

# Rapport de stage

Ophir LOJKINE

Été 2014

## Contexte

### L'entreprise

AF83 est une agence web française, qui a des bureaux à Paris et San Francisco. J'ai travaillé dans les locaux parisiens, situés rue Poissonnière. On y travaille sur de nombreux projets, utilisant les technologies du web pour créer des applications pour des clients tels que Canal+, France Télévision, ou EDF.

### Le projet

L'entreprise mène de nombreux projets. Je n'ai travaillé que sur un seul: MAP. C'est le nom donné en interne au projet monalbumphoto.fr. Monalbumphoto est une société d'impression d'albums photos destinée aux particuliers. Les albums photos sont entièrement personnalisés par le client, originellement à l'aide d'une application à installer avant de pouvoir commencer la création.

L'application native, disponible uniquement pour Microsoft Windows, privait la société de clients potentiels, et augmentait le temps entre la découverte du site et le début de la création d'un album.

Monalbumphoto.fr a donc fait appel à AF83 afin de créer une application de création d'albums photos qui tourne directement dans le navigateur web du client, sur le même site qui présente les produits.

[Cette application](#) est déjà en place, et est proposée par défaut aux consommateurs sous Mac OS ou Linux sur certains produits. À l'heure de l'écriture de ce rapport, l'application native est encore le seul choix disponible pour les utilisateurs de Windows, mais cela est amené à changer.

## Déroulement du stage

### Découverte de l'entreprise

La première chose qui frappe lorsque l'on rentre chez AF83, c'est l'atmosphère décontractée qui y règne, à des lieues de ce l'on nous a décrit bien souvent comme le monde du travail.

Il n'y a pas dans cette entreprise

### Ce que j'ai fait

J'ai eu la chance de ne jamais me faire confier de tâches ingrates, et ai pu travailler sur des sujets qui m'intéressaient vraiment.

En conséquence, j'ai travaillé sur plusieurs domaines différents, j'ai pu me pencher sur l'architecture d'un projet complexe, mais ai aussi effectué du débogage. Je ne présenterai donc ici qu'une partie des tâches que j'ai effectuées, les plus intéressantes et représentatives de mon travail, je l'espère.

**Les calendriers** L'une des tâches sur lesquelles j'ai le plus travaillé est la mise en place des calendriers. MAP propose des calendriers personnalisables à ses clients, sur lesquels ils peuvent intégrer leurs propres photos.

Ce type de produit constituait un petit défi technique, car contrairement aux autres, il contient un "fond", le calendrier lui-même, qui doit être dessiné indépendamment des images ajoutées par l'utilisateur.

Il s'agissait tout d'abord de choisir un format de description des grilles de calendrier, ce qui est en soi assez délicat. Si l'on choisit un format trop générique, alors il y a potentiellement beaucoup de code à écrire pour gérer toutes les possibilités offertes par le format. Si l'on choisit un format trop restreint, alors on risque de se retrouver bloqué dans le futur, lorsque MAP demandera une nouvelle grille plus originale.

Le format SVG s'est finalement imposé, car bien que très générique, il est géré nativement dans tous les navigateurs modernes, ce qui permet d'économiser du travail de développement.

Comme j'avais travaillé avec ce format auparavant pour d'autres projets, j'ai été très heureux de pouvoir être utile à l'équipe, en participant à l'écriture de la première grille en SVG, et en écrivant du code pour sa manipulation.

**Promesses** Les [promesses](#) constituent une technique de programmation asynchrone intéressante que j'ai apprise et utilisée pendant mon stage.

Travailler sur une application qui tourne dans le navigateur nous met face à des problèmes que l'on ne rencontre pas sur des applications de bureau classiques,

et que l'on peut parfois résoudre de manière élégante avec certains schémas de programmation.

Notamment, lorsque l'utilisateur charge une page, on ne peut pas se permettre de lui faire attendre plusieurs minutes avant d'afficher l'application. On ne peut donc pas non plus se permettre de télécharger à l'avance, "au cas où", tous les fichiers dont l'application pourrait avoir besoin. Les ressources doivent être chargées dynamiquement, dès qu'elles sont nécessaires.

Il arrive donc que l'application se retrouve dans un état où elle a besoin d'une ressource qui n'a pas encore été chargée. Elle peut alors utiliser une *promesse*, un objet qui symbolise la ressource, qui peut être chargée ou non.

Alors, au lieu de

- regarder si la ressource est chargée,
- si elle l'est, faire une opération dessus,
- sinon, programmer une opération pour qu'elle soit effectuée lorsque la ressource sera chargée,

on peut simplement

- dire qu'il faut faire telle opération sur telle ressource.

Le [système de promesses](#) s'occupe alors de gérer le chargement de la ressource.

Cette technique de programmation devient particulièrement intéressante lorsque l'on manipule plusieurs ressources qui peuvent avoir différents état de chargement. On [agrége](#) alors leurs promesses, on manipule la promesse résultante.

Dans MAP, les promesses sont utilisées notamment pour le chargement des événements de calendrier, des polices de caractère et des pictogrammes associés aux grilles, qui doivent tous être chargés pour pouvoir dessiner les pages de calendrier.

## Organisation du travail, gestion de projet

Le projet est développé en utilisant la [méthode agile](#), censée accélérer le développement et permettre de suivre au plus près les besoins du client, même si ceux-ci changent.

Au lieu d'avoir, par exemple une [méthode dite "en V"](#), où l'on passe un certain temps à élaborer l'architecture du code, mais où l'on ne programme rien, on s'intéresse à l'implémentation concrète des concepts au fur et à mesure qu'on les élabore.

**Communication avec le client** Cela nécessite bien sûr une très bonne organisation, pour ne pas perdre du temps à écrire du code jetable, mais cela permet d’avancer rapidement sur un projet, et de s’adapter aux demandes du client au fur et à mesure qu’elles arrivent.

Le projet MAP est ainsi organisé en “sprints” de quelques semaines, séparés par des réunions avec le client. J’ai eu l’occasion d’assister à deux réunions, tout d’abord par téléphone parce que j’étais à Nantes, puis physiquement à Paris. Ces réunions, organisées dans un cadre assez peu formel, commencent par une démonstration des fonctionnalités qui ont été codées lors du sprint précédent, suivie d’une discussion avec le client et de l’établissement de la liste des points à traiter lors du sprint suivant.

Le projet est associé à un chef de projet, qui est très proche des développeurs. Loin d’être une autorité qui pèse sur les développeurs ou leur impose ses choix, il s’occupe principalement de la gestion de la communication avec le client, du déploiement et du test du code, et de la gestion des erreurs qui peuvent survenir sur les serveurs en production ou chez les clients.

**Communication interne** Tout est fait pour faciliter la communication entre développeurs, et avec le chef de projet. Les bureaux d’AF83 sont en fait un grand open-space, et les développeurs s’adressent les uns aux autres lorsqu’ils en ont besoin.

La communication interne est également facilitée par plusieurs outils:

- [Trello](#), qui est utilisé comme un gestionnaire de tâches. À chaque action à faire on associe une “carte trello”, à laquelle on peut associer une ou plusieurs personnes.
- [Github](#), et plus généralement [git](#) pour gérer les versions, développer de nouvelles fonctionnalités sans casser le code existant pour les autres, faire de la revue de code, et beaucoup plus. . .

## En dehors du travail

Ca stage a également été l’occasion de découvrir la vie sociale en entreprise.

Chez AF83, l’organisation est très “horizontale”. Il n’y a pas vraiment de pression managériale qui vienne du dessus, et cela facilite le rapprochement entre les employés, à 10 heures le matin avant de commencer le travail, lorsque l’on mange ensemble, autour d’une partie de babyfoot, au le soir avant de quitter les bureaux.

Cette absence de pression contribue à créer cette ambiance de travail que j’ai beaucoup appréciée, et qui permet de garder sa motivation à travailler.

## Déceptions

En dépit de tous les points qui font que ce stage a été une période agréable, il y a un point sur lequel j'ai été déçu.

**Rémunération** Ce point est malheureusement très courant lors de stages, quels qu'ils soient. J'ai bien conscience d'avoir été formé pendant ce stage, et d'avoir dans les premiers jours plus coûté que rapporté à l'entreprise.

Cependant, j'ai l'impression d'avoir finalement apporté une certaine plus-value au projet, qui je pense valait plus que les 600€ environ que j'ai reçus pour mon travail d'un mois et demi.

**Répétition des tâches** Lors de mon stage d'un mois et demi, je n'ai jamais eu l'occasion de m'ennuyer, car j'avais plein de choses à découvrir. Cependant, je pense que ce genre de projet peut devenir lassant, lorsque l'on connaît bien le code et l'on doit faire régulièrement des changements qui se ressemblent. De plus, lorsque la base de code grossit, on se retrouve avec du code que l'on n'ose plus faire évoluer radicalement car cela représente trop de travail. Dans ces conditions, j'ai peur que le travail devienne trop répétitif, et moins excitant qu'il ne l'était pour moi qui n'ai passé que peu de temps au contact du code.

## Ce que j'ai appris

J'ai appris à utiliser vim, à jouer au babyfoot, et à ouvrir des b\*\*\*\*\* avec un briquet. Mais je dois encore me perfectionner dans tous ces domaines.

Sinon, j'ai aussi appris à utiliser [backbone](#), je me suis perfectionné en [CoffeeScript](#), j'ai eu l'occasion de faire de la [programmation en binôme](#) pour la première fois, ce qui est très formateur, et de manière générale, j'ai appris à travailler en équipe.