

Travail de bachelor 2010

Filière Informatique de gestion

Outil web pour la gestion de projet selon les méthodes agiles

**AGILE
WEB
PROJECT
MANAGEMENT
TOOL**



Etudiant-e : Thomas Correia

Professeur : Pierre-Yves Guex

Préface

Au cours de ma formation d'informaticien de gestion, j'ai eu l'opportunité de connaître plusieurs méthodes et outils de gestion de projet. La planification a un rôle très important au sein d'une entreprise et, vu que ce sujet porte sur l'implémentation d'un de ces outils, cela m'intéresse beaucoup.

Comme j'ai un grand intérêt pour le développement web et que je suis désireux d'améliorer mes connaissances sur le langage Ruby On Rails, cela a aussi appuyé mon choix pour ce sujet.

De plus, au moment de choisir un travail de Bachelor j'étais employé comme stagiaire au sein de l'institut Icare qui m'a alors proposé ce thème. Le fait que cet outil de gestion de projet est un réel besoin pour les membres de l'institut m'a permis de m'investir pleinement à la réalisation de ce travail.

Remerciements

Je tiens vivement à remercier les membres de l'institut Icare, et tout particulièrement Pierre Kenzelmann, Jean-Jérôme Sarrasin, Gilles Cherix, Christophe Abbet, Hugo Torres, Philippe Bürgisser, Stéphane Savioz et Cédric Michelet, pour avoir donné de leur temps lors des séances et pour leurs précieux conseils.

Je tiens à remercier également mon amie Tamara Gomez et mes parents, pour leur soutien moral et pour avoir relu mon document final, ainsi que mon professeur responsable de la HES-SO, Jean-Pierre Guex, pour sa disponibilité à mon égard.

Structure du document

Dans ce document vous trouverez tout d'abord un résumé qui explique brièvement l'ensemble du projet. La suite du document contient une présentation générale et une explication détaillée du travail réalisé. À la fin du document se trouvent les annexes, la table d'illustrations ainsi que la bibliographie.

Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué mon travail de Bachelor seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées.

Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe des responsables de la HES-SO de Sierre, du professeur responsable du suivi du travail de Bachelor, Monsieur Jean-Pierre Guex et de l'institut Icare.

Signature :

Résumé

Ce travail de Bachelor est réalisé au sein de l'institut Icare situé au TechnoArk de Sierre. Il est nécessaire à la finalisation de ma formation d'informaticien de gestion à la HES-SO Valais.

L'objectif du travail est d'implémenter un outil de gestion de projet pour les membres de l'institut afin de résoudre les problèmes liés au logiciel actuellement utilisé.

Cette application doit être de langage et de Framework Ruby On Rails, mais doit aussi intégrer des caractéristiques d'une méthodologie Agile. Une méthode Agile est une pratique qui introduit des notions spécifiques lors d'un développement informatique pour un groupe de personnes. Ruby étant un langage open source, l'application se doit d'être libre d'utilisation, d'exploitation et de modification.

Les différentes étapes à la réalisation de ce travail consistent à effectuer une recherche et une analyse de plusieurs solutions disponibles sur le marché, sans que toutes ne soient forcément open source. Pour élargir l'horizon et offrir une comparaison totale, il est possible d'introduire des outils payants dans l'état de l'art.

La solution choisie est Redmine, un outil open source qui permet de créer des tâches qui sont assignées à un des membres de l'institut. Une planification s'en suit pour répartir ces demandes selon l'emploi du temps de la personne. Redmine fournit également une certaine quantité d'autres fonctionnalités, comme un wiki ou un reporting du temps effectué par les ressources.

Cependant et afin de parfaire cette implémentation, les membres de l'institut doivent pouvoir aisément utiliser cette application. Une analyse des besoins est effectuée afin de récolter, de concevoir et d'ajouter les fonctionnalités manquantes à cet outil.

Les fonctionnalités principales comprennent un plan de charges, un calendrier, un budget et un reporting par semaine. Des fonctionnalités secondaires sont également développées. Elles se composent d'une notification par email et d'une gestion plus souple des heures prévues. Afin d'améliorer la planification, un Drag&Drop est implémenté dans le plan de charges. Les utilisateurs peuvent également mettre à jour des demandes lors d'un commit de code source.

Finalement, la solution implémentée est fonctionnelle et utilisée par tous les membres de l'institut. Tous les projets courants sont migrés sur Redmine et une utilisation quotidienne de cet outil s'en suit.



Table des matières

1. INTRODUCTION	1
1.1. Cadre du projet	1
1.2. Description détaillée	1
1.3. Problématique	2
1.4. Objectifs du projet	2
1.4.1. Objectif principal	2
1.4.2. Objectif secondaire	2
2. MÉTHODOLOGIE	3
2.1. Préambule	3
2.2. Caractéristiques	3
2.3. Cycle	4
2.3.1. Backlog de produit	4
2.3.2. Backlog de sprint	4
2.3.3. Sprints	5
2.3.4. Release	5
2.3.5. Quotidien	5
2.4. Rôles	6
2.4.1. Directeur de produit	6
2.4.2. Équipe	6
2.4.3. Facilitateur	6
2.4.4. Intervenants	6
2.5. Synthèse	7
3. TECHNOLOGIES	8
3.1. Ruby	8
3.1.1. Définition	8
3.1.2. Caractéristiques	8
3.1.3. RubyGems	9
3.1.4. Quelques exemples Ruby	10
3.1.5. Ruby on Rails (RoR)	10
3.1.6. Utilité de RoR	11
3.1.7. Environnement de travail de Rails	11
3.1.8. Architecture Modèle-Vue-Contrôleur	11



3.1.9.	Conclusion	12
3.2.	Ajax	12
3.3.	RJS	13
3.4.	Prototype	14
3.5.	Script.Aculo.Us	14
3.6.	jQuery	14
4.	ETAT DE L'ART	15
4.1.	Introduction	15
4.2.	Critères	16
4.3.	Outils de gestion existants	17
4.3.1.	Introduction	17
4.3.2.	Redmine	17
4.3.3.	Icescrum	18
4.3.4.	Intra-Know	18
4.3.5.	ExplainPMT	19
4.3.6.	VersionOne	20
4.3.7.	Trac	20
4.4.	Synthèse	21
4.4.1.	Compte rendu	21
4.4.2.	Choix de la solution à implémenter	22
5.	DESCRIPTION DU PROJET	23
5.1.	Redmine	23
5.1.1.	Définition	23
5.1.2.	Fonctionnalités existantes	23
5.2.	Fonctionnalités à ajouter	23
5.2.1.	Introduction	23
5.2.2.	Fonctionnalités principales	24
5.2.3.	Fonctionnalités secondaires	25
5.3.	Prototypes écran	26
5.3.1.	Plan de charges	26
5.3.2.	Reporting semaine	28
5.3.3.	Calendrier	29
5.3.4.	Budget	30
6.	DÉVELOPPEMENT DU PROJET	31
6.1.	Introduction	31



6.2. Configuration	31
6.2.1. Environnement de travail	31
6.2.2. Ruby On Rails	31
6.2.3. SGBD	31
6.2.4. Installation de Redmine	32
6.3. Planification du développement	33
6.4. Réalisation	34
6.4.1. Fonctionnalités principales	34
6.4.2. Fonctionnalités secondaires	43
7. PROBLÈMES RENCONTRÉS	47
7.1. Trouver plusieurs outils Ruby	47
7.2. Attentes différentes de l'entreprise	47
7.3. Ruby On Rails	47
7.4. Coder sur Redmine	49
8. FEEDBACKS	51
8.1. Avis des membres de l'institut Icare	51
8.2. Améliorations possibles	52
9. GESTION DU PROJET	53
10. CONCLUSION	54
11. ANNEXES	55
11.1. Cahier des charges	55
11.2. Planification initiale	58
11.3. Planification détaillée	59
11.4. Détail des heures réalisées	60
11.5. Fonctionnalités existantes	67
12. TABLE DES ILLUSTRATIONS	79
13. BIBLIOGRAPHIE	81



1. Introduction

1.1. Cadre du projet

Ce projet a été proposé par l'institut Icare, dans lequel je suis stagiaire à 20%. Le thème du travail porte sur l'implémentation d'un outil de gestion de projet.

Cet institut d'informatique spécialisé dans le développement d'applications web est désireux d'améliorer quotidiennement l'environnement de travail de ses employés.

Le chef de projet m'a donc assigné comme objectif de fournir une solution livrable qui soit utilisable et réponde aux besoins des membres de l'institut.

1.2. Description détaillée

Ce travail de Bachelor consiste à trouver, concevoir et implémenter une application fonctionnelle pour l'institut Icare : en arborant une analyse détaillée des outils existants sur le marché, en récoltant et en définissant précisément les fonctionnalités nécessaires à l'équipe de développement, en fournissant un outil utilisable qui respecte toutes les conditions définies dans le cahier des charges.

Actuellement, les développeurs et le chef de projet utilisent plusieurs outils pour gérer les différents projets en cours ; Intra-Know, Redmine et divers autres logiciels et services comme Google Agenda, Microsoft Excel ou encore Trac.

Ces outils sont complémentaires les uns des autres, mais aucun n'est capable actuellement de subvenir à tous les besoins de l'entreprise par sa seule présence.

L'intérêt de n'utiliser qu'un seul outil est d'éviter au maximum la redondance de saisie d'informations, par exemple en utilisant le Google Agenda pour la gestion des vacances qui devront aussi être reportées dans une feuille Excel commune à tous.

Un autre intérêt est de réunir les fonctionnalités les plus intéressantes de ces outils sur une seule application.



1.3. Problématique

La problématique consiste à résoudre les ennuis résultant de l'utilisation de différents outils de gestion de projet au sein de l'institut Icare, en analysant le domaine, récoltant les suggestions et avis des membres et en concevant et livrant une solution utilisable.

Il est également nécessaire que l'outil adopte des caractéristiques propres aux méthodologies agiles et qu'il soit basé sur Ruby On Rails, qui est le langage et Framework le plus utilisé dans les différents projets de l'institut.

Le challenge majeur de ce travail se trouve donc au niveau du langage, sur lequel je possède très peu d'expérience et qui, de prime abord, reste assez complexe. Il faut également comprendre et modifier un code existant.

1.4. Objectifs du projet

1.4.1. Objectif principal

L'objectif principal, comme cité dans la description détaillée cf. page 1, est de fournir une application fonctionnelle, qui soit d'une part simple d'utilisation, qui impose le moins possible de contraintes aux utilisateurs et qui remplisse les conditions spécifiées dans le cahier des charges.

Cet outil doit en priorité répondre aux besoins des membres, être basé sur une méthodologie Agile et langage Ruby afin que l'équipe d'Icare puisse le modifier facilement.

1.4.2. Objectif secondaire

L'objectif secondaire est d'implémenter des fonctionnalités de moindre importance, qui sont prioritaires en rapport à leur fréquence d'utilisation au sein de l'application.

2. Méthodologie

2.1. Préambule

Dans le cadre de ce projet où il s'agit de baser un outil de gestion sur une méthode Agile, il n'a pas été question de choisir une méthodologie précise, étant donné qu'il n'était pas censé impliquer des changements majeurs dans la gestion des projets au sein de l'institut, en modifiant la manière de travailler du chef de projet et des développeurs.

Cependant, afin d'expliquer et de déterminer les points intéressants d'une méthode Agile, il fut nécessaire d'en choisir une et celle-ci fut la méthodologie Scrum, car c'est l'une des plus simples à interpréter, qui comporte une majorité de fonctionnalités intéressantes pour un outil de gestion de projets et qui fut créée pour des petites équipes de développement comme celle de l'institut Icare.

2.2. Caractéristiques¹

« Le terme Scrum est emprunté au rugby et signifie mêlée. Ce processus s'articule en effet autour d'une équipe soudée, qui cherche à atteindre un but, comme c'est le cas en rugby pour avancer avec le ballon pendant une mêlée. »

Source : Wikipedia.org

La méthodologie Scrum a été inventée pour simplifier et améliorer la productivité dans des équipes de développement et peut être utilisée dans n'importe quel contexte, par exemple pour la gestion de recherches scientifiques ou simplement pour planifier un mariage.

Le principe de Scrum est de définir une liste de fonctionnalités à réaliser au sein d'une itération, ou sprint, d'une durée fixe (une à quatre semaines). Chaque itération est consacrée à obtenir un but final que désigne le directeur de produit ou « Product Owner », et dans chacune d'entre elles on implémentera alors les différentes fonctionnalités à réaliser.

Une des forces de Scrum est de faire participer le client à la planification, car il peut définir les priorités des fonctionnalités à développer, mais également modifier ou ajouter des fonctionnalités en cours de projet.

¹ **Source :** Wikipedia, Scrum, consulté le 13/08/2010, <http://fr.wikipedia.org/wiki/Scrum>

2.3. Cycle²

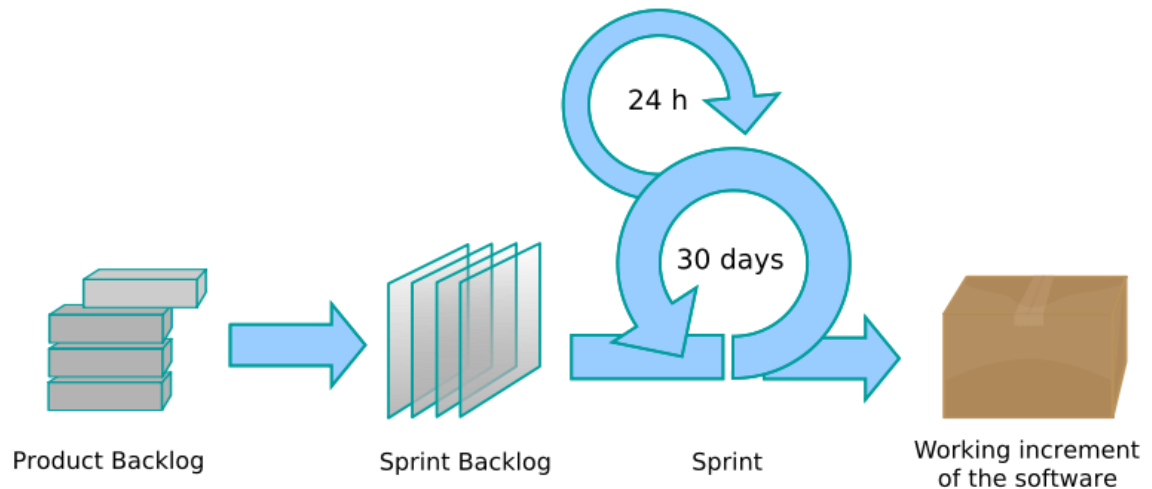


Figure 1 : Schéma du cycle de Scrum³

2.3.1. Backlog de produit

Le « backlog de produit » est une liste de fonctionnalités à réaliser, qui représentent les besoins des utilisateurs et à chaque fonctionnalité on associe deux attributs: une estimation en points et une valeur client. Celle-ci est définie par le directeur de produit, par exemple le « retour sur investissement ».

Ce dernier définit également l'ordre dans lequel ces fonctionnalités seront réalisées, ordre qui peut être changé en cours de projet en ajoutant, en modifiant voir en supprimant certaines de ces fonctionnalités.

La somme des estimations permet finalement de connaître ce qu'il reste à réaliser dans le projet, mais aussi de générer des graphiques nommés « Release Burndown Charts ».

2.3.2. Backlog de sprint

Dans une itération, il est nécessaire de choisir les fonctionnalités du backlog product qui seront développées. Ces fonctionnalités sont alors découpées en plusieurs tâches

² **Source** : Wikipedia, Scrum, consulté le 13/08/2010, <http://fr.wikipedia.org/wiki/Scrum>

³ **Source** : Document PPT SCRUM, cours 616 de Müller Henning, consulté le 13/08/2010



élémentaires pour lesquelles le développeur doit estimer le temps nécessaire à leur réalisation. Le « Sprint Backlog » est donc une liste de tâches découpées et leurs estimations sont faites en heures et non en points, de 4 à 16 heures. Ces estimations permettent de produire un graphique prénommé « Sprint Burndown Chart » représentant le nombre d'heures restantes à effectuer pour une itération.

2.3.3. Sprints

Les itérations, ou sprints, durent théoriquement 30 jours. En pratique, on va plutôt définir ces itérations entre une à quatre semaines, mais ces choix dépendent du type de projet à réaliser et du type de personnes ou sociétés qui les utilisent.

Chaque tâche est composée d'un nombre d'heures pour sa réalisation, ainsi que d'une description à titre d'explications. Lorsqu'une itération et ses fonctionnalités ont été définies, on ne peut plus les modifier si le sprint est en cours, mais on peut le faire dans une prochaine itération si besoin. Il est tout de même possible de modifier une fonctionnalité en cours, d'un commun accord entre l'équipe de développement et le directeur du produit.

2.3.4. Release

En principe, on va regrouper toutes les itérations dans une « release », ou un livrable, afin de permettre une meilleure lisibilité du projet. Cette phase n'est pas obligatoire dans Scrum, mais permet de marquer plus facilement le livrable réalisé à partir de différents sprints.

2.3.5. Quotidien

« Au quotidien, une réunion, le ScrumMeeting, permet à l'équipe et au ScrumMaster de faire un point d'avancement sur les tâches et sur les difficultés rencontrées. »

Source : Wikipedia

Cette caractéristique de la méthode Scrum est très importante, car elle permet d'apporter une indication rapide et quotidienne sur l'avancement des tâches et itérations en cours de développement, mais surtout de maintenir un certain état d'esprit professionnel.

En effet, un employé n'est pas supposé connaître les détails de tous les projets, itérations ou tâches réalisées par ses collègues, mais cela permet au moins de recentrer l'intérêt des membres sur l'entreprise et de renforcer l'esprit d'équipe, en

impliquant tout le monde sur tous les projets en cours.

2.4. Rôles⁴

Dans Scrum, il existe trois rôles principaux qui sont le directeur de produit, l'équipe et le facilitateur, mais ponctuellement des intervenants peuvent aussi être assignés à des projets.

2.4.1. Directeur de produit

Le directeur de produit est l'entité qui représente les clients et utilisateurs et définit l'ordre du développement des fonctionnalités. Le mot « directeur » n'est pas à prendre dans le sens de la hiérarchie, mais dans le sens où c'est à lui de prendre la majorité des décisions concernant l'orientation du projet.

Dans le meilleur des cas, le directeur de produit doit être intégré à l'équipe de développement, afin qu'il puisse très rapidement répondre aux questions et donner son avis sur certaines phases du projet, par exemple au niveau des prototypes écrans.

2.4.2. Équipe

L'équipe représente tous les développeurs et ne doit pas comporter de hiérarchie interne. En effet, chaque membre du groupe possède les mêmes responsabilités et personne n'est censé donner d'ordre à un autre membre. En outre, l'équipe s'adresse au directeur de produit, qui doit être informé le plus souvent possible de l'avancement du développement afin de pouvoir effectuer des changements sur certaines phases en temps voulu.

2.4.3. Facilitateur

Le facilitateur ou « ScrumMaster » a un rôle capital dans Scrum, car c'est lui qui doit protéger l'équipe contre les perturbations extérieures et résoudre le mieux possible les problèmes non techniques, par exemple tout ce qui concerne l'administration. Par contre il n'est pas chef et ne sert pas d'intermédiaire avec le client.

2.4.4. Intervenants

Les intervenants ou « Stakeholders » sont des personnes externes qui peuvent avoir

⁴ Source : Wikipedia, Scrum, consulté le 13/08/2010, <http://fr.wikipedia.org/wiki/Scrum>

un aperçu de l'avancement d'un projet sans y être actives, par exemple des experts ou des membres de la direction de l'entreprise.

2.5. Synthèse

Pour conclure ce chapitre sur la méthodologie, il est nécessaire d'avoir une récapitulation des caractéristiques à implémenter dans l'outil de gestion.

D'une part, il faut veiller à ce que l'application permette une gestion en « sprints », sans pour autant suivre à la lettre toutes les caractéristiques propres à Scrum comme le « Product Backlog ».

Par exemple, il est intéressant d'imaginer, dans un projet en cours, pouvoir définir des itérations qui correspondent à des phases importantes de celui-ci et dans ces itérations, définir une série de tâches à réaliser pour parvenir au bout du sprint.

Une autre caractéristique intéressante de Scrum est le « ScrumMeeting », qui est déjà pratiqué au sein de l'institut Icare. En effet, pratiquement chaque jour un meeting a lieu, dans lequel chaque membre exprime le travail effectué et à venir.

Même si cette caractéristique n'a rien à faire dans une application, elle reste cependant intéressante pour un outil de gestion dans le sens où chaque membre doit pouvoir analyser de manière rapide les tâches à venir et celles déjà réalisées, afin d'en faire part de façon aisée pendant une séance de groupe, voire à tout autre moment.

3. Technologies

3.1. Ruby⁵⁶

3.1.1. Définition

Ruby fut inventé au Japon, par Yukihiro "Matz" Matsumoto en 1993 et la première version fut publiée en 1995. L'auteur souhaitait avoir un langage à la fois simple et flexible comme Python et Perl. Par ailleurs, le nom Ruby est un clin d'œil au mot Perl, mais pas un acronyme.

Ruby est un langage de script orienté objet, reste un langage de programmation conçu pour le web et est totalement libre. Une autre caractéristique de ce langage est qu'il est entièrement orienté objet, mais reste considéré comme un langage de programmation procédurale.

La programmation procédurale est un paradigme de programmation qui se base sur le concept des appels procéduraux. Une procédure est une série d'étapes à réaliser et n'importe quelle procédure peut être appelée à tout moment.

La programmation procédurale est souvent un meilleur choix qu'une simple programmation séquentielle ou programmation non-structurée. Elle a l'avantage de permettre une réutilisation d'un même code à différents emplacements dans l'application ou de créer un code plus modulaire et structuré.

3.1.2. Caractéristiques

Ruby se trouve à l'adresse suivante : http://rubyforge.org/frs/?group_id=167 et ses fonctionnalités principales sont les suivantes :

- Class représentée par « class » à « end »
- Module représenté par « module » à « end »
- Fonction représentée par « def » à « end »
- Orientation objet

⁵ **Source** : aysoon.fr, Ruby On Rails – dossier complet pour tout comprendre sur cette techno phare du web2.0, consulté le 14/08/2010, <http://aysoon.fr/66-ruby-on-rails-dossier-complet-pour-tout-comprendre-sur-cette-techno-phare-du-web20>

⁶ **Source** : wikipedia.org, Ruby, consulté le 14/08/2010, <http://fr.wikipedia.org/wiki/Ruby>

- Libération automatique de la mémoire
- Héritage simplifié
- Inclusion de modules dans des classes, ou héritage multiple
- Extensions en C relativement simple
- Entiers de taille illimitée avec conversion automatique
- Les variables :
 - « var » est une variable locale
 - « @var » est une variable d'instance
 - « @@var » est une variable de classe
 - « \$var » est une variable globale
 - « Var » est une constante
 - ...

3.1.3. RubyGems⁷

RubyGems est le standard de Ruby pour la gestion et l'édition de bibliothèques tierces et se trouve à l'adresse suivante : http://rubyforge.org/frs/?group_id=126

Ses fonctionnalités les plus importantes sont :

- Installation à distance de paquets
- Gestion des dépendances
- Désinstallation facile et fiable
- Fonctionnalités de requête, locale et distante
- Possibilité d'avoir plusieurs versions du même paquet installées
- Interface web pour visualiser la documentation des gems

Un Gem est un paquet d'une bibliothèque ou application Ruby, et chaque Gem possède un nom, par exemple « rails » et une version « 2.3.5 ». Ces Gems sont gérés directement dans le shell, par exemple pour installer le Gem « rails », il suffit de lancer la commande « gem install rails ». De même que pour le désinstaller en faisant « gem uninstall rails ».

⁷ **Source** : rubyfrance.org, RubyGems - Introduction, consulté le 14/08/2010, <http://www.rubyfrance.org/documentations/rubygem---introduction/>

Plusieurs autres possibilités sont offertes grâce à RubyGems, comme le fait de pouvoir « freezer » un Gem. En effet, lorsque l'utilisateur installe un Gem, celui-ci s'installe dans un dossier prédéfinis sur le disque local de l'utilisateur. Mais il doit également être installé sur le serveur de production où le site est hébergé.

Cependant, en freezant un Gem, il est directement inclus dans les fichiers du site web. Cela permet de modifier des Gems à sa convenance et de facilement pouvoir les utilisés tels quels sur n'importe quel autre environnement.

3.1.4. Quelques exemples Ruby

- Afficher « Hello World » :

```
puts "Hello World"
```

- Objets :

```
"ruby is cool".length -> 12  
"Rick".index("c")     ->  2
```

- Fonction :

```
def eat  
  puts "I'm eating !"  
end
```

- Héritage :

```
module AnimalSkills  
  def eat  
    puts "I'm eating !"  
  end  
end  
class Animal  
  include AnimalSkills # héritage multiple  
end  
class Zebra < Animal # héritage  
end
```

3.1.5. Ruby on Rails (RoR)

Malgré toutes les qualités de Ruby, il lui manquait un Framework puissant pouvant exploiter au mieux toutes ses fonctionnalités : Rails est né.

Rails provient d'un outil de gestion de projets appelé « Basecamp », développé par David Heinemeier Hansson, et datant de juillet 2004. La première version stable de Rails fut lancée le 14 décembre 2005.

3.1.6. Utilité de RoR

RoR permet de créer des sites web rapidement en fournissant divers outils, mais permet surtout d'améliorer la productivité grâce à :

- Un code plus concis, évolutif et produit rapidement
- À une configuration minimale
- À des technologies déjà intégrées comme « Ajax »

Par exemple, il est possible grâce à Rails de créer un petit site internet sans écrire une seule ligne de code. Simplement en utilisant des outils qui automatisent les tâches les plus classiques comme la création de formulaires ou la gestion de la base de données.

3.1.7. Environnement de travail de Rails

Rails est entièrement interopérable, car il est compatible Windows, MacOS et Linux. Tout sur Rails se fait en mode console, shell ou terminal, mais l'on trouve également des logiciels tel que Aptana RadRails permettant un développement simplifié sur de plus gros projets.

3.1.8. Architecture Modèle-Vue-Contrôleur

Rails s'articule autour d'une architecture MVC ou Model-View-Controller et fournit des outils pour la construire très facilement. Cette architecture permet une réutilisation facile du code, qui est séparé de manière logique selon la fonction qu'on lui attribue.

Modèle :

Les modèles s'occupent de la gestion des données et sa structure est déterminée automatiquement par rapport à une table de la base de données portant le même nom que le modèle. On peut cependant spécifier les relations dans ces modèles, par exemple en indiquant qu'un « utilisateur » possède un ou plusieurs « livres » en utilisant la relation « has_many » (user has_many books), ce qui permet de récupérer tous les livres d'un utilisateur.

Vue :

Les vues concernent les interfaces utilisateur de l'application, et permettent de mélanger du HTML et du Ruby dans les fichiers qui portent l'extension « .RHTML » ou « .HTML.ERB ». En plus d'HTML, il est possible d'utiliser tout autre langage web comme du Javascript, de l'Ajax, ou des bibliothèques comme jQuery ou Prototype.

Contrôleur :

Les contrôleurs contiennent toutes les actions possibles de l'utilisateur et sont la liaison entre les vues et les modèles. Plus précisément, les contrôleurs récupèrent les données des modèles et les fournissent aux vues.

3.1.9. Conclusion

Ayant suivi la majorité de ma formation sur le langage Java, il n'est pas très aisé de passer à Ruby étant développeur Java, car même si ces deux langages sont orientés objets, ils n'ont pas beaucoup de points communs.

En effet, afin d'utiliser au mieux toutes les fonctionnalités de RoR, il est inévitable d'avoir effectué au préalable un apprentissage approfondi. Heureusement, RoR possède une communauté qui grandit de jour en jour et qui permet au développeur de s'y former beaucoup plus facilement qu'auparavant.

Pour finir, les côtés open source et flexibles de Ruby lui permettent d'être remarquablement efficace pour installer une bibliothèque existante. Cette faculté peut aussi se retourner contre soi dans le sens où c'est la communauté RoR qui gère ces bibliothèques, avec parfois des problèmes de compatibilité entre différents Gems. Cela implique une bonne gestion des différentes versions d'un Gem.

http://aysoon.fr/66-ruby-on-rails-dossier-complet-pour-tout-comprendre-sur-cette-techno-pahre-du-web20#ruby_histoire

3.2. Ajax⁸

Ajax signifie « Asynchronous Javascript and XML » et permet de mettre à jour dynamiquement une page web. Ajax n'est pas vraiment une technologie à proprement parler, mais représente la fusion de plusieurs technologies du web :

⁸ **Source :** [wikipedia.org, Asynchronous JavaScript and XML](http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_and_XML), consulté le 14/08/2010, http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_and_XML

- HTML (ou XHTML) pour la structure sémantique des informations ;
- CSS pour la présentation des informations ;
- DOM et JavaScript pour afficher et interagir dynamiquement avec l'information présentée ;
- l'objet XMLHttpRequest pour échanger et manipuler les données de manière asynchrone avec le serveur Web.
- XML pour remplacer le format des données informatives (JSON) et visuelles (HTML).

3.3. RJS⁹

Les RJS, qui signifient « Remote JavaScript », sont des vues écrits en Javascript et en Rails et servent à mettre dynamiquement à jour une page grâce à Ajax. Ils sont considérés comme des vues dans l'architecture MVC et sont définis avec l'extension « .RJS ».

Afin d'en faire usage, il faut utiliser dans une vue une fonction se prénommant « link_to_remote » en lui spécifiant le nom du « .rjs » souhaité. Ce lien lance une requête Ajax qui va d'abord lire la fonction du contrôleur portant le même nom que le .rjs, et, une fois la fonction effectuée, va lire le « .rjs » correspondant.

De ce fait, la fonction du contrôleur met à jour les données disponibles pour la vue, et les données affichées dans la vue sont mises à jour à l'aide du « .rjs » sans que toutes les autres données ne soient rechargées.

Cela permet d'avoir des interfaces extrêmement dynamiques, par exemple en imaginant qu'un utilisateur souhaite rajouter ou effacer des lignes dans un tableau, grâce à Ajax et RJS. Cela mettra à jour la vue en ajoutant ou en supprimant cette même ligne sans avoir besoin de recharger toutes les autres lignes déjà affichées.

Bien que cette technologie soit efficace et très intéressante pour une interface web, elle reste plus compliquée à mettre en place que la technique habituelle consistant à recharger la page, car elle nécessite du code supplémentaire afin de mettre à jour uniquement certaines données de la vue.

⁹ **Source** : pbnaigeon.developpez.com, Les RJS (Remote Javascript), consulté le 14/08/2010, <http://pbnaigeon.developpez.com/tutoriel/rails/ajax-facile-RJS/#LIII>

3.4. **Prototype**

Prototype est un Framework Javascript rendant son utilisation plus facile, en fournissant une grande quantité de méthodes, propriétés et fonctions au développeur. Une liste de ces fonctionnalités se trouve à l'adresse suivante : <http://prototype.js.le-developpeur-web.com/index.php>

3.5. **Script.Aculo.Us**

Scriptaculous est aussi un Framework Javascript, qui utilise en partie certaines fonctionnalités de Prototype. Cette librairie a pour but principal de fournir au développeur une aide afin de créer des effets, de générer des éléments HTML et DOM, d'utiliser de l'Ajax facilement, de rendre déplaçable certains éléments d'une page web avec du Drag&Drop, ainsi que d'autres fonctionnalités très intéressantes. Plus d'informations sont disponibles sur cette adresse : <http://scriptaculous.le-developpeur-web.com/>

3.6. **jQuery**

Tout comme Prototype et Scriptaculous, jQuery est également un Framework Javascript et est en quelque sorte le concurrent direct de Prototype. jQuery fournit également une grande quantité de fonctionnalités permettant d'améliorer les interfaces utilisateurs, par exemple en utilisant des librairies d'effets visuels. On trouve une liste de fonctionnalités à cette adresse : <http://jquery.com/>

4. Etat de l'art

4.1. Introduction

Dans cette phase du projet, il est nécessaire de déterminer si un outil de gestion de projet existant est adapté à la méthodologie choisie et aux technologies décrites ci-dessus, et s'il convient pour une implémentation idéale au sein de l'institut Icare.

Pour cela, il faut effectuer un état de l'art dans ce domaine afin de définir une série de solutions disponibles sur le marché, en les analysant et en les comparant. Afin de réaliser cette étape importante du travail de Bachelor, il faut d'abord définir les critères qui vont influencer le choix final.

Bien que le langage Ruby soit une condition requise pour cet outil de gestion, certaines applications sélectionnées dans l'état de l'art ont été développées sur un autre langage. C'est par exemple le cas pour l'outil de gestion « Intra-Know », développé sur le langage PHP et qui est tout de même analysé, car certaines de ses fonctionnalités sont intéressantes.

Ces applications ne sont donc pas toutes considérées comme potentiellement acceptables pour une implémentation dans ce projet mais servent d'exemples, voire de contre-exemples aux outils développés.

Bien que les applications Ruby soient modifiables à souhait, il est important d'examiner le niveau d'extensibilité des logiciels afin de déterminer si ceux-ci peuvent être plus ou moins facilement ajustés en fonction des demandes.

Un autre critère important pour les applications open source est la quantité et la qualité de la communauté autour de l'outil, car cela permet un avancement plus aisé au niveau de l'implémentation, tout en garantissant que l'outil jouisse d'une certaine avancée technique et d'une stabilité éprouvée.

D'autres critères sont également pris en compte dans l'analyse, comme le niveau d'agilité, la complexité de l'ergonomie ou encore la quantité des fonctionnalités existantes au sein de ces outils.

4.2. Critères

Les critères sélectionnés pour l'état de l'art sont donc les suivants :

- Agilité
- Extensibilité
- Ergonomie
- Fonctionnalités
- Communauté

Ces différents critères sont pondérés d'une note allant de 0 à 10, 0 étant la note la plus mauvaise et 10 la meilleure.

Certains de ces critères peuvent aussi être susceptibles d'être liés à d'autres critères, par exemple la quantité des fonctionnalités qui peut influencer sur l'ergonomie de l'application ou encore le niveau d'extensibilité de l'outil qui peut dépendre de la communauté présente.

D'autre part, le niveau d'agilité élevé d'une application ne signifie pas que celui-ci soit optimal, étant donné que le but premier est d'avoir la possibilité d'utiliser certaines fonctionnalités de la méthodologie Scrum, mais en aucun cas de vouloir utiliser toutes ses caractéristiques.

Il est donc nécessaire de déterminer l'ordre et la priorité de ces critères afin de faire un choix final. L'ordre d'importance des critères est donc le suivant : le niveau d'extensibilité et la communauté disponible, le niveau d'agilité et de l'ergonomie et finalement les fonctionnalités conformes à l'attente du client.

Ce choix se fait par rapport aux objectifs que l'on souhaite atteindre avec cet outil de gestion : Extensibilité, Agilité, Simplicité, Rapidité et Stabilité.

4.3. Outils de gestion existants

4.3.1. Introduction

Parmi les outils disponibles sur le marché, il n'est pas forcément évident de les noter selon des critères très précis. Cependant et afin d'arriver à un résultat réaliste, l'estimation de ces notes s'est faite après avoir testé les applications, dans la mesure du possible, tout en ayant regardé des vidéos et lu diverses explications et spécifications.

Le choix des outils analysés s'est fait par rapport aux recommandations de l'institut Icare, qui a cité quelques noms comme Redmine, Intra-Know ou VersionOne. Les autres outils choisis l'ont été par jugement personnel et suite à de nombreuses recherches sur internet.

4.3.2. Redmine¹⁰

Redmine est une application web développée sous langage Ruby et Framework Rails, entièrement open source et disponible à l'adresse suivante : <http://www.redmine.org/>

Cet outil possède une énorme communauté qui l'améliore constamment. Le site web de Redmine est en fait l'application en elle-même, et étant donné que ce projet soit entièrement libre, chacun peut participer à son évolution en contribuant au développement de nouvelles fonctionnalités.

Pour preuve d'exemple, en visitant le site web, on peut apercevoir l'évolution des prochaines versions de l'outil ainsi que les fonctionnalités, bugs et imperfections qui vont être ou ont été développées.

Tout un chacun peut également proposer des améliorations ou des corrections, et l'on trouve actuellement 2'258 demandes ouvertes. Bien évidemment, toutes ces tâches ne sont pas susceptibles d'être utilisées, certaines étant farfelues ou tout simplement incompréhensibles. Néanmoins, cela prouve le nombre de communauté présente autour de l'outil.

En allant explorer les forums, il saute également aux yeux que la qualité de la communauté est aussi présente, car elle est principalement composée de développeurs qui travaillent en commun à l'élaboration de nouvelles fonctionnalités et autres corrections et qui s'organisent grâce à l'outil de gestion. En somme, les tâches sont

¹⁰ Source : redmine.org, consulté le 13/08/2010, <http://www.redmine.org>

planifiées et attribuées à certains bénévoles, comme s'il s'agissait d'une entreprise.

Le niveau d'agilité de l'outil est par contre peu évolué. Néanmoins il reste suffisant en proposant l'utilisation de versions qui peuvent être comparées à des itérations et au fait qu'il soit multi-projets.

Redmine propose également sur son site une explication détaillée des différentes possibilités de modification de l'outil, tout simplement en permettant de créer ses propres plugins.

L'ergonomie de l'application est très basique et simple d'utilisation, même si elle demande à être améliorée sur certaines parties, comme par exemple le fait de trouver le bon lien qui nous amènera sur la page souhaitée.

Pour finir, le nombre de fonctionnalités disponibles est bien fournie, mais sans être obsolète, puisque l'outil propose des fonctionnalités comme un Wiki, un forum, la gestion des demandes par projet, la gestion des utilisateurs / groupes et rôles, mais aussi des outils de reporting, d'aperçu d'avancement ainsi que la gestion des emails ou encore des repositories.

4.3.3. Icescrum¹¹

IceScrum est une application J2EE (Java) et basée sur Scrum. Son niveau d'agilité est par conséquent très élevé, mais l'ergonomie est, quant à elle, peu évidente et rend difficile une utilisation du logiciel. L'application propose beaucoup de fonctionnalités, mais uniquement basées sur la méthodologie agile et la communauté est très peu présente, le forum étant pratiquement vide.

Par contre, il est assez facile de récupérer le code source de l'application et de pouvoir la modifier : <http://www.icescrum.org/>

4.3.4. Intra-Know¹²

Intra-Know est une solution intranet collaborative, mais fonctionnant sur un site externe <http://www.intraknow.com/>, développé en PHP étant propriétaire et payant.

Malgré tout, il est analysé car c'est l'outil qu'utilise actuellement l'institut Icare pour la gestion de ses projets. Cette application propose beaucoup de fonctionnalités, comme le planning, la gestion du calendrier (absence / congé), un plan de charges, un reporting détaillé ou encore la gestion des documents et du budget.

¹¹ **Source** : icescrum.org, consulté le 13/08/2010, <http://www.icescrum.org/>

¹² **Source** : intra-know.com, consulté le 13/08/2010, <http://www.intraknow.com/>

L'interface est plutôt bien réalisée, avec des effets visuels et une bonne qualité graphique. Par contre l'ergonomie est à revoir car il est assez compliqué de s'y retrouver et certains points pourraient être améliorés, comme la partie du planning, qui se trouve être assez complexe.

La communauté autour de l'outil est quant à elle peu présente, étant donné que l'outil est propriétaire et payant. Malgré cela, il jouit évidemment d'un support de la part de la société le proposant, mais cela n'entre pas en ligne de compte. De même que l'extensibilité qui est inapplicable.

Finalement, le niveau d'agilité semble suffisant, car il est possible de placer les tâches d'un projet dans des étapes (itérations) et la présence d'un plan de charges facilite grandement la vision future du travail à effectuer. Malgré cela, toutes les estimations sont faites en jours et non pas en heures, ce qui ne permet pas une précision pour des tâches de 4 à 16 heures.

4.3.5. ExplainPMT¹³

ExplainPMT est un outil open source, sous Framework Ruby On Rails, permettant de planifier un projet avec un niveau d'agilité très poussé. La méthodologie agile utilisée est XP, ou eXtreme Programming, et ressemble de près à la méthodologie Scrum. Malgré le fait que ce niveau d'agilité soit très évolué, il ne correspond pas à ce qui est recherché, c'est-à-dire un niveau d'agilité plus basique et n'entraînant pas de changements majeurs pour l'institut Icare.

D'autre part, l'interface de l'outil est trop simpliste et austère, et il y a très peu de fonctionnalités disponibles, mis à part celles destinées uniquement à un usage restrictif de la méthode agile, comme par exemple le fait de pouvoir créer des « user stories ».

La communauté autour d'ExplainPMT reste active, car l'on trouve des versions récentes, malheureusement celle-ci n'est pas très grande et l'outil ne possède même pas de site web dédié. Afin de récupérer cette application, il faut aller à l'adresse suivante : <http://github.com/explainpmt/explainpmt>

Pour finir, on ne trouve aucune information sur l'extensibilité de l'outil, mis à part le fait qu'il soit open source et sur langage Ruby, ce qui permet tout de même de pouvoir le modifier à sa convenance.

¹³ **Source** : github.com, ExplainPMT, consulté le 13/08/2010, <http://github.com/explainpmt/explainpmt>

4.3.6. VersionOne¹⁴

VersionOne est le dernier outil analysé et comparé, disponible à cette adresse : <http://www.versionone.com>. Cette application est très évoluée et propose une multitude de fonctionnalités afin de gérer plusieurs projets. Ce logiciel est propriétaire, mais il est possible de l'utiliser gratuitement pour un groupe de dix personnes au maximum, tout en ayant une limitation dans le nombre de fonctionnalités utilisables.

Cet outil possède un site web dédié à la communauté, qui semble être très active et de très bonne qualité. Il est également possible de télécharger le code source de l'application, pour lui apporter des modifications. Malheureusement, il n'est possible de le faire qu'en Java et .NET, et non pas en Ruby.

Le niveau d'agilité de l'application est très impressionnant, puisque l'outil permet un support de la plupart des méthodologies agiles courantes comme Scrum, XP, Kanban, AgileUP ou encore DSDM. Malgré cela, l'interface et l'utilisation de l'outil est très complexe et demande beaucoup de temps d'adaptation afin d'être capable d'utiliser toutes ses fonctionnalités. Cet outil est puissant, stable et fournis en possibilités, mais le fait qu'il le soit autant ne le rend pas optimal pour une implémentation dans l'institut Icare, qui recherche une solution plus simple et plus rapide à utiliser.

4.3.7. Trac¹⁵

Trac est un outil de gestion développé en Python, open source et disponible à cette adresse : <http://trac.edgewall.org/>

Cette application est plutôt minimaliste, et comporte assez peu de fonctionnalités, uniquement un Wiki et la possibilité de lister des tâches avec des milestones (itérations). L'outil est très proche de Redmine au niveau ergonomique et fonctionnel, car il propose également un Roadmap ou un Timeline, et l'interface ressemble parfois à celle de Redmine.

Malheureusement, cet outil ne propose pas de gestion multi-projets, ce qui est un gros manque et compromet sa possible implémentation pour ce travail. Malgré cela, la communauté semble bien active, le site web étant également le logiciel en lui-même, sur lequel il est possible de voir l'avancement des prochaines versions.

¹⁴ **Source** : versionone.com, consulté le 13/08/2010, <http://www.versionone.com>

¹⁵ **Source** : trac.edgewall.org, consulté le 13/08/2010, <http://trac.edgewall.org/>

4.4. Synthèse

4.4.1. Compte rendu

Voici la répartition des points :

	Extensibilité	Agilité	Fonctionnalités	Ergonomie	Communauté
Redmine	8	5	6	6	9
IceScrum	5	8	7	4	2
Intra-Know	0	6	8	5	1
ExplainPMT	4	8	2	3	2
VersionOne	6	9	9	5	6
Trac	7	2	4	6	7

Et ci-après un graphique permettant de comparer le résultat obtenu :

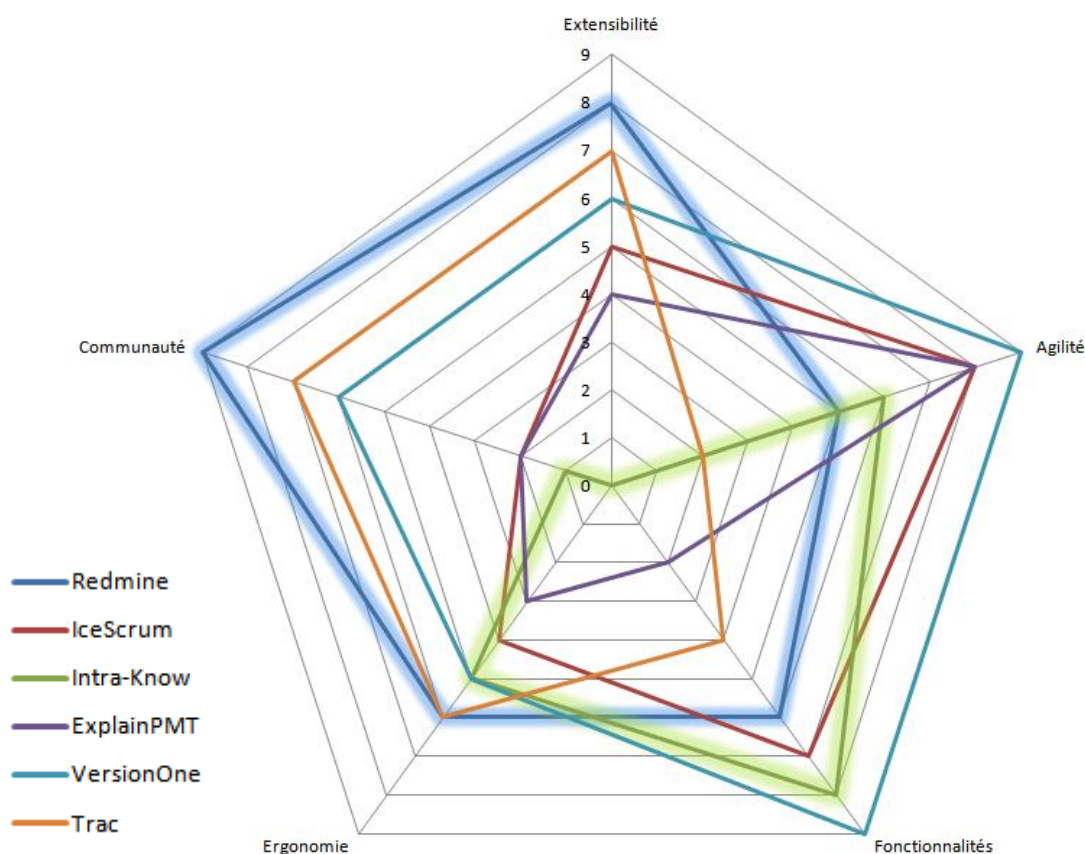


Figure 2 : Graphique radar des résultats de l'état de l'art

Le trait bleu prononcé identifie Redmine comme étant l'outil remportant la majorité des points, et le trait vert l'outil actuellement utilisé par Icare qui est Intra-Know

4.4.2. Choix de la solution à implémenter

Grâce aux résultats, on constate que l'outil le plus adapté à une implémentation est Redmine, étant donné qu'il remplit la majorité des critères, excepté le niveau d'agilité. Mais comme expliqué plus haut, ce critère est un peu spécial étant donné que le but premier d'un outil est la saisie de demandes avec une utilisation d'itérations ou de milestones, ce qu'il est possible de faire.

Le critère le plus important de Redmine est donc sa communauté qui est de très bonne qualité et très présente, mais également la faculté de pouvoir lui greffer des plugins personnalisés.

Le nombre de ses fonctionnalités est également bon et le fait qu'il soit quelque peu en dessous de certains autres outils comme VersionOne ou Intra-Know ne pose pas de problèmes. Il est possible d'ajouter celles qui manquent cruellement.



5. Description du projet

5.1. Redmine¹⁶¹⁷¹⁸

5.1.1. Définition

Comme expliqué dans l'état de l'art page 17, Redmine est un outil web de gestion de projet, incluant quelques caractéristiques des méthodologies agiles. Cette application Ruby On Rails, créée par Jean-Philippe Lang, date déjà de plusieurs années et la dernière version est la 1.0.0 Release Candidate.

Le menu de Redmine est plutôt simple et permet d'accéder aux interfaces principales de l'application, de rechercher ou d'aller directement à un projet et de se connecter ou déconnecter.

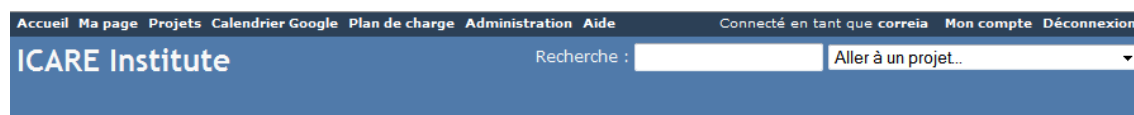


Figure 3 : Menu principal de Redmine

Fondamentalement, Redmine est une application permettant de lister des demandes à réaliser sur un projet, et de les assigner aux ressources disponibles. Cependant, cet outil propose toute une gamme de fonctionnalités supplémentaires, permettant une gestion de projet plus souple et plus complète.

5.1.2. Fonctionnalités existantes

Afin de mieux comprendre les possibilités qu'offre Redmine, se référer aux fonctionnalités existantes dans les annexes pour une explication détaillée.

5.2. Fonctionnalités à ajouter

5.2.1. Introduction

Quelques séances ont eu lieu avec les développeurs et le chef de projet de l'institut Icare, afin de discuter des attentes de chacun pour une implémentation optimale de

¹⁶ Source : redmine.org, consulté le 13/08/2010, <http://www.redmine.org>

¹⁷ Source : projet-plume.org, Redmine : gestionnaire de projets, consulté le 13/08/2010, <http://www.projet-plume.org/fr/fiche/redmine>

¹⁸ Source : wikipedia.org, Redmine, consulté le 13/08/2010, <http://fr.wikipedia.org/wiki/RedMine>



l'outil.

Comme ils utilisent quotidiennement Intra-Know, certains développeurs ont des avis tranchés sur les points positifs et négatifs de cet outil, ce qui permet d'établir une liste de fonctionnalités à ajouter sur Redmine.

Les informations glanées sont essentielles pour mon travail, car d'une part elles permettent de préciser le cahier des charges, mais surtout de planifier la phase de développement du projet.

5.2.2. Fonctionnalités principales

Plan de charges

Le chef de projet d'Icare a signalé que la fonctionnalité la plus importante est celle du plan de charges. On trouve ce plan de charges dans l'application Intra-Know et cette interface permet de visualiser l'estimation du travail à réaliser pour les semaines à venir.

Cette interface est essentielle, car elle permet de contrôler le temps de travail prévu pour toutes les ressources. Il a également été signalé que cette fonctionnalité est prioritaire, et qu'une fois qu'elle sera implémentée, tous les projets pourront être migrés sur l'application Redmine en lieu et place d'Intra-Know.

Reporting par semaine

Le reporting est également essentiel, mais existe déjà dans l'outil Redmine. Grâce à lui, il est possible d'avoir un aperçu des heures réalisées par chaque ressource, tout en ayant la possibilité d'utiliser une gamme de filtres pour trier les données affichées. Cependant, il n'existe pas d'interface permettant de visualiser rapidement le nombre d'heures effectuées pour une semaine donnée, cette fonctionnalité est donc à rajouter.

Calendrier

Il faut également prévoir une interface pour la gestion des congés et des absences. Redmine ne permet pas une gestion de la disponibilité des ressources, et c'est un gros manque pour une planification réaliste dans un plan de charges.

Budget

Le chef de projet souhaite également pouvoir gérer le budget des projets et des ressources grâce à Redmine, comme c'est le cas avec l'outil Intra-Know. Bien que des plugins existent déjà pour une gestion d'un budget sous Redmine, ceux-ci ne sont



pas conformes à l'attente, c'est-à-dire une gestion simple du budget, par projet et par ressource, avec la possibilité d'ajouter des coûts annexes.

5.2.3. Fonctionnalités secondaires

Notifications par email

La possibilité de notifier par email les ressources lors de changements dans le planning est une fonctionnalité présente dans Redmine, seulement à l'heure actuelle elle ne fonctionne pas.

Mise à jour des tâches depuis un commit

Certains membres aimeraient pouvoir mettre à jour des tâches ouvertes, par exemple en indiquant le nombre d'heures passées sur une tâche ou l'avancement de celle-ci, directement depuis le message log d'un commit SVN. Un commit SVN est le fait de mettre à jour le code source sur une zone de partage afin que plusieurs ressources puissent travailler sur un même code.

Drag&Drop

La fonctionnalité du Drag&Drop, ou cliquer-glisser-déposer, serait très pratique afin de réorganiser la planification dans le plan de charges de manière rapide, et serait un gros atout pour une partie des membres.

Graphiques

L'ajout de graphiques, par exemple un Burndown Chart, serait également une fonctionnalité possible à implémenter, mais elle reste non prioritaire car ce n'est pas d'une importance capitale aux yeux de la plupart des membres de l'institut.

Modifications diverses

D'autres personnes ont signalé quelques modifications à apporter à l'outil Redmine, comme par exemple la possibilité de rajouter des liens personnalisés ou d'améliorer l'ergonomie de l'outil.



5.3. Prototypes écran

Des prototypes écran ont été réalisés pour les fonctionnalités principales et furent validés auprès de tous les membres d'Icare afin d'obtenir le feu vert pour le développement.

5.3.1. Plan de charges

Par ressource

Le prototype écran du plan de charges par ressource permet de visualiser toutes les tâches ouvertes d'une ressource. Les tâches sont réparties par projet, par version (itération) et le détail affiché indique le temps déjà effectué sur la tâche, le temps planifié et l'avancement.

Pour chaque semaine et à partir de la semaine courante, le temps planifié moins le temps passé est réparti en fonction des dates de début et/ou d'échéance.

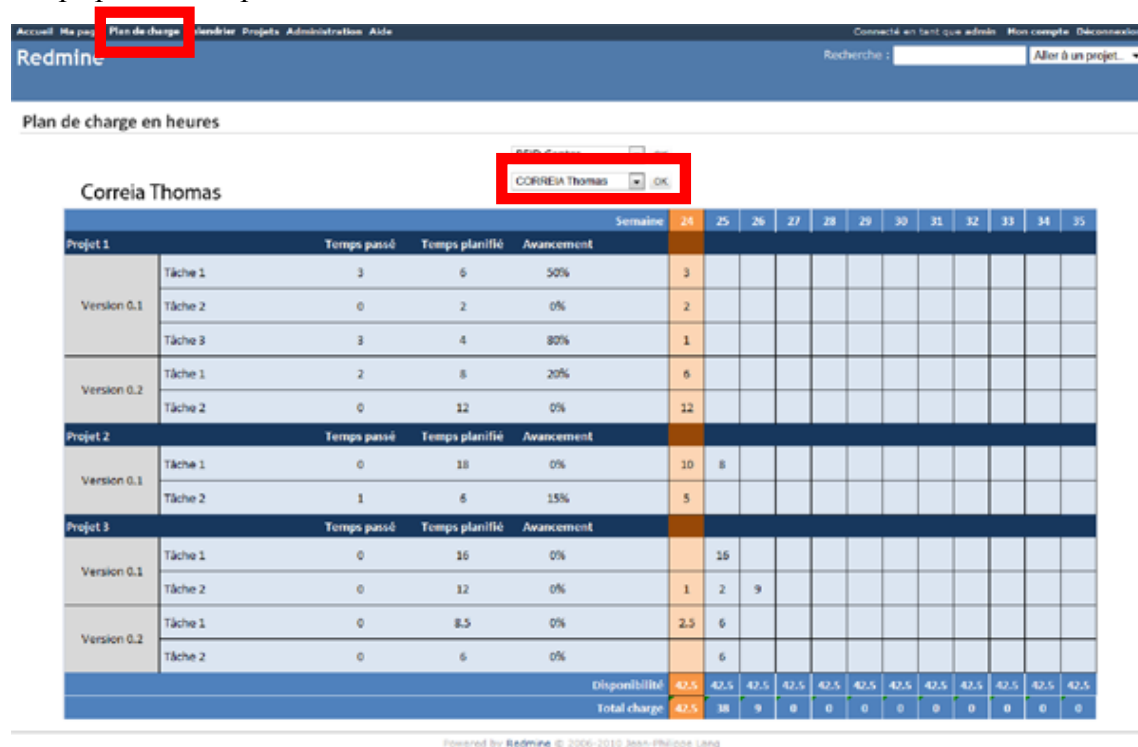


Figure 4 : Prototype “Plan de charges – Par ressource”



Par groupe

Toujours pour le plan de charges, cette deuxième interface représente tout un groupe de ressources. Ici, le détail des tâches n'est plus affiché, mais uniquement l'ensemble des projets dans lesquels la ressource a des tâches ouvertes et assignées à sa personne. Le temps passé et planifié ainsi que l'avancement représentent alors la concaténation des différentes tâches de ce projet.

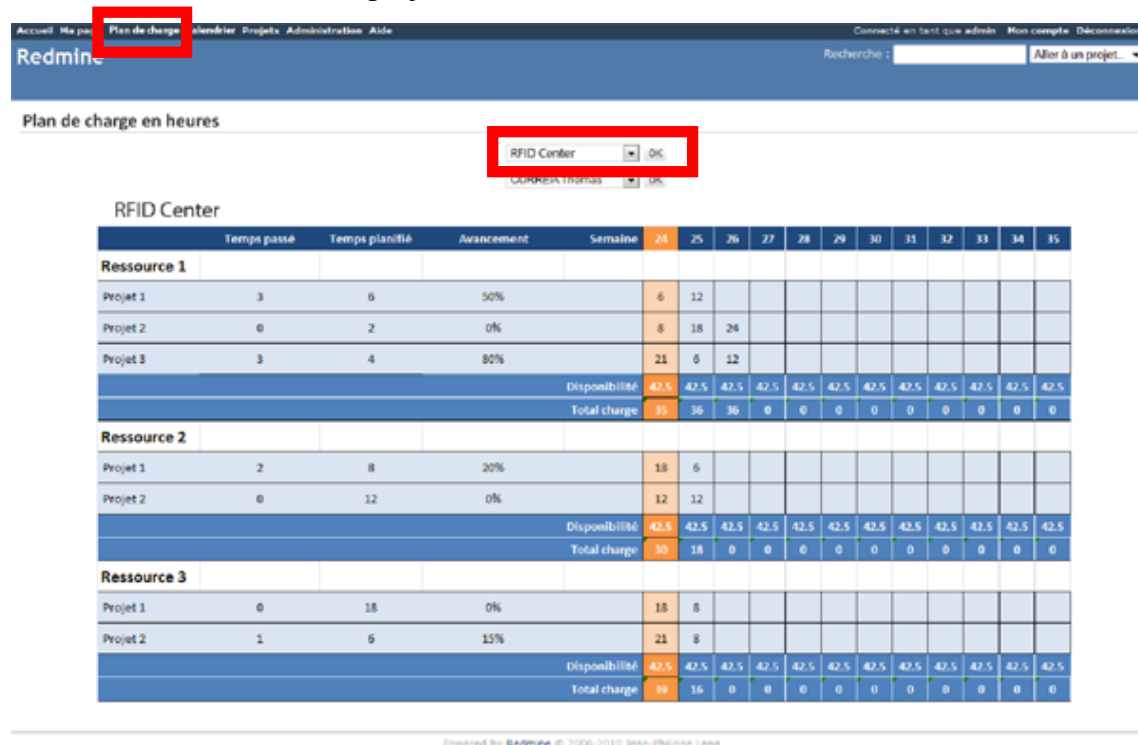


Figure 5 : Prototypage "Plan de charges – Par groupe"

Après validation, ces modifications ont été demandées :

- Savoir plus précisément quel type (groupe ou ressource) est sélectionné, en somme ne plus avoir deux listes déroulantes mais une seule
- Pouvoir sélectionner le nombre de semaines à afficher
- Date du lundi en plus des numéros de semaine
- Gestion des couleurs pour retard de tâches et dépassement de disponibilité



5.3.2. Reporting semaine

L'interface « Ma Page » est déjà présente dans Redmine et permet à l'utilisateur de la personnaliser en ajoutant des modules existants comme le calendrier, le temps passé, les demandes soumises, etc... Le but est donc de créer un nouveau module « Heures de la semaine » qui permet de sélectionner facilement une semaine X et de visualiser le temps effectué pour chaque jour de cette semaine.

The screenshot shows the Redmine 'Ma page' interface. The 'Heures de la semaine' module is highlighted with a red box. It displays a table for 'Semaine 24' (June 14-18, 2010) with columns for days of the week and hours passed. Below the table is a calendar view for June 2010. The interface also includes other modules like 'Temps passé', 'Demandes surveillées', 'Demandes qui me sont assignées', and 'Demandes soumises'.

Semaine 24	Lundi 14.06.2010	Mardi 15.06.2010	Mercredi 16.06.2010	Judi 17.06.2010	Vendredi 18.06.2010
Temps passé	38	59	26	36	0

Semaine : 14.06.2010 - 18.06.2010

Calendar view for June 2010 showing days 1 through 30.

Figure 6 : Prototype “Ma page – Heures de la semaine”

5.3.3. Calendrier

Cette vue permet d'insérer des absences ou des congés pour une plage de dates et ainsi de mettre à jour la disponibilité du plan de charges.

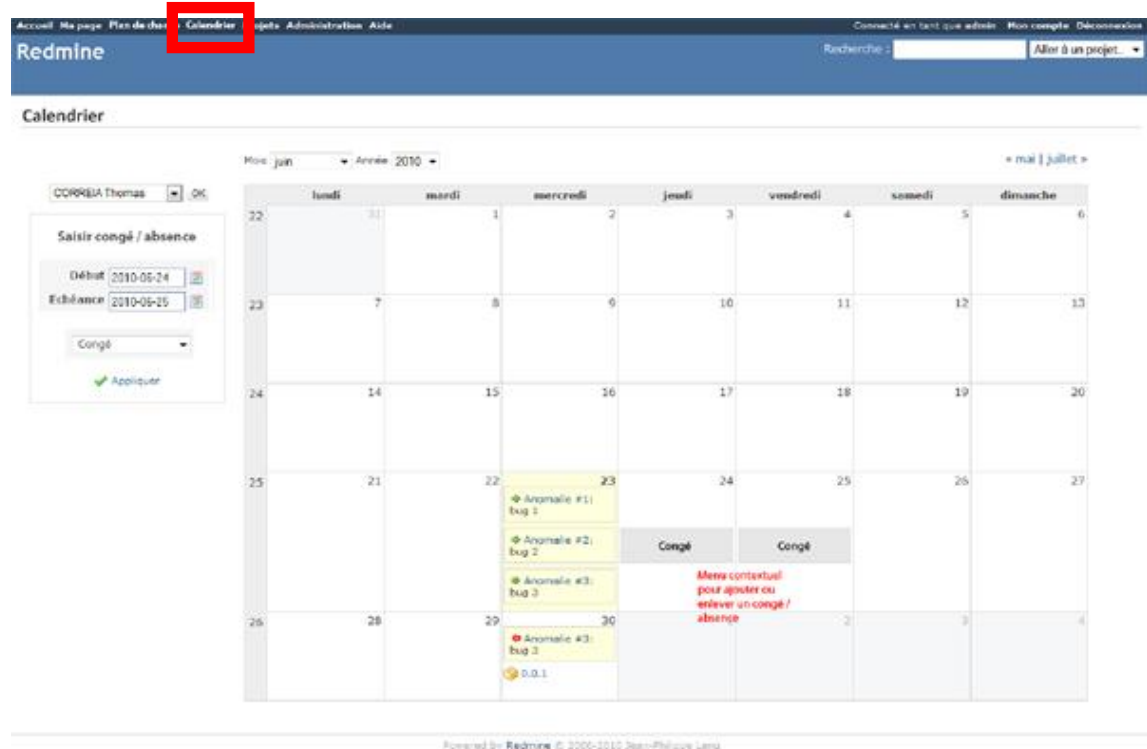


Figure 7 : Prototypé “Calendrier”

Après validation, ces modifications ont été demandées :

- Permettre de synchroniser le calendrier avec l’agenda de Google
- Rajouter la gestion des fériés, qui est un input pour toutes les ressources



5.3.4. Budget

Le budget est accessible seulement dans certains projets et non pas partout comme les autres interfaces, car le but est de pouvoir gérer le coût des ressources par projet. En effet, les salaires horaires sont susceptibles de changer selon le type de projet réalisé. De plus, il doit être possible d'ajouter des coûts annexes, comme par exemple du matériel.

Projet 1

Ressources	Salaires horaires	Temps passé (heures)	Coûts cumulés
Ressource 1	120	30	CHF 3'600.00
Ressource 2	100	38	CHF 3'800.00
Ressource 3	80	25	CHF 2'000.00
Ressource 4	80	12	CHF 960.00
Ressource 5	80	17	CHF 1'360.00
Coûts ressources			CHF 11'720.00

Annexes	Prix
Matériel 1	CHF 540.00
Matériel 2	CHF 210.00
Matériel 3	CHF 890.00
Matériel 4	CHF 1'650.00
Coûts annexes	CHF 3'290.00
Total	CHF 15'010.00

Sauvegarder

Figure 8 : Prototype “Budget”

6. Développement du projet

6.1. Introduction

La partie développement du projet concerne l'ajout des fonctionnalités principales et secondaires, décrites ci-dessus, dans l'outil Redmine. Cette phase du projet est un réel challenge, car il faut à la fois analyser la structure du code et la base de données de Redmine, comprendre le fonctionnement des plugins, et les coder dans un langage peu connu.

En effet, Ruby On Rails n'a jamais été utilisé durant ma formation à la HES-SO, mais uniquement pendant mon stage chez Icare, ce qui a demandé beaucoup plus de temps de recherches et a ralenti le développement.

6.2. Configuration

6.2.1. Environnement de travail

Le système d'exploitation utilisé pour le développement est Microsoft Windows 7, mais il est également possible de développer sur Linux ou MacOS.

6.2.2. Ruby On Rails¹⁹

La dernière version de Ruby (1.9) n'est pas compatible avec Redmine, c'est pourquoi la version 1.8.6 est utilisée.

Lien de téléchargement: http://rubyforge.org/frs/?group_id=167

Concernant Rails, la version installée est la 2.3.5 et pour ce faire il suffit de taper dans le shell :

```
gem install rails -v=2.3.5
```

6.2.3. SGBD

Le SGBD, ou Système de Gestion de Base de Données, utilisé est MySQL 5.0.67.

Lien de téléchargement: <http://www.brothersoft.com/mysql-220613.html>

¹⁹ Source : redmine.org, Installing Redmine, consulté le 14/08/2010, <http://www.redmine.org/wiki/redmine/RedmineInstall>



Sur Windows, il faut encore installer le gem MySQL en faisant :

```
gem install mysql
```

6.2.4. Installation de Redmine²⁰

Le début du développement a commencé sur la version 0.9.3 de Redmine, car cette version était déjà déployée sur le serveur de production d'Icare, c'est pourquoi elle a été utilisée comme base.

Par la suite, une migration a été effectuée lors de la sortie de la version 1.0.0 Release Candidate datant du 18/07/2010, version stable et finalisée de l'outil.

Afin d'installer correctement Redmine, il est nécessaire d'effectuer d'abord la configuration système décrite ci-dessus et ensuite, il faut télécharger le code source de Redmine.

Lien de téléchargement: <http://www.redmine.org/wiki/redmine/Download>

Une fois Redmine téléchargé, il faut renommer le fichier « database.yml.example » en « database.yml » dans le dossier config du code source. Ce fichier représente la configuration des tables de production, de développement et de test du SGBD pour Redmine. Il dépend de la configuration de MySQL, ou autre, lors de son installation, c'est-à-dire le nom du SGBD, le nom de la table, l'url de MySQL (par défaut localhost) et enfin le nom d'utilisateur et le mot de passe.

Par la suite, il faut ouvrir le shell et se rendre dans le répertoire du code source de Redmine et taper :

```
rake db:setup
```

Cette commande permet de créer les tables et colonnes dans la base de données. Toujours dans le répertoire de Redmine, il faut encore générer une clef de session, qui est fait en tapant :

```
rake generate_session_store
```

Afin de démarrer Redmine en mode développement, il faut encore se rendre dans le répertoire du code source et taper :

```
ruby script/server
```

²⁰ **Source** : redmine.org, Installing Redmine, consulté le 14/08/2010, <http://www.redmine.org/wiki/redmine/RedmineInstall>

Pour finir, il suffit de se rendre sur <http://localhost:3000/> et de se logger avec le nom d'utilisateur « admin » et le mot de passe « admin ».

6.3. Planification du développement

Voici une planification des étapes du développement effectuée avant le début de celui-ci :



Figure 9 : Planification du développement

L'estimation de ces heures est faite en fonction de la difficulté des plugins et de leur taille :

Fonctionnalités	Heures planifiées
Plan de charges	48
Ma page – Heures de la semaine	16
Calendrier	20
Budget	12
Notifications par email	4
Commit SVN Saisie des heures	8
Drag&Drop plan de charges	20
Graphiques	16
Améliorer l'ergonomie	16
Total	160

6.4. Réalisation

Voici la phase concrète du projet, celle qui concerne le développement des plugins. Cette présentation de la réalisation est divisée en deux étapes ; Les fonctionnalités principales et les fonctionnalités secondaires.

6.4.1. Fonctionnalités principales

Plugins

Voici les plugins développés. Ils sont au nombre de trois car la fonctionnalité « Ma page – Reporting des heures » est incluse dans le plugin « Workload » (Plan de charges).

La liste des plugins installés affiche le nom du plugin, une courte description, l’auteur ainsi que la version actuelle.

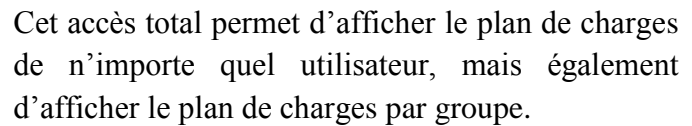
Sur cette interface « Plugins », on voit que seul le plugin du plan de charges est configurable et possède une version avancée. Nous reviendrons plus tard sur ce point.

Plugins

Budget plugin This is a plugin for Redmine which adds a budget for each projects to set a rate to resources and alternative costs	Thomas Correia	0.0.1
Calendar plugin This is a plugin for Redmine which adds a google calendar and interaction with workload disponibilities	Thomas Correia	0.0.1
Workload plugin This is a plugin for Redmine which show workload of a ressource or a group	Thomas Correia	0.0.2 Configurer

Figure 10 : Liste des plugins dans “Administration” de Redmine

Les autres plugins sont également configurables, mais pas depuis cette interface.



Interface plan de charges par ressource

Ces tâches sont listées par rapport à leur version et ordrées selon la date de début et la date d'échéance.

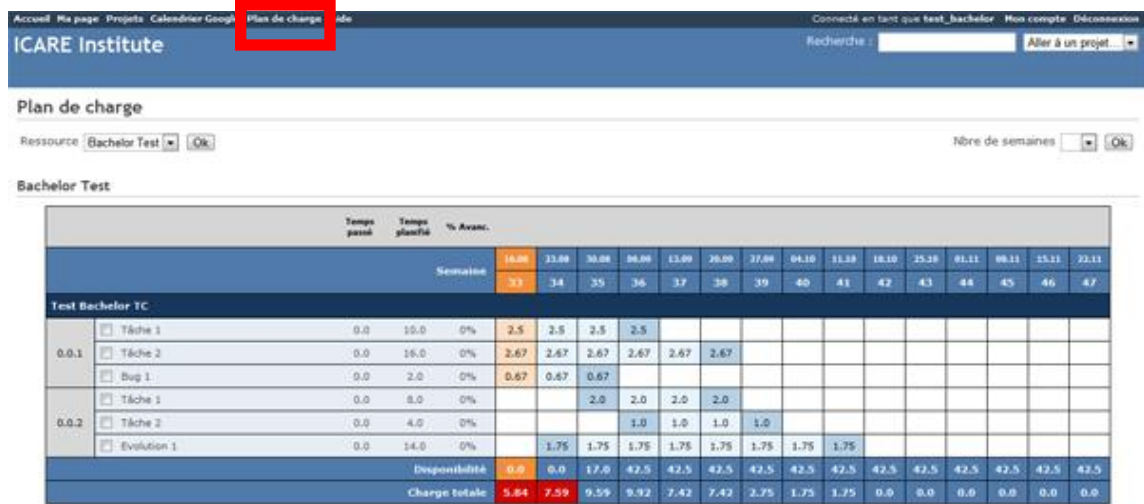


Figure 12 : Interface du plugin “Plan de charges – Par resource”

Sur chaque tâche (partie gauche) le menu contextuel de Redmine a été implémenté. Toutes les tâches sont sélectionnables et grâce au menu, il est très facile de les modifier.

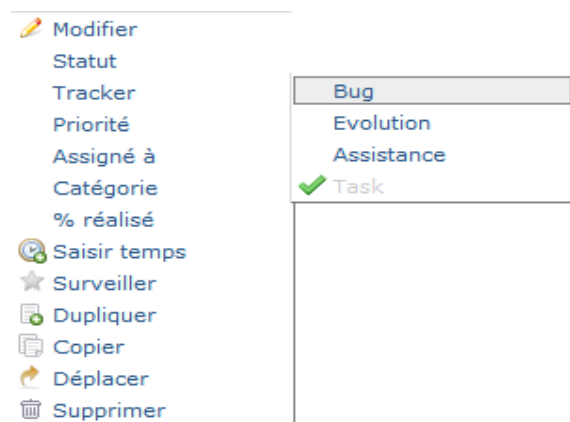


Figure 13 : Menu contextuel de Redmine

En plus de tout cela, une liste déroulante permet de sélectionner la ressource à afficher et une autre permet d’augmenter le nombre de semaines visibles.

Afin de mieux visualiser sur quelle semaine on se trouve, le plugin jQuery TipTip²¹ a été utilisé afin de rendre plus dynamique les tooltips.

Un Drag&Drop a également été implémenté. Il est décrit dans les fonctionnalités secondaires.

²¹ Source : <http://code.drewwilson.com/entry/tiptip-jquery-plugin>



Interface plan de charges par groupe

Cette interface ressemble au plan de charges par ressource, à la différence qu'elle n'intègre ni le menu contextuel, ni le Drag&Drop. La raison de cela vient du fait que les demandes ne sont plus listées, mais uniquement les projets.

Toutes les demandes ouvertes de chaque projet listé ont leurs heures additionnées et concaténées afin d'offrir une vision globale du temps planifié pour tout un groupe.

Cette interface est très utile pour le chef de projet qui peut rapidement voir qui est en retard ou qui n'a pas rempli son planning.

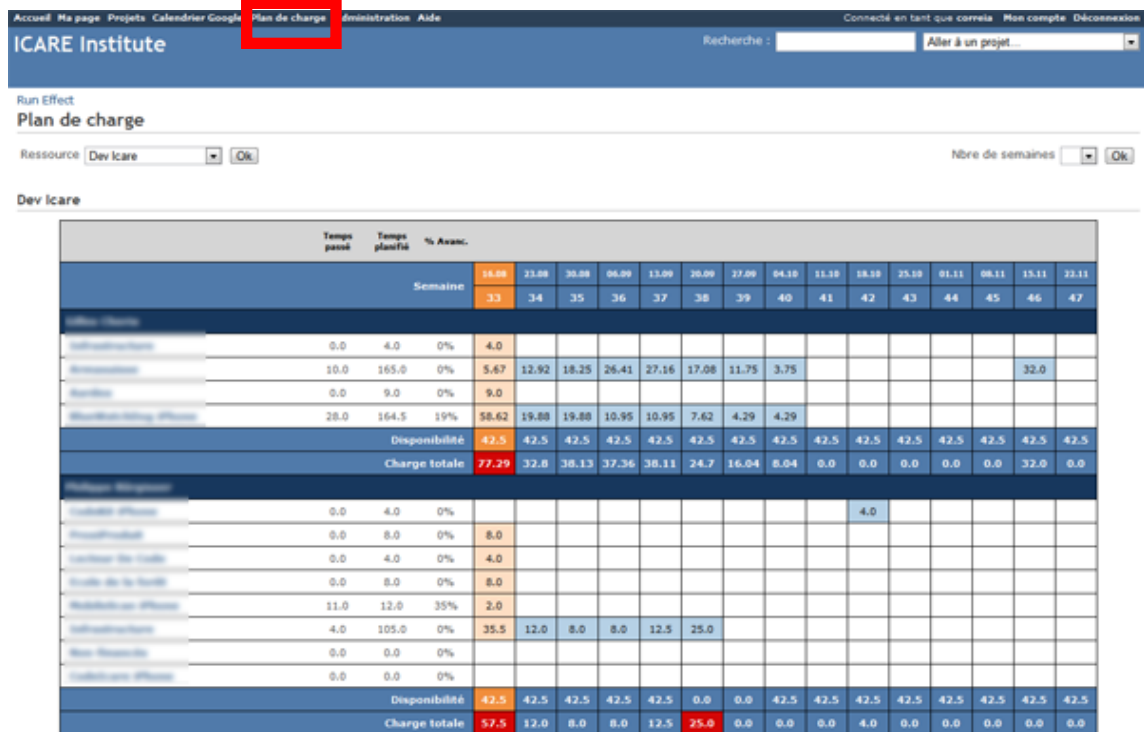


Figure 14 : Interface du plugin “Plan de charges – Par groupe”

Afin de comprendre un peu mieux ce plan de charge, voici une explication plus détaillée de ce qui est présent :

Cette sorte de calendrier affiche semaine par semaine le planning des ressources. Chaque colonne représente une semaine différente. L'objectif souhaité est de fournir une vision « macro » et agile pour les développeurs et chef de projet. Grâce à cette forme d'interface, le détail n'est pas très présent car l'on ne connaît pas précisément les dates de début et d'échéance des demandes ou projets. Cependant, il est nettement plus facile de se rendre compte du planning et ainsi de prévoir une nouvelle



planification le cas échéant. Etant donné que les heures d'un projet sont concaténées, il n'est pas possible de modifier directement sur cette interface le planning. Pour cela, il faut cliquer sur le nom d'une ressource, ce qui nous amène vers le plan de charges par ressource de cette personne.

En bas du tableau, on trouve le total des heures planifiées pour chaque semaine, ainsi que la disponibilité pour cette semaine. Cette disponibilité est mise à jour grâce au plugin « Calendrier ».

Interface ma page heures de la semaine

Cette interface est basique et a demandé peu de temps de développement. Son principal atout est de fournir rapidement une visualisation des heures réalisées pour une semaine choisie.

Le petit calendrier permet de sélectionner une semaine entière et a été implémenté à l'aide d'un plugin jQuery qui se prénomme « Date Picker »²².

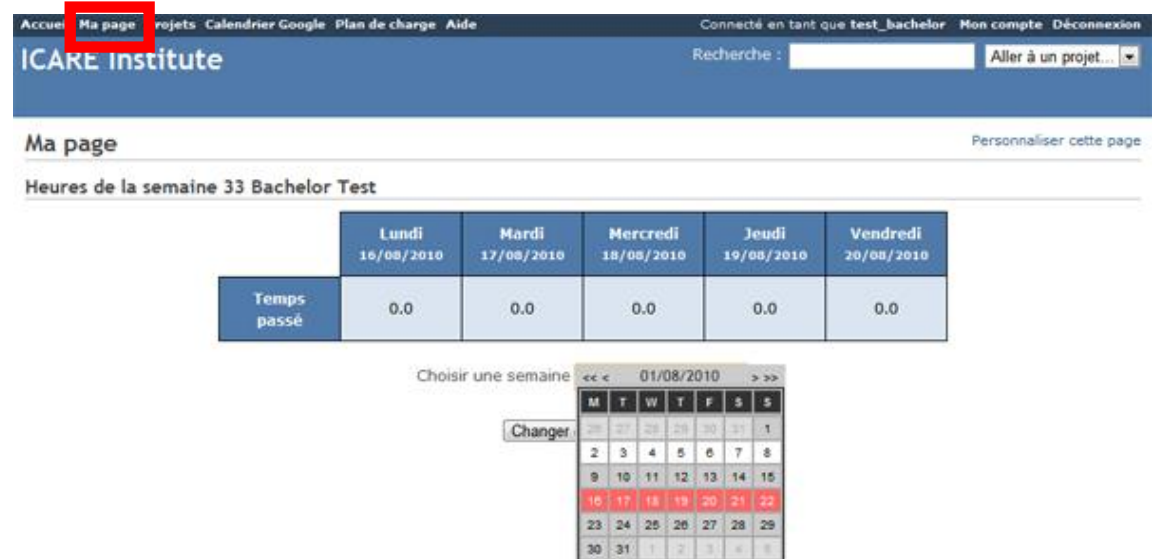


Figure 15 : Interface plugin “Ma page – Reporting semaine”

La seule difficulté de cette interface est de l'implémenter dans le code de Redmine. Etant donné qu'il s'agit d'une interface existante, il a fallu créer un nouveau module et faire en sorte qu'il soit utilisable comme un des autres déjà présents.

²² **Source :** kelvinluck.com, Date Picker, consulté le 12/08/2010,
<http://www.kelvinluck.com/assets/jquery/datePicker/v2/demo/index.html>



Interface calendrier

Le calendrier permet donc de mettre à jour la disponibilité du plan de charges. Sachant que tout l'institut Icare utilise l'agenda de Google pour planifier ses congés et autres absences, il est logique de synchroniser ces disponibilités par rapport à cet agenda, afin d'éviter une saisie à double des absences.

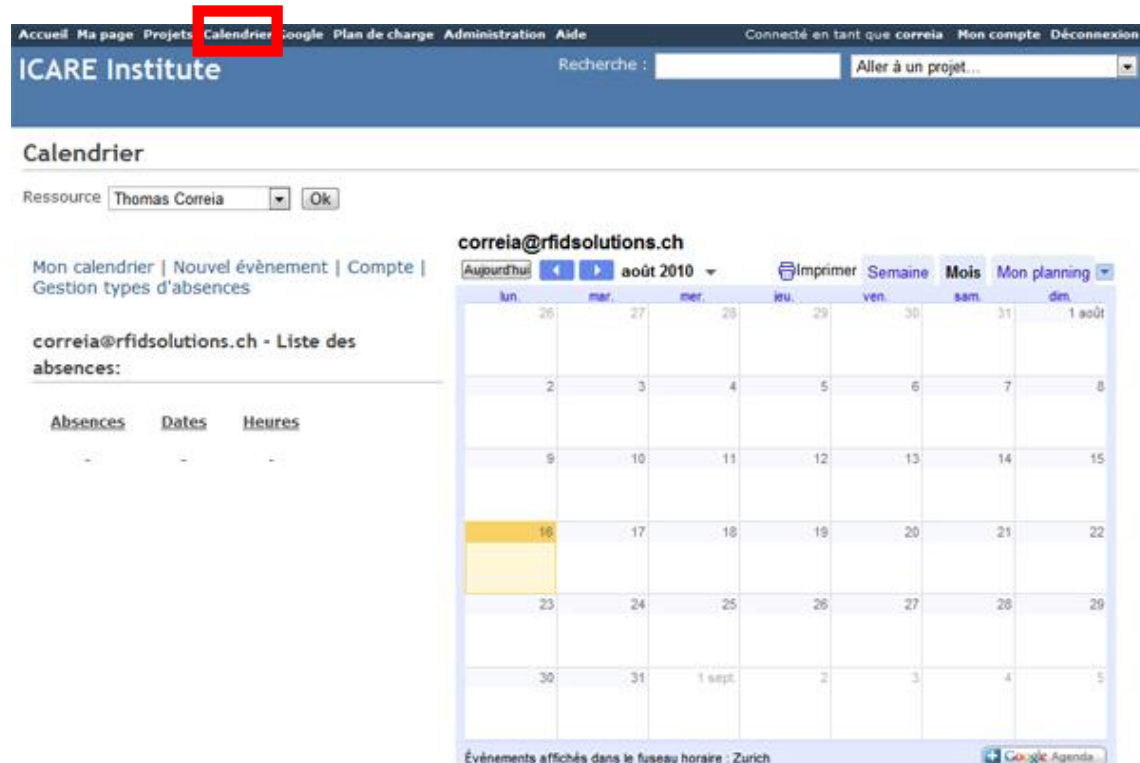


Figure 16 : Interface plugin “Calendrier”

C'est justement ce qu'autorise cette interface, en utilisant un gem de Rails appelé GCal4Ruby²³. Grâce à cette API, il est possible de se connecter à un compte Gmail et d'y récupérer le calendrier associé. Le calendrier affiché sur la figure 16, est fourni par Gmail²⁴ et ne permet que d'afficher les événements de l'agenda, sans avoir la possibilité d'interaction, le but n'étant pas de recréer entièrement l'agenda de Google.

²³ Source : cookingandcoding.com, GCal4Ruby, consulté le 13/08/2010, <http://cookingandcoding.com/gcal4ruby/>

²⁴ Source : google.com, Google Agenda, consulté le 13/08/2010, <https://www.google.com/calendar>



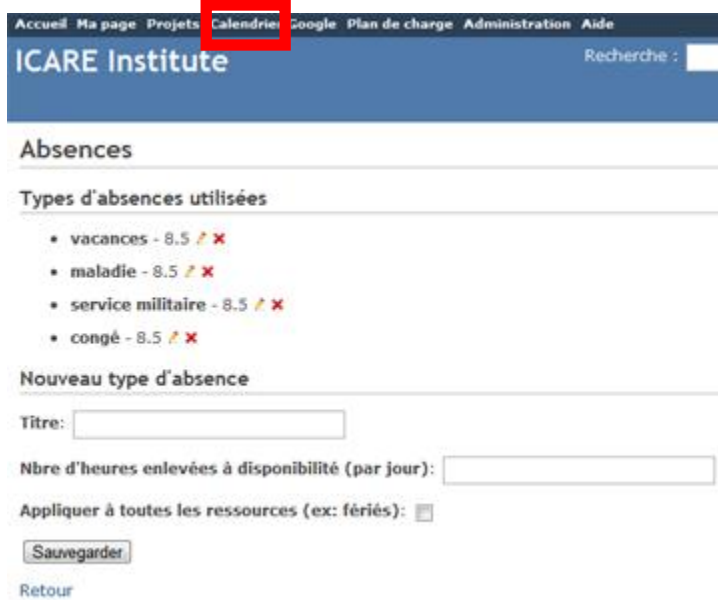
Le plugin “calendrier” permet également de créer de nouveaux évènements pour l’agenda de Google. Sur cette interface, on peut spécifier un nouvel évènement avec sa plage de dates, s’il a lieu toute la journée, le lieu de l’évènement, s’il se répète et si oui, de quand à quand, ainsi que la fréquence de répétition (chaque lundi de chaque semaine par exemple).

La seule différence avec Google et que cet évènement ne peut avoir comme titre que ceux spécifiés dans la prochaine interface du calendrier.

Figure 17 : Interface plugin “Calendrier – nouvel évènement”

Sur l’interface d’absences du plugin « calendrier », il est possible d’ajouter des types d’absences personnalisées. Les absences spécifiées dans la figure 18 modifieront les disponibilités du plan de charges, par rapport au nombre d’heures inscrites.

Pour tous les types d’absences listées, chaque évènement comportant le même titre diminuera de « 8.5 » heures chaque jour d’une semaine du plan de charges selon la plage de dates de l’évènement.



Accueil Ma page Projets **Calendrier** Google Plan de charge Administration Aide

ICARE Institute Recherche :

Absences

Types d'absences utilisées

- vacances - 8.5 ⚡ ✖
- maladie - 8.5 ⚡ ✖
- service militaire - 8.5 ⚡ ✖
- congé - 8.5 ⚡ ✖

Nouveau type d'absence

Titre:

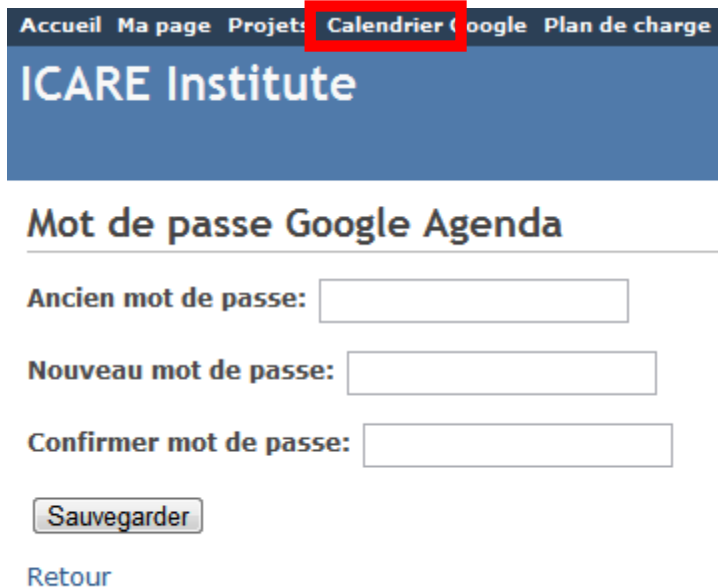
Nbre d'heures enlevées à disponibilité (par jour):

Appliquer à toutes les ressources (ex: fériés): ☐

[Retour](#)

Figure 18 : Interface plugin “Calendrier - Ajout d’absence”

La dernière interface du plugin « calendrier » permet de modifier le mot de passe de son compte Gmail.



Accueil Ma page Projets **Calendrier** Google Plan de charge

ICARE Institute

Mot de passe Google Agenda

Ancien mot de passe:

Nouveau mot de passe:

Confirmer mot de passe:

[Retour](#)

Figure 19 : Interface plugin “Calendrier – modification du mot de passé”



Interface budget

L'interface du budget est également assez simple et n'a demandé que quelques heures de développement. Grâce à elle, il est possible dans chaque projet de spécifier le coût horaire des ressources.

Il est également possible d'ajouter des coûts annexes, comme par exemple du matériel.

La seule difficulté de ce plugin est qu'il a été réalisé en Ajax, technologie nouvellement utilisée, afin de mettre à jour rapidement l'interface lorsque l'on ajoute ou supprime un coût annexe.

The screenshot shows the 'Budget' plugin interface for a project named 'Test Bachelor TC'. The interface includes a navigation bar with links like 'Accueil', 'Ma page', 'Projets', 'Calendrier Google', 'Plan de charge', 'Administration', and 'Aide'. The 'Budget' link is highlighted with a red box. Below the navigation bar, there is a search bar and a dropdown menu to 'Aller à un projet...'. The main content area displays a table with the following data:

Ressources	Coûts horaires	Temps passé	Coûts cumulés
Bachelor Test	80.00	0.0	CHF 0.00
Coûts des ressources			CHF 0.00
Annexes			Coûts
Matériel 1			100.0
Matériel 2			2100.0
Coûts des annexes			CHF 2200.00
Coûts totaux			CHF 2200.00

There are plus and minus icons to the right of the 'Annexes' section, and a 'Sauvegarder' button at the bottom right.

Figure 20 : Interface plugin "Budget"

On peut également gérer les autorisations d'accès à ce plugin, en spécifiant quels rôles (manager, développeur, rapporteur,...) peuvent y accéder. Par défaut, tout administrateur y a accès, mais comme cette interface est destinée au chef de projet, aucun accès n'est actuellement autorisé pour tous les rôles présents.

The screenshot shows the 'Budget' plugin permissions configuration. It includes a section for 'Budget' with a green checkmark and the text 'Accès budget'. Below this, there are five checkboxes, all of which are currently unchecked.

Budget					
✓ Accès budget	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 21 : Autorisations pour plugin "Budget"



6.4.2. Fonctionnalités secondaires

Notifications par email²⁵

Cette fonctionnalité secondaire ne comporte pas de code mais uniquement une configuration système de Redmine. En effet, afin de faire fonctionner ces notifications par email, il a simplement été nécessaire de suivre un tutoriel expliquant comment résoudre le problème qui empêchait l'envoi des emails depuis Redmine.

Pour ce faire, il faut utiliser un compte Gmail, le protocole TLS ainsi qu'un plugin Rails s'appelant « `action_mailer_optional_tls_plugin` ».

Ce plugin s'installe en tapant dans le shell :

```
ruby script/plugin install  
git://github.com/collectiveidea/action_mailer_optional_tls.git
```

Il se peut que cela ne fonctionne pas sur Windows, il faut alors télécharger l'archive sur http://github.com/collectiveidea/action_mailer_optional_tls et l'extraire dans le dossier « `vendor/plugins/` » de l'application Redmine.

Une fois le plugin installé, il faut encore modifier le fichier « `email.yml` » et le configurer comme suit :

```
# File: config/email.yml  
development:  
  delivery_method: :smtp  
  smtp_settings:  
    tls: true  
    address: "smtp.gmail.com"  
    port: '587'  
    domain: "smtp.gmail.com"  
    authentication: :plain  
    user_name: "your_email@gmail.com"  
    password: "your_password"
```

²⁵ **Source** : [redmineblog.com](http://redmineblog.com/articles/setup-redmine-to-send-email-using-gmail/), tutorial setup redmine send email using gmail, consulté le 12/08/2010, <http://redmineblog.com/articles/setup-redmine-to-send-email-using-gmail/>



Il faut encore ajouter une librairie externe²⁶, qui permet d'activer le TLS. Cette librairie se trouve à cette adresse : http://knk.album.googlepages.com/tls_smtp.rb et est à placer dans le dossier « lib » de Redmine.

Afin de pouvoir l'utiliser, il est nécessaire de rajouter la ligne « require 'tls_smtp' » dans le fichier « environnement.rb » du dossier « config ».

Finalement, il ne reste qu'à aller dans l'administration de Redmine et y spécifier l'email de Gmail.

Figure 22 : Configuration “notifications par mail” dans Redmine

Commit SVN saisie des heures

Le commit SVN pour saisir des heures sert à mettre à jour une demande lorsqu'un utilisateur met à jour du code source d'un projet.

Un commit est un outil primordial dans le développement d'applications. Il permet de partager du code source sur un serveur, très pratique lorsque plusieurs ressources travaillent sur un même code.

Par exemple, lorsque l'on corrige un bug dans un projet quelconque, on va commiter les changements sur le serveur de source, et une autre personne pourra mettre à jour son code et ainsi récupérer les modifications réalisées.

Lors de chaque commit, il est possible d'écrire un message. Redmine autorise la configuration d'un de ces serveurs de source pour chaque projet. Ainsi, lorsque l'on spécifie un dépôt de source dans un projet, Redmine est informé lorsqu'un commit a lieu.

²⁶ **Source :** [errorhelp.com](http://www.errorhelp.com), Redmine email STARTTLS, consulté le 12/08/2010, <http://www.errorhelp.com/index.php/search/details/78026/redmine-rails-must-issue-a-starttls-command-first>



Grâce au message du commit, l'utilisateur peut spécifier le numéro d'identification d'une demande, et ainsi celle-ci sera automatiquement mise à jour.

Afin d'améliorer cette fonctionnalité, il existe déjà différents patches créés par la communauté de Redmine. Un de ceux-ci s'appelle « Automatic spent time logging from commit messages » et est disponible sur le site web²⁷ de l'outil.

En utilisant ce patch, il devient alors possible de spécifier des options supplémentaires dans le message, par exemple :

```
refs #1 @2h30m
```

Ce message permet de notifier à Redmine que 2h30 de temps ont été effectuées sur la demande #1. En utilisant Rubular²⁸, un éditeur de Regex, il devient possible de modifier ce patch afin de rajouter des fonctionnalités supplémentaires.

Une d'entre elles est de pouvoir spécifier en plus de la saisie des heures, le pourcentage d'avancement d'une demande, en faisant :

```
refs #1 @2h30m %50
```

Ce qui placera le pourcentage d'avancement de la demande #1 à 50%.

Drag&Drop plan de charge par ressource

Le Drag&Drop est la dernière fonctionnalité développée et implémentée. C'est une fonctionnalité qui a demandé beaucoup d'heures de recherche et de conception afin d'arriver à un résultat probant.

Afin d'améliorer la gestion de la planification sur le plan de charges, certains membres de l'institut Icare souhaitent pouvoir directement modifier les heures planifiées en les déplaçant d'une semaine à l'autre.

Ainsi, l'interface du plan de charges par ressource devient entièrement dynamique et chaque demande peut être re planifiée avec beaucoup plus de facilité qu'auparavant. Il n'est désormais plus nécessaire de cliquer sur une demande, d'aller dans ses détails et d'y changer la date de début et / ou la date d'échéance.

²⁷ **Source** : redmine.org, automatic spent time logging from commit messages, consulté le 12/08/2010, <http://www.redmine.org/issues/4155>

²⁸ **Source** : <http://rubular.com/>



Ceci est réalisé grâce aux Frameworks Javascripts Prototype²⁹ et Script.Aculo.Us³⁰, qui fournissent un ensemble de méthodes permettant de spécifier des objets HTML comme étant draggables et / ou droppables.

Chaque case (partie droite) du plan de charges devient alors « droppable », c'est-à-dire qu'elle accepte un objet « draggable » à y être déposé, et chaque case comportant une heure devient donc « draggable ».

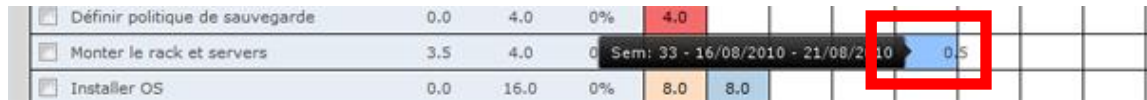


Figure 23 : Drag&Drop plugin “Plan de charges par ressource”

Sur la figure 23, on distingue qu'une heure planifiée de la demande « Monter le rack et servers » est en train d'être déplacée vers une nouvelle semaine. La gestion des couleurs signalant qu'un Drag&Drop est effectué n'est pas non plus évidente à mettre en place. Beaucoup de paramètres entrent en ligne de compte et à chaque modification de code, un nouveau problème apparaît, jusqu'à trouver le bon compromis.

Il est également possible d'implémenter différentes sortes de Drag&Drop pour ces heures. Par exemple, si une demande est répartie sur plusieurs semaines, les deux extrémités de celle-ci modifient soit sa date de début, soit sa date d'échéance, tandis que si l'on déplace une heure se trouvant entre ces deux extrémités, tout le bloc d'heures se déplacera.

Une dernière fonctionnalité du Drag&Drop a été ajoutée, lorsqu'une demande n'est planifiée que sur une seule et unique semaine. A ce moment-là, en laissant presser la touche « CTRL » ou « MAJ », la demande sera étendue à la semaine choisie, au lieu d'y être déplacée.

²⁹ Source : <http://prototype.js.le-developpeur-web.com/index.php>

³⁰ Source : <http://scriptaculous.le-developpeur-web.com/>



7. Problèmes rencontrés

7.1. Trouver plusieurs outils Ruby

Une des difficultés en début du projet fut la recherche d'outils existants sous langage Ruby. Il existe plusieurs petits projets en développement, mais très peu étaient suffisamment stables pour être susceptibles d'être utilisés.

C'est le cas de l'outil Retrospectiva³¹ qui semblait être un outil performant et stable de prime abord, mais après quelques heures de tentatives d'installation, il n'a jamais été possible de le tester sur un environnement Windows. Afin de ne pas perdre trop de temps, il a fallu l'abandonner et en rechercher d'autres, étant donné que certains outils utilisés dans l'état de l'art ne sont pas développés en Ruby.

7.2. Attentes différentes de l'entreprise

Grâce aux séances de groupe autour de Redmine, il m'a été possible de récolter de précieuses informations sur les besoins et attentes des membres. Seulement, il n'a pas toujours été simple de trouver un compromis qui satisfasse tout le monde, certains sujets déviant parfois sur des détails complexes comme le calcul du pourcentage d'avancement d'une tâche.

Certaines personnes voyaient cette fonctionnalité inutile, alors que d'autres imaginaient un calcul de l'avancement au prorata du nombre d'heures passées et planifiées et par rapport à la différence de jours entre la date de début et la date d'échéance d'une tâche. D'autres n'étaient pas d'accord sur ce principe et voyaient juste le pourcentage comme une information rapide permettant de notifier le restant de travail sur une tâche sans tenir compte du temps passé et planifié.

Il n'a donc pas été possible de mettre à profit toutes les suggestions et avis proposés par les membres pendant ces séances, c'est pourquoi elles n'ont pas été retenues pour ce travail de Bachelor.

7.3. Ruby On Rails

Une autre problématique fut le langage et Framework Ruby On Rails. Comme déjà expliqué dans ce document, il n'est pas simple de modifier un code existant sur un

³¹ <http://retrospectiva.org/overview>



langage peu connu et plutôt complexe lorsqu'on débute.

Cela m'a demandé beaucoup d'heures de recherches afin de comprendre la manière de procéder pour certaines fonctionnalités dans Rails. Par exemple, il y a les formulaires qui sont simplement une série de champs, de listes déroulantes ou de cases à cocher et qui sont envoyés depuis une vue vers un contrôleur.

Jusque-là rien de très compliqué, seulement dans Ruby il existe plusieurs types de formulaires permettant de créer un nouvel objet (imaginer un formulaire pour créer un livre, par exemple avec des champs « nom », « auteur », « année de parution », etc...). Certains sont automatisés par rapport aux données d'un modèle alors qu'un autre type de formulaire en Rails permet de simplement créer des champs vides, sans aucune interaction avec la base de données.

Cela semble plutôt simple, mais en pratique c'est une toute autre histoire car selon le type de formulaire utilisé, les options des champs et autres listes déroulantes changent également, et parfois on a besoin d'un champ précis d'un formulaire automatisé alors que l'on utilise un formulaire simple.

D'autres problèmes sont survenus par un manque de connaissance de Ruby On Rails, comme la liaison des tables dans les modèles de la structure MVC. Il existe plusieurs possibilités de créer des liaisons entre les tables, simplement en utilisant des termes tels que « belongs_to », « has_one » ou encore « has_many », qui ont chacun un sens et une utilité différente. Par exemple la table « disponibilités » comporte un numéro d'identification d'un utilisateur de la table « users », car un utilisateur possède une ou plusieurs disponibilités. Faut-il alors que le modèle « user » contienne une relation « has_many » disponibilités ou que le modèle « disponibilité » contienne une relation « belongs_to » voir « has_one » vers un utilisateur ?

Afin de trouver le bon terme, il est nécessaire de les comprendre, et en recherchant sur le web³² on apprend la différence entre un « belongs_to » et un « has_one » est que le premier permet de signaler que la table « disponibilités » possède un numéro d'identification de la table « users », sans que la table « users » ne soit au courant de cette liaison. Avec un « has_one » la table « disponibilités » a également un numéro d'identification d'un utilisateur, sauf que la table « users » contient également un numéro d'identification de la disponibilité correspondante. Et avec un « has_many », une table intermédiaire est créée afin de stocker ces relations.

³² **Source** : duanesbrain.blogspot.com, Ruby On Rails : Has_One versus Belongs_To, consulté le 14/08/2010, <http://duanesbrain.blogspot.com/2006/05/ruby-on-rails-hasone-versus-belongsto.html>



Un dernier point qui a posé quelques problèmes avec le langage Ruby, est la récupération des données des modèles depuis un contrôleur. Avec Ruby On Rails, il existe plusieurs manières de procéder afin de trouver les données correspondantes à ce que l'on souhaite, par exemple en écrivant une requête SQL basique tel que « `SELECT * FROM users WHERE id = '1'` » qui permet de récupérer l'utilisateur portant le numéro d'identification « 1 ». Mais Rails propose aussi des outils afin d'améliorer cela, par exemple en faisant « `Users.find(:all, :condition => { :id => 1 })` » ou encore « `Users.find_all_by_id(1)` », ce qui donne le même résultat.

Cependant, cela demande parfois du temps afin de trouver la bonne syntaxe pour des requêtes beaucoup plus complexes, comme par exemple celle qui permet de récupérer tous les projets visibles et actifs, en incluant les demandes liées à un utilisateur :

```
Project.visible.active.find(:all, :include => [ :issues, :versions ], :conditions =>
  "#{Issue.table_name}.id IN #{@issues.map{|i| i.id}.join(', ')})")
```

Il est possible de procéder de manière beaucoup moins élégante et performante, alors que cette façon de faire est propre et respecte la norme Ruby On Rails, mais demande plus de recherche lorsque l'on débute.

7.4. Coder sur Redmine

Une des dernières difficultés rencontrées fut le développement des plugins sur Redmine. Le but fondamental était de ne jamais modifier le code source de cette application, de la laisser brute afin de pouvoir à tout moment migrer vers une nouvelle version. Pour cela, il était important de créer des plugins venant se greffer sur l'application et ainsi ne modifient pas le code source de base.

Bien que le site de Redmine propose une aide détaillée pour la création de ces plugins, certaines étapes ont posé problème.

Tout d'abord, lorsque l'on développe en Ruby On Rails, il y a une spécificité de ce Framework qui permet de mettre à jour automatiquement les interfaces lorsque des modifications sont apportées dans un modèle, un contrôleur ou une vue.



Cela permet un développement plus rapide, car il n'y a pas besoin de recompiler l'application pour afficher à l'écran les modifications apportées. Seulement, de base les plugins ne permettent pas cette fonctionnalité, et il est nécessaire de rajouter une ligne³³ dans le fichier de configuration du plugin (init.rb) afin de pouvoir le faire :

```
if RAILS_ENV == 'development'
  ActiveSupport::Dependencies.load_once_paths.reject!{|x| x =~
/^#{Regexp.escape(File.dirname(__FILE__))}/}
end
```

Bien que ce problème soit résolu, les plugins impliquent un autre problème beaucoup plus important. Il est parfois nécessaire de modifier certains fichiers sources de Redmine, par exemple dans le cadre de la saisie des heures d'une demande. Sur cette interface, certains membres souhaitent pouvoir également mettre à jour le pourcentage d'avancement d'une demande en plus du temps passé.

Le tutoriel fourni par Redmine explique qu'il est possible de modifier une vue existante, mais pour les contrôleurs et modèles on ne peut qu'ajouter une nouvelle méthode, ou modifier une méthode existante en utilisant une fonction de Rails qui permette d'envelopper une méthode existante et de lui rajouter du code. Cette fonction reste très complexe et pour la saisie des heures cela ne suffit pas.

Comme les vues sont facilement modifiables, il est aisé de rajouter un champ « % avancement ». Malheureusement, le contrôleur qui réceptionne le formulaire de ces champs implémente une autre fonction de Rails qui va automatiquement lire le nom du champ et sa valeur et l'inscrire dans la colonne de la base de données correspondante, selon la table indiquée.

Or, le pourcentage d'avancement d'une demande ne se trouve pas dans la table « TimeEntries » qui stocke les saisies des heures. Afin de pouvoir rajouter ce champ, il est donc nécessaire de réécrire entièrement la méthode du contrôleur, afin que la sauvegarde dans la base de données se fasse de manière manuelle et non plus automatique.

Ce problème est récurrent lorsque l'on souhaite modifier un contrôleur ou un modèle, et cela demande une plus grande réflexion et plus de travail à sa réalisation.

³³ **Source** : strd6.com, Can't dup nil class, consulté le 14/08/2010, <http://strd6.com/2009/04/cant-dup-nilclass-maybe-try-unloadable/>

8. Feedbacks

8.1. Avis des membres de l'institut Icare

Afin de faire partager les avis des membres de l'institut Icare, voici leurs commentaires suite à l'utilisation des nouvelles fonctionnalités développées :

- Que pensez-vous des différentes interfaces implémentées, est-ce qu'elles vous paraissent utiles et sont-elles à votre goût ?

« Pas encore utilisé le calendrier, je me sers plutôt d'iCal sur mon mac. Plan de charges, vraiment bien. C'est déjà le jour et la nuit depuis Intra-Know. Vraiment cool. Heures de la semaine: parfait, le résumé vraiment pratique pour savoir où on en est. »

« Interfaces sympas, tu as réussi à bien intégrer les modifications développées. Le plan de charges est très bien fait, simple d'utilisation, le Drag&Drop est vraiment pratique à l'utilisation, c'est un gain de temps considérable par rapport à l'outil précédant. Le calendrier, je n'ai pas encore essayé, mais je l'utiliserai principalement pour saisir mes congés »

- En l'état actuel, est-ce que Redmine vous permet de facilement mettre à jour votre planning ?

« Tout est là. Tout ce qui serait ajouté est du pur bonus. Niveau ergonomie c'est déjà suffisant. Niveau fonctionnalité, c'est bon aussi. »

« Oui, il permet de mettre à jour le planning de manière facile, dans le sens où si l'on veut déplacer une tâche à une semaine ultérieure, c'est très simple. Par contre la saisie/mise à jour des tâches demeure un peu longue, car il y a toujours plusieurs étapes pour mettre à jour une tâche, clic droit, modifier, rentrer les heures, on est obligé de sélectionner un scénario pour la tâche (développement, conception...),... »

8.2. Améliorations possibles

Quelques améliorations possibles et souhaitées par certains membres de l'institut Icare sont listées ci-après :

Plan de charges

- Implémentation de l'Ajx dans le plan de charges
- Déplacer automatiquement des demandes en retard pour qu'elles se retrouvent les semaines suivantes tant qu'elles ne sont pas closes.
- Affichage des demandes non estimées dans le plan de charges
- Feedback visuel du Drag&Drop (changement de forme du curseur en fonction de ce qui va se passer)

Heures de la semaine

- Savoir en quoi consiste les heures d'un jour en cliquant ou en laissant la souris dessus
 - Nom du projet, nom de la tâche, temps ou liste des temps passés comme dans le résumé "temps passé"
- Alerte visuelle des heures non remplies

Divers

- Pouvoir saisir le temps passé lors de la création d'une nouvelle tâche
- Plugin pour ajouter de liens personnalisés
- Assigner une demande à une personne non affiliée au projet
- Date d'échéance d'une demande automatiquement remplie par rapport à la date de début
- Améliorer la saisie du temps d'une demande pour gain de temps

9. Gestion du projet

Voici une comparaison des heures planifiées et des heures effectuées, ainsi qu'une indication sur les fonctionnalités réalisées et non réalisées par rapport au cahier des charges :

Fonctionnalités principales

Fonctionnalités	Heures planifiées	Heures réalisées	Status
Plan de charges	48	105	✓
Ma page – Heures de la semaine	16	5	✓
Calendrier	20	57	✓
Budget	12	16.5	✓
Total	96	183.5	

Fonctionnalités secondaires

Fonctionnalités	Heures planifiées	Heures réalisées	Status
Notifications par email	4	2	✓
Commit SVN Saisie des heures	8	11	✓
Drag&Drop plan de charges	20	46	✓
Graphiques	16	0	✗
Améliorer l'ergonomie	16	0	✗
Total	64	59	

Total

Sujet	Heures planifiées	Heures réalisées
Recherche / Analyse / Séances...	80	63
Documentation	120	97
Fonctionnalités principales	96	183.5
Fonctionnalités secondaires	64	59
Total	360	402.5



10. Conclusion

L'utilisation d'un outil de gestion de projet au sein d'une entreprise informatique est indispensable. Implémenter un tel outil et répondre aux besoins des utilisateurs ont été un vrai challenge.

Développer des fonctionnalités sur un langage dans lequel on possède peu d'expérience demande également un gros investissement, de même que les implémenter dans un code préexistant. Tout repose sur une bonne compréhension des attentes, des possibilités à disposition et du temps imparti pour les réaliser. Cela requiert également une recherche approfondie et une bonne motivation afin d'arriver aux buts souhaités.

L'important dans un tel travail demeure les volontés du client. Chaque choix entrepris et solutions proposées restent légitimes des besoins et désirs du demandeur. Ses exigences ne signifient pas non plus qu'il faille suivre un chemin tout tracé, car même s'il a des idées arrêtées sur une solution finale, c'est à moi de savoir si cela est réalisable ou non.

Il est parfois aussi difficile de refuser une demande, car la solution lui est destinée. Néanmoins, il faut aussi savoir dire non, ou proposer une solution alternative comblant ses besoins et résolvant de possibles problèmes rencontrés.

L'implémentation de Redmine ne fut pas toujours évidente, mais au final toutes les fonctionnalités principales ont été réalisées ; Les fonctionnalités secondaires également, mais sans avoir eu le temps de toutes les implémenter. Cependant, les non réalisées tels que les graphiques demeurent peu intéressantes pour le client et n'entravent en rien l'utilisation de l'outil.

Ce travail de Bachelor m'a permis d'améliorer mon expérience dans un milieu professionnel, mais surtout de développer mes facultés à réaliser un projet de A à Z. Personnellement, je suis satisfait du travail fourni et je pense avoir pleinement répondu aux attentes du client.

11. Annexes

11.1. Cahier des charges

Sujet

Outil web pour la gestion de projet selon les méthodes agiles

Description

Ce travail de Bachelor consiste à mettre en place un outil web de gestion de projets, et de ses ressources, pour un groupe d'une dizaine de personnes au sein de l'institut « Icare ».

L'outil offrira une gestion multi-projets, sera utilisable sur internet et reposera sur certains standards des méthodologies agiles. Cette application permettra une visualisation du plan de charges hebdomadaire des ressources (temps planifié), le reporting (temps effectué) par semaine, la gestion du budget et des congés / absences, ainsi que d'autres fonctionnalités secondaires.

Objectifs du projet

- Définition des fonctionnalités de l'outil
- Analyse de solutions existantes (open source, ou non)
- Evaluation et analyse des processus entreprises
- Définition de la solution technique
- Développement et/ou mise en place de la solution arrêtée
- Test et validation

Buts de l'outil

- Améliorer la gestion des projets au sein de l'institut Icare :
 - Qualité de la solution permettant une utilisation simple, rapide et efficace
 - Qualité des livrables, planifiés grâce à l'outil
 - Planning réalisé à l'aide d'itérations (Sprints)
 - Graphiques
- Gestion des ressources et des projets

- Reporting et planification en fonction ressources disponibles et des projets actifs, avec gestion des heures (temps effectué, temps planifié)
- Satisfaire les besoins des différents membres de l'institut Icare
 - Analyse des besoins
- Outil développé sur Ruby On Rails pour une intégration aux processus métiers de l'institut

Fonctionnalités de l'outil

Fonctionnalité requise

- Développer une application sur le Framework Ruby On Rails (RoR)
 - L'institut Icare développant en majorité ses projets sur « RoR », fournir un outil basé sur le même langage est d'une importance capitale

Fonctionnalités principales

- Plan de charges - Futur
 - Vision « macro » des heures planifiées par semaine
 - De x à x semaines (par exemple minimum 6 semaines maximum 20 semaines)
 - Pour une ressource ou un groupe de ressources
 - Graphique - fonctionnalité optionnelle
- Reporting pour semaine x - Passé
 - Aperçu des heures effectuées pour une ressource
- Calendrier
 - Gestion des absences, congés, fériés,...
- Gestion du budget
 - Choix du coût horaire des ressources par projet
- Agilité
 - Permettre aux membres de planifier des projets à l'aide d'itérations
 - Tâches représentées dans une itération, et plusieurs itérations dans un projet

Fonctionnalités secondaires

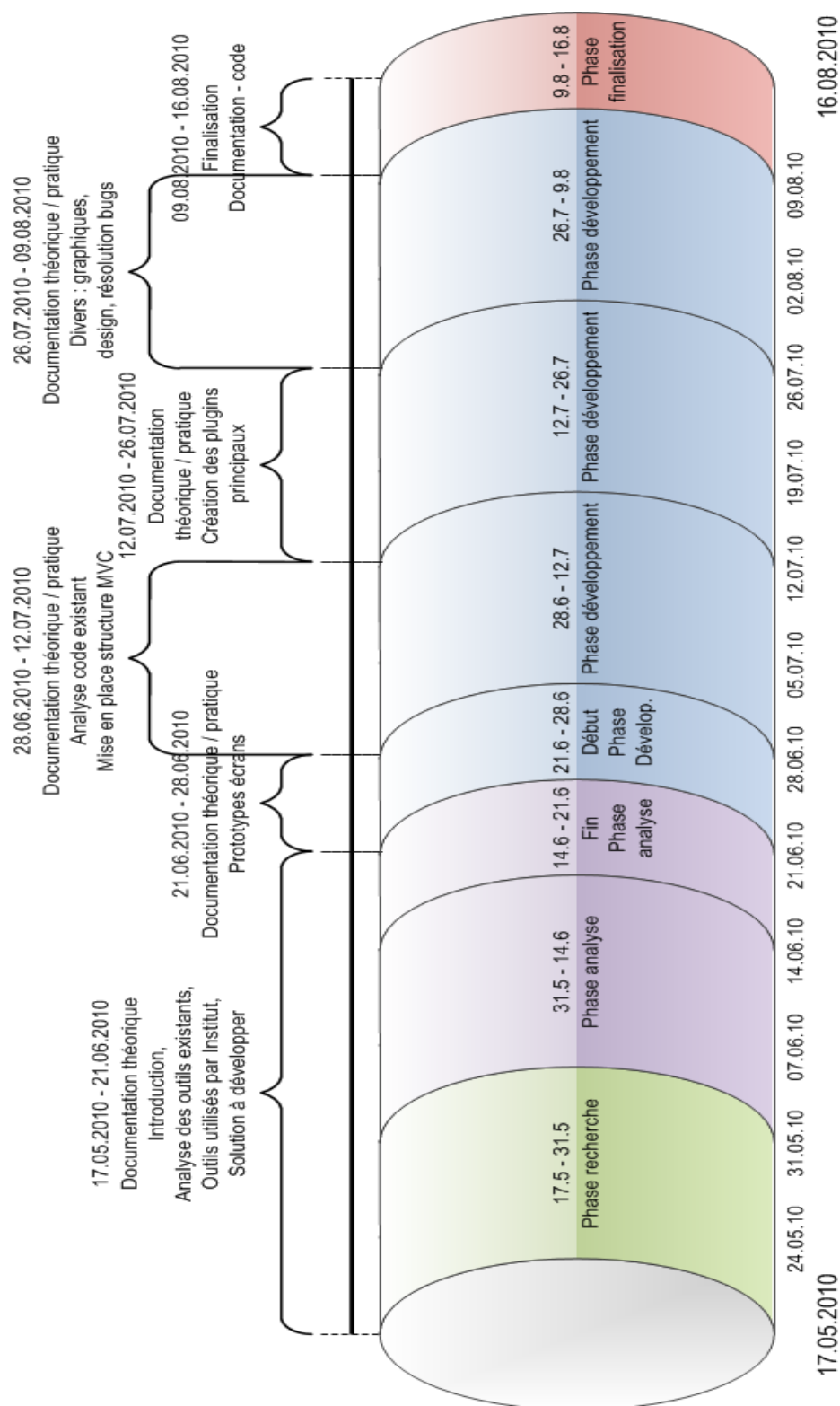
- Divers
 - Alertes par email lors de dépassements des heures planifiées
 - Intégration au SVN (Sources des projets en cours de développement)

- Graphiques
 - Représentation graphique du travail qu'il reste à faire en fonction du temps
 - Par ex : Sprint Burndownchart ou plan de charges
- Ergonomie
 - Implémenter des fonctionnalités comme le Drag&Drop, Ajax, JQuery,...
 - Améliorer l'ergonomie pour un usage quotidien
 - Graphisme / design de l'outil

Démarche projetée afin d'atteindre les objectifs fixés

- Analyse des outils existants et ceux utilisés actuellement par l'institut Icare, que sont « Intra-Know » et « Redmine »
 - Comparaison des outils, compte-rendu avec points positifs / négatifs
- Interview et séances avec tous les membres de l'institut Icare, afin de définir plus précisément les points négatifs et positifs des outils utilisés, et des fonctionnalités nécessaires manquantes actuellement
 - Points positifs / négatifs des applications « Intra-Know » et « Redmine »
- Déterminer la meilleure solution à développer, soit :
 - Créer des plugins pour « Redmine » fournissant les différentes fonctionnalités nécessaires
 - Trouver une autre application existante à modifier / améliorer
 - Partir de zéro en développant une nouvelle application sur « RoR »
- Définir, avec priorité, les différentes étapes nécessaires au développement de l'outil et estimer le temps nécessaire sous un langage peu connu (Ruby) et son Framework (Rails)
- Développer une solution de gestion de projet et de ressources qui remplit les fonctionnalités demandées
 - Création des prototypes écrans
 - Modèles-Vues-Contrôleurs (MVC) sous « Ruby On Rails »
 - Design HTML / CSS, interfaces web classiques
 - Base de données SQL (MySQL, PostgreSQL, SQLite,...)
- Fournir un livrable fonctionnel accordé aux besoins de l'institut

11.2. Planification initiale



11.3. Planification détaillée

Projet Travail de Bachelor Correia - Outil web AGILE																											
Semaine		Mai					Juin					Juillet					Août										
		20	21	22	23	24	25	26	27	28	29	30	31														
▲ 1 - Documentation																											
	1	Partie introduction		0j/2j		TC	21		▲ 0%																		
	2	Partie théorique		0j/6j		TC			26		▲ 0%																
	3	Partie pratique		0j/6j		TC					30		▲ 0%														
	4	Partie finale		0j/1j		TC					32		▲ 0%														
▲ 2 - Recherche																											
	1	Etat de l'art des outils de gestion de projet		0j/2.5j		TC	21		▲ 0%																		
▲ 3 - Analyse																											
	1	Comparaison des outils		0j/2.5j		TC	22		▲ 0%																		
▲ 4 - Déterminer solution à développer																											
	1	Analyse du choix de l'outil		0j/5j		TC			24		▲ 0%																
▲ 5 - Développement d'un outil de gestion de projets																											
	2	Analyse de la bdd Redmine		0j/1j		TC			26		▲ 0%																
	3	Analyse de la structure MVC Redmine		0j/1j		TC			26		▲ 0%																
	4	Tutorial création plugin Redmine		0j/0.5j		TC			26		▲ 0%																
	5	Fonctionnalités principales		0j/0j		TC			29		▲ 0%																
	6	Fonctionnalités optionnelles		0j/0j		TC			31		▲ 0%																
▲ 6 - Dév. des fonctionnalités principales																											
	1	Plugin "Plan de charge"		0j/6j		TC			28		▲ 0%																
	2	Plugin "Ma page"		0j/2j		TC			28		▲ 0%																
	3	Plugin "Calendrier"		0j/2.5j		TC			29		▲ 0%																
	4	Plugin "Budget"		0j/1.5j		TC			29		▲ 0%																
▲ 7 - Dév. des fonctionnalités optionnelles																											
	1	Drag&Drop		0j/3.5j		TC			30		▲ 0%																
	2	Divers - Commit SVN / Emails / Graphiques / Ergonomie		0j/4.5j		TC			31		▲ 0%																
▲ 8 - Fournir solution fonctionnelle																											
	1	Correction des bugs		0j/4j		TC			32		▲ 0%																

11.4. Détail des heures réalisées

Thomas Correia		8h-10h	10h-12h	13h-15h	15h-17h
Total heures : 402:29:00					
Semaine 1 -					
Heures : 6:15:00					
17.05.2010	0	Séance d'information sur le travail de Bachelor	Présentation d'un projet pour l'école	Présentation d'un projet pour l'école	Présentation d'un projet pour l'école
18.05.2010	0	Cours	Cours	Préparation de deux présentations pour des cours	Préparation de deux présentations pour des cours
20.05.2010	6:15:00	Visionnage de vidéos sur VersionOne / Trial de VersionOne - test de l'application	Installation de Redmine en local / recherche de plugins Scrum pour Redmine / installation de msysGit + plugin Redmine Scrum PM	Test et configuration de Scrum PM sous Redmine / Divers préparations (dropbox, google docs,...)	Mise en place de différents documents (journal de bord, sources, rapport hebdomadaire, squelette TB, cahier des charges)
Semaine 2 -					
Heures : 11:00:00					
24.05.2010	0	Féfé	Féfé	Féfé	Féfé
25.05.2010	2:30:00	Cours	Cours	Séance avec client - Discussion sur les objectifs à réaliser, fonctionnalités de l'outil, cahier des charges	Rédaction du cahier des charges (objectifs, fonctionnalités, buts, démarche projetée)
27.05.2010	8:30:00	Finalisation de la rédaction du cahier des charges	Test de l'application VersionOne - Début documentation, première itération de la table des matières du document	Installation / Essai / Analyse de différents outils AGILE - Redmine Scrum PM - VersionOne - Intra-know	Installation / Essai / Analyse de différents outils AGILE - Icescrum - Basecampe - Début compte-rendu
Semaine 3 -					
Heures : 10:15:00					
31.05.2010	8:15:00	Modifications sur le cahier des	Rédaction document final -	Rédaction document final -	Planification du projet sur



		charges / Rédaction document final (Technologie, méthodologie Scrum,...)	Modifications des textes de Wikipedia sur Scrum / Ruby / RoR	Modifications des textes de Wikipedia sur Scrum / Installation Microsoft Visio / Daemon Tools	outil Intra- Know / Planification globale sur Visio
					Analyse de l'outil Redmine / Test de compatibilité entre différentes versions et plugins
01.06.2010	2:00:00				
03.06.2010	0	Férié	Férié	Férié	Férié
Semaine 4 -					
Heures :		11:50:00			
			Analyse Redmine / Configuration des emails sous Redmine en passant par Gmail	Installation plugin Budget pour Redmine - Traduction du plugin pour langue FR	Analyse de différents plugins pour Redmine - Workload Manager Redmine Plugin
07.06.2010	6:20:00				
					Rendez-vous avec la team de développeur pour interviews - Tests plugin Treeview
08.06.2010	2:30:00	Cours	Cours		
10.06.2010	3:00:00	Job	Job	Séance avec client	Séance avec client
Semaine 5 -					
Heures :		33:20:00			
14.06.2010	6:00:00	Documentation	Documentation	Documentation	Documentation
15.06.2010	6:30:00	Documentation	Documentation	Documentation	Documentation
16.06.2010	9:50:00	Documentation	Documentation	Documentation / Séance avec client	Documentation
					Workflow - Prototypes écrans / Test plugin Schedule / Documentation / Présentation PPT pour team Icare
17.06.2010	11:00:00	Schéma mindmap propositions développeurs	Documentation	Workflow - Prototypes écrans / Séance avec client	



Semaine 6 -					
Heures :		32:15:00			
21.06.2010	0	Révision examens	Révision examens	Révision examens	Révision examens
22.06.2010	10:45:00	Prototypes écrans / Documentation	Prototypes écrans / Documentation	Prototypes écrans / Documentation	Documentation
23.06.2010	7:30:00	Prototypes écrans	Prototypes écrans	Prototypes écrans / Planification détaillée	Prototypes écrans / Séance avec responsable / Planification détaillée
24.06.2010	9:00:00	Révision examens	Révision examens	Prototypes écrans / Test Redmine "commit issue with time spent & ratio % for an id #"	Prototypes écrans / Test Redmine "commit issue with time spent & ratio % for an id #"
25.06.2010	5:00:00	Révision examens	Commit issues with ratio	Commit issues with ratio / Tests compatibilité	Commit issues with ratio / Tests compatibilité
Semaine 7 -					
Heures :		11:25:00			
28.06.2010	0	Examen	Examen	Examen	Examen
29.06.2010	6:55:00	Regex commit log message / Modifications des prototypes écrans	Modifications des prototypes écrans	Modifications des prototypes écrans / Subversion avec Redmine 0.9.5	Modifications des prototypes écrans / Liste des fonctionnalités / Validation par la team ICare
30.06.2010	0	Examen	Examen	Examen	Examen
01.07.2010	0	Examen	Examen	Examen	Examen
02.07.2010	1:00:00	Job	Séance Icare sur prototypes écrans	Job	Job
03.07.2010	3:30:00	Développement du plugin "Plan de charges"	Commit log / Documentation	Commit log / Documentation	Job
Semaine 8 -					
Heures :		61:25:00			
05.07.2010	7:25:00	Documentation / Début développement plugin "Plan de charges"	Réinstallation du système d'exploitation de la machine de travail	Réinstallation des logiciels nécessaires (Office, Ruby, MySQL,...)	Test plugin Redmine - Override de methodes existantes / Dev des



					models et contrôleurs nécessaires pour le plugin "plan de charges"
06.07.2010	11:30:00	Développement plugin "plan de charges" - Base du plugin / Models et contrôleurs	Développement plugin "plan de charges" - Base du plugin / Models et contrôleurs	Développement plugin "plan de charges" - Base du plugin / Models et contrôleurs	Développement plugin "plan de charges" - Base du plugin / Models et contrôleurs
07.07.2010	12:30:00	Développement plugin "plan de charges" - Algorithme calculs heures	Développement plugin "plan de charges" - Algorithme calculs heures	Développement plugin "plan de charges" - Algorithme calculs heures	Développement plugin "plan de charges" - Algorithme calculs heures
08.07.2010	7:30:00	Développement plugin "plan de charges" - Correction des bugs	Développement plugin "plan de charges" / Mise à jour du cahier des charges	Développement plugin "plan de charges" - Correction des bugs	Développement plugin "plan de charges" - Correction des bugs
09.07.2010	3:00:00	Développement plugin "plan de charges" - Correction des bugs	Développement plugin "plan de charges" - Correction des bugs		
10.07.2010	8:30:00	Développement plugin "plan de charges" - Réécriture de l'algorithme de calculs des heures	Développement plugin "plan de charges" - Réécriture de l'algorithme de calculs des heures	Développement plugin "plan de charges" - Réécriture de l'algorithme de calculs des heures	Développement plugin "plan de charges" - Réécriture de l'algorithme de calculs des heures
11.07.2010	11:00:00	Développement plugin "plan de charges" - Design de l'interface	Développement plugin "plan de charges" - Design de l'interface	Développement plugin "plan de charges" - Design de l'interface	Développement plugin "plan de charges" - Design de l'interface
Semaine 9 - Heures :		48:15:00			
12.07.2010	3:00:00	Développement plugin "plan de charges" - Fonctionnalités manquantes	Développement plugin "plan de charges" - Fonctionnalités manquantes		
13.07.2010	12:00:00	Développement plugin "plan de charges" - Fonctionnalités manquantes	Développement plugin "plan de charges" - Fonctionnalités manquantes	Développement plugin "plan de charges" - Fonctionnalités manquantes /	Développement plugin "plan de charges" - Fonctionnalités manquantes /



			Autorisation des accès	Autorisation des accès
				Début développement d'un menu contextuel pour le plan de charges / Début développement plugin "Heures de semaine"
14.07.2010	9:15:00	Développement plugin "plan de charges" - Modification d'un controller existant dans le core de Redmine	Développement plugin "plan de charges" - Modification d'un controller existant dans le core de Redmine	
15.07.2010	11:30:00	Implémentation du contexte menu existant de Redmine dans plugin "plan de charges" / Réécriture du controller "plan de charges"	Implémentation du contexte menu existant de Redmine dans plugin "plan de charges" / Réécriture du controller "plan de charges"	Implémentation du contexte menu existant de Redmine dans plugin "plan de charges" / Réécriture du controller "plan de charges"
16.07.2010	6:00:00	Corrections de bugs pour le plugin "plan de charges"	Mise en production du plugin "plan de charges" / Développement du plugin "Heures de la semaine" - Implémentation du calendrier jQuery DatePicker pour sélectionner une semaine entière	Implémentation du controller et model "disponibilités" pour le plugin "Heures de la semaine"
18.07.2010	6:30:00		Corrections de bugs pour le plugin "plan de charges"	Corrections de bugs pour le plugin "plan de charges"
Semaine 10 - Heures : 40:15:00				
19.07.2010	6:15:00	Début développement plugin "Calendrier" / Migration version 1.0.0 sur production RFID - Séance lundi RFID	Migration version 1.0.0 sur production RFID	Développement plugin "Calendrier" / Utilisation du googlecalendar gem
20.07.2010	13:30:00	Développement plugin "Calendrier" - Divers tests	Développement plugin "Calendrier" - Divers tests	Développement plugin "Calendrier" - Implémentation



		avec différentes API	avec différentes API	avec différentes API	de l'API GCal4Ruby avec un calendrier dynamique JQuery
21.07.2010	7:00:00	Développement plugin "Calendrier" - Implémentation de l'API GCal4Ruby avec un calendrier dynamique JQuery	Développement plugin "Calendrier" - Implémentation de l'API GCal4Ruby avec un calendrier dynamique JQuery	Développement plugin "Calendrier" / Abandon du calendrier JQuery / Implémentation de l'API GCal4Ruby avec IFrame Google	Développement plugin "Calendrier" / Abandon du calendrier JQuery / Implémentation de l'API GCal4Ruby avec IFrame Google
22.07.2010	7:00:00	Développement plugin "Calendrier" / Gestion des utilisateurs, roles,...	Développement plugin "Calendrier" / Gestion des utilisateurs, roles,...	Développement plugin "Calendrier" / Création de la vue	Développement plugin "Calendrier" / Création de la vue
23.07.2010	6:30:00	Développement plugin "Calendrier" / Gestion des disponibilités	Développement plugin "Calendrier" / Gestion des disponibilités	Développement plugin "Calendrier" / Gestion des disponibilités	Développement plugin "Calendrier" / Gestion des disponibilités
Semaine 11 - Heures :		59:30:00			
26.07.2010	6:45:00	Correction de bugs plugin "calendrier" liés à la vue	Correction de bugs plugin "calendrier" liés à la vue	Correction de bugs plugin "calendrier" liés à l'API	Mise en production du plugin "Calendrier"
27.07.2010	9:40:00	Correction de bugs plugin "calendrier" liés à l'API, à la gestion des disponibilités	Correction de bugs plugin "calendrier" liés à l'API, à la gestion des disponibilités	Tests du plugin "Calendrier" / Mise en production du plugin "Calendrier" mis à jour	Début développement plugin "Budget" / Création de la vue
28.07.2010	10:30:00	Développement plugin "Budget" - création de la vue	Développement plugin "Budget" - création de la vue	Développement plugin "Budget" - Implémentation de Ajax RJS et Prototype pour insertion de données	Développement plugin "Budget" - Implémentation de Ajax RJS et Prototype pour insertion de données
29.07.2010	6:00:00		Développement plugin "Budget"	Développement plugin "Budget"	Développement plugin



			- Edit des champs et validation formulaire	- Correction de bugs liés aux IDS	"Budget" - Correction des bugs liés à Ajax et RJS
30.07.2010	7:00:00	Implémentation du drag & drop pour le plan de charges / Apprentissage de Script Aculo Us (Prototype)	Implémentation du drag & drop pour le plan de charges / Apprentissage de Script Aculo Us (Prototype)	Implémentation du drag & drop pour le plan de charges / Tests du drag & drop Script Aculo Us (Prototype)	Implémentation du drag & drop pour le plan de charges / Tests du drag & drop Script Aculo Us (Prototype)
31.07.2010	9:35:00	Implémentation du drag & drop pour le plan de charges / Modification du code pour utilisation correcte du drag & drop (Implémentation des IDS pour les zones draggables et droppables)	Implémentation du drag & drop pour le plan de charges / Modification du code pour utilisation correcte du drag & drop (Implémentation des IDS pour les zones draggables et droppables)	Implémentation du drag & drop pour le plan de charges / Modification du code pour utilisation correcte du drag & drop (Implémentation des IDS pour les zones draggables et droppables)	Implémentation du drag & drop pour le plan de charges / Implémentation des effets lors d'un dragdrop
01.08.2010	10:00:00	Implémentation du drag & drop pour le plan de charges / Implémentation des effets lors d'un dragdrop	Implémentation du drag & drop pour le plan de charges / Implémentation des effets lors d'un dragdrop	Implémentation du drag & drop pour le plan de charges / Ajout d'une box pour choix d'options	Implémentation du drag & drop pour le plan de charges / Ajout d'une box pour choix d'options
Semaine 12 - Heures :		34:14:00			
02.08.2010	10:15:00		Modification du drag & drop pour le plan de charges / Retrait du box et implémentation d'une nouvelle façon de drag	Modification du drag & drop pour le plan de charges / Retrait du box et implémentation d'une nouvelle façon de drag	Modification du drag & drop pour le plan de charges / Retrait du box et implémentation d'une nouvelle façon de drag
03.08.2010	5:45:00	Développement drag & drop	Développement drag & drop	Développement drag & drop	Développement drag & drop
04.08.2010	7:10:00	Développement drag & drop	Développement drag & drop	Implémentation finale des plugins dans le mode Production / Tests globaux	Implémentation finale des plugins dans le mode Production / Tests globaux



05.08.2010	6:34:00	Reprise du document	Document TB / Séance avec responsable	Document TB	Document TB
06.08.2010	4:30:00		Document TB	Document TB	Document TB
Semaine 13 -					
Heures :	42:30:00				
09.08.2010	0	job	job	job	job
10.08.2010	6:30:00	Documentation	Documentation	Documentation	Documentation
11.08.2010	7:00:00	Documentation	Documentation	Documentation	Documentation
12.08.2010	7:00:00	Documentation	Documentation	Documentation	Documentation
13.08.2010	6:30:00	Documentation	Documentation	Documentation	Documentation
14.08.2010	3:00:00	Documentation	Documentation	Documentation	Documentation
15.08.2010	12:30:00	Documentation	Documentation	Documentation	Création des CD / Finalisation du document

11.5. Fonctionnalités existantes

Mon compte

Permet de modifier les informations liées au compte : Nom, prénom, adresse mail, la langue de l'outil, les notifications par email, pouvoir cacher son email, le fuseau horaire, l'affichage des commentaires et la réinitialisation des clefs RSS et API.

Figure 24 : Interface “Mon compte” de Redmine

Accueil

La page d'accueil ne possède rien de très intéressant, mise à part le fait qu'elle est personnalisable dans la configuration de l'outil et que l'on a un aperçu des derniers projets créés.

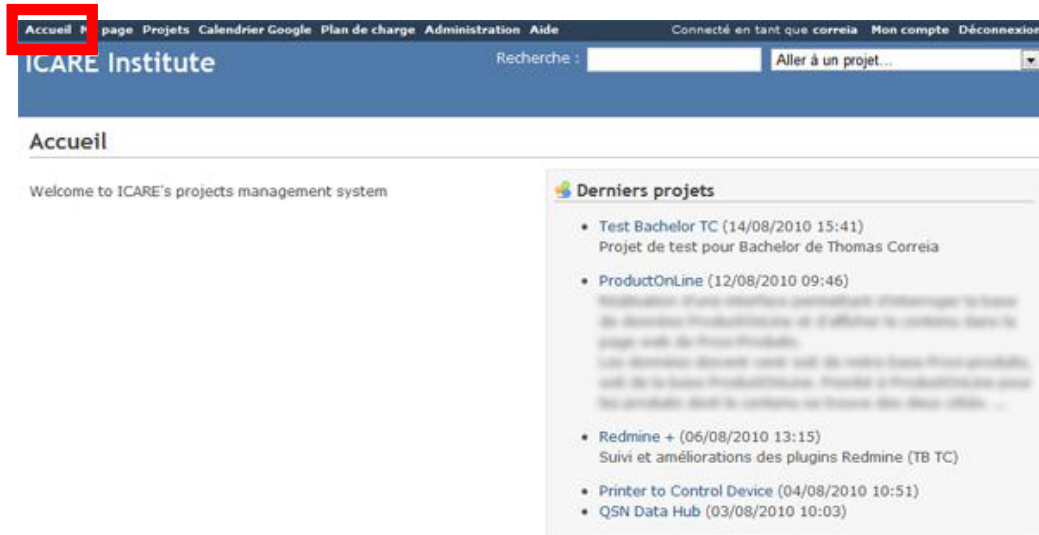


Figure 25 : Interface “Accueil” de Redmine

Ma Page

Chaque utilisateur peut personnaliser cette interface en y ajoutant un ou plusieurs modules qui sont déplaçables et parmi lesquels on trouve :

- Demandes qui me sont assignées
- Demandes surveillées
- Temps passé
- Documents
- Demandes soumises
- Dernières annonces
- Calendrier



Accueil **Ma page** Projets Calendrier Google Plan de charge Administration Aide Connecté en tant que correia Mon compte Déconnexion

ICARE Institute Recherche : Aller à un projet...

Ma page Personnaliser cette page

Demandes qui me sont assignées (4)

#	Projet	Tracker	Sujet
1039	Infrastructure	Task	Modifier le code et services (Banc)
1127	ProductOnLine	Task	Intégration des données avec le
1126	ProductOnLine	Task	Modifications de l'interface avec
928	Non-financés	Task	Prendre en compte avec un

Voir toutes les demandes

Demandes surveillées (2)

#	Projet	Tracker	Sujet
1072	LeClerc	Task	Resoudre le problème (Banc)
894	TB Correia	Bug	afficher que les développeurs et les managers dans "Plan ... (Resolved)

Voir toutes les demandes

Dernières annonces

Documents

Figure 26 : Interface “Ma page” de Redmine

Liste des projets

La liste des projets affiche tous les projets actifs et permet d’en créer de nouveaux, d’afficher toutes les demandes, de voir le temps passé et les activités de toutes les ressources.

Accueil Ma page **Projets** Calendrier Google Plan de charge Administration Aide Connecté en tant que correia Mon compte Déconnexion

ICARE Institute Recherche : Aller à un projet...

Projets Nouveau projet | Voir toutes les demandes | Overall spent time | Activité globale

Nouveaux
Description du projet à créer (à partir d'un modèle existant)

Existant
Rechercher le projet existant par nom
Il y a 100 projets existants, voici un aperçu des 10 premiers projets existants

Statut
Statut du projet (à partir d'un modèle existant)

Statut
Statut du projet (à partir d'un modèle existant)

Statut
Statut du projet (à partir d'un modèle existant)

Figure 27 : Interface “Liste des projets” de Redmine

Nouveau projet

Permet de créer un nouveau projet en spécifiant son nom, s’il est sous-projet d’un autre projet existant, une description, un identifiant unique, une adresse de site web, s’il est public (si tout le monde peut y accéder), les trackers associés au projet et les modules disponibles.

Accueil Ma page Projets Calendrier Google Plan de charge Administration Aide Connecté en tant que correia Mon compte Déconnexion

ICARE Institute Recherche : Aller à un projet...

Nouveau projet

Nom *
30 caractères maximum.

Sous-projet de

Description **B** **I** **U** **S** **C** **H1** **H2** **H3** **≡** **≡** **≡** **pre** **img** **Formatage du texte: Aide**

Identifiant *
Longueur comprise entre 1 et 20 caractères. Seuls les lettres minuscules (a-z), chiffres et tirets sont autorisés.
Un fois sauvegardé, l'identifiant ne pourra plus être modifié.

Site web

Public ☐

Trackers

☒ Bug ☒ Evolution ☒ Assistance

☒ Task

Modules

☒ Suivi des demandes ☒ Suivi du temps passé ☒ Publication d'annonces

☒ Publication de documents ☒ Publication de fichiers ☒ Wiki

☒ Dépôt de sources ☒ Forums de discussion ☐ Budget

Figure 28 : Interface “Nouveau projet” de Redmine

Temps passé

Le temps passé est un reporting des heures effectuées par toutes les ressources. La partie « Détails » affiche pour chaque membre le temps passé pour chaque demande

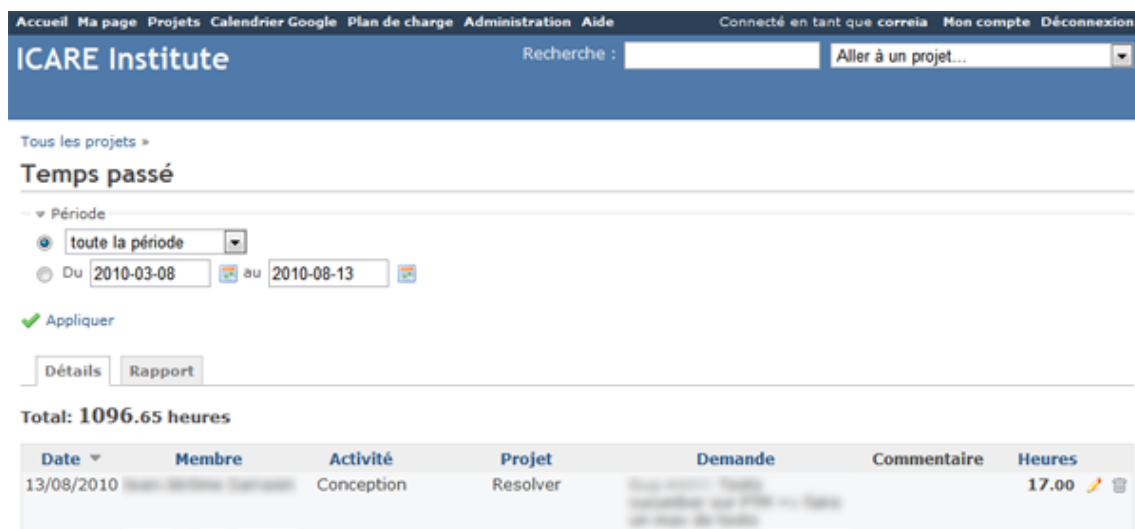


Figure 29 : Interface “Temps passé - Détails” de Redmine

La partie « Rapport » permet de choisir les catégories à afficher et se trouve être très flexible et pratique

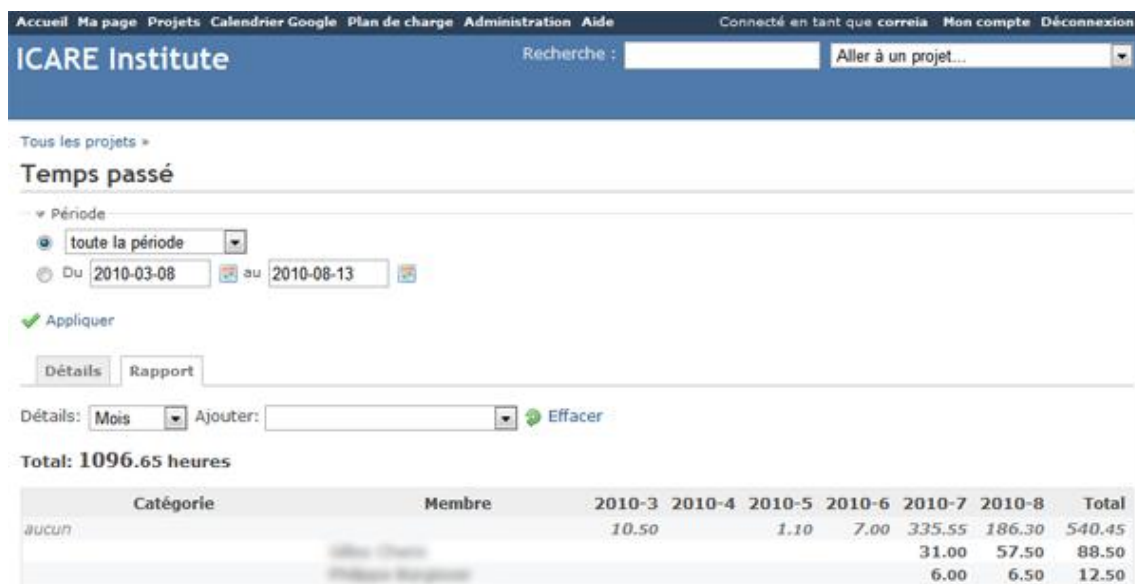


Figure 30 : Interface “Temps passé - Rapport” de Redmine



Activité

Affiche toutes les activités effectuées sur l'application (saisie d'heures, nouvelles demandes,...)



Figure 31 : Interface “Activité” de Redmine

Administration

Sur cette page, on peut administrer entièrement l'application si l'on est désigné comme « Administrateur ».

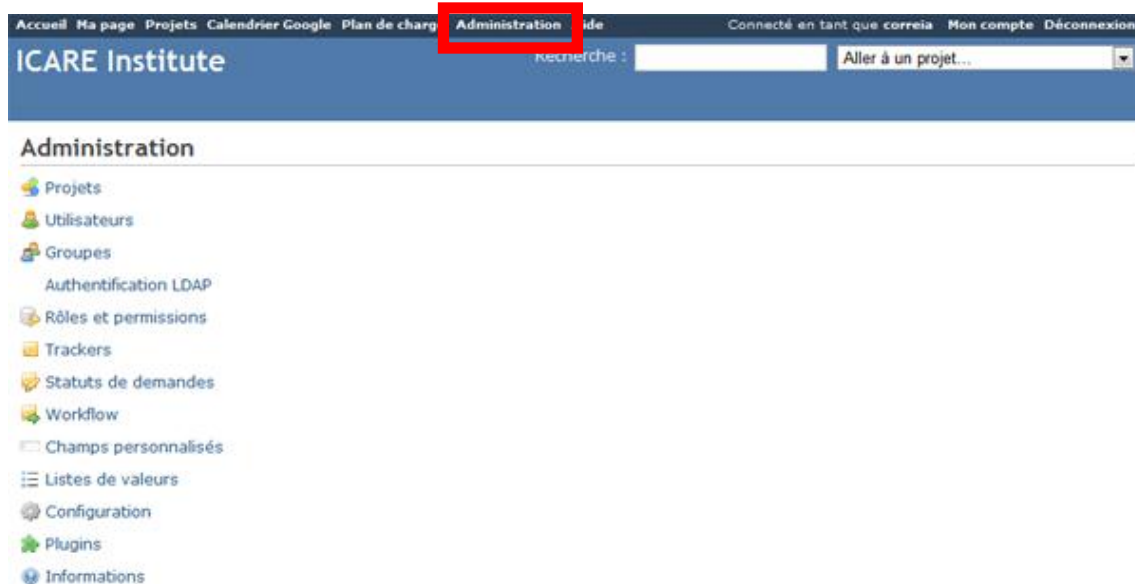


Figure 32 : Interface principale “Administration” de Redmine

Les possibilités sont :

- Projets
 - Lister, archiver, copier, créer et modifier des projets
- Utilisateurs
 - Lister, filtrer, verrouiller, créer et modifier des utilisateurs
 - Spécifier les groupes de l'utilisateur et ses rôles dans chaque projet
- Groupes
 - Lister, créer et supprimer des groupes
- Gérer l'authentification LDAP
- Rôles et permissions
 - Lister, créer et modifier des rôles (manager, développeur,...) et assigner des permissions à ces rôles (pour tous les modules de projets)
- Trackers
 - Lister, créer, modifier et supprimer des trackers (types de demandes, par exemple bug, tâche, évolution,...)
- Status des demandes
 - Lister, order, créer, modifier et supprimer les status des demandes (Nouveau, résolu, en attente,...)
- Workflow
 - Le workflow est une interface très complète permettant de spécifier pour chaque tracker et chaque rôle, quels sont les status possibles. Par exemple, une nouvelle demande est enregistrée avec le tracker « bug » et un développeur souhaite changer son status de « nouveau » à « résolu ». Cette modification dépendra alors de la configuration du workflow et ne seront affichés que les choix possibles pour le développeur
- Champs personnalisés
 - L'interface des champs personnalisés est très complexe et permet de créer de nouveau champs pour la plupart des interfaces disponibles. Par exemple, il est possible de créer un nouveau champ « Description détaillée » dans un nouveau projet.
- Liste des valeurs
 - Lister, order, créer, modifier et supprimer les valeurs des Activités (suivi du temps), des priorités des demandes et des catégories des documents

- Configuration
 - Interface très complète et complexe qui permet de gérer les paramètres de l'application comme les informations générales, l'affichage, l'authentification, les options des projets, le suivi des demandes, les notifications par email, la lecture des emails entrants et les dépôts des codes sources
- Plugins
 - Lister et configurer si possible les plugins installés. Nous y reviendrons plus tard dans la partie développement
- Informations
 - Liste des informations cruciales comme la version de l'application par exemple « Redmine 1.0.0.stable (MySQL) », si le compte administrateur par défaut a été changé (admin/admin), si le répertoire de stockage des fichiers est accessible en écriture, si le répertoire public des plugins est accessible en écriture et si la bibliothèque RMagick est présente (gem de Ruby)

Aide

Renvoie l'utilisateur vers <http://www.redmine.org/guide> où l'on trouve une aide sur Redmine avec des explications très détaillées

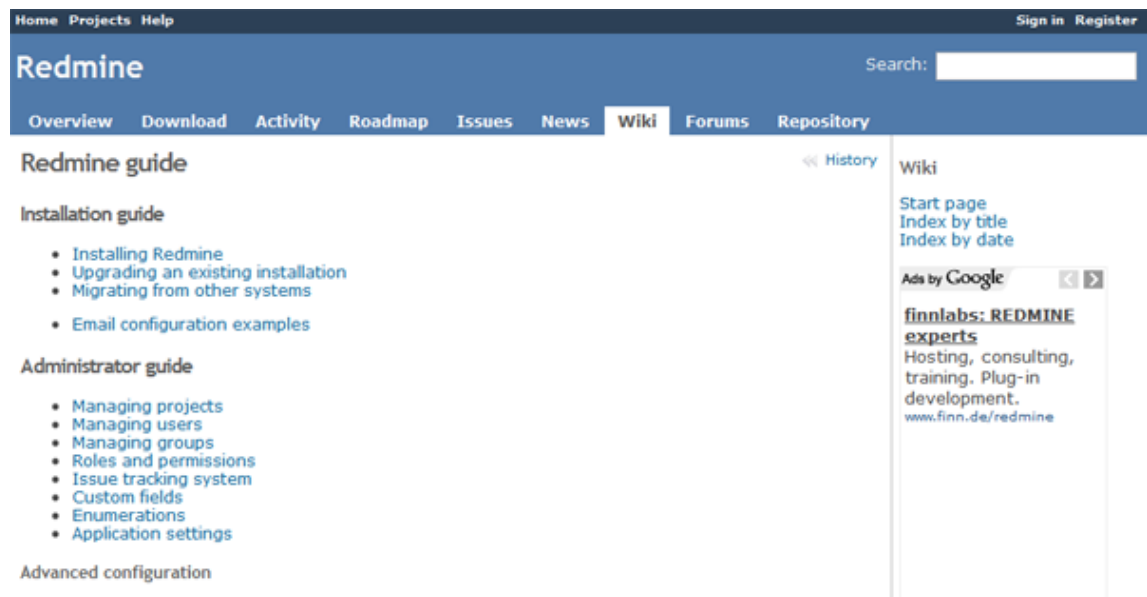


Figure 33 : Interface “Aide” de Redmine³⁴

Source : redmine.org

³⁴ Source : redmine.org, consulté le 14/08/2010, <http://www.redmine.org/guide>



Menu projet

Le menu d'un projet se décompose en plusieurs modules dont certains sont optionnels comme le suivi des demandes, le suivi du temps passé, la publication d'annonces, la publication de documents, la publication de fichiers, le wiki, le dépôt de sources et les forums de discussion.



Figure 34 : Menu "Projet" de Redmine

Projet / Aperçu

L'aperçu d'un projet permet de rapidement voir les ressources associées, le temps passé sur le projet (selon utilisateur connecté) et le suivi des demandes



Figure 35 : Interface "Projet – Aperçu" de Redmine

Projet / Roadmap

Le Roadmap permet de visualiser les demandes ouvertes et fermées, par itération (version) et également avoir un précieux pourcentage d'avancement de cette itération.



Accueil Ma page Projets Calendrier Google Plan de charge Administration Aide Connecté en tant que correia Mon compte Déconnexion

Test Bachelor TC

Recherche : Aller à un projet...

Aperçu Activité **Roadmap** Demandes Nouvelle demande Budget Annonces Documents Wiki Fichiers Configuration

Roadmap

0.0.1

itération 1

zéro fermé (0%) 3 ouverts (100%)

Demandes liées

Bug #1152: Bug 1
Task #1150: Tâche 1
Task #1151: Tâche 2

0.0.2

itération 2

zéro fermé (0%) 3 ouverts (100%)

Demandes liées

Evolution #1155: Evolution 1
Task #1153: Tâche 1
Task #1154: Tâche 2

Roadmap

☒ Bug
☒ Evolution
☐ Assistance
☒ Task

☐ Voir les versions passées

Appliquer

Versions

0.0.1
0.0.2

Figure 36 : Interface “Projet – Roadmap” de Redmine

Projet / Liste des demandes

Cette interface liste les demandes du projet. On peut également accéder à un résumé peu utile, à un calendrier qui affiche les demandes par date de début et date d’échéance, et à un graphique Gantt pour une aide à la planification.

Accueil Ma page Projets Calendrier Google Plan de charge Administration Aide Connecté en tant que correia Mon compte Déconnexion

Test Bachelor TC

Recherche : Aller à un projet...

Aperçu Activité Roadmap **Demandes** Nouvelle demande Budget Annonces Documents Wiki Fichiers Configuration

Demandes

Filtres

☒ Statut Ajouter le filtre :

Options

Appliquer Effacer Sauvegarder

#	Tracker	Statut	Priorité	Sujet	Assigné à	Catégorie	Début	Échéance	Temps estimé	% réalisé
1155	Evolution	New	Normal	Evolution 1	Bachelor Test	24/08/2010	12/10/2010	14.0		14
1154	Task	New	Normal	Tâche 2	Bachelor Test	10/09/2010	01/10/2010	4.0		14
1153	Task	New	Normal	Tâche 1	Bachelor Test	31/08/2010	21/09/2010	8.0		14
1152	Bug	New	Normal	Bug 1	Bachelor Test	17/08/2010	31/08/2010	2.0		14
1151	Task	New	Normal	Tâche 2	Bachelor Test	18/08/2010	22/09/2010	16.0		14
1150	Task	New	Normal	Tâche 1	Bachelor Test	16/08/2010	06/09/2010	10.0		14

(1-6/6) | Par page : 25, 50, 100

Formats disponibles : Atom | CSV | PDF

Demandes

Voir toutes les demandes
Résumé
Calendrier
Gantt

Figure 37 : Interface “Projet – Liste des demandes” de Redmine



Projet / Nouvelle demande

Chaque demande possède un tracker, un sujet, une description, un statut, une priorité, une assignation, une catégorie, une date de début, une date d'échéance, un temps estimé, un pourcentage d'avancement, un ou plusieurs fichiers, une description facultative et un ou plusieurs observateurs.

Figure 38 : Interface “Projet – Nouvelle demande” de Redmine

Projet / Annonces

Les annonces permettent d’afficher un message pour toutes les ressources associées au projet.

Projet / Wiki

Le Wiki sert de plateforme d’échange pour livrer des explications sur un sujet ou thème concernant le projet.

Projet / Fichiers

L’interface des Fichiers permet aux utilisateurs de rajouter des fichiers qui seront



disponibles dans le projet.

Projet / Configuration

La configuration d'un projet permet de remplir ses informations, de sélectionner les modules disponibles, de sélectionner les membres pouvant accéder au projet, de spécifier les versions (itérations) du projet, mais également les catégories de demandes, de créer un wiki, un dépôt de code source, un ou des forums et les types d'activités disponibles dans le projet.

The screenshot shows the 'Configuration' page for a project named 'Test Bachelor TC'. The interface includes a top navigation bar with links like 'Accueil', 'Ma page', 'Projets', 'Calendrier Google', 'Plan de charge', 'Administration', and 'Aide'. Below this is a search bar and a dropdown to 'Aller à un projet...'. The main content area has tabs for 'Informations', 'Modules', 'Membres', 'Versions', 'Catégories de demandes', 'Wiki', 'Dépôt', 'Forums', and 'Activités (suivi du temps)'. The 'Informations' tab is active, showing fields for 'Nom' (Test Bachelor TC), 'Sous-projet de', 'Description' (Projet de test pour Bachelor de Thomas Correia), 'Identifiant' (test-bachelor-tc), 'Site web', and a 'Public' checkbox. At the bottom, there are 'Trackers' (Bug, Task, Evolution, Assistance) and a 'Sauvegarder' button.

Figure 39 : Interface “Projet – Configuration” de Redmine



12. Table des illustrations

Figure 1 : Schéma du cycle de Scrum	4
Figure 2 : Graphique radar des résultats de l'état de l'art	21
Figure 3 : Menu principal de Redmine	23
Figure 4 : Prototype "Plan de charges – Par ressource"	26
Figure 5 : Prototype "Plan de charges – Par groupe"	27
Figure 6 : Prototype "Ma page – Heures de la semaine"	28
Figure 7 : Prototype "Calendrier"	29
Figure 8 : Prototype "Budget"	30
Figure 9 : Planification du développement	33
Figure 10 : Liste des plugins dans "Administration" de Redmine	34
Figure 11 : Configuration du plugin "Plan de charges"	35
Figure 12 : Interface du plugin "Plan de charges – Par ressource"	36
Figure 13 : Menu contextuel de Redmine	36
Figure 14 : Interface du plugin "Plan de charges – Par groupe"	37
Figure 15 : Interface plugin "Ma page – Reporting semaine"	38
Figure 16 : Interface plugin "Calendrier"	39
Figure 17 : Interface plugin "Calendrier – nouvel évènement"	40
Figure 18 : Interface plugin "Calendrier - Ajout d'absence"	41
Figure 19 : Interface plugin "Calendrier – modification du mot de passé"	41
Figure 20 : Interface plugin "Budget"	42
Figure 21 : Autorisations pour plugin "Budget"	42
Figure 22 : Configuration "notifications par mail" dans Redmine	44
Figure 23 : Drag&Drop plugin "Plan de charges par ressource"	46
Figure 24 : Interface "Mon compte" de Redmine	67
Figure 25 : Interface "Accueil" de Redmine	68
Figure 26 : Interface "Ma page" de Redmine	69
Figure 27 : Interface "Liste des projets" de Redmine	69
Figure 28 : Interface "Nouveau projet" de Redmine	70
Figure 29 : Interface "Temps passé - Détails" de Redmine	71
Figure 30 : Interface "Temps passé - Rapport" de Redmine	71
Figure 31 : Interface "Activité" de Redmine	72
Figure 32 : Interface principale "Administration" de Redmine	72
Figure 33 : Interface "Aide" de Redmine	74
Figure 34 : Menu "Projet" de Redmine	75
Figure 35 : Interface "Projet – Aperçu" de Redmine	75
Figure 36 : Interface "Projet – Roadmap" de Redmine	76
Figure 37 : Interface "Projet – Liste des demandes" de Redmine	76



Figure 38 : Interface “Projet – Nouvelle demande” de Redmine _____	77
Figure 39 : Interface “Projet – Configuration” de Redmine _____	78

13. Bibliographie

Scrum

- <http://fr.wikipedia.org/wiki/Scrum>
- Document PPT SCRUM, cours 616 de Müller Henning

Ruby On Rails

- <http://strd6.com/2009/04/cant-dup-nilclass-maybe-try-unloadable/>
- <http://duanesbrain.blogspot.com/2006/05/ruby-on-rails-hasone-versus-belongsto.html>
- <http://aysoon.fr/66-ruby-on-rails-dossier-complet-pour-tout-comprendre-sur-cette-techno-pahre-du-web20>

Ruby

- <http://fr.wikipedia.org/wiki/Ruby>
- <http://www.rubyfrance.org/documentations/rubygem---introduction/>

Asynchronous JavaScript and XML

- http://fr.wikipedia.org/wiki/Asynchronous_JavaScript_and_XML

RJS (Remote Javascript)

- <http://pbnaigeon.developpez.com/tutoriel/rails/ajax-facile-RJS/#LIII>

Redmine

- <http://www.redmine.org>
- <http://www.projet-plume.org/fr/fiche/redmine>
- <http://fr.wikipedia.org/wiki/RedMine>
- <http://www.redmine.org/wiki/redmine/RedmineInstall>

Icscrum

- <http://www.icescrum.org/>

Intra-Know

- <http://www.intraknow.com/>

ExplainPMT

- <http://github.com/explainpmt/explainpmt>

VersionOne

- <http://www.versionone.com>

Trac

- <http://trac.edgewall.org/>

Retrospectiva :

- <http://retrospectiva.org/overview>

Date Picker

- <http://www.kelvinluck.com/assets/jquery/datePicker/v2/demo/index.html>

GCal4Ruby

- <http://cookingandcoding.com/gcal4ruby/>

Google Agenda

- <https://www.google.com/calendar>

Tutorial setup redmine send email using gmail

- <http://redmineblog.com/articles/setup-redmine-to-send-email-using-gmail/>

Redmine email STARTTLS

- <http://www.errorhelp.com/index.php/search/details/78026/redmine-rails-must-issue-a-starttls-command-first>

Automatic spent time logging from commit messages

- <http://www.redmine.org/issues/4155>

Rubular

- <http://rubular.com/>

TipTip jQuery

- <http://code.drewwilson.com/entry/tiptip-jquery-plugin>