

PPH

Pierre Laurac

9 avril 2012

Table des matières

1	Introduction	3
2	Le commencement	3
2.1	Description du fil rouge	3
2.2	Klaim	3
2.2.1	Le contexte	3
2.2.2	Description du projet	4
3	Un projet complet assurant un apprentissage continu	5
3.1	Les problèmes liés au développement	5
3.1.1	La problématique du développement multiplateforme	7
3.1.2	Un cahier des charges évoluant constamment	8
3.2	Les problèmes liés au management de l'équipe	8
3.2.1	Une avancé difficile	8
3.2.2	Peu de connaissance en la matière	8
4	La gestion du temps et du projet	8
4.1	Le temps, notre pire ennemi	8
4.1.1	Un projet long a mener de front avec d'autres	8
4.1.2	L'écart des concurrents	8
4.2	Continuer le projet après la fin	8
4.2.1	La démotivation au sein de l'équipe	9
4.2.2	Près d'un an sans release	9
4.3	Les outils mis en place	9
4.3.1	Les outils de code	10
4.3.2	Les outils de management	10
5	Vers l'infini et au delà	10
5.1	L'équipe actuelle	11
5.2	L'avancement du projet	11
5.3	Le devenir du projet	11
6	Conclusion	11
7	Annexes	12

1 Introduction

Le Projet Personnel en Humanités (PPH) est un projet que tout futur ingénieur INSA doit réaliser lors de son cursus du second cycle. C'est une réflexion personnelle que nous pouvons réaliser sur presque tous les sujets et ne doit pas forcément être en phase avec notre parcours ou le département auquel nous appartenons.

Ce projet est encadré par un enseignant tuteur, et le sujet que nous choisissons doit être validé par une commission. La réalisation de ce projet doit être soutenue devant un jury d'au moins personnes dont le tuteur.

L'intérêt de ce projet ici est double : il permet de mener une réflexion sur un sujet qui nous tient à coeur, avec lequel nous avons une affinité particulière, mais également de faire un retour sur un projet, une problématique plus grande, qui a été la gestion d'un projet de groupe, la réalisation d'un travail difficile, sans méthodes particulières.

2 Le commencement

2.1 Description du fil rouge

Tout a commencé en début de troisième année au département informatique de l'INSA de Lyon lors d'une présentation en amphi d'un projet appelé "fil rouge", optionnel, regroupant de 5 à 8 étudiants, de départements différents ou non, sur un thème non imposé. De plus, ce projet étant facultatif, il ne donne lieu à aucune notation mais permet cependant d'éviter de réaliser un autre TP imposé par le corps enseignant.

Ce concept m'a immédiatement séduit, car rare sont les opportunités en école où nous pouvons nous lancer dans un projet ou nous définissons notre cahier des charges. Je me suis donc mis à la recherche d'une idée originale, où je pouvais écrire une application iPhone, étant très intéressé par cela, mais ne trouvant jamais le temps pour le faire.

L'INSA ne possédant pas d'application iPhone réellement intéressante, je me suis lancé sur cette voie, puis je me suis renseigné s'il était possible de créer une application permettant d'accéder aux notes, aux cours, à l'emploi du temps, etc. Cependant, cette gestion est actuellement réalisée par département et n'est pas toujours disponible en ligne. La DSI¹ du département m'a donc encouragé à chercher une autre idée, moins complexe à mettre en place et à maintenir, ne serait-ce que de leur côté.

Je n'arrivais pas à trouver d'autres idées qui en vaille la peine, et c'est finalement le projet qui est venu à moi, puisqu'un groupe d'amis est venu me parler, me demandant de rejoindre leur projet, où il leur manquait justement des développeurs iPhone. J'ai été séduit par l'idée, dont la description se trouve ci-dessous.

2.2 Klaim

2.2.1 Le contexte

Avec le développement des 'Smartphones', ces téléphones possédant un système d'exploitation complexe permettant de réaliser de plus en plus de tâches, les développeurs de tous horizons se sont mis à développer des applications plus ou moins utiles. Suite au lancement de l'iPhone d'Apple, ce fut le tour à Google de réaliser Android et ainsi de suite.

1. Direction des Services Informatiques

Ces sociétés ont donc mis à disposition des professionnels des outils permettant de réaliser des applications de plus en plus complexes afin de réaliser les programmes qu'ils désiraient.

Nous connaissons tous les prix démesurés que les fournisseurs d'accès aux téléphones proposent à leur client malgré l'arrivée récente des forfaits dits « illimités ». Dans un contexte où la technologie mobile se développe de plus en plus, et ce pour tous les âges, il est aberrant pour nous de devoir payer les SMS². Si les forfaits actuels démocratisent les SMS illimités, leur envoi à des destinataires venant d'autres pays est toujours surtaxé.

BlackBerry tente de répondre à cette problématique grâce à son très connu BBM³ mais connaît de très nombreux problèmes (serveurs crashant constamment par exemple). De plus, l'autre très grande problématique d'aujourd'hui à mes yeux est le fait que ces services ne fonctionnent qu'avec une seule plateforme. Ainsi, un utilisateur de BBM ne peut discuter qu'avec des utilisateurs de BlackBerry possédant un compte BBM, ce qui réduit considérablement le nombre d'utilisateurs potentiels.

Apple décida de réaliser le même genre de service, appelé iMessage, mais se confronte aux même problème : l'unicité de la plateforme supportée. Apple a l'avantage de se trouver sur la vague du succès en ce moment et vend de nombreux iPhone, iPad ainsi que des ordinateurs possédant iMessage. Le lancement a été réalisé avec iCloud, mais la migration des utilisateurs sur ce système provoque encore de nombreux problèmes et l'utilisation d'iMessage reste parfois complexe.

C'est afin de répondre aux deux problématiques citées ci-dessus que notre projet trouve tout son sens.

2.2.2 Description du projet

Notre idée est donc simple et permet de répondre aux problèmes cités ci-dessus. Développer des applications mobiles et Web permettant l'envoi de messages en utilisant internet. Ce système, multiplateforme, permet d'éviter les coûts d'envois relatifs aux opérateurs téléphoniques. En effet, si ces derniers font souvent payer les SMS, les données cellulaires 3G sont comprises dans les forfaits pour les Smartphones.

Nous avions à l'époque quelques concurrents au service, mais notre projet s'étendait au delà de ce qu'ils proposaient. Si leur système propose l'envoi de messages comme le notre, ces derniers ne sont pas sauvegardés sur leurs serveurs. Cela leur pose alors deux problèmes principaux :

- Ils ne peuvent proposer un service Web permettant la lecture des messages envoyés et reçus, ainsi que l'envoi de nouveaux messages. C'est un atout très important à nos yeux, et nous essayons de nous démarquer entre autre grâce à ce point.
- Les messages sont stockés sur le téléphone des personnes utilisant le service uniquement. Si une personne change de téléphone, ils risquent alors de perdre les conversations existantes. Avec Klaim, ceci ne peut exister puisque si vous changez de téléphone, vous pouvez retrouver tous les messages que vous aviez précédemment, étant stockés sur nos serveurs.

Le point central de notre service est notre serveur, gardant tous les messages (sauf s'ils sont supprimés par l'utilisateur bien entendu) permettant donc d'apporter une solution viable aux deux points expliqués ci-dessus. Lorsqu'un utilisateur souhaite envoyer un message, il utilise l'une de nos applications, communiquant avec le serveur qui va alors enregistrer le message dans ses bases de données, puis le transférer à l'utilisateur voulu.

Le sujet était donc lancé : notre service devait permettre d'envoyer des messages depuis les plateformes suivantes :

- Apple (iPhone, iPod, iPad)

2. SMS, abréviation de 'Short Message System', système permettant d'envoyer un message court sur le téléphone d'un destinataire.

3. BlackBerry Messenger

- Android
- Windows Phone Mobile
- BlackBerry
- Site web

Ces plateformes se synchronisent avec le serveur principal permettant de récupérer et où mettre à jour les informations modifiées d'un compte depuis telle ou telle plateforme. Mais si ces services que les concurrents ne possèdent pas permettent de créer un avantage comparatif, ils nous ont en réalité énormément retardé sur la date de sortie, créant des problèmes que nous n'envisagions pas au premier abord. En effet, grâce à notre système, un compte utilisateur peut-être accédé par plusieurs plateformes. Si un utilisateur possède un iPhone et un iPad par exemple, il doit pouvoir utiliser les deux à sa convenance. Se pose alors le problème de la synchronisation ! En effet, si je n'utilise pas mon iPad pendant quelques jours, mais activement l'application sur mon téléphone, au moment du rallumage de l'iPad, il sera momentanément désynchronisé. Toute la difficulté sera donc de rapatrier uniquement les bonnes informations provenant du serveur et de mettre à jour l'iPad.

Afin d'effectuer tout ce travail, nous étions initialement huit personnes, dont les noms et la description de leur tâche se trouve ci-dessous :

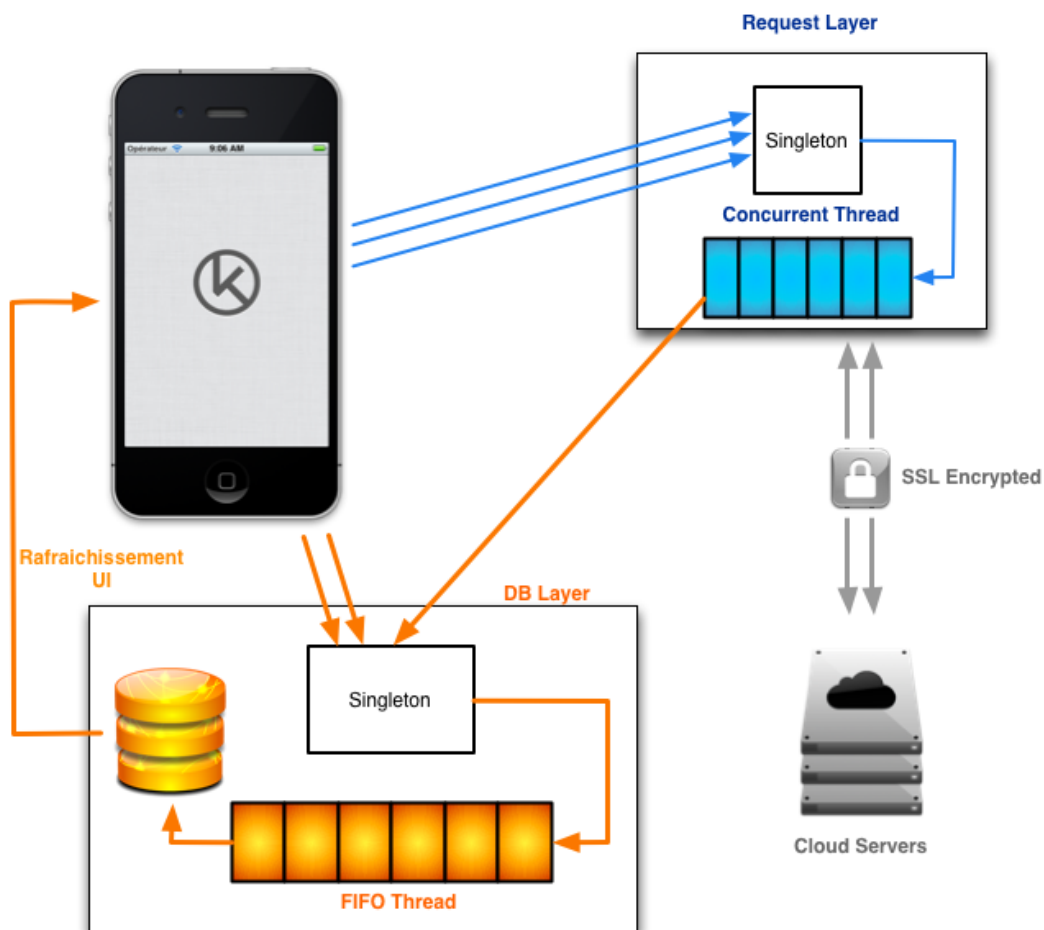
- **Quentin Calvez** : Initiateur du projet, c'est la personne possédant le plus de connaissances techniques. C'est également le concepteur du serveur et du projet sans qui rien n'aurait commencé. Il été également en charge de l'application Windows Phone 7 étant amoureux des technologies microsoft.
- **Marc Viricel** : Co-initiateur, il s'occupait entre autre du développement sous Blackberry.
- **Benjamin Bouvier** : Co-développeur Blackberry, en binôme avec Marc.
- **Alpha Barry** : Développeur Android - il était seul ayant déjà de bonnes connaissances dans la plateforme.
- **Romain Arnaud** : Co-développeur iPhone.
- **Benjamin Planche** : Développeur site web.
- **Pierre Laurac** : Co-développeur iPhone.

3 Un projet complet assurant un apprentissage continu

Le sujet est simple, mais nous avons cependant soulevé un certain nombre de problèmes en très peu de temps. En plus de ceux énoncés dans la section précédente, intervient également les langages de programmation. En effet, chaque plateforme possède son propre langage, et il a donc fallu que chaque équipe de développeur du projet se spécialise dans un langage particulier.

3.1 Les problèmes liés au développement

Mis à part une ou deux personnes dans le groupe de travail, nous nous étions tous assignés dans un langage que nous souhaitions apprendre car outre le service que nous voulions commercialiser, notre but premier était ici l'apprentissage, et c'était celui de l'iPhone qui m'intéressait plus particulièrement.



Voici un plan général très synthétique du fonctionnement de notre application. Nous avons essayé de trouver une structure modulaire, utilisant une structure en couche nous permettant de pouvoir s'adapter à l'ensemble des plateformes le plus simplement possible. Dans ce cas très précis, il me serait très simple de supprimer la partie iPhone, mais garder les moteurs pour les requêtes et la base de données afin de créer un client lourd pour OS X ⁴.

Chaque plateforme possède donc un langage spécifique (l'Objective-C et Cocoa pour Apple) et si nous connaissons les grands aspects et concepts de la programmation, il m'a fallu apprendre les spécificités de ce langage. Nous pouvons dégager trois grands aspects qu'il m'a fallu maîtriser dans le cadre de ce projet :

- Les requêtes au serveur (Request Layer sur le schéma)
- La base de données (DB Layer sur le Schéma)
- L'interface Graphique

Nous passerons un peu de temps à expliquer la complexité de chacune des parties citées ci-dessus.

Les requêtes au serveur Cette partie a certainement été la moins complexe dans sa réalisation mais pour autant celle qui a reçue le plus de modifications. Le premier système de requêtes en place était fonctionnel, mais le moindre changement à effectuer était un véritable cauchemar. Après de nombreuses manipulations, j'ai donc décidé de le changer entièrement dans sa structure. Nous pouvons donc retrouver actuellement une majeure partie du code écrit en premier lieu, mais son utilisation en est grandement simplifié aujourd'hui.

4. OS X est le nom du système d'exploitation d'Apple, anciennement Mac OS puis Mac OS X

L'un des plus gros changements est survenu lorsque j'ai du programmer la reprise de requêtes. Dans un environnement aussi mobile, nous travaillons énormément avec le serveur. Cependant, que se passe-t-il quand aucun réseau n'est disponible ? Certaines requêtes doivent-elles être sauvegardées, d'autres mises de côté pendant un certain temps. Pouvons-nous les enregistrer en base de données ? Quelle est la durée de vie de ces requêtes, et comment le savoir ? Nous sommes parvenus à un système pour l'ensemble de l'application ou nous traitons avec des objets génériques possédant pouvant être sauvegardé en BD. Selon le type de requête, sa durée de vie est changée accordement (au moment où l'application est tuée, ou quand l'utilisateur se délogue, etc.)

C'est dans ce cas très précis que nous pouvons voir l'importance des projets d'urbanisation que nous avons lors de notre quatrième année. Au début du projet, puisque nous n'avions pas fait de conception générale ni détaillée, nous nous sommes lancé dans un code peu propre, non maintenable, et peu enclin aux changements. Je pense qu'il est cependant impossible de se rendre compte de l'importance d'une bonne conception sans s'être une fois confronté au problème.

La base de données Les bases de données sont un moyen efficace et simple à mettre en oeuvre afin de sauvegarder des informations de manière statique. Une première approche aurait donc été de reprendre une base dite classique avec l'approche que nous avons vu en cours. Ce n'est cependant pas le cas ici.

Toujours dans une volonté d'apprendre ce langage, j'ai décidé d'utiliser Coredata, qui est l'outil de prédilection d'Apple pour les bases de données. Coredata est une base de données déguisée, en ce sens où Apple met à disposition des développeurs des moyens "simplifiés" mais utilise en réalité une base de données. Nous utilisons donc des appels de fonctions mis à notre disposition par Apple, qui offre une surcouche à une base de données. Cette simplification a été dans un premier temps un vrai casse-tête, tellement la logique est différente de celle d'une base normale. Cependant, aujourd'hui je ne regrette en rien ce choix de développement.

L'interface Graphique L'interface graphique a de tout temps été ce que je préférais de moins mais cet élément est cependant indispensable puisque sans elle, nous n'aurions aucun utilisateur. Il est tout à fait envisageable de réaliser une interface non complexe, et c'est ce qui a été fait dans un premier temps. Cependant, nous souhaitons, dans un souci de simplification d'utilisation pour l'utilisateur final, de reprendre entièrement l'interface de l'application des SMS de la plateforme sur laquelle nous nous trouvons. Ceci représente un challenge de taille puisqu'Apple est réellement maniaque à ce niveau là.

Nous avons cependant aujourd'hui une interface réellement identique, et quand j'utilise l'un des deux services (SMS ou Klaim) il m'est difficile de savoir quelle application je suis en train d'utiliser.

3.1.1 La problématique du développement multiplateforme

Ceci est l'un des points les plus importants du document, puisqu'il parle d'une problématique d'envergure dans le monde de l'informatique d'aujourd'hui. De plus en plus de personnes ont des smartphones. Les développeurs souhaitent donc développer des applications pour chacune des plateformes, mais cela demande énormément de temps et de connaissance. Si j'ai de bonnes connaissances aujourd'hui pour iOS, je serais incapable de développer pour Android sans me plonger dans la documentation et de nombreux tutoriels.

C'est pour cette raison que nous étions autant de monde sur le projet. Il y avait initialement deux personnes par plateforme afin de combler le manque de connaissances que nous avions initialement. C'est aujourd'hui l'une des plus grandes problématiques du moment avec la poussée de téléphones mobiles au sein de la société. Devant ce problème, des solutions ont été développées mais posent encore certains problèmes :

- Réaliser un site web mobile permet de réduire considérablement le développement puisque les technologies à utiliser sont connues et nous n'avons qu'un seul code à écrire pour l'ensemble des plateformes. Cependant, nous ne pouvons rien sauvegarder sur la mémoire interne du téléphone ce qui peut être réellement problématique et critique selon le type d'applications que nous souhaitons développer, et nous n'avons surtout pas accès aux notifications, ces messages permettant d'indiquer à l'utilisateur un événement particulier.
- Il existe aujourd'hui des Framework permettant de réduire la charge de travail mais le résultat obtenu n'est pas aussi satisfaisant que celui d'une application dite native.

3.1.2 Un cahier des charges évoluant constamment

Un autre soucis important auquel nous avons été confronté initialement était le manque d'information sur ce que nous devons faire. Nous connaissions les idées directrices mais c'était tout. Nous apprenons lors de nos projets en école qu'il est important de bien étudier l'ensemble du projet afin de réaliser une conception répondant à toutes les exigences fonctionnelles comme non fonctionnelles. Une simple modification peut avoir d'importantes répercussions.

C'est exactement le problème que nous avons eu pour diverses raisons. Premièrement parce que nous avons eu de nombreux problèmes auxquels il a fallu trouver une solution. Ces dernières apportaient parfois les modifications dont nous parlions précédemment. Deuxièmement car nous n'avons pas réellement effectué de conception générale pour l'ensemble du projet. Nous demandions uniquement au chef de projet quelle était l'étape suivante une fois que nous avons terminé une partie. Cette façon de faire a posé de nombreux soucis et nous avons du constamment apporté des modifications à ce qui avait déjà été fait.

Nous n'imaginions tout simplement pas l'ampleur du projet et c'est un peu pour cette raison que cela s'est passé ainsi.

3.2 Les problèmes liés au management de l'équipe

3.2.1 Une avancé difficile

3.2.2 Peu de connaissance en la matière

4 La gestion du temps et du projet

Nous maintenant passé à une analyse que nous pouvons maintenant réalisé puisque nous sommes suffisamment avancé dans notre quatrième année, qui est rappelons le centré autour de la gestion de projet et d'équipe.

4.1 Le temps, notre pire ennemi

4.1.1 Un projet long a mener de front avec d'autres

Une autre problématique importante de notre projet était que celui était extra-scolaire bien que dans le cadre du fil rouge. La troisième année reste une année chargée avec les projets et les examens de fin d'année et devant un projet de cette taille le temps que nous avons afin de programmer est très important. J'ai passé la majeure partie de mes week-end à avancer sur le projet. J'aurai pu être bien plus efficace si j'avais connu le développement iOS avant mais ce n'était pas le cas.

4.1.2 L'écart des concurrents

Un aspect que nous n'avons pas encore réellement étudié dans ce document est les concurrents. Ces derniers existent déjà sur ce marché et proposent des produits fonctionnement très bien mais comme nous

l'avons déjà énoncé, notre application possèdent quelques points pouvant réellement faire la différence.

Commencer notre projet avec un tel retard est cependant dommageable puisqu'en attendant, leur nombre de client augmente considérablement. Obtenir de nouveaux clients est bien plus facile que de les récupérer lorsqu'ils utilisent déjà un service plus ou moins identique.

4.2 Continuer le projet après la fin

Ce projet s'est commencé dans le cadre du fil rouge et après une présentation plus ou moins réussie, nous avons enfin terminé! La pression allait enfin pouvoir retomber un peu après tant de travail, pour un résultat qui était prometteur mais encore peu concluant. Se posait alors la question "Qu'allons nous faire maintenant?", qui était évidente pour certains, moins pour d'autres.

4.2.1 La démotivation au sein de l'équipe

L'avantage du projet scolaire facultatif nous libérant de la place sur notre plage horaire a été perdu à la fin des cours. Ce projet est devenu plus un fardeau pour certaines personnes du groupe puisque pour continuer, il fallait se remettre à travailler alors que les beaux jours arrivaient, et que la charge de travail pour les cours diminuait.

Nous nous sommes donc réunis afin de demander qui était partant pour continuer, et toute l'équipe avait l'air favorable à cette idée. Cependant, nous avons rapidement perdu 4 développeurs officiellement, et le cinquième sans que nous le sachions vraiment. Pour les quatre premiers, je pense qu'ils n'en pouvait plus. Personne ne s'attendait réellement à un tel projet demandant un travail aussi intense sur une telle durée. Les différents problèmes auxquels ils ont été confrontés ont eu raison d'eux probablement. De plus, pour continuer, il était nécessaire de s'investir d'autant plus puisque le nombre de personne diminuait, et que le stage allait commencer, ce qui réduisait nos sessions de travail au soir et au week-end. La cinquième personne quant à elle devait continuer pendant l'été mais n'a pas réellement travaillé, sans pour autant nous le dire. Nous pouvons comprendre la démotivation ou l'envie de partir, mais pas de rester et mentir en nous disant que l'on continue.

Bref, mi-août, nous n'étions plus que deux. Quentin et moi. Comment continuer en partant d'ici? J'avais toujours énormément de problèmes avec iOS, que je commençais cependant à mieux maîtriser. Quentin avait désormais le serveur, la version WP7 et les site web à gérer, sans parler de la plateforme Blackberry abandonnée.

4.2.2 Près d'un an sans release

L'un des principes que l'on tente de nous apprendre dans notre formation, est celui des jalons. Il est important de définir des objectifs pour un version d'un produit, et d'avoir une date butoir et de s'y tenir. Notre problème est que nous n'avons jamais eu de jalons bien défini, et nous avançons un peu comme nous pouvions, au fil de l'eau avec le temps qui nous restait en dehors des cours et des projets.

Le site a été lancé le et l'application WP7 le, ce qui définissait un premier jalon important à nos yeux, puisqu'il représentait bientôt un an de travail. Bien que Quentin était responsable de ces deux plateformes, cela m'a redonné un coup de fouet pour m'y remettre dur. Nous venions de trouver deux personnes supplémentaires pour reprendre le projet Android, ce qui nous déchargeait d'un poids important!

Je n'avais maintenant plus qu'à terminer l'application pour iOS. Elle était un peu plus stable, mais les graphismes n'étaient pas encore terminés, et de nombreux crash surgissaient encore, sans compter les manquements aux spécifications. Il y avait encore énormément de travail, et la quatrième année au département nous laissait vraiment trop peu de temps, d'autant plus que je m'occupais des relations entreprises pour

l'AEDI⁵.

Au moment de l'écriture de ces lignes, l'application pour iOS est en cours de validation chez Apple et nous attendons leurs retours. Suite à la publication de cette version, nous allons pouvoir commencer à développer la version 2, puisqu'il y a énormément d'améliorations que nous voulons apporter afin de réduire l'écart avec nos concurrents.

4.3 Les outils mis en place

En un an, nous avons utilisés différents outils, aussi bien pour le management des équipes, que pour la gestion des lignes de code, ce qui est absolument essentiel chez nous. Je vais dans cette section en parler, puisque c'est ce que nous avons appris cette année à réaliser. Nous utilisons donc déjà les outils, sans avoir pour autant connaissance de leur potentiel.

4.3.1 Les outils de code

Rapidement, nous nous sommes rendu compte qu'il serait important d'utiliser des outils afin de nous aider à nous guider ou travailler en groupe. Voici une liste des outils que nous avons utilisés (et que nous utilisons encore pour certains) :

- SVN : Subversion est un outil permettant de travailler collaborativement sur un projet. Il permet en plus de versionner le projet c'est-à-dire journaliser l'ensemble du code écrit, ce qui est d'autant plus important lorsque nous travaillons à plusieurs. Nous n'utilisons plus ce service à ce jour.
- GIT : C'est le remplaçant pour nous de SVN et propose certains avantages par rapport à SVN.
- Github : Héberger un serveur permettant d'utiliser GIT peut-être contraignant. C'est pourquoi nous utilisons Github nous mettant à disposition des serveurs pour cela.
- - Serveurs

4.3.2 Les outils de management

Les outils cités ci-dessus permettent uniquement d'améliorer la vie des développeurs. Afin de gérer un projet informatique de cette envergure, il faut cependant bien plus. C'est pourquoi nous utilisons les outils suivants :

- Redmine : est un outils de gestion de projet en ligne que nous utilisons depuis le début du projet. Il permet de déclarer des projets et sous-projets, mais également de découper ces derniers en tâches et sous-tâches afin de les assigner à des ressources. Il est en quelque sorte une version webisée de Microsoft Project, mais avec des fonctionnalités cependant moins avancées. Cela nous permet cependant de suivre l'avancement des projets de façon claire et précise.
- Klaim : maintenant que notre projet est assez avancé, nous l'utilisons énormément entre nous afin de nous donner des informations que nous voulons faire passer (i.e. petits bugs détectés ne demandant par forcément l'ouverture d'un ticket sous redmine).
- Jabbr : c'est un outil en ligne nous permettant de communiquer avec l'ensemble de l'équipe. Chaque personne peut voir ce que nous avons écrit et nous pouvons également consulter l'historique. Cela permet d'apporter de la connaissance générale sur l'ensemble des sous-projets.

5 Vers l'infini et au delà

Après plus d'un an d'existence, ce projet à énormément évolué et les équipes également. Après la fin officielle du fil rouge, toutes les personnes étaient encore motivées pour continuer. L'équipe BlackBerry a cependant cédé en premier annonçant qu'ils se confrontaient à trop de problèmes techniques, et que le projet n'était pas assez avancé malgré les efforts qu'ils avaient déjà déployés. Marc et Benjamin ont donc décidé

5. Association Etudiante du Département Informatique

d'arrêter. Ce fut le premier coup dur pour nous. L'autre Benjamin a également arrêté rapidement, n'arrivant pas réellement à travailler avec Quentin sur le site web. Nous n'étions donc plus que trois.

Travailler à New York pendant mon stage, et essayer de rentrer le soir pour continuer le développement m'a demandé énormément de concentration et de volonté. J'ai cependant tenu le coup et continué d'avancer malgré moi. Je croyais au projet, et me disais que je ne pouvais pas réellement arrêter. En effet, j'étais le seul développeur iOS du moment, et bien que Quentin effectuait son stage sur cette plateforme, il n'avais pas la capacité de supporter ce projet en plus.

Alpha quant à lui nous avait indiqué qu'il continuait de développer sur le projet. En réalité, ce n'était plus vraiment le cas et nous avons pris énormément de retard sur cette plateforme.

A partir du mois d'août, nous n'étions plus que deux, et une motivation tombée bien bas, malgré un concept et des projets commençant à fonctionner correctement. Nous avons cependant réussi à trouver des personnes motivées début septembre pour reprendre le développement Android.

5.1 L'équipe actuelle

Avec le temps, je me suis aperçu que Quentin était un excellent développeur, cependant l'aspect de gestion de projet et de gestion des équipes ne l'intéressait absolument pas. N'étant plus que deux pendant l'été, cela ne posait pas réellement de problèmes. Cependant, avec l'arrivée de nouvelles personnes dans l'équipe, je me suis proposé de m'occuper de ce poste, ce pour quoi Quentin avait l'air favorable.

5.2 L'avancement du projet

5.3 Le devenir du projet

6 Conclusion

Ce PPH a été écrit dans plusieurs buts :

- Montrer dans un premier lieu notre inexpérience dans un projet d'aussi grande envergure, avec autant d'éléments à gérer, que ce soit au niveau de la gestion des ressources humaines, mais également de la conduite d'un projet.
- Porter un regard sur notre formation de quatrième année, fortement axée sur la gestion de projet.

Ce premier point a été largement ressenti tout au long de l'année dernière, d'autant plus que Quentin n'était pas réellement intéressé par cette gestion - c'est en tout cas ainsi que je le percevais. L'ensemble des modifications qui ont été effectuées au cahier des charges a été plus que préjudiciable sur la date de rendu de projet. Nous ne pouvions malheureusement pas imaginer de façon claire et précise ce dans quoi nous nous jetions. C'est pour moi à ce jour le projet qui m'a fait acquérir le plus de compétences aussi bien dans le domaine technique, que dans le domaine des relations humaines, puisqu'il a fallu apprendre à communiquer au sein de notre groupe, bien que nous étions des amis.

La quatrième année au sein du département informatique est réputée pour être exigeante, aussi bien pour l'ensemble de projet qu'il nous est demandé de rendre, et incidemment pour la charge de travail induise. Elle nous confronte étrangement à l'ensemble des problèmes que nous avons eu avec Klaim puisqu'il fallait gérer une équipe de six personnes (un hexanome) pour chaque projet. Il y avait cependant deux différences notables. Premièrement, chaque personne travaillait sur le même projet, et il était donc plus facile de contrôler l'avancement de chaque personne. Deuxièmement, ces projets étaient dans le cadre des études, et nous étions donc "obligés" de les réaliser.

Peut-être que mon approche des projets en groupe été déjà différente en début de cette année, mais je sais qu'elle a énormément évolué en un an. C'est aussi pourquoi je m'occupe de la coordination des différentes

équipes de Klaim à ce jour, puisqu'il est important de le faire, afin de ne pas s'éparpiller comme nous l'avons fait auparavant.

Je pense donc qu'il aura été plus qu'intéressant de voir comment nous nous en serions sorti avec nos connaissances actuelles. Ce projet reste bien entendu extra-scolaire d'où la difficulté de l'avancer aussi vite que nous le souhaitons, mais c'est avec plaisir que nous continuons le développement. Nous sommes actuellement à la recherche de fonds ainsi que de ressources humaines. Nous avons les idées ainsi que le potentiel requis, et nous nous battons pour avancer.

7 Annexes

- capture version iPhone
- capture version web
- capture version WP7