

PPH

Pierre Laurac

16 mars 2012

Table des matières

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

1 Introduction

Le Projet Personnel en Humanités (PPH) est un projet que tout 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 de deux 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. Nous avons eu 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 où 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.

C'est en réalité le projet qui est venu à moi, puisqu'un groupe d'amis est venu me parler, me demandant de rejoindre leur projet, et il leur manquait 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 tout 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.

Ces sociétés ont donc mis à disposition des professionnels des outils permettant de réaliser des applications de plus en plus complexes afin de réaliser tel ou tel programme.

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 SMS¹. 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 BlackBerry Messenger (BBM) mais connaît de très nombreux problèmes (serveurs crashant constamment par exemple). De plus, l'autre très grande problématique d'aujourd'hui est le fait que ces services ne fonctionnent qu'avec une seule plateforme. Ainsi, un BBM ne peut-être utilisé qu'avec des utilisateurs de BlackBerry possédant un compte BBM.

Apple décida de réaliser le même genre de service, appelé iMessage, mais se confronte aux même problème : l'unicité de la plateforme supportée. C'est afin de répondre aux deux problématiques citées ci-dessus que notre projet débute.

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

2.2.2 Description du projet

Notre projet est donc simple et permet de répondre aux problèmes cités ci-dessus. Développer des applications mobiles et Web permettant l'envoi de messages utilisant la 3G. 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 Smartphone.

Nous avions à l'époque quelques concurrents au service fonctionnant correctement, 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.
- 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.

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.

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

- Apple (iPhone, iPod, iPad)
- 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 l'écart, 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.

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 domaine 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. Je continuerai d'ailleurs par développer uniquement cet aspect dans cette partie.

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
- La base de données

- 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 complexe dans sa réalisation mais pour autant celle qui a reçue le plus de modifications. Le premier système de requêtage 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 simplifiée aujourd’hui.

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 puisque 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

4.1 Le temps, notre pire ennemi

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

4.1.2 L'écart des concurrents

4.2 Continuer le projet après la fin

4.2.1 La démotivation au sein de l'équipe

4.2.2 Près d'un an sans release

4.3 Les outils mis en place

4.3.1 Les outils de gestion

4.3.2 Les outils de management

5 Conclusion