# Réalisation d'un classeur pédagogique numérique

## *Méthodologie et contexte*

## The realization of an educational workbook software

### *Methodology and context*

**Ali Sadiqui**
*ISTA Meknes, OFPPT, Meknes, Maroc*
**sadiqui2000@yahoo.fr**

**Résumé**

Utilisées pour concevoir des applications en impliquant pas à pas le client, les méthodes agiles sont de plus en plus adoptées par les équipes de développement. C'est dans ce cadre théorique qu'a été conçu et piloté le projet objet de cette contribution.
Ledit projet est motivé par le fait que depuis quelques années les formateurs appartenant à l'Office de la Formation Professionnelle et de la Promotion du Travail (Maroc) recourent quotidiennement à un classeur pédagogique sur support papier pour organiser les activités pédagogiques relatives à la formation, à l'évaluation et à la présence des étudiants. Étant sur support papier, l'utilisation dudit classeur se révèle fastidieuse en ce sens qu'elle exige un temps considérable et, de ce fait, affecte négativement la progression de l'apprentissage. Il a fallu donc penser à alléger son usage.
Nous avons alors, en collaboration avec une équipe de praticiens et avec le client effectif, conçu et expérimenté une version numérique de cet outil pédagogique. Le produit final a impacté positivement la gestion de la classe.

**Abstract**

*Used to design applications involving step by step the customer, agile methods are increasingly adopted by development teams. It is within this theoretical framework that has been designed and piloted the project purpose of this contribution.*
*The said project is motivated by the fact that in recent years the trainers belonging to the Office of Vocational Training and Work Promotion (Morocco) daily have recourse to a paper educational workbook to organize educational activities related to training, evaluation and student's presence. Being on paper, it requires considerable time and, thus, affects negatively the learning progress. It was, therefore, necessary to think about simplifying its use.*
*So we, in collaboration with a team of practitioners and the actual customer, have designed and tested a digital version of this educational tool. The final product has impacted positively classroom management.*

**Mots-clés**

classeur pédagogique, logiciels pédagogiques, logiciels libres, Outils didactiques, outils pédagogiques, méthodes Agiles, Extreme-Programming.

**Keywords**

paper educational workbook, educational software, free software, didactic tool, educational tools, Agile Extreme Programming.

# 1. Introduction

Placer le client au cœur du processus de développement des logiciels est le fondement des méthodes de développement dites «agiles» (Houy *et al*, 2013) ; (Cantone et Marchesi, 2014) ; (Vickoff, 2009) ; (Abbas *et al*, 2008). Plusieurs méthodes se regroupent sous cette bannière. Elles sont fondées sur un développement itératif et incrémental dans lequel la recherche de solutions aux problèmes rencontrés s'appuie sur une collaboration de pair à pair. Ces méthodes définissent des groupes de pratiques qui visent à favoriser le travail avec les spécificités de tout un chacun intégré dans la création du logiciel. Le développement d'un système informatique devient alors une activité motivante où chaque membre se voit confier une part de responsabilité.
La présente étude a pour objectif la réalisation d'un logiciel informatique qui permet d'élaborer un classeur pédagogique selon les exigences de la démarche agile et de l'organisme demandeur.
Dans la seconde section de cet article, nous présenterons les méthodes agiles et plus particulièrement la méthode Extreme Programming. Puis, dans la troisième section, nous présenterons le contexte de notre projet, incluant une brève présentation de l'organisme concerné, son système de gestion et le classeur pédagogique utilisé. Présenter la solution adoptée pour créer ledit logiciel et les résultats obtenus fera l'objet de notre quatrième et dernière section.

# 2. Les méthodologies Agiles

Les méthodes traditionnelles se fondent sur un enchaînement séquentiel des différentes étapes de développement, depuis les spécifications jusqu'à la validation du système, selon un planning préétabli. Elles s'efforcent à définir toutes les exigences et les spécifications dès le début supposant que celles-ci restent immuables. Cette vision rassurante est cependant bien loin de la réalité des projets. En effet, des changements peuvent intervenir suite à une modification des besoins du client, ou bien suite à des erreurs découvertes lors de la phase de conception ou lors de l'implémentation du système. Il devient évident qu'un logiciel est a priori un exercice difficile, sauf dans le cas d'applications extrêmement simples ou de rares contextes connus et maîtrisés.
Les méthodologies classiques sont fondées sur le principe que le coût relié à une modification du logiciel augmente d'une manière exponentielle avec le temps. Et par conséquent, il est judicieux de définir tous les aspects du produit et de concentrer la plupart des décisions avant de procéder à sa réalisation.

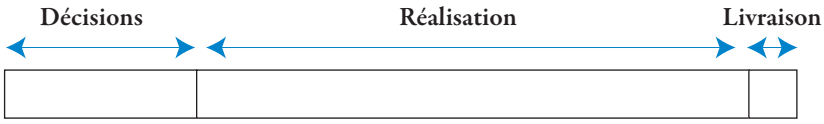


*Figure 1. Processus de développement dans les approches classiques*

Si les méthodologies classiques sont peu ouvertes au changement, les méthodes agiles, par contre, se proposent de réserver un accueil favorable au changement partant du fait que ce dernier est une composante incontournable dans le processus de développement. Elles partent du fait qu'il est plus rentable et plus pertinent de prendre les décisions progressivement et le plus tard possible au lieu de chercher à les spécifier et les concevoir complètement dès le départ. Pour ce faire, la prise de décision sera faite tout au long du projet grâce à des cycles de réunions appelés itérations.

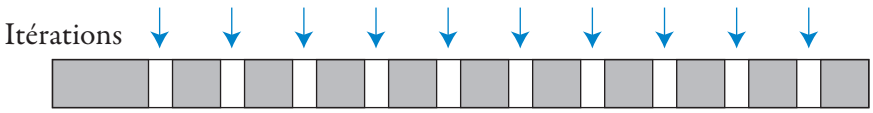


*Figure 2. Processus de développement dans les approches agiles*

En conséquence, le grand gagnant de cette approche est d'abord le client qui se trouve intégré dans tout le processus de développement. A chaque itération, il établit lui-même les fonctionnalités à implémenter, collabore avec l'équipe pour définir tous les détails et reçoit une nouvelle version du logiciel qui intègre les mises à jour en question.
L'une des méthodes Agiles les plus connues est la méthode Extreme Programming (XP) (Lassenius *et al*, 2015) ; (Rao *et al*, 2014) ; (Beck et Andres, 2005) ; (Cros, 2004) ;. Elle a été principalement développée par Kent Beck et Ward Cunningham. Cette méthode est définie par un ensemble de pratiques qui visent à coordonner le travail d'une équipe de développement. Elle se caractérise par les quatre éléments suivants : des cycles de développement courts, de tests intenses et une validation en continu, en plus d'une collaboration totale du client tout au long du processus de développement.

Notre projet, baptisé «PLANIFORM» (la planification de la formation), est une application concrète de ladite méthodologie. En effet, il a fait intervenir plusieurs participants dans les domaines de l'éducation et de l'informatique et a tracé pour objectif l'élaboration d'un projet informatique qui doit répondre à certaines exigences, d'ordre administratif, technique et pédagogique, de l'organisme demandeur, et selon la démarche Extreme Programming dont l'intérêt n'est plus à démontrer.

# 3. Le contexte du projet

Nous allons, en quelques lignes, faire une brève présentation du contexte qui nous a permis de concevoir et d'expérimenter notre projet.

## 3.1. Présentation de l'OFPPT

Créé en 1974, l'Office de la Formation Professionnelle et de la Promotion du Travail (OFPPT) est un établissement public ayant pour vocation d'accompagner les jeunes et les entreprises en matière d'apprentissage et de développement de compétences.
Le parcours de formation qu'il procure permet à son public cible de construire les compétences nécessaires permettant une insertion facile et rapide dans le monde du travail ou de la création d'entreprises.
Il s'adresse également aux demandeurs d'emploi pour les aider à actualiser ou à élargir leurs compétences professionnelles via des formations qualifiantes et aux entreprises, via des formations continues, pour développer les compétences de leurs employés.

## 3.2. Le système de formation à l'OFPPT

Le régime de formation à l'OFPPT est modulaire et les formations sont proposées selon la hiérarchie «domaine – filière – spécialité». Par exemple la spécialité «Développent Web et Multimédia» est une option de la filière «Techniques de Développement Informatique» appartenant au domaine «NTIC».
Il est à noter que certaines filières sont fusionnées avec leurs spécialités pour des raisons relatives à la simplification de leur gestion.
Chaque module est un ensemble d'informations et de connaissances qui permettent d'acquérir une compétence professionnelle précise reliée à une filière ou une spécialité. Les modules font partie d'un ensemble plus global : la filière ou la spécialité.
Un module est le découpage d'une compétence donnée en un certain nombre d'objectifs qui forment un tout autonome. Les objectifs sont divisés en objectifs de premier niveau et en objectifs de second niveau et commencent toujours par un verbe d'action (Présenter, Définir, etc.). Un module peut être acquis via plusieurs modalités : cours théoriques, travaux pratiques, travaux dirigés, projets ou stages, le tout organisé de manière cohérente, selon une masse horaire prédéfinie et une logique de progression. Ces modules visent l'acquisition d'une ou de plusieurs compétences ciblées.
C'est à travers ces divers modules que les stagiaires acquièrent de nouvelles connaissances qui leur permettent de finaliser l'apprentissage d'un métier donné.
Chaque module s'achève par un contrôle des compétences, une sorte d'examen appelée examen de fin de module (EFM) qui peut être théorique ou pratique.

## 3.3. Le système de gestion

Dans un établissement relevant de l'OFPPT, l'encadrant est un formateur qui met en œuvre des actions de formation et d'évaluation du public ciblé. Il dispense un ou plusieurs modules, qui relèvent d'une ou de plusieurs filières. Chaque formateur a un emploi du temps hebdomadaire variable durant l'année. Lors de chaque séance, de 2 heures 30 minutes en général, il doit aborder les objectifs préétablis du module assuré.
Le formateur est tenu de préparer une fiche pédagogique pour chaque séance, où il indique les objectifs abordés, les documents et les méthodes pédagogiques utilisés. Il doit aussi mentionner, dans le cahier de textes (cahier journal) les mêmes objectifs ainsi que l'état d'absence des stagiaires. Chaque module est évalué sous forme de contrôles continus et d'un EFM. La formule pour calculer la note du module est donnée comme suit :
Note = ((moyenne des notes des Contrôles Continus)+(2*note EFM))/3
On désire gérer le suivi de la progression du programme ainsi que le suivi des évaluations des modules et des absences des stagiaires.
Toutes ces actions pédagogiques se font actuellement de façon manuelle à travers un outil appelé «le classeur pédagogique» complètement basé sur le support papier.

## 3.4. Le classeur pédagogique, éléments de composition

Parmi les éléments clés de ce classeur pédagogique, on trouve la fiche **«Planification et suivi de la réalisation des modules de formation»** (Figure 3). Elle est composée de trois zones essentielles :

**Zone 1** : indique des informations générales sur le module : filière et groupe concerné, masse horaire, objectifs du module, etc.

**Zone 2** : intitulée **«Prévision par séance»**, elle indique le numéro et la date ainsi que les objectifs prévus pour cette séance. Chaque objectif commence par un verbe d'action (présenter, définir, etc.). Les objectifs mentionnés peuvent être des objectifs de premier niveau, ou détaillés en objectifs secondaires. Cette partie devrait faire l'objet d'une conception pédagogique selon un modèle préétabli. Le formateur doit y préciser minutieusement les démarches pédagogiques à mettre en œuvre pour atteindre les objectifs ciblés.

**Zone 3** : intitulée **«Réalisation par séance»**, elle indique les objectifs réalisés pour chaque séance. Dans le meilleur des cas, tous les objectifs prévus seront réalisés. Dans le cas contraire (un rythme lent de l'avancement de la séance, un problème technique ou un empêchement imprévu lors de déroulement de la séance, etc.) les objectifs réalisés seront mentionnés et le reste sera programmé pour la prochaine séance.

On établit aussi dans cette zone le cumul des heures réalisées pour ce module, cela a un intérêt particulier étant donné qu'il permet d'indiquer la progression de la réalisation du module.

Dans la zone réservée, on indique la liste des stagiaires absents.

### Planification et suivi de la réalisation des modules de formation

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Filière : | | | | | | Module : | | | |
| Masse horaire du module : | | | | | | Nombre de séances : | | | |
| 1ère année | | 2ème année | | 3ème année | | Groupe : | Nombre de stagiaires : | | |
| Objectif du module : **ZONE 1** | | | | | | | | | |

| Prévision par séance | | | | Réalisation par séance | | | | |
|---|---|---|---|---|---|---|---|---|
| N° | Date de la séance | Objectif opérationnel de la séance | Durée | Date | Contenu réalisé | Durée en heures réalisée | Durée en heures cumul | Stagiaires absents |
| | | | | | A prévoir pour la prochaine séance : | | | |
| | | **ZONE 2** | | | **ZONE 3** A prévoir pour la prochaine séance : | | | |
| | | | | | A prévoir pour la prochaine séance : | | | |
| | | | | | A prévoir pour la prochaine séance : | | | |

*Figure 3. la fiche «Planification et suivi de la réalisation des modules de formation»*

## 3.5. Cahier des charges du projet

Il fallait penser à la conception et la mise en œuvre d'un logiciel permettant de mieux gérer toutes ces informations pendant toute l'année scolaire et pour tous les publics ciblés. Notre référence est la norme ISO 9126, qui définit la qualité d'un logiciel, que nous avons essayé d'appliquer en prenant en considération le contexte présenté auparavant. Cela a permis de fixer certaines de ses fonctionnalités :

**La capacité fonctionnelle**

C'est la capacité d'un logiciel à répondre aux exigences et aux besoins explicites ou implicites des usagers. En effet, le projet doit respecter tous les éléments existant dans le classeur pédagogique exigé par la Direction Générale, sans aucune modification, en respectant aussi le contenu et la forme de tous les états de sorties.

**La facilité d'utilisation**

C'est la capacité d'un logiciel à être manipulé sans demander beaucoup d'efforts qui pourraient entrainer son rejet. Pour cela, le projet doit être facile à apprendre et à exploiter. En plus, une utilisation incorrecte d'un débutant ne doit pas entraîner son dysfonctionnement.

**La fiabilité**

C'est la capacité d'un logiciel à rendre des résultats corrects : tous les calculs effectués (le cumul des heures supplémentaires, les moyennes des notes des modules des stagiaires, etc.) doivent respecter les particularités et les exigences de l'OFPPT.

**La maintenabilité**

C'est la capacité d'un logiciel à être facilement modifiable. Le projet doit être extensible et nécessiter peu d'efforts pour celui qui veut y ajouter de nouvelles fonctions.

**La portabilité**

C'est la capacité d'un logiciel à fonctionner dans un environnement matériel ou logiciel différent de son environnement initial.

# 4. La conduite du projet

## 4.1. Présentation de l'équipe

Notre équipe fut formée de cinq personnes, toutes exerçant au sein de l'OFPPT. Chacun des membres a pris un rôle en conformité avec la méthode XP. Le projet a été piloté par un coordonnateur (Manager et Coach), qui avait pour rôle de planifier les réunions et de coordonner les efforts. Faisaient aussi partie de l'équipe un vérificateur (Tracker), deux développeurs et un testeur.

## 4.2. Expérimentation du logiciel

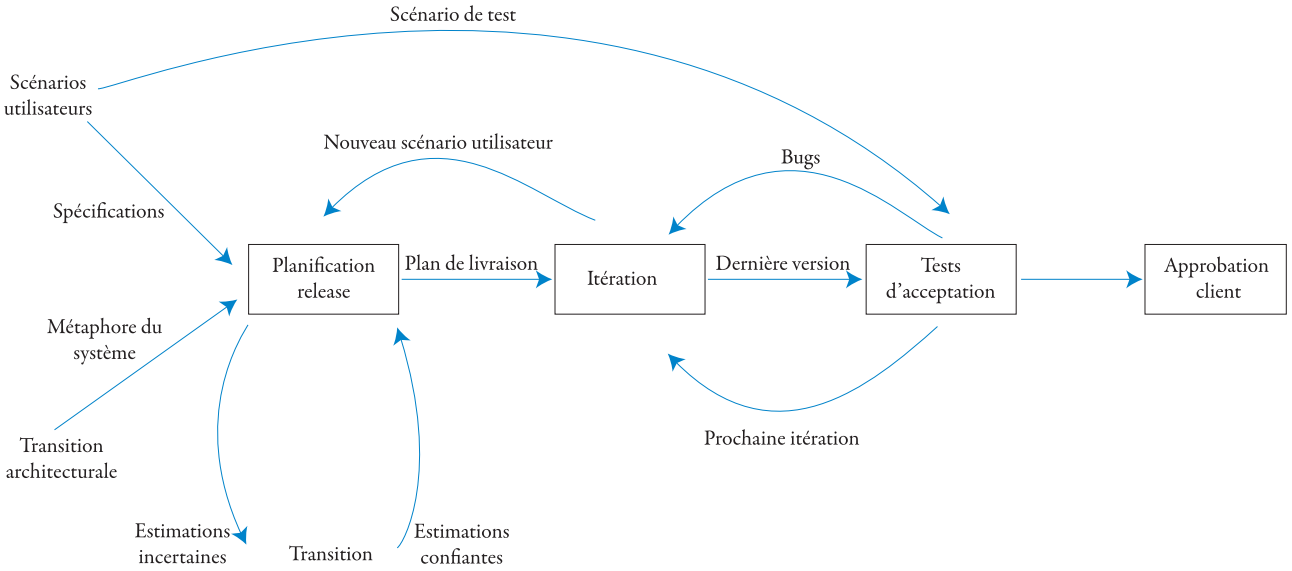


*Figure 4. Cycle de développement de la méthode «Extreme Programming».*

Le projet a fait aussi intervenir une trentaine de formateurs, tous exerçant à l'OFPPT. Leur domaine d'intervention au sein de notre organisme inclut toutes filières et tous niveaux confondus.

Le projet avait démarré par une phase d'exploration d'un mois, qui avait pour objectifs de définir le contenu fonctionnel de l'application, établir un premier plan de développement pour le projet, et produire la toute première version du logiciel. Le fait que notre équipe ainsi que les formateurs étaient tous disponibles à l'établissement durant toute la semaine nous a beaucoup facilité le pilotage du projet.

La planification du projet a été réalisée conjointement avec les formateurs au cours de séances dédiées, organisées régulièrement tout au long du projet.

Les itérations ont été fixées à deux semaines et les réunions ont été établies dans les établissements de formation. L'équipe a adopté des horaires qui lui permettent de conserver tout au long du projet l'énergie nécessaire pour produire un travail de qualité et mettre en œuvre efficacement les régulations qui s'imposaient.

## 4.3. Résultats obtenus

### *4.3.1. Le feed-back des formateurs*

Notre projet a été développé et amélioré en référence à la fois aux objectifs sus cités et à l'analyse des pratiques des formateurs. Durant toute une année, notre équipe leur livrait des versions du logiciel (frequent releases) pour qu'il soit adapté et optimisé à leurs besoins et pour leur offrir une réactivité et un confort d'utilisation maximal sans oublier la correction des erreurs et des anomalies produites.
Il est à noter que les documents du classeur pédagogique, sur support papier, restent toujours des éléments exigés par l'administration pour tout contrôle et suivi d'avancement du programme. Cependant, ils sont imprimables sous forme d'états de sorties pour être classés sur ledit classeur.
Notre projet ne visait donc pas à remplacer l'ancien système, mais se fixait plutôt comme objectif de doter les formateurs d'une solution souple pour la gestion et la production desdits documents.
En outre, le produit final a permis à tous les formateurs impliqués dans cette expérience de gérer leurs emplois du temps, de faire le suivi des heures supplémentaires et de planifier les dates des EFM, entre autres.

### *4.3.2. Présentation de PLANIFORM*

Les différentes réflexions menées lors de la conception de ce projet nous ont orientés vers le choix de le doter de 6 modules : Vues, Groupes, Séquences, Listes des stagiaires, États, et Évaluations.
Les modules ont été développés simultanément, car ils définissent le contenu fonctionnel minimal d'une application utilisable par les formateurs. Cependant, ils ont subi des améliorations ainsi que des corrections durant la période du projet.
Le développement de ces modules s'est basé sur des composants libres et dont le code source est disponible. Ce dernier a été modifié et adapté selon les besoins de notre projet.

**Le module «Vues»**

Il permet de gérer les emplois de temps, les jours d'absence, les jours fériés et les congés de maladie, etc. il permet aussi de gérer les contenus des séances. En effet, nous avons la possibilité de mettre en forme le texte saisi, de lui ajouter des schémas, des documents, des liens, etc., de réexploiter ce contenu d'un groupe à l'autre, et de marquer les absents.

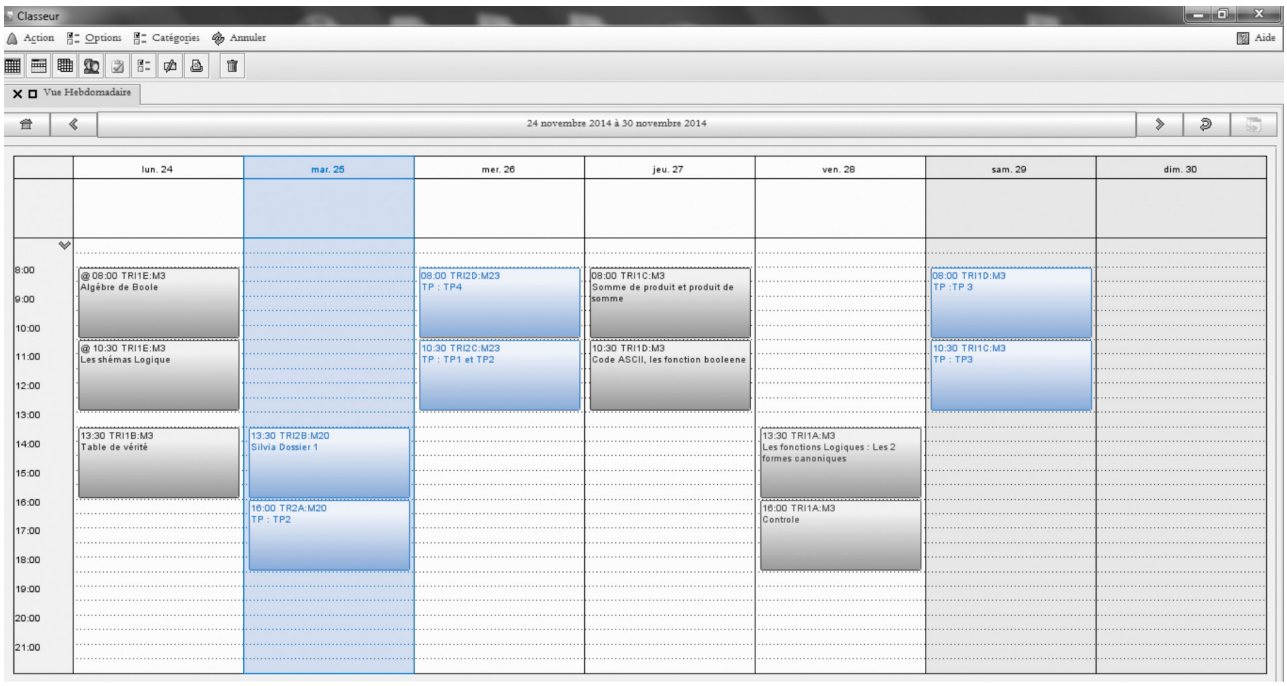


*Figure 5 : Capture écran de la vue «hebdomadaire»*

**Le module «Séquences»**

À partir des objectifs établis par la Direction de la Recherche et de l'Ingénierie de la Formation (DRIF), les formateurs doivent élaborer des séquences pédagogiques qu'ils vont suivre tout au long de l'année, les segmenter et les organiser en activités pédagogiques. Ce module leur permet de faciliter la création et l'utilisation des séquences pédagogiques. En effet, nous avons la possibilité de mettre le lien avec le guide pédagogique élaboré par la DRIF. Ce module permet aussi d'échanger les préparations des séquences avec d'autres collègues pour les améliorer.

**Le module «Groupes»**

Il permet de gérer les groupes et les modules, planifier les dates des EFM et faire le suivi de la progression du programme annuel.

**Le module «Listes des stagiaires»**

Il permet de gérer les stagiaires, importer la liste des stagiaires pour chaque groupe et avoir un accès rapide aux absences des stagiaires et à leurs évaluations.

**Le module «États»**

Il permet d'imprimer les préparations pédagogiques pour chaque module, les documents du classeur pédagogique, le bilan d'absence annuelle ou par module des stagiaires et l'emploi du temps d'une période définie.

**Le module «Évaluations»**

Il permet de gérer les évaluations, saisir les notes des évaluations et les imprimer.

## 4.4. Les composants utilisés

Deux composants libres essentiels ont été utilisés pour le développement de notre projet à savoir «Borg» et «Jasperreports». Ce choix est justifié par le fait que ces derniers ont répondu à plusieurs recommandations notamment le type de licence, la disponibilité du code source et d'autres facteurs comme la plateforme requise, le langage de programmation ayant servi à sa création, le type de base de données mis en œuvre, etc.
En conséquence, le logiciel qui constitue la pierre angulaire de notre projet et qui a été développé en langage Java est un logiciel libre. Nous avons veillé à ce que le logiciel et les outils sus cités suivent les mêmes termes des licences General Public License (GNU).

## 4.5. Présentation de l'outil «Borg»

L'outil Borg est un outil libre qui combine un calendrier et un système de suivi de projets personnels. Le calendrier prend en charge toutes sortes de rendez-vous, de gestion des événements et des listes de tâches simples à réaliser. Il fait partie des logiciels appelés les gestionnaires d'informations personnelles ou de projets (PIM), tels que Microsoft Outlook, Mozilla Calendar, Palm Desktop, Yahoo Calendar, etc. Ces logiciels proposent des fonctionnalités agenda, bloc-notes, post-it, rendez-vous, carnet d'adresses. Les versions professionnelles peuvent aussi être associées à des fonctionnalités de courrier et de téléphonie (VoIP, SMS), couplés à des logiciels de gestion de la relation client et de cartographie. Elles peuvent également servir les multi-utilisateurs de façon à partager des messageries, des calendriers et des emplois du temps de réunions.
Notre projet hérite de certaines des fonctionnalités de cet outil. Cependant, son code source était largement modifié et revu étant donné que les objectifs escomptés diffèrent en grande partie de ceux ayant fait l'objet de sa création.
Plusieurs fonctionnalités lui ont été intégrées et d'autres ont été abandonnées. Les attributs de classes ont été également revus et, par conséquent, la conception de la base de données a été modifiée. Et cela pour l'adapter à nos besoins et pour mener à bien notre projet. Nous estimons que seulement 30% du code source original a été conservé.
C'est sur la base de ce composant que nous avons développé les modules : Vues, Groupes, Séquences, Listes des stagiaires et Évaluations.

## 4.6. Présentation de JasperReports

JasperReports est un outil libre de création d'état de sortie, offert sous forme d'une bibliothèque qui peut être intégrée dans tous types d'applications Java. Cet outil se base sur des fichiers XML (dont l'extension est en général .jrxml) pour la présentation des états. Il peut être couplé à d'autres outils (iReport ou JasperStudio par exemple) pour faciliter sa mise en œuvre dans une application Java, classique ou orientée web.

Cet outil nous a servi pour produire les documents du classeur pédagogique. Ces derniers peuvent être lus, directement dans l'application, imprimés ou exportés dans une variété de formats de documents, y compris HTML, PDF, Excel, OpenOffice et Word. Sa facilité d'utilisation et d'intégration nous a permis de réduire considérablement le temps et l'effort de développement.
Ce composant nous a aidés à développer le module États.

### 4.7 Autres détails à préciser

- Le projet est disponible sur les liens :
  - `http://www.mediafire.com/download/46gjogchlu67jpo/PlaniformSetup.exe`
  - `https://mon-partage.fr/f/nSiVKEeT/`
- La documentation de PLANIFORM est directement intégrée au logiciel. Un fichier pdf permet d'expliquer les différentes étapes pour son utilisation. De plus, la vidéo disponible sur le lien : `https://www.youtube.com/watch?v=bUTJJXnwn6c` permet de combler tout besoin d'information. Deux fichiers Excel sont aussi disponibles pour faciliter la procédure d'importation.
- La base de données choisie est le «H2» vu sa facilité d'être intégrée dans notre projet. Elle est déjà intégrée dans l'outil «Borg» et suit les mêmes termes de licence.
- Une copie des termes de licence est intégrée dans la rubrique «Licence» sous le menu Aide.
- Avant la version 2, le projet a pris le nom de «Classeur».

## 5. Conclusion

PLANIFORM est un outil simple à utiliser, pratique et intuitif, destiné à l'élaboration du classeur pédagogique selon les exigences de notre établissement. Par ailleurs, le projet prend en considération les paramètres théoriques auxquels nous nous sommes référés. Il permet de réduire considérablement le temps et l'effort pour la réalisation des documents afférents et, par conséquent, permet au formateur de se focaliser sur l'approfondissement des contenus pédagogiques et sur l'amélioration des compétences des étudiants.
Ce projet peut facilement être intégré dans d'autres contextes similaires (enseignement général, supérieur, etc.) étant donné que dans sa conception, il a été tenu compte de plusieurs facteurs qui peuvent se retrouver partiellement ou totalement dans ces contextes.
Le projet va, dans la prochaine étape, bénéficier d'autres fonctionnalités visant à offrir plus de satisfaction aux différents intervenants. Nous projetons, en fait, de continuer à nous investir dans des recherches participatives qui permettraient d'élargir et d'optimiser son utilisation dans divers contextes universitaires.

## 6. Références

Abbas, N., Gravell, A., Wills, G. (2008). *Historical roots of Agile methods: where did «Agile Thinking» come from?* In Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L., Wang X. Agile processes in software engineering and extreme programming. 9th International Conference XP2008, Limerick, Ireland, June 2008, Proceedings in Software Engineering. Limerick: 10 - 14 Juin 2008, Berlin: Springer. 94-103.

Beck, K., Andres, C. (2005). *Extreme programming explained: embrace change* (2e édition). Upper Saddle River NJ : Pearson Education.

Cantone, G., Marchesi, M. (2014). *Agile processes in software engineering and extreme programming*. Proceedings of 15th international conference, XP 2014. Berlin : Springer.

Cros, T., (2004). *Maîtriser les projets avec l'extreme programming : pilotage par les tests-client.* Toulouse : Éditions Cépadues.

Houy T., Fernandez V., Khalil C., (2013). *Les méthodes agiles de développement informatique*. Paris: Presses des Mines.

Lassenius, C., Dingsøyr, T., Paasivaara, M. (2015). *Agile processes, in software engineering, and extreme programming*. Proceedings of 16th international conference, XP 2015, Helsinki: 25-29 Mai 2015, Berlin : Springer.

Rao, G. S., Krishna, C. V., Rao, K. R. (2014). *Extreme Programming for service-based application development architecture*. Proceedings of the 2014 Conference on IT in Business, Industry and Government (CSIBIG). Indore: 8-9 Mars 2014. Mishra, D. K., Sheikh, R., Excellent Publishing Services.

Vickoff, J.P. (2009). *Méthode agile, Les meilleures pratiques, Compréhension et mise en œuvre*. Vanves (Hauts-de-Seine) : Editions QI.