3.1.4 et 3.1.5 ont été développés. Le chapitre 3.1.4 correspond à la validation du sprint n°2. Il ressort ce que le client a trouvé de bien et ce qui est nécessaire de modifier. Le chapitre 3.1.5 n’est autre que la review du sprint n°3. Tout ce qui a été effectué par l’équipe de développement y est recensé.

A la fin de la mise à jour de la documentation, Jonathan repris la création du PowerPoint.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## Dossier des tests

On dresse le bilan des tests effectués (qui, quand, avec quelles données…) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats attendus et obtenus, ainsi que les actions à entreprendre en conséquence (et une estimation de leur durée).

Si des tests prévus dans la stratégie n'ont pas pu être effectués :

raison, décisions, etc.

Liste des bugs répertoriés avec la date de découverte et leur état:

Corrigé, date de correction, corrigé par, etc.

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différences entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Pour le journal de travail, le tableau créer sur Trello fera foi. Chaque point est précisé et le nom des personnes affectées à une tâche y sont notées.

Lien vers Trello : <https://trello.com/b/vcPlEByr/042-p-gesproj2-dulexsa>

## Webographie

Références des sites Internet consultés durant le projet.

# Annexes

Cahier des charges

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.