



Gestion d'un projet Web e-commerce : Pièces Avenue.com

Hakan Ebabil

► To cite this version:

Hakan Ebabil. Gestion d'un projet Web e-commerce : Pièces Avenue.com. Web. 2012. dumas-01077422

HAL Id: dumas-01077422

<https://dumas.ccsd.cnrs.fr/dumas-01077422>

Submitted on 24 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE STRASBOURG

MEMOIRE

Présenté en vue d'obtenir

Le DIPLOME D'INGENIEUR CNAM

SPECIALITE : INFORMATIQUE

OPTION : Informatique Système d'information

Par

EBABIL Hakan

Gestion d'un projet Web e-commerce: Pièces Avenue.com

Soutenu le 12 juin 2012

JURY

PRESIDENT : Mme. Isabelle WATTIAU

MEMBRES : M. Yoann NUSSBAUMER

M. Cédric KLEINPETER

Remerciements

Je voudrais tout d'abord remercier une personne très spéciale à mes yeux et qui m'a beaucoup inspiré lors de mon passage pendant ces quatre années chez Skalpel, puisqu'il s'agit de **Yoann Nussbaumer**, mon dirigeant et patron au moment de la réalisation du projet Pièces Avenue. Yoann m'a toujours été d'un soutien sans faille. Il m'a toujours accordé sa confiance. Cette confiance, je l'ai encore plus ressentie pendant le projet Pièces Avenue. Je voudrais dire que, plus qu'un supérieur hiérarchique, Yoann a été pour moi un mentor. Il m'a appris tous ce que je sais aujourd'hui dans le domaine du Web et je souhaite lui exprimer toute ma gratitude dans ce mémoire.

Je voudrais aussi remercier l'ensemble de l'équipe Skalpel pour son travail extraordinaire et sans qui ce projet n'aurait très certainement pas pu voir le jour. Merci donc à **Maxime Castanié, Amaury Chedotal, Killian Ebel, Nicolas Girardin, William Guth, Mickael Knauer, Julien Lavault, Coralie Nussbaumer, Thanh Tri Phan, Louis Saunders**, et enfin **Hervé Weltzer**. Vous avez fait un travail phénoménal malgré les délais et les contraintes du projet. Bravo à vous tous !

Si l'équipe interne Skalpel a été exemplaire, je n'oublie pas non plus l'aide apportée par nos partenaires et par l'ensemble des prestataires qui nous ont accompagnés lors de la réalisation de ce site Internet. En particulier, je voudrais dire un grand merci à **Jimmy Raheriarisoa**, graphiste au sein de l'agence Cube X Agency ainsi qu'à notre contact chez Tecdoc Informations Services, **Karine Ray**.

Ce projet n'aurait évidemment pas pu voir le jour sans l'impulsion de nos clients, **Stéfan Edet et François Xavier Huan** et je voudrais les remercier d'avoir fait confiance à Skalpel pour la réalisation de leur site Internet !

Enfin, je voudrais remercier les personnes qui m'ont aidé dans la rédaction ainsi que dans la relecture de ce mémoire : en premier lieu, je pense à mon directeur d'étude M. **Cédric Kleinpeter**, mais aussi à **Burcu Çetinkaya** et à mon grand ami **Valéry Thomas** qui ont eu la gentillesse de prendre le temps de relire la centaine de page que composent ce mémoire.

Liste des abréviations et acronymes

ORM (Anglais) : Object-Relational Mapping / Mapping objet-relationnel

AGL : Atelier de Génie Logiciel

SVN : SubVersioN

PHP (Anglais) : Php : Hypertext Preprocessor / Préprocesseur hypertexte

HTML (Anglais) : HyperText Markup Language / Langage de balise hypertexte

CSS (Anglais) : Cascading Style Sheets / Feuille de style en cascade

PAQ : Plan d'assurance qualité

UML (Anglais) : Unified Modelling Language / Langage de Modelisation Unifiée

MCD : Modèle Conceptuel de Données

MLD : Modèle Logique de Données

MPD : Modèle Physique de Données

DOM (Anglais) : Document Object Model / Modèle, Structure d'objet du document

W3C (Anglais) : World Wide Web Consortium

FEVAD : Fédération de l'E-commerce et de la Vente À Distance

Glossaire

A

AdWords : Programme publicitaire de la société Google permettant aux entreprises d'acheter des mots clés et ainsi d'apparaître dans le moteur de recherche.

Arborescence (web) : Représentation schématique, logique et hiérarchisée des pages d'un site ainsi que des liens qui peuvent exister entre ces pages.

B

Backlink (anglais) : Lien hypertexte entrant vers une page d'un site Internet. Le lien peut être effectué depuis le même domaine ou depuis un domaine différent du site.

Back Office (anglais) : Interface graphique permettant d'administrer et de gérer le contenu d'un site Internet. A opposer au *Front Office*.

Business plan (anglais) : Document décrivant la nature d'une entreprise, ses stratégies de vente et marketing, son *background* financier ainsi que les estimations de profits ou pertes.

C

Client (web) : Logiciel qui envoie des demandes au serveur. Dans le domaine du web, le client est généralement le navigateur Internet de l'utilisateur.

CMS (anglais) : *Content Management System*. Voir « Gestionnaire de contenu ».

Commit (subversion) : Opération qui permet de valider et d'archiver les modifications d'un ou plusieurs fichiers dans l'outil de contrôle de code source Subversion.

CSS (anglais) : Langage qui permet de décrire la représentation graphique d'une page web écrite en HTML.

D

Dead Line (anglais) : Date butoir à laquelle une tâche doit être terminée. Par exemple, date à laquelle le projet doit être livré.

Découpage (HTML) : Voir « Intégration ».

DOM (web) : Interface de programmation fournissant un jeu d'objets standards pour l'accès, la représentation, la manipulation et la combinaison des balises dans les langages HTML et XML.

E

Easter Egg (anglais) : Fonctionnalité cachée et ludique dans un programme informatique, accessible via une combinaison de touches particulière.

E-commerce Analytics : Outil de la société Google permettant d'analyser l'audience d'un site Internet et de recouper les informations avec les ventes effectuées sur le site.

F

Flash : Technologie d'animation vectorielle de la société Adobe permettant la création d'animations intégrables sur une page internet.

Fork (anglais) : Nouveau logiciel créé à partir des sources d'un logiciel existant.

Framework (anglais) : Ensemble de bibliothèques, de classes, de fonctions et conventions permettant de faciliter et d'accélérer le développement d'application.

Front Office (anglais) : Partie du site Internet visible des internautes. A opposer au *Back Office*.

G

Gabarit (web) : Mise en page pré formatée, vide de contenu, qui sert de modèle pour la structure des pages d'un site Internet.

Gestionnaire de contenu (web): Logiciel permettant de gérer et d'administrer le contenu d'un site Internet.

H

HTML (anglais) : Langage de balises permettant de créer des pages web pouvant être interprétées par le navigateur Internet.

I

Index (SQL) : Objet d'une base de données permettant d'indexer des données dans le but d'en améliorer les temps d'accès.

Instant gagnant (mercatique web) : Forme de jeu de hasard dans lequel les internautes gagnent en se connectant à des tranches de dates / heures précises.

Intégration (HTML) : Activité qui consiste à concevoir les fichiers HTML et CSS depuis des maquettes, le plus souvent fournies au format Adobe Photoshop ou Fireworks.

J

Javascript (web) : Langage de script exécuté au sein du navigateur web de l'internaute pour interagir avec celui ci.

L

Load Balancing (anglais) : Technique permettant de répartir la charge de travail sur plusieurs serveurs (à l'insu du client) dans le but de réduire les temps de réponse.

Logigramme : Outil d'analyse et de description qui permet de décomposer une activité en évènements et de montrer les relations logiques qui les unissent.

M

Mapping objet-relationnel : Technique de programmation qui consiste à faire correspondre des objets du langage de programmation aux tables d'une base de données.

Maquette (web) : Photo numérique, représentation graphique du site Internet à créer. Très souvent fournie au format Adobe Photoshop ou Fireworks.

Merise : Méthode française d'analyse, de conception et de gestion de projet informatique. Merise décompose l'analyse sur trois niveaux : conceptuel, logique ou organisationnel, et physique.

Métadonnées : Ensemble structuré d'informations permettant de décrire une ressource. Dans le domaine de l'Internet, il s'agit très souvent de la description d'une page web.

O

ORM (anglais) : voir « Mapping objet-relationnel ».

P

Panier moyen (mercatique web) : Montant moyen d'une commande enregistrée sur un site e-commerce.

PHP (anglais) : Langage de script libre utilisé pour générer des pages HTML depuis un serveur web.

Plugin (anglais) : Composant logiciel permettant d'ajouter des fonctionnalités à un autre logiciel. Par exemple, le plugin Flash Player pour les navigateurs web.

Proof of concept (anglais) : Preuve qu'un produit, qu'une technologie, qu'un programme informatique ou qu'un système d'information est capable de résoudre une problématique.

Pure player (anglais) : Entreprise exerçant son activité uniquement via Internet, à l'image d'Amazon ou eBay.

R

Référencement (web) : Activité qui consiste à référencer un site Internet dans les moteurs de recherche.

Repository (subversion) : Entrepôt ou répertoire de données soumis au contrôle de code source. Le *repository* (ou dépôt) contient la liste des fichiers sources du projet dans leurs différentes versions.

S

Serveur (web) : Ordinateur connecté à Internet, qui héberge des données ou fichiers, dont le but est de répondre aux requêtes et de fournir un service aux clients.

Sitemap (anglais) : Fichier XML définissant la structure et le plan du site. Utilisé par les moteurs de recherche tel que Google dans le cadre de l'indexation d'un site Internet.

SOAP (web) : Protocole basé sur la technologie XML permettant l'appel de méthodes et de procédures à distance. Utilisé dans le cadre d'architectures orientées services (SOA).

T

Taux de conversion (mercatique web) : Pourcentage d'internaute réalisant effectivement un achat sur un site e-commerce.

TecDoc (anglais) : Référentiel mondial de pièces pour automobiles. Il s'agit de la base de données utilisée dans le cadre du projet Pièces Avenue.com.

Template (web) : Voir « Gabarit ».

Ticket (web) : Fichier, généralement intégré à un outil de suivi d'anomalies, contenant des informations relatives à un incident.

U

UML (anglais) : Langage d'analyse et de conception orienté objet défini par l'OMG (*Object Management Group*).

Update (subversion) : Opération de mise à jour des fichiers sources d'un poste client à partir d'un serveur subversion.

W

Web 2.0 : Concept d'utilisation du réseau Internet qui a pour but de valoriser l'internaute et ses relations avec les autres.

Wiki : Site web dont les pages sont modifiables par les utilisateurs afin de permettre un travail collaboratif sur les documents numériques qu'il contient.

Wireframe (anglais) : Document qui permet de représenter de façon schématique et hiérarchique les différentes zones des pages d'un site Internet.

Z

Zoning (anglais) : Voir « Wireframe ».

Table des matières

Remerciements	2
Liste des abréviations et acronymes	3
Glossaire.....	4
Table des matières.....	8
Introduction.....	11
<i>I Skalpel.....</i>	12
I.1 La société.....	13
I.2 L'expérience	15
I.2.1 Les Z'animousses.....	15
I.2.2 Moto 85	15
I.2.3 Motoshopping	15
I.2.4 Top's Cars.....	16
I.2.5 Bilan.....	17
I.3 L'organisation.....	18
I.4 L'équipe.....	19
I.4.1 Les concepteurs.....	19
I.4.2 Les développeurs.....	20
I.4.3 L'intégrateur	21
I.4.4 L'expert en référencement	22
I.4.5 Le chef de projet.....	22
<i>II Présentation du projet.....</i>	23
II.1 Le contexte.....	23
II.2 Le projet et sa réalisation	25
II.2.1 La durée du projet, le planning et la composition de l'équipe.....	25
II.2.2 L'organisation et la gestion de projet.....	26
II.2.3 L'architecture.....	26
II.2.4 Aspects techniques.....	27
II.2.4.1 Les choix matériels	28
II.2.4.2 Les langages et outils utilisés	29
II.2.4.3 Le contrôle de code source.....	32
II.2.4.4 Les environnements de développement	32
II.3 L'évolution du site	33
<i>III Les étapes de la conception et de la réalisation.....</i>	34
III.1 La conception	35
III.2 La réalisation	36
<i>IV La phase de conception.....</i>	37
IV.1 L'étude de faisabilité	37
IV.1.1 L'étude des interfaces	38
IV.1.2 TecDoc	39
IV.1.2.1 Le catalogue produit TecDoc.....	39
IV.1.2.2 Les Web Services	40
IV.1.2.3 L'étude	40
IV.1.2.4 Récapitulatif des interfaces	41
IV.2 La gestion de la qualité	42
IV.2.1 Le plan d'assurance qualité	42
IV.2.1.1 La démarche du projet.....	43
IV.2.1.2 Les livrables.....	43
IV.2.1.3 L'organisation	43
IV.2.1.4 Les livraisons et installations	45
IV.2.1.5 Le bilan projet.....	45
IV.2.2 La gestion des évènements.....	45
IV.2.3 La gestion de configuration.....	47

IV.2.3.1	Les documents du projet	48
IV.2.3.2	Les éléments graphiques	49
IV.2.3.3	Le code source de l'application	49
IV.2.4	Bilan.....	51
IV.3	La rédaction du cahier des charges fonctionnel.....	53
IV.3.1	Le contenu	53
IV.3.2	La méthodologie d'écriture	55
IV.3.2.1	L'analyse descendante	55
IV.3.2.2	Le langage et les outils.....	56
IV.3.2.3	La communication	59
IV.4	La conception d'une arborescence.....	60
IV.4.1	Les facteurs clés	60
IV.4.2	Exemple du processus de commande.....	60
IV.4.3	Retour d'expérience	61
IV.4.4	Le résultat.....	62
IV.5	L'annexe à l'arborescence	64
IV.5.1	Une étude ergonomique	64
IV.5.2	Un guide pour le développement.....	64
IV.5.3	La rédaction de l'annexe à l'arborescence	65
IV.6	Le zoning	66
IV.6.1	Définition du Zoning.....	66
IV.6.2	Mise en pratique	66
IV.6.3	Mise en pratique	70
IV.7	Le pré-briefing graphique	71
IV.8	L'estimation et le planning	73
IV.8.1	Le découpage en tâches	73
IV.8.2	L'enchaînement des tâches	74
IV.8.3	L'estimation	75
IV.8.4	L'affectation des ressources.....	75
IV.9	Bilan.....	77
V	<i>La réalisation du projet Pièces Avenue.</i>	78
V.1	La création du logotype	80
V.2	L'élaboration des maquettes	81
V.2.1	La prestation	81
V.2.2	L'organisation	82
V.2.3	L'expérience	86
V.3	Le mapping objet relationnel	87
V.3.1	Deux visions de la problématique	87
V.3.2	Le format YML	88
V.4	Le développement du back office	90
V.4.1	Le FrameWork Serum	90
V.4.1.1	Les besoins et contraintes	90
V.4.1.2	Le développement.....	91
V.4.2	Les spécificités de Pièces Avenue.....	92
V.5	Le découpage HTML / CSS.....	94
V.5.1	Les techniques de découpage	94
V.5.2	La méthodologie employée	95
V.6	L'intégration du Front Office	97
V.6.1	La hiérarchisation des templates	97
V.6.2	La création des gabarits	98
V.6.3	Les traitements et assignations	99
V.7	La recette	100
V.8	La post production et le référencement	101
V.8.1	L'après mise en ligne	101
V.8.2	L'optimisation du référencement.....	102
V.8.2.1	Le sitemap et le fichier robot.txt.....	102
V.8.2.2	Google e-commerce Analytics	102
V.8.2.3	Google e-commerce AdWords.....	103

V.8.2.4	Le référencement naturel	103
V.8.3	Les indicateurs du e-commerce	104
V.9	Bilan	105
V.9.1	La gestion du processus	105
V.9.2	La gestion de l'équipe	106
V.9.3	La gestion du client.....	107
Conclusion	109
Bibliographie	111
Liste des figures	112
Liste des tableaux	115

Introduction

Le choix d'un projet ou d'un sujet pour le mémoire d'ingénieur n'est pas toujours une chose aisée. D'une part parce que du point de vue du contenu et de la prestation réalisée, il doit être compatible avec la qualification « d'ingénieur ». D'autre part, je pense que le projet choisi doit être l'aboutissement du cursus universitaire et doit donc impliquer la totalité des connaissances acquises au cours de la formation au CNAM.

Mon choix s'est porté sur la « Gestion d'un projet Web E-commerce » car j'estime que parmi l'ensemble des projets auxquels j'ai participé dans ma carrière, il s'agit de celui qui répond le mieux à ces deux critères. Premièrement, le rôle de « chef de projet » est un des rôles qu'un ingénieur peut être amené à endosser. De plus, le projet Pièces Avenue a été complet d'un point de vue des connaissances à mettre en œuvre pour son bon déroulement : analyse du besoin, estimation, gestion humaine, gestion des conflits, qualité...

Ce mémoire va relater mon travail sur le projet Pièces Avenue. Je vais mettre en avant les techniques ou technologies employées dans le cadre du projet, mais je vais surtout insister sur le retour d'expérience, les difficultés rencontrées, ainsi que les solutions apportées et le ressenti personnel tout au long du projet. Il sera axé sur cinq parties :

- La présentation de la structure Skalpel, de son expérience dans le domaine du web et présentation de l'équipe chargée de réaliser le projet,
- Un aperçu général et rapide du projet Pièces Avenue : son contexte, les technologies employées, son architecture et son évolution,
- Un schéma représentant les différentes étapes de la conception d'un site web e-commerce. Il s'agit également des étapes suivies dans le cadre du projet Pièces Avenue,
- L'assistance à maîtrise d'ouvrage : la rédaction du cahier des charges, l'aide apportée à Pièces Avenue en amont de la réalisation du projet,
- La réalisation du projet : les différentes étapes de la réalisation ainsi que mon rôle dans chacune des étapes jusqu'à la livraison du site web.

Enfin, la conclusion me permettra de faire un point plus général sur le projet Pièces Avenue : ce que j'ai appris, ce que j'ai pu mettre en pratique et, avec le recul, ce qui aurait pu être amélioré. La conclusion sera également l'occasion de faire le lien entre les compétences à mettre en œuvre pour gérer un projet web e-commerce et la formation dispensée au CNAM. Nous verrons en effet que chaque unité d'enseignement joue un rôle et apporte sa pierre dans l'éducation et la formation du chef de projet. Enfin, cette dernière partie donnera un point de vue plus personnel sur la gestion de projet en général dans le domaine informatique et ce qui, d'après moi, fait un bon chef de projet.

I Skalpel

La première partie de ce document sera l'occasion de présenter la société au sein de laquelle je travaillais lors de la réalisation du projet Pieces Avenue. En effet, pour de nombreuses raisons il me semble primordial de présenter la société Skalpel avant d'entrer plus en profondeur dans le projet dont fait l'objet ce mémoire.

Tout d'abord, Skalpel a un historique et son activité a évolué entre sa création et la réalisation du projet Pieces Avenue. Indéniablement, chaque projet achevé, chaque prestation réalisée, chaque client accompagné a apporté une **expérience ainsi qu'un savoir faire dont a bénéficié Pièces Avenue**. Présenter ce parcours permettra ainsi au lecteur de mieux comprendre comment le projet a été pris en charge.

Mais cette expérience est aussi celle de ses collaborateurs, que je considère comme la « matière première » du projet. En effet, j'ai l'intime conviction que, au delà de la méthodologie ou de la nature de la prestation, la clef de voûte d'un projet est l'équipe chargée de sa réalisation.

Le « chef de projet » doit connaître les personnes avec lesquelles il travaille pour pouvoir mieux communiquer avec elles, leur affecter des tâches ou encore éviter les conflits et former des groupes travaillant en émulation. Une bonne connaissance de l'équipe permet alors d'augmenter considérablement la productivité dans le projet. Cela implique de connaître non seulement leurs **aptitudes techniques** et leurs **expériences passées**, mais aussi **leur profil psychologique** ou même leurs activités extra professionnelles.

Enfin, je présenterai l'évolution de mon rôle pendant les quatre années où j'ai travaillé au sein de la société Skalpel.

I.1 La société



Figure 1 - Logo Skalpel

Skalpel était une SSII spécialisée dans le domaine de l'Internet. Traditionnellement, ce type de structure est également appelé « agence web ». La société fut fondée en mai 2005 par Yoann Nussbaumer et dépose le bilan en juillet 2009, en raison de la grave crise financière et économique qui secoue le monde à cette époque.

La force de Skalpel résidait dans la qualité de ses solutions de gestion de contenu (d'abord le Skalpel Manager, puis son évolution « Serum ») qui lui permettait de se différencier singulièrement de la masse des prestataires existants sur Internet. Ces deux outils étaient particulièrement adaptés aux clients PME / PMI de Skalpel. En effet, ils avaient l'avantage d'être simples à manipuler, ergonomiques et fonctionnels. De plus, ils étaient modulaires, ce qui permettait des évolutions en fonction des nouveaux besoins des clients.



Figure 2 - Logo Infomoto

A ses débuts, Skalpel travaillait principalement en tant que sous-traitant pour la société « Infomoto.org », un portail dont le thème principal était la moto et les deux roues. Les prestations consistaient à réaliser des sites Internet pour les partenaires d'Infomoto.org. A titre d'exemple, nous avons réalisé les sites

des pilotes moto Philippe Donischal ou encore celui du pilote Moto GP Randy de Puniet. Son activité en déclin, Infomoto.org décide de réorienter sa stratégie, change de nom et devient Cube-X Agency. Cependant, les deux sociétés garderont un lien historique avec le domaine de la moto et continueront à travailler dans ce secteur avec notamment la refonte du site leader de petites annonces moto « Moto 85.com ».

Du fait de leur complémentarité, il y a toujours eu une proximité entre Skalpel et Cube-X Agency. C'est pourquoi la collaboration entre les deux sociétés a continué après l'ère « Infomoto.org ». Dans la pratique, Cube-X Agency réalisait les maquettes des sites web tandis que Skalpel installait son gestionnaire de contenu et s'occupait de l'aspect dynamique des sites (le code côté serveur). La proximité entre les deux sociétés était aussi liée au fait qu'elles ont partagé pendant deux années les mêmes locaux. Cette particularité nous permettait une communication beaucoup plus fluide et aisée. Ensemble, ces deux structures ont travaillé pour des groupes et clients ayant une notoriété nationale, tel que Air Caraïbes, Brossard Cakes, Hager, Le Petit Marseillais, Les camemberts Rustiques, le groupe Dubreuil, le groupe Motor Presse etc.

La demande étant de plus en plus forte, l'activité de Skalpel s'est lentement mais sûrement orientée vers le commerce en ligne et la conception de sites sur mesure. Deux types de sites émergent alors : les sites de petites annonces et ceux de ventes en ligne. La complexité de ces types de sites est évidemment plus élevée que celle d'un site institutionnel ou d'un site événementiel.

Après la réalisation de la boutique Labyrinth Bikes et du site de petites annonces Top's Cars, deux projets que l'on peut légitimement considérer comme des échecs, nous nous sommes rendus compte que la gestion de projet revêtait une importance capitale dans notre métier et qu'il ne fallait surtout pas la prendre à la légère, sous peine de perdre beaucoup d'argent. En effet, malgré leur aboutissement, ces deux projets ont eu beaucoup de retard à la livraison et ont nécessité beaucoup de maintenance corrective. Ces retards étaient

clairement dus à l'absence d'une politique de gestion des projets. Nous avons donc décidé de remettre en questions nos méthodes de travail pour les projets Pièces Avenue.com et Solargie.

Malgré des premiers retours positifs et une situation qui s'améliorait au fil du temps, la crise de 2009/2010 frappe de plein fouet Skalpel. Plusieurs prospects ont alors purement et simplement annulé leurs projets. Ces annulations, qui représentaient une perte de plusieurs dizaines de milliers d'euros, ont été fatales à Skalpel qui ferme définitivement ses portes en juillet 2009 après deux mois sans projet. Le dernier projet réalisé aura été le site de la société Shining Karma... auparavant connue sous le nom Cube-X Agency.

I.2 L'expérience

Entre sa création et le développement du projet Pièces Avenue, Skalpel a réalisé un très grand nombre de sites Internet. Cette partie va présenter succinctement les réalisations marquantes de la société, avec pour objectif de faire comprendre au lecteur l'évolution de l'activité et de l'expérience de la société au travers de ces différents projets.

I.2.1 Les Z'animousses

Le premier projet que j'ai souhaité présenter est un projet en collaboration avec CubeX Agency pour le compte de l'agence de communication « Espadon ». Il s'agissait d'un site web promotionnel pour un produit de la marque « Le petit Marseillais » avec la particularité d'avoir un jeu basé sur le principe des « instants gagnants ». L'internaute gagne un prix s'il se connecte au bon moment sur le site web. Le front office était en Full Flash, Skalpel a réalisé le Back Office et la logique du jeu. Il s'agissait des prémisses de l'orientation e-commerce de Skalpel.

I.2.2 Moto 85

Moto 85 est un site de petites annonces dans le domaine du deux roues. Avec ce projet, Skalpel a mis **un pied dans le domaine du commerce électronique**, avec la mise en place des moyens de paiement en ligne pour régler les annonces, la conception d'un espace sécurisé pour les membres leur permettant de visualiser leurs factures, leurs annonces, etc. Ce projet a aussi été un atelier pour le perfectionnement de nos compétences en référencement, puisque grâce aux optimisations dans ce domaine, le trafic du site a triplé entre l'ancienne et la nouvelle version.



Figure 3 - Capture d'écran du site Moto 85

I.2.3 Motoshopping

Avec Motoshopping, Skalpel reste dans le domaine de la moto. Cependant, il ne s'agit plus d'annonces mais bel et bien de **la première boutique de vente en ligne** développée par la société. Nous avons du assimiler et intégrer des concepts fondamentaux du commerce électronique : catalogue produit, panier, bons de réductions, livraison et transporteur,

paiement et retours... Ce fut une expérience importante pour Skalpel puisque nous avons pu bénéficier du retour d'expérience sur les multiples projets e-commerce qui ont suivi tel que Sirokipik ou encore Pièces Avenue.

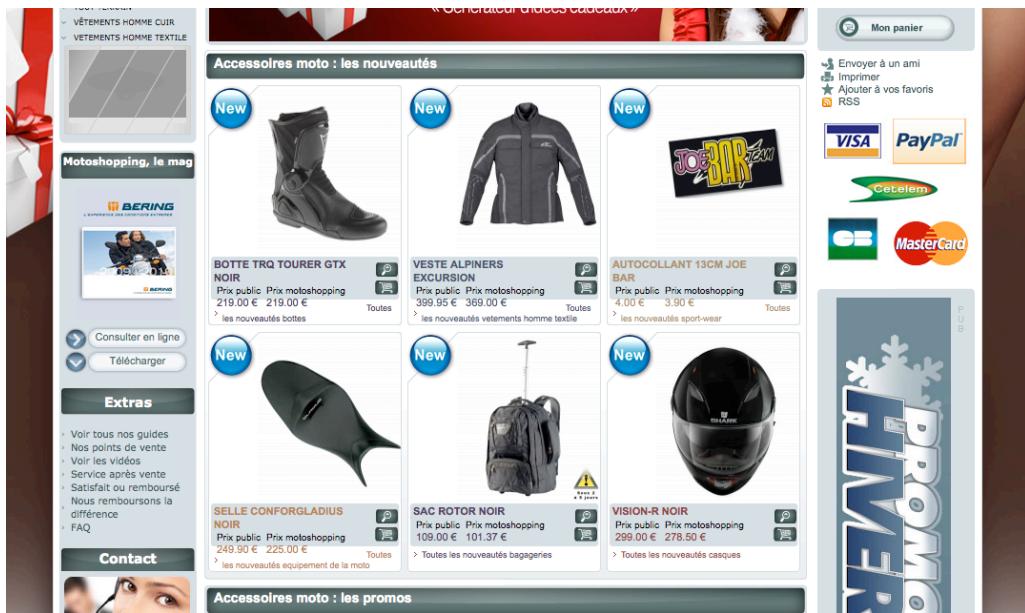


Figure 4 - Capture d'écran du site Motoshopping

I.2.4 Top's Cars

Top's Cars est un autre projet de petites annonces, cette fois ci dans le domaine de l'automobile de luxe. Si la société s'est appuyée sur l'expérience acquise grâce à Moto85, elle a quand même beaucoup appris de ce projet. En effet, la gestion du client était particulièrement complexe du fait de la différence de taille et de poids entre les deux structures. De plus, cela a été l'occasion de nous rendre compte de la nécessité de mettre en place des **process pour la réalisation de nos sites web**. En particulier, nous avons pris la décision de mettre l'accent sur la spécification des besoins, la description des méthodologies employées et donc la nomination d'un vrai chef de projet pour les futurs projets.



Figure 5 - Capture d'écran du site Top's Cars

I.2.5 Bilan

L'activité de la société Skalpel a **progressivement** évolué de la réalisation de site web institutionnels/promotionnels vers le commerce électronique. Lors de cette mutation, nous avons pu bénéficier du retour d'expérience à chaque palier atteint.

Cela peut paraître assez anecdotique, mais nous nous sommes toujours familiarisé avec un domaine d'activité en commençant par la réalisation d'un site de petites annonces dans celui-ci. L'expérience emmagasinée ainsi fut à mon sens très utile lors du passage au développement des vrais sites marchands...

Mais j'ai aussi évolué d'un **point de vue personnel**. Pendant mes trois premières années en tant que développeur web, j'ai pu profiter des **connaissances techniques** acquises en travaillant sur de nombreux sites tels que Top's Cars, Motoshopping ou encore Moto 85. Sur ces projets, j'ai été au contact des technologies utilisées dans le domaine du web et notamment les langages PHP, HTML, CSS, ou Javascript.

C'est aussi grâce à ces trois années de développement que j'ai compris que la **gestion de projet** était une activité fondamentale et que sans elle, n'importe quel projet était voué à l'échec. J'ai ensuite décidé de m'orienter dans cette voie car je ne souhaitais plus vivre ces échecs. C'était aussi pour moi une sorte « d'évolution naturelle », car je souhaitais continuer ma carrière dans le domaine du web.

I.3 L'organisation

La société Skalpel était organisée sur **un modèle matriciel**. Chaque membre de l'équipe était affecté à un ou plusieurs projets à la fois. Par exemple, Louis Saunders, un architecte, travaillait en même temps sur la conception Merise de Pièces Avenue et sur la finalisation d'un autre projet.

Cela n'est pas toujours évident puisqu'il faut savoir jongler avec les *dead-lines* ou le planning imposé par les autres projets. Notre chance était la petite taille de notre structure : en effet, les chevauchements de planning étaient finalement assez rares, avec au maximum deux ou trois projets menés simultanément.

J'ai compris qu'au sein d'une structure pareille, il est important de savoir **négocier les temps de travail** de chaque membre de l'équipe. Pour le chef de projet, il est impératif **d'argumenter** pour mettre en avant l'importance de son projet et convaincre les autres responsables de libérer des ressources.

Ces **compétences** ne sont pas innées. Elles peuvent s'acquérir sur le « terrain », mais j'estime qu'elles nécessitent aussi une personnalité particulière. Un chef de projet ne peut pas être une personne passive, il doit être capable de défendre ses intérêts et son projet, qu'il s'agisse d'une négociation en interne ou avec le client. Ce sont des choses que l'on n'apprend pas forcément à l'école.

Nous verrons plus loin comment cette négociation a été menée pour les ressources affectées au projet Pièces Avenue, en particulier dans le cas de Louis.

I.4 L'équipe

L'objectif de cette partie est de présenter les membres de l'équipe de développement du projet Pièces Avenue.

La réalisation d'un site Internet est une activité pluridisciplinaire. Parmi les corps de métier qui ont participé à la réalisation du site, on retrouve des **analystes-concepteurs** (Louis Saunders et Killian Ebel), des **développeurs** (Hervé Weltzer, Nicolas Girardin, Thanh Tri Phan et William Guth), un **intégrateur HTML** (Mickael Knauer) et un **expert en référencement** (Maxime Castanié).

Cette partie ne va pas uniquement lister les membres de l'équipe, leurs qualités et leurs expériences passées. En effet, cela n'aurait aucun intérêt si elle ne cherchait pas à analyser l'impact que cela peut avoir sur le projet.

Ce chapitre va donc montrer comment la connaissance des membres de l'équipe va **influer sur la répartition et l'affectation des tâches** dans un projet, sur la création de **groupes de travail** ou même sur le style de **communication à adopter** avec un membre. Elle va expliquer en quoi il est crucial de **connaître ses équipiers** pour un chef de projet. Enfin, un des rôles du chef de projet étant de recruter son équipe, je ferais part de mon retour d'expérience à ce sujet.

I.4.1 Les concepteurs



Figure 6 - Avatar
Louis Saunders

Louis Saunders a rejoint la société Skalpel peu après mon embauche. Avant d'intégrer le projet Pièces Avenue, Louis a notamment travaillé seul sur le site « Votre-Electricien » pour le client Hager, ainsi que sur plusieurs boutiques de vente en ligne tel que « Sirokipik » ou encore « e-bikershop ».

Sur ces projets, Louis a démontré qu'il était capable de faire preuve d'**autonomie** pour réaliser un site Internet à partir de son cahier des charges et jusqu'à sa recette. Avec le recul, j'ai même la conviction qu'il avait besoin de cette autonomie et d'un certain niveau de challenge pour s'épanouir professionnellement.

Un des points communs que nous partagions est notre préférence pour la méthodologie « **Merise** ». En effet, nous avions une démarche commune dans l'analyse et la résolution d'un problème : « MCD », « MLD » puis « MPD ». De plus, en plus de nos connaissances dans le web, Louis et moi avions une certaine expérience dans la plateforme .NET de Microsoft, ce qui nous permettait par exemple d'avoir des références communes lors de nos discussions.



Figure 7 - Avatar
Killian Ebel

Killian Ebel a eu un parcours assez atypique puisqu'il a d'abord suivi un cursus de type « DUT Services & Réseaux de communications » avant d'intégrer la Licence Professionnelle d'Informatique à l'IUT Robert Schumann d'Illkirch.

Il n'a donc pas un profil de développeur classique. Cependant, je n'ai jamais considéré son parcours universitaire comme un handicap, car Killian avait une **capacité** impressionnante à **assimiler** de nouveaux

concepts tel que la programmation orientée objet ou l'UML.

Cela nous a été très utile car nous travaillions avec des ORMs (outil de *mapping* objet-relationnel) et donc nous raisonnions en « objet ». Cette particularité nous a aussi donné la possibilité de confronter deux analyses différentes du même problème : l'approche Merisienne menée par Louis et celle basée sur UML de Killian. En tant que chef de projet, j'étais vraiment satisfait de pouvoir avoir deux perceptions complémentaires d'un même problème.

Dans tous les cas, j'ai souvent considéré le parcours de Killian comme un atout pour Skalpel, puisqu'il avait une sensibilité graphique et un bagage technique varié (Flash, etc.) que l'on ne retrouve pas forcément chez tous les développeurs issus de la filière informatique.

I.4.2 Les développeurs

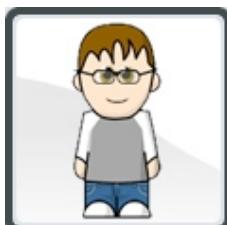


Figure 8 - Avatar
Hervé Weltzer

Hervé Weltzer avait un profil plus « traditionnel » d'ingénieur en informatique. Il a d'abord commencé sa formation par un DUT en informatique. Il a ensuite validé son diplôme d'ingénieur « SupInfo » en alternance pendant trois ans au sein de la société Skalpel avant d'intégrer définitivement la société en tant que développeur.

C'était un développeur expérimenté qui a notamment joué un rôle important dans la refonte du site « Moto85 » et dans la réalisation de la boutique en ligne « Motoshopping.fr ».

Hervé avait une bonne connaissance du web et de ses technologies. Il effectuait une veille constante. D'un point de vue psychologique, c'était un développeur plutôt solitaire, qui aimait travailler seul. Il avait la particularité de rester calme quelque soit la situation. Il était plutôt lent, mais son travail était toujours **réfléchi** et d'une rare **qualité**.

C'est une des raisons pour lesquelles je pouvais toujours faire confiance à Hervé. Avec Killian, c'est d'ailleurs Hervé qui a hérité des parties les plus complexes du projet, comme nous le verrons dans le planning présenté plus loin dans ce document.

Thanh Tri Phan était le second développeur sur le projet Pièces Avenue. Comme Hervé, il avait un profil d'ingénieur en informatique, une formation qu'il a suivie à l'Exia de Strasbourg. Nous avions déjà eu l'occasion de travailler avec Thanh Tri sur plusieurs projets dont « Air Caraïbes » et « Solargie ».

Contrairement à Hervé, Thanh Tri était **plutôt rapide** et se projetait très vite dans le code. Cependant, il était aussi **moins méticuleux** qu'Hervé. C'est la raison pour laquelle je préférais lui affecter des tâches de développement dont l'aspect algorithmique n'était pas trop complexe.

William Guth était lui aussi élève ingénieur à l'Exia. Je ne le connaissais pas avant son stage chez Skalpel mais un rapide entretien m'a conforté dans ses aptitudes techniques. Le fait que Thanh Tri et William s'entendaient bien a également été un facteur important pour moi puisque je souhaitais leur affecter des tâches communes et j'avais besoin qu'il y ait une émulation entre ces deux développeurs.

Le dernier binôme était composé d'**Amaury Chedotal** et de **Nicolas Girardin**, deux développeurs en apprentissage (respectivement en Licence Professionnelle et à l'école d'ingénieur SupInfo) avec lesquels nous n'avions pas travaillé auparavant et que je ne connaissais pas du tout. Lors de leur entretien, ils ont toutefois montré beaucoup **d'envie** et de **motivation** par rapport au projet Pièces Avenue. Et si par la suite Amaury s'est effectivement impliqué dans le projet, Nicolas a été souvent distant voire même **nonchalant**. C'est d'ailleurs pour cela que nous avons du prendre la décision de le retirer du projet.

Finalement, j'ai pu vraiment me rendre compte a quel point il pouvait être difficile de faire confiance et donner des responsabilités à des développeurs en apprentissage et plus généralement à des collaborateurs que l'on ne connaît pas encore. Malgré tout, pour un chef de projet, cette confiance est absolument nécessaire pour déléguer des tâches ou travaux. Avec le recul, je me suis aussi rendu compte que cette confiance ne s'acquiert qu'en faisant l'effort de vraiment connaître ses collègues et en ayant une bonne vision de leurs aptitudes techniques et de leurs personnalités.

Le recrutement des stagiaires s'est lui aussi avéré être très long et fastidieux. Les entretiens étaient composés d'un entretien oral et d'un test d'aptitude technique. Mon retour d'expérience est que le test d'aptitude n'a servi à rien et que la motivation ne peut pas être évaluée sur la base d'un seul entretien présentant le projet ou le poste. Il en est de même pour l'intégration du candidat à l'équipe. Et s'il existe des tests (tel que celui de M. Belbin) pour donner des indications sur la personnalité d'un candidat ou sur son rôle au sein d'une équipe, aucun ne peut réellement garantir qu'une **alchimie va s'opérer**.

I.4.3 L'intégrateur



Figure 9 - Avatar
Mickael Knauer

Sur ce projet, le rôle d'intégrateur était occupé par **Mickael Knauer**. Avant son embauche par Skalpel, Mickael travaillait au sein de l'agence de communication Reymann. Nous avions déjà eu l'occasion de travailler ensemble dans le cadre d'une prestation réalisée pour la société de distribution « Cora ».

Mickael a rejoint Skalpel bien avant le lancement du projet Pièces Avenue. Il a travaillé sur le découpage de nombreuses maquettes tel que « Votre-Electricien » ou « Top's Cars ». Techniquement, Mickael était **expérimenté** et avait une très bonne expertise sur le *framework* Javascript Mootools. Dans son travail, Mickael était aussi **minutieux** et **autonome**. Ces trois particularités étaient importantes à mon niveau pour deux raisons.

La première était que le budget ne permettait pas d'embaucher un intégrateur supplémentaire et il fallait donc que le rôle soit occupé par une personne ayant déjà une bonne expérience et qui soit capable d'évoluer seul.

L'autre raison est que l'intégration n'est pas mon domaine de prédilection. J'avais besoin de pouvoir compter sur ses compétences et ses conseils dans mes choix et il aurait été compliqué pour moi d'accorder du crédit à un développeur moins expérimenté.

I.4.4 L'expert en référencement

Maxime Castanié nous a conseillé dans **l'optimisation du référencement** (ou SEO, un acronyme pour *Search Engine Optimisation*) du site. Nous verrons que cette discipline porte une importance capitale tout au long de la vie d'un site web.

En effet, un site Internet peut être très bien conçu d'un point de vue technique ou avoir un design et une ergonomie adaptée à sa cible, s'il n'a aucune **visibilité** sur le réseau Internet il n'aura absolument aucun **intérêt** pour le commerçant.

Concrètement, cette discipline consiste à faire ressortir le site web lorsqu'un internaute effectue une recherche sur un site tel que *Google* ou encore *Bing*. Il y a plusieurs techniques pour cela et nous les détaillerons plus loin dans ce document.

Maxime était un expert tout à fait autonome dans son domaine. Nous verrons que nous l'avons consulté très tôt dans la réalisation du site, dans la phase de conception de « l'annexe à l'arborescence ». Toutefois, ses interventions étaient ponctuelles et il a surtout joué un rôle de conseiller et de formateur.

I.4.5 Le chef de projet

J'ai occupé le rôle de **chef de projet** sur le projet Pièces Avenue.

J'ai évolué vers ce poste de manière très naturelle. En effet, après avoir développé des sites web pendant **trois années sans réelle gestion de projet**, nous nous sommes rendus compte que notre productivité pouvait être grandement améliorée. Nous perdions beaucoup de temps car le **besoin n'était pas clairement défini**, le client changeait d'avis très fréquemment et il nous **manquait un cadre et une méthodologie** de travail. Le déclencheur fut le projet Top's Cars, un échec à imputer principalement à l'absence d'un réel chef de projet.

Pour le projet Pièces Avenue, nous avons donc décidé de nous remettre en question et de définir un process de travail que nous avons détaillé dans le **Plan d'Assurance Qualité**. En effet, la qualité est grandement dépendante des méthodologies de travail.

Ce document était présenté au client pour validation et faisait entièrement partie du contrat qui lie le prestataire au client. Nous verrons plus loin dans ce mémoire de quoi le PAQ était composé et quelles sont les étapes du process de création d'un site web.

La seconde décision fut de donner une plus grande importance à la **définition du besoin** et à **l'assistance** à la définition du besoin. Le mémoire détaillera aussi les différentes étapes pour définir le besoin : fonctionnel, ergonomique et enfin graphique.

Enfin, la **gestion humaine** a été le dernier point que nous devions améliorer. Cela était valable d'un point de vue interne, où il a fallu apprendre à gérer et motiver l'équipe de développement, mais aussi avec le client (gestion du changement, négociation, etc.).

J'ai pris énormément de plaisir à occuper ce poste. Malheureusement, les événements ont fait que cette expérience n'a pas pu être répétée sur d'autres grands projets... Mais elle reste celle qui me motive aujourd'hui à retrouver un poste similaire et orienter ma carrière dans cette direction.

II Présentation du projet

II.1 Le contexte

Le projet Pièces Avenue a été initié en octobre 2008. En effet, c'est à cette date que M. Stéfan EDET et M. François-Xavier HUAN, les deux entrepreneurs à l'origine du projet, décident de prendre contact avec la société Skalpel pour la réalisation d'un site Internet dont le but est de commercialiser des pièces automobiles (filtres à huile, balais d'essuie glace, etc.) sur Internet. Pièces Avenue est ce que l'on appelle un « pure-player ». En d'autres termes, il s'agit d'une entreprise qui réalise son chiffre d'affaire exclusivement sur le réseau Internet (à l'image d'Amazon par exemple).

A cette époque (et encore aujourd'hui), le marché du commerce en ligne est en plein essor et le secteur de la pièce automobile ne compte que quelques acteurs majeurs sur Internet. Ces sites concurrents ne semblaient toutefois pas aussi aboutis que ce à quoi le consommateur final pouvait s'attendre et une analyse poussée a fait rapidement émerger un certain nombre de faiblesses telles que les délais de livraison, l'absence de support et de services, etc.

Fort de ce constat et de plusieurs années d'expérience dans le secteur de la pièce automobile, M. EDET et M. HUAN décident de saisir cette opportunité pour créer la structure Pièces Avenue et font appel à Skalpel pour la réalisation du projet.

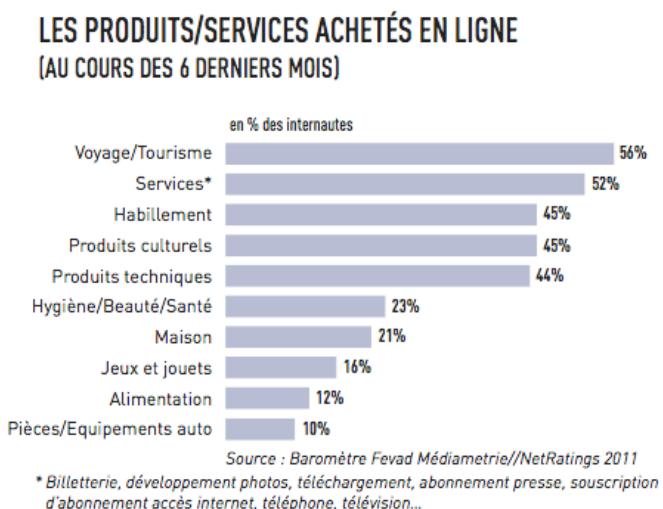


Figure 11 - Produits / Services achetés en ligne

LE MARCHÉ E-COMMERCE ET VENTE À DISTANCE

Vente de produits et services
en milliards d'Euros

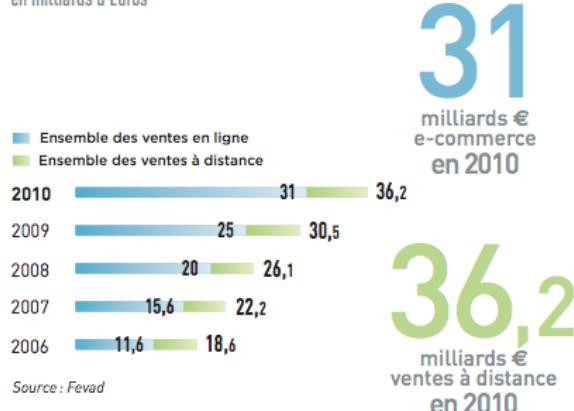


Figure 10 - Statistiques VAD en 2010

L'objectif était de faire du site Pièces Avenue une marque forte et identifiée par le grand public comme le premier site de vente en ligne de pièces automobiles. Les points forts du projet sont la variété ainsi que le nombre de références du catalogue produit, les délais d'expédition et de livraison, les services et le support associés.

Pour cela, il était impératif de mettre en ligne un site clair, simple à utiliser et agréable pour l'internaute.

L'utilisation des dernières tendances en matière de marketing et de conception de site Internet était indissociable des ambitions affichées. Pièces Avenue a donc fait appel à l'expertise de la société Skalpel pour :

- L'analyse **financière** et l'élaboration d'un *business plan* pour la société Pièces Avenue,
- L'**'assistance à maîtrise d'ouvrage'**, ou AMOA ainsi que l'aide à l'**expression du besoin**,
- La **conception, le développement** et la maîtrise **technique** du projet,
- L'acquisition de trafic et l'**optimisation « SEO »** (*Search Engine Optimisation*).

Mon intervention aura plus particulièrement impacté les deux points que sont l'**assistance à maîtrise d'ouvrage** (aide à la rédaction du cahier des charges) ainsi que la **gestion du projet**.

Je vais développer ces deux aspects dans la suite de ce mémoire. Je détaillerai les aspects managériaux, techniques, organisationnels, économiques et budgétaires, mais ce mémoire traitera aussi de la nécessité en formation, de la maintenance ou encore de la qualité des livrables qui sont fournis dans le cadre d'un projet de site Internet E-commerce.

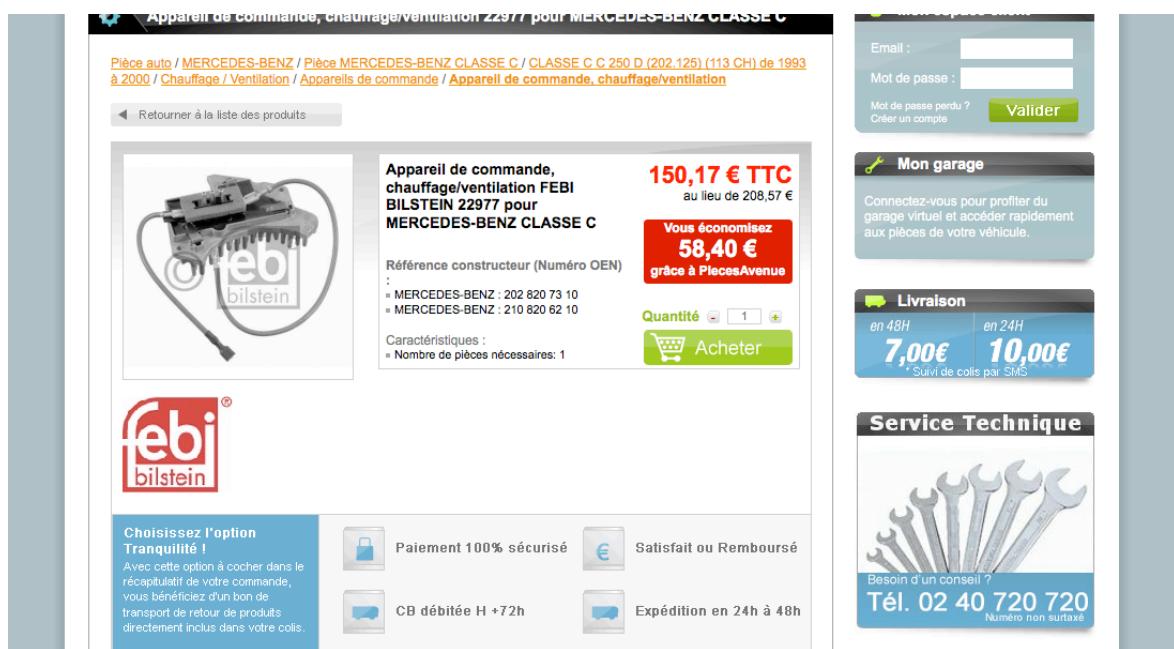


Figure 12 - Capture d'écran d'une page de détail sur Pièces Avenue

II.2 Le projet et sa réalisation

Je vais aborder ici quelques aspects généraux du projet Pièces Avenue et de sa réalisation. Je rentrerai plus en détail dans chacun des points dans les parties suivantes du mémoire. L'idée ici est de donner une vision globale du projet, sans véritablement être exhaustif. Quatre aspects seront plus particulièrement traités ici :

1. La durée du projet et la composition de l'équipe,
2. L'organisation et la gestion de projet,
3. L'architecture du site ainsi que les interfaces,
4. Les aspects techniques.

II.2.1 La durée du projet, le planning et la composition de l'équipe

Habituellement, pour estimer les coûts, délais, et charges nécessaires à la réalisation d'un projet, il faut d'abord rédiger et analyser un cahier des charges que l'on va supposer comme complet. Dans le cas du projet Pièces Avenue, ce n'est pas tout à fait ce qui a été fait. En effet, lorsque j'ai hérité du dossier Pièces Avenue en décembre 2008, le budget ainsi que les délais et dates de livraison du projet avaient déjà été définis par l'ingénieur Technico-commercial en charge du projet, à partir des grandes lignes du projet: une boutique en ligne, une foire aux questions dynamique, une section « petites annonces » et une interface d'administration.

En gestion de projet nous avons l'habitude d'utiliser le triangle « coûts », « qualité », « délais ». Ici les coûts étaient imposés par le budget et les délais par le business plan. Mon rôle a surtout consisté à satisfaire au mieux aux besoins du client dans ces limites.

La date limite de livraison du projet a été fixée au **premier mars 2009**. Le budget estimé par l'ingénieur technico-commercial nous a imposé le développement du site en **trois mois**, ce qui s'est révélé être trop court.

Les grandes étapes du projet ont été les suivantes. D'octobre à décembre 2008, l'ingénieur technico-commercial a pris en main le dossier et estimé le budget ainsi que les délais. De décembre 2008 à fin janvier 2009, je suis intervenu en assistance à maîtrise d'ouvrage pour la rédaction des documents du projet. Dans le même temps, l'équipe de développement était chargée de réaliser le nouvel outil de gestion de contenu de la société Skalpel (appelé Serum) qui sera également utilisé pour le projet Pièces Avenue. Le développement du site à proprement parler a débuté en février pour finir au début du mois d'avril, soit avec un mois de retard par rapport au planning initial.



Figure 14 - Logo CXA

Comme je l'ai mentionné précédemment, l'équipe chargée du projet était composé d'un chef de projet, d'un intégrateur HTML / Javascript, de trois développeurs expérimentés, de quatre développeurs en

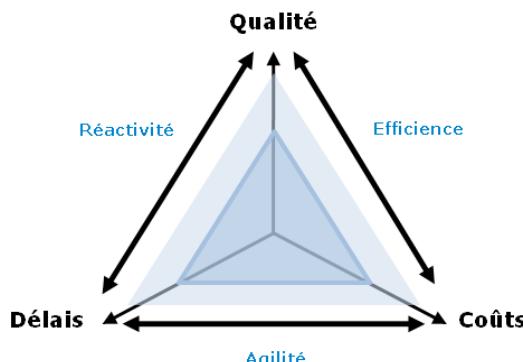


Figure 13 - Triangle de la qualité

apprentissage, d'un expert en optimisation de référencement. Nous avons également fait appel à une société extérieure partenaire de Skalpel (Cube-X Agency) pour la réalisation des maquettes et du logotype.

II.2.2 L'organisation et la gestion de projet

Après plusieurs années de travail dans le secteur de l'IT, force est de constater que les agences web sont considérées comme des « sous sociétés de services » en informatique. Peut-être est-ce dû à l'absence de maturité du domaine, à l'absence de méthodologie de certaines structures ou encore à la (toute relative) simplicité des langages utilisés qui donnent aux acteurs de l'industrie cette impression...

Cependant la réalité est toute autre. La réalisation d'un projet web, de surcroit « e-commerce », nécessite une méthodologie particulière. Les enjeux sont ici plus importants que pour un simple site institutionnel. L'aspect managérial n'est pas moins complexe que pour un autre projet d'application lourde par exemple. L'équipe était composée d'une dizaine de personnes et de plusieurs sociétés externes ou partenaires qu'il a fallu gérer. Les estimations de charges, délais, coûts sont autant fondamentaux que pour n'importe quelle autre application. Pour que le projet web soit un succès, l'ensemble des composantes de la gestion de projet doivent être mise œuvre.

Nous verrons dans la suite du mémoire quelles sont ses composantes et comment je les ai mises en œuvre dans le cadre de ce projet.

II.2.3 L'architecture

Ce qui fait la particularité du domaine du web est son aspect « interconnecté ». Les interfaces entre systèmes d'informations sont ici omniprésentes.

Lorsqu'un internaute navigue sur le site Pièces Avenue, il affiche les produits issus du catalogue produit « **TecDoc** ». Ce catalogue au format électronique est un des référentiels de pièces automobiles les plus utilisés au monde. C'est en fait une base de données qui contient les pièces auto mais aussi les différents véhicules, affectations, visuels, caractéristiques des produits, etc.

A ces produits ont été assignés des **prix**, fournis par les constructeurs de pièce automobile dans un format « excel ». L'utilisateur ajoute ces produits au panier et peut ensuite payer en ligne sur l'**interface Paybox** (auparavant **Atos SIPS**).

La société Pièces Avenue travaille en flux tendus. C'est à dire qu'elle ne gère pas ou très peu de stocks. C'est pourquoi, une fois la commande de l'internaute effectuée sur le site, les pièces sont automatiquement commandées au sein des sociétés **Adipa** et **Doyen Auto**. Ces sociétés sont en fait des logisticiens, qui possèdent un stock important de pièces auto et livrent la société Pièces Avenue dès qu'une commande est réalisée et réglée sur le site.

A la réception de ces pièces, les préparateurs de commandes emballent les colis et expédient les pièces via les transporteurs « **La poste** » ou « **Chronopost** », puis un message est envoyé au téléphone mobile de l'internaute depuis la **plateforme SMS OVH**. Enfin, une fois par mois, un export vers le logiciel de comptabilité **Cegid** est effectué...

Le site doit donc communiquer avec l'ensemble de ces systèmes d'information. Nous détaillerons les interactions plus loin dans ce document. L'image ci-dessous permet de se rendre compte des différents échanges que nous avons étudié et mis en place.

Enfin, les interfaces ont aussi la particularité de devoir être interchangeables. En effet, l'ajout d'un nouveau logisticien ou d'un nouveau transporteur doit être faisable facilement. Depuis le lancement du site, il y a d'ailleurs eu une modification du paiement en ligne (il s'agissait d'ATOS SIPS auparavant), l'ajout du logisticien « Doyen Auto », l'envoi de SMS ou encore la livraison en point de retrait « Ici Relais ». Tout ceci est bien évidemment transparent pour l'internaute mais implique une conception dont les maîtres mots sont « **évolutivité** » et « **modularité** ».

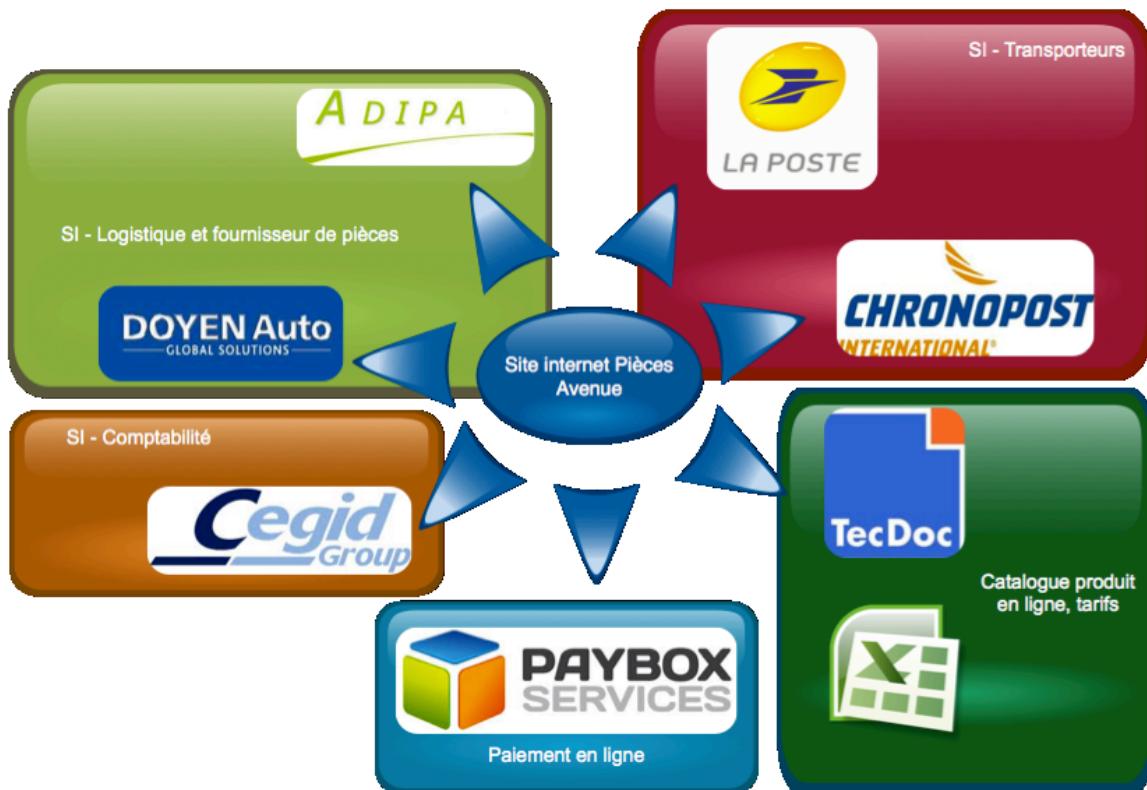


Figure 15 - Interconnexion des systèmes d'information

II.2.4 Aspects techniques

Historiquement, Skalpel a toujours travaillé avec des langages et des outils provenant du monde du logiciel **libre**. Ce choix était motivé par deux grandes raisons.

Il y a tout d'abord le point de vue **économique**. Au sein d'une petite structure, l'achat de licences, de forfaits d'assistances ou de formations sont des dépenses très lourdes qu'il faut essayer de réduire au maximum. Le choix de plateformes libres permet à ces petites entreprises d'avoir à disposition des outils performants, gratuits et dont la communauté est active et permet d'avoir rapidement l'aide nécessaire.

L'autre grande raison est le caractère particulier du **marché du travail Français** dans le domaine de l'informatique. Contrairement à des pays comme les Etats-Unis où les technologies prédominantes dans le web sont des approches propriétaires tel qu'ASP ou

.NET, la France et les développeurs français apprécient beaucoup ces technologies libres et il est alors plus facile de trouver des développeurs compétents.

Les aspects concernant le choix de l'environnement technique sont les suivants :

- Le choix du matériel,
- Les langages de programmation employés dans le cadre du projet,
- Les outils et logiciels utilisés pendant la phase de développement,
- Le contrôle de code source.

En réalité, le contexte technologique que je vais présenter ici n'est pas propre au projet Pièces Avenue. Il s'agissait d'un environnement que nous utilisions de base pour tous les projets, avec quelques variantes d'un projet à l'autre.

II.2.4.1 Les choix matériels

Fondamentalement, tout site web est basé sur une architecture client/serveur. Le poste client accède à l'aide d'un navigateur Internet au contenu du site web, qui est stocké sur un ou plusieurs serveurs. La figure ci-dessous représente une architecture de ce type.

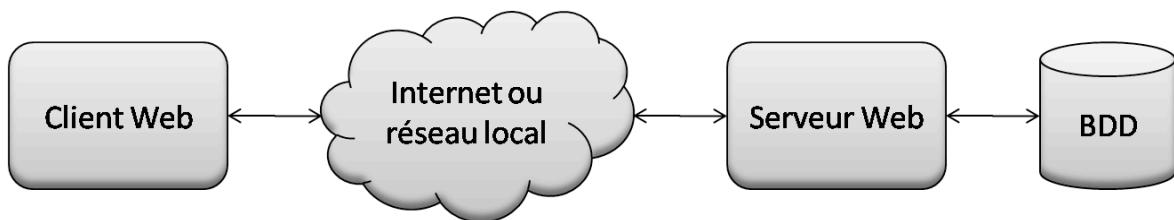


Figure 16 - Architecture Web

Le choix du serveur est une étape qui s'est opérée rapidement puisque la société PiècesAvenue avait déjà acheté au préalable le matériel nécessaire pour son site. Il s'agissait d'un serveur de base de données et d'un serveur frontal web.

En tant que chef de projet, j'ai dû valider que ce matériel répondait à notre besoin d'hébergement, ce qui était effectivement le cas pour le lancement de la boutique. Mais ces deux serveurs se sont avérés n'être pas assez puissants lors des montées en charges, en particulier le serveur frontal web.

Voici les configurations initiales des machines :

Serveur Web	Serveur de base de données
<ul style="list-style-type: none">• HP ProLiant DL380 G5• 1 x Quad-Core Xeon E5310 / 1.6 GHz• RAM 1 Go• Disque dur - 72 Go - SAS - 10000 tours/min	<ul style="list-style-type: none">• HP ProLiant DL360 G5• 1 x Quad-Core Xeon E5335 / 2 GHz• RAM 2 Go• Disque dur 2 x 72 Go SAS hot swap 2.5"• Gigabit Ethernet



Figure 17 - Logo Celeonet

Les deux machines étaient parfaitement fonctionnelles mais il manquait simplement le branchement sur le réseau Internet et la location de la bande passante pour le site. Ces prestations ont été réalisées par l'hébergeur Céléonet, un partenaire historique de la société Skalpel.

II.2.4.2 Les langages et outils utilisés

Dans le cadre du projet PiècesAvenue, la génération des pages côté serveur était réalisée à l'aide du langage de programmation **PHP 5**. En effet, cette version est celle qui respecte le plus les paradigmes de la programmation orientée « objet ».



Figure 18 - Logo PHP

En tant que chef de projet, je n'ai pas eu à faire un réel choix quant au langage de programmation. En effet, PHP était parfaitement en mesure de répondre aux besoins du projet. De plus, il était déjà bien maîtrisé par l'ensemble de l'équipe de développement. Choisir un autre langage aurait introduit un grand risque car l'équipe n'aurait pas eut la même maturité sur ce nouveau langage. Sans compter la perte de temps engendrée par l'apprentissage d'un nouveau langage...

D'un point de vue fonctionnel et technique, Pièces Avenue était un projet d'une grande ampleur pour une équipe de sept développeurs. Il était impossible de tout réaliser dans un délai de trois mois et nous avons donc opté pour l'utilisation de plusieurs boîtes à outils que l'on appelle des « *frameworks* ».

L'avantage indéniable de ces outils est qu'ils évitent à des petites structures telles que Skalpel l'écriture, la maintenance et l'évolution d'une partie du code de l'application. Ce gain se traduit par la **réduction des temps de développement** et donc **des coûts de réalisation**.



Figure 19 - Logo Doctrine

Le premier *framework* que j'ai choisi d'utiliser est l'outil de *mapping* objet relationnel « Doctrine for PHP ». Le *mapping* objet relationnel est la technique qui permet de lier des « objets » au sens langage de programmation à des systèmes de gestion de bases de données relationnelles. C'est aussi ce que l'on appelle plus communément **la persistance des données**.

Cet outil propose trois grands avantages. Le premier est l'abstraction du système de stockage. Il est théoriquement possible de migrer la base de données d'un système à l'autre (de MySQL à Oracle par exemple) sans aucune modification du code de l'application.

De plus, le développeur n'a plus à se soucier de l'écriture de lignes de code SQL. Il manipule uniquement des objets sans se soucier des requêtes de sélection, mise à jour ou suppression dans la base de données.

Enfin, le dernier gros avantage est que l'outil de *mapping* objet relationnel (ou « ORM ») permet de garantir la validité des informations enregistrées dans la base de données. Il permet par exemple de contrôler les types de valeurs, longueurs de chaînes. Mieux encore,

il permet de sécuriser sa base de données contre certaines attaques du type « Injection SQL » en contrôlant les données à enregistrer.

J'ai choisi Doctrine car il s'agissait d'un ORM très prometteur. Certes, il nécessitait plus de ressources CPU que ses concurrents mais les fonctionnalités et les perspectives d'évolution ont fait pencher la balance en faveur de cet outil. Il fut par la suite adopté par beaucoup d'autres utilisateurs dans le monde et il fut notamment intégré au *framework* Symphony 2.0 dont la philosophie est très similaire à ce que nous avons fait avec Pièces Avenue.



Figure 20 - Logo Zend Framework

Le second *framework* côté serveur est le **Zend framework**, que nous avons réellement utilisé comme boîte à outils: manipulation de dates, envoi de mails, génération de PDF, etc. Il s'agissait de piocher dans les classes mises à disposition par le Zend *framework* afin de traiter un problème en écrivant le moins de lignes de code possible.

Le choix du Zend *framework* n'est pas innocent. Je l'ai choisi car je connaissais bien ce *framework*, je savais ce dont il était capable et ce pour quoi nous allions l'utiliser. En effet, je l'avais étudié dans le cadre d'un autre projet qui fut finalement annulé. D'autant plus que son concurrent, PEAR, était trop lourd à installer, utiliser, et surtout, son architecture ne correspondait pas à ce que nous souhaitions faire avec Pièces Avenue. Finalement, le Zend *framework* a été choisi pour ses fonctionnalités et sa flexibilité.

Enfin, quelques librairies supplémentaires ont été utilisées ponctuellement, notamment pour la génération de code barre, la lecture de flux RSS en PHP à l'aide de SimplePie, ou encore l'interaction avec les services web avec NuSOAP.



Figure 21 - Logo Dwoo

Pour la partie visible par les internautes (Front Office), la couche de présentation est réalisé à l'aide du couple « **xHTML** » / « **CSS** », en s'appuyant sur le moteur de *template* **Dwoo**. L'avantage dans l'utilisation d'un moteur de *template* est qu'il permet de séparer la couche de « contenu » de la couche de « présentation » et ainsi d'architecturer l'application en respectant un modèle n-tiers, idéal pour l'évolutivité. La séparation est réalisée en assignant des variables (il s'agit de scripts PHP dans la couche de « contenu ») et en les affichant à l'aide de fichiers TPL dans la couche de « présentation ».

Dwoo est le successeur de Smarty, un moteur de *template* que nous avions notamment utilisés dans le cadre du projet Moto 85 et que nous connaissions donc très bien. C'est d'ailleurs une des raisons pour laquelle je l'ai choisi. L'équipe s'est très rapidement adaptée à ce nouvel outil car il était rétro-compatibile. L'autre raison était qu'en tant qu'évolution de

Smarty, Dwoo apportait plusieurs fonctionnalités très intéressantes telles que l'héritage de *template*. Ces fonctions nous permettaient de gagner encore plus de temps en écrivant moins de ligne de HTML. Nous verrons dans la suite de ce mémoire comment cela fut possible.



Figure 22 - Logo Mootools

L'aspect dynamique et web 2.0 du site est obtenu à l'aide du *framework* javascript **Mootools**. L'intérêt principal dans l'utilisation d'un *framework* Javascript est de s'abstraire du navigateur. En effet, la principale problématique est que chaque navigateur

possède une interprétation propre du langage Javascript. Certains navigateurs acceptent des syntaxes particulières ou différentes des standards du W3C. L'utilisation de Mootools nous a évité un long travail de compatibilité inter-navigateurs et a apporté un gain de temps non négligeable.

C'est Mickael Knauer qui a choisi cet outil. En effet, au sein de l'équipe et en tant qu'intégrateur XHTML / Javascript, c'était le membre qui avait le plus de légitimité pour le choix du *framework* Javascript. Il a préféré Mootools car il l'avait déjà utilisé à maintes reprises auparavant. Aujourd'hui, Jquery me semble être un meilleur choix car il est adopté par une grande communauté alors que Mootools semble être sur le déclin. Si l'adoption d'un *framework* était une chose obligatoire (qu'il s'agisse de l'un ou de l'autre), elle a aussi imposé un apprentissage à l'équipe. C'est pourquoi j'ai préféré valider le choix de Mickael. Ce dernier a d'ailleurs formé les autres développeurs.



Figure 23 - Logo Ext JS

Pour la partie Back Office **Serum**, il s'agit d'une couche de présentation basée sur **ExtJS**. Le choix de cette technologie nous a permis de créer rapidement une interface d'administration dont le look est similaire à celui d'une application lourde et offrant des fonctionnalités qui ne sont pas disponibles avec les *framework* javascript «classiques» (tel que le clic droit, les effets d'accordéons, les graphiques, etc.).

Ext faisait partie d'un projet d'entreprise et cette technologie nous a été imposée par notre dirigeant, Yoann Nussbaumer. En effet, les nouveaux outils de gestion de contenus livrés à nos clients devaient être codés à l'aide de Ext.



Figure 24 - Capture d'écran de l'interface d'administration

II.2.4.3 Le contrôle de code source



Figure 25 - Logo
SubVersioN

Le contrôle de code source faisait partie intégrante des processus qualité chez Skalpel. Le système utilisé était **Subversion** (ou SVN).

J'avais choisi cet outil bien avant Pièces Avenue, principalement parce qu'il s'agissait d'un outil libre et gratuit. La majorité des interfaces de développement ont aujourd'hui un plugin pour intégrer les opérations de subversion (commit, update etc.) dans l'IDE, permettant ainsi de faire gagner du temps aux développeurs.

L'administration du serveur subversion (entre autres, la gestion des « répositories », des comptes utilisateurs et l'affectation des droits) était réalisée à l'aide de l'outil **uSVN**, un projet d'interface d'administration web de subversion développé initialement par des étudiants français comme projet de fin d'études.

A ma connaissance, il n'y avait à l'époque pas d'autres outils de ce type. En tout cas je n'en avais pas trouvé. Le choix de uSVN s'est donc imposé par lui-même. Finalement, ce fut vraiment une aubaine d'avoir trouvé cet outil. En effet, il m'a évité de perdre du temps dans la modification des fichiers de configuration des repositories. Alors que j'aurais du utiliser des protocoles tel que SSH pour me connecter à distance à la machine et modifier manuellement les fichiers de configuration, uSVN m'a permis de réaliser les opérations directement depuis un site web, de manière totalement sécurisée.

II.2.4.4 Les environnements de développement

L'ensemble des membres de l'équipe de développement utilisait l'environnement de développement intégré NetBeans avec la prise en charge du langage PHP, HTML, CSS, Javascript et de Subversion. Un plugin supplémentaire pouvait être installé pour la prise de contrôle à distance et le travail collaboratif.

Le graphisme du site web a été réalisé puis découpé à l'aide du logiciel **Adobe Fireworks**.

Le choix des « AGLs » (ateliers de génie logiciel) et autres outils n'était pas imposé. De même, le choix du système d'exploitation était laissé à l'appréciation et aux affinités de chaque développeur puisque Netbeans est compatible avec Linux, Windows et Mac OSX.

En tant que chef de projet, j'ai toujours pensé que l'essentiel était que chacun puisse travailler dans les meilleures conditions. J'ai souvent entendu des développeurs critiquer tel ou tel environnement de développement, ou tel système d'exploitation.

Je pense que dans la mesure où cela n'impacte pas la sécurité du système informatique d'une entreprise, chaque développeur devrait être en mesure de choisir les outils avec lesquels il souhaite travailler, car il s'agit souvent de ceux avec lesquels il est le plus performant et productif.

J'admet toutefois que si cette politique est facilement adaptable dans une petite structure, la mettre en œuvre dans des entreprises plus grandes peut poser des problématiques tel que la gestion du parc informatique, des licences, etc. qui peut être difficile à traiter au cas par cas.

II.3 L'évolution du site



Figure 26 - Logo Ti-Pièces

Pièces Avenue a été mis en ligne en avril 2009 avec un mois de retard sur le planning initial. La société Skalpel a continué d'effectuer la maintenance corrective du site jusqu'à sa liquidation qui a eu lieu en mi-juillet 2009.

Suite à cette liquidation, j'ai pris la décision de devenir auto-entrepreneur et de continuer à maintenir seul le site, en marge de ma nouvelle activité principale que j'exerçai au sein de la société Technology and Strategy. Cette période a durée environ un an et demi, jusqu'au mois de mars 2011 où j'ai finalement accepté la proposition de travail à temps plein offerte par Pièces Avenue.

Depuis mars 2011, je travaille donc seul et à temps plein pour la société Pièces Avenue / FAST PR. Mon travail consiste à effectuer la maintenance corrective et évolutive du site Pièces Avenue, mais aussi à mettre en place des sites satellites à Pièces Avenue tel que Ti-Pièces, un « fork » de Pièces Avenue qui est spécialisé dans la vente de pièces automobiles pour les départements et territoires d'outre mer.

Depuis le lancement en 2009, beaucoup de choses ont évolué sur le site. Correction des bogues, mais aussi ajout de nouveaux logisticiens, ajout et modification des moyens de paiement, ajout de nouveaux modes de transport, ajout de nouvelles fonctionnalités tel que l'envoi de SMS à l'expédition de la commande ou encore amélioration de l'ergonomie du site, ajout de nouveaux serveurs web frontaux et d'un système de « *load balancing* » (répartition de charge)... Le site est en constante mutation, même si cela ne se retrouve pas forcément directement aux yeux des internautes.

Avec le recul, je pense que le fait de travailler seul implique de grandes responsabilités et nécessite un niveau de connaissances techniques plutôt élevé. D'autant plus que le web est un domaine pluridisciplinaire et un bon chef de projet doit en connaître l'ensemble des technologies. Plus exactement, il ne doit pas forcément être un expert pour chacune d'elles, mais il doit savoir quelles sont les possibilités et limites de chaque technologie. Travailler seul implique cependant de devoir connaître toutes ces technologies. Et de bien les connaître.

Il est vrai toutefois que du point de vue de la « gestion de projet », les enjeux sont moins importants aujourd'hui. Pas de gestion humaine, de résolution de conflits, des plannings beaucoup plus souples, une architecture déjà en place sont autant de facteurs qui facilitent la gestion des évolutions et les corrections sur le site.

III Les étapes de la conception et de la réalisation

Ce chapitre du mémoire va expliciter quelles sont les étapes de la conception et de la réalisation d'un site web e-commerce, ainsi que les enchaînements entre celles-ci. Mais, en premier lieu, il est important de comprendre la **différence** entre ces deux termes.

La **conception** est la phase d'étude, de réflexion, de prise de choix qui va mener par la suite à la phase de **réalisation**, où l'on troque les crayons et la calculatrice contre un clavier et une souris. La réalisation consiste donc à mettre en pratique ce qui a été écrit et imaginé dans la phase de conception.

Nous pouvons faire le rapprochement entre ces deux étapes et la Roue de Deming. Alors que la phase de conception est le « **Plan** » de la roue de Deming, la phase de réalisation représente le « **Do** ». Nous verrons que les phases Check et Act sont aussi abordées dans le cadre du projet.

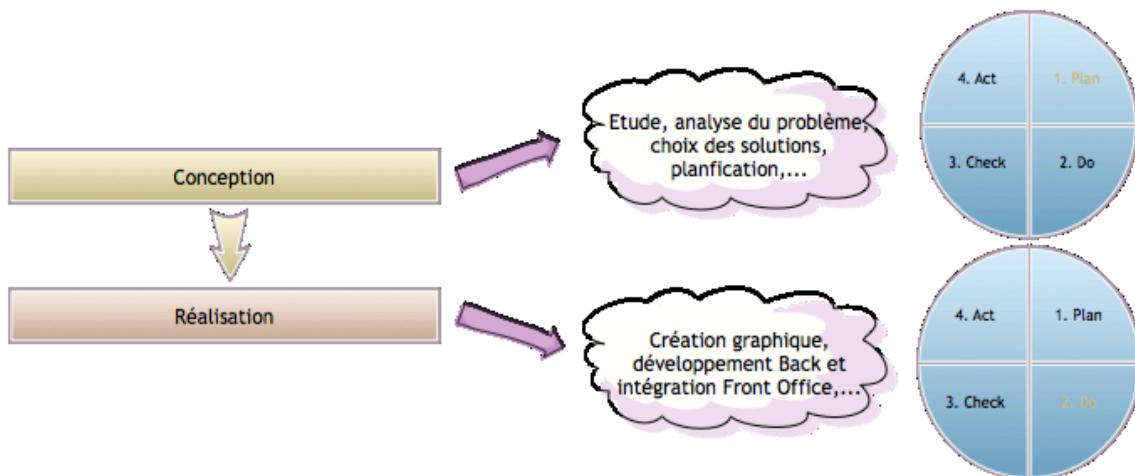


Figure 27 - Les deux phases de la réalisation d'un site Web

Ce chapitre va présenter ces deux phases sous forme de schémas et graphiques agrémentés de commentaires.

La démarche qui va être présentée ici est le fruit du retour d'expérience que j'ai acquis durant les nombreux projets auxquels j'ai participé en tant que développeur. C'est la démarche que j'ai jugé la plus logique pour un projet de création de site Internet. Mon point de vue était qu'il fallait d'abord concevoir le site sous tous ses angles (fonctionnel, graphique et ergonomique) avant de passer au développement.

Aujourd'hui encore, je pense qu'il s'agit de la meilleure méthode pour développer un site Internet. Suite au projet Pièces Avenue, j'ai eu l'occasion de mettre en œuvre cette méthodologie pour le cas « Solargie » et le résultat fut encore une fois un succès.

Chaque étape de ces deux phases sera détaillée dans les parties correspondantes du mémoire. Je décrirais aussi plus en détail mon rôle dans ces deux étapes, mais aussi les difficultés rencontrées ainsi que les solutions que j'ai du mettre œuvre pour résoudre ces problèmes.

III.1 La conception

La première étape de la phase de conception est l'étude de faisabilité. Cette étude est cependant optionnelle et uniquement effectuée en cas de besoin (par exemple, s'il y a une nouvelle technologie à tester). Puis vient ensuite la rédaction et la diffusion du plan d'assurance qualité, la rédaction du cahier des charges, l'arborescence et son annexe, le zoning des pages et enfin le briefing graphique. Une fois que l'ensemble de ces éléments a été étudié, il est possible de faire une estimation et un planning avant de se lancer dans la réalisation.

Pendant chacune de ces étapes, il faut respecter le plan d'assurance qualité. Enfin, la particularité de la phase de conception est qu'elle implique un grand nombre d'échanges avec la maîtrise d'ouvrage.

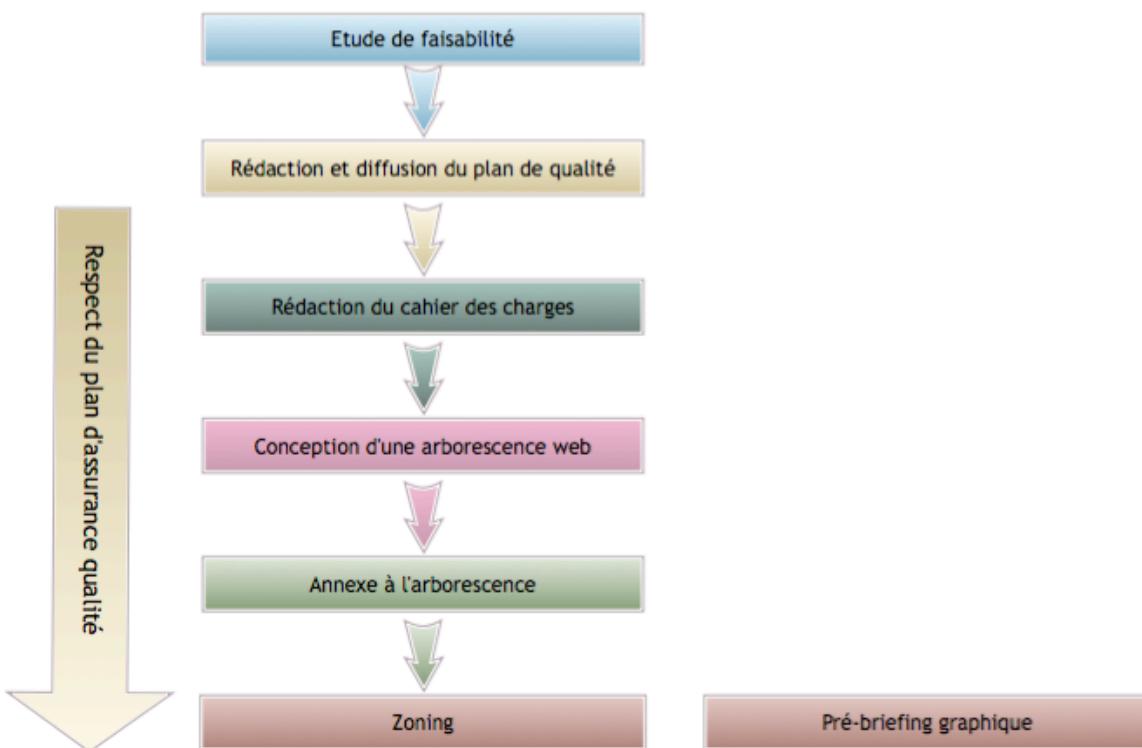


Figure 28 - Etapes de la phase de conception d'un site Web

III.2 La réalisation

La phase de réalisation débute par l'élaboration d'une maquette et la création d'un logotype. Il est ensuite possible de faire le découpage HTML ainsi que l'intégration du front office. En parallèle, les développeurs peuvent travailler sur le *mapping* objet-relationnel et le back office.

Enfin, la phase de recette et de débogage (respectivement **Check** et **Act** de la roue de Deming) permettent de clôturer le projet. J'ai également ajouté ici l'étape d'optimisation du référencement, qui a lieu en fait durant toute la durée de vie du site web.

Encore une fois, il faut respecter le plan d'assurance qualité pour l'ensemble des étapes de cette phase de réalisation. Ici, ce n'est plus avec le client que les échanges sont nombreux, mais avec les différents prestataires et partenaires (graphismes, SI externes, etc.) qui interviennent dans la réalisation.

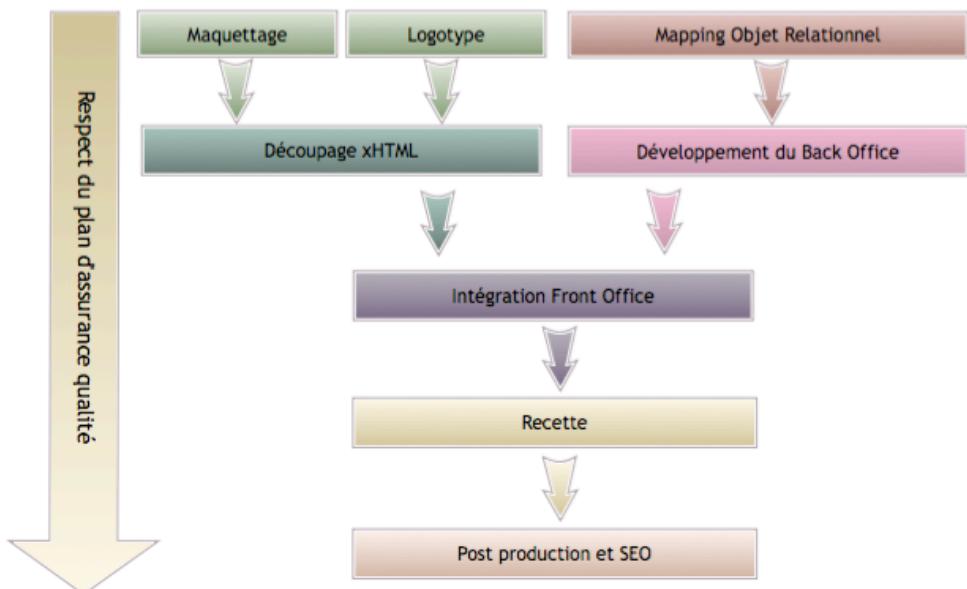


Figure 29 - Etapes de la phase de réalisation d'un site Web

IV La phase de conception

Cette partie du mémoire va détailler de manière chronologique l'ensemble des activités de la **phase de conception**. Chaque étape fera l'objet d'un chapitre dont le contenu sera constitué d'une description détaillée de l'activité et, le cas échéant, d'un argumentaire sur les solutions apportées ainsi que l'expérience que j'en ai retiré.

IV.1 L'étude de faisabilité

La première étape dans la création d'un site Internet complexe tel que Pièces Avenue est l'**étude de faisabilité**.

L'**Étude de faisabilité** dans la gestion de projet est une étude qui tend à prouver que le projet est techniquement faisable et économiquement rentable. Dans une optique plus large, on distingue les volets suivants dans une étude de faisabilité: étude technique, commerciale, économique, juridique et d'organisation.

Source: Wikipedia

Je vais me concentrer ici sur l'**aspect technique** de l'étude de faisabilité. En effet, lorsque le dossier m'a été remis, les aspects commerciaux, économiques et juridiques avaient déjà fait l'objet d'une étude menée par le conseiller commercial de la société Skalpel, Julien Lavault.

Dans cette étude, on retrouve notamment une rapide **analyse concurrentielle** du projet qui permet de visualiser quels sont les concurrents de Pièces Avenue, et quels sont les points faibles et les points positifs de chaque site Internet dans ce secteur d'activité. Elle comporte aussi un **business plan** (en français, « plan d'affaire ») sur une ou deux années, qui permet de s'assurer de la viabilité économique du projet et de fixer des objectifs de rentabilité. Ce business plan est notamment basé sur une estimation de l'évolution du nombre de ventes et du panier moyen. Je ne m'attarderai pas sur ces études car je ne les ai pas réalisés.

Une fois ces « études d'affaires » validées, le projet est confirmé comme étant valable économiquement. Et c'est à ce moment là que le dossier m'est transmis. Mon objectif est alors de valider que les demandes du client sont réalisables techniquement, notamment via le développement de prototypes qui serviront de *Proof of Concept* (preuve de la viabilité du concept).

A ce stade du projet, nous n'avons évidemment pas à notre disposition l'ensemble des demandes du client. Mon étude se base donc sur certains points clés et les grandes lignes du projet. Quels étaient ces points clés pour Pièces Avenue ?

Pour la boutique, je connais les partenaires choisis par la société Pièces Avenue : la banque de notre client, les prestataires chargés de la logistique et du transport des marchandises, les référentiels à utiliser pour le catalogue produit, les logiciels utilisés pour la comptabilité, etc.

Je connais aussi les grandes fonctionnalités du site Internet. Celles ci m'ont été transmises par le conseiller commercial: un site Internet e-commerce avec pour particularité le

catalogue produit TecDoc, une foire aux questions dynamique (semblable au concept de forum), un espace membre sécurisé ainsi qu'une interface d'administration.

La grande question que l'on se pose à ce moment est donc : à partir de ces informations, **comment puis-je valider la faisabilité** technique du projet et, pour cela, quels points allons nous étudier puis valider ?

IV.1.1 L'étude des interfaces

Dans le cadre du projet Pièces Avenue, il était capital de pouvoir assurer les **interfaces** du site avec l'ensemble des partenaires, des prestataires et des systèmes d'information. Par exemple, je devais m'assurer qu'il serait possible d'exporter à partir du site un fichier des ventes pour le progiciel Cegid.

Les interfaces sont souvent des aspects sensibles des projets car ils impliquent une connexion avec **un système d'information imaginé par d'autres développeurs**.

Ces derniers n'ont pas nécessairement la même perception du monde réel et donc la modélisation peut (c'est **souvent** le cas dans la réalité) varier entre les deux systèmes d'information. Sur le projet Pièces Avenue, les interfaces ont été les suivantes :

Pour les transporteurs: Nous avions déjà travaillé auparavant avec les sociétés « La Poste » et « Chronopost » dans le cadre du projet Motoshopping. L'étude de faisabilité n'a donc pas concerné ces points.

Pour les logisticiens et pour la comptabilité: l'étude a consisté à analyser un modèle de fichier, à valider qu'il serait possible de le générer puis le télécharger sur un serveur distant. Il s'agit en fait d'opérations que nous avions déjà réalisé pour de nombreux sites donc encore une fois, l'étude a été très rapide.

Pour le paiement en ligne: initialement, la banque choisie par notre client était la banque « BNP-Paribas ». Cette banque propose la solution de paiement en ligne SIPS, développée par la société Atos Origin. Cette solution était déjà connue puisque je l'avais installé sur Motoshopping. Après le lancement du site, le client est passé à la solution Paybox System, que nous (Hervé Weltzer) avions aussi eu l'occasion de mettre en place sur Moto85. Ici aussi l'étude a été très rapide.

Pour le catalogue produit: finalement, la seule zone d'ombre du projet. Je ne savais pas comment le catalogue produit TecDoc fonctionnait. C'est assez compréhensible dans la mesure où il s'agit du cœur du métier du client, métier qui n'est évidemment pas le même d'un projet à l'autre. Le client souhaitait que nous mettions en place la solution TecDoc Web Services. Je savais que cette intégration était possible, car plusieurs concurrents de Pièces Avenue travaillaient déjà avec ce système. Il fallait simplement savoir comment fonctionnaient les échanges entre les deux systèmes d'information. La partie suivante détaillera les travaux menés dans le cadre de cette étude.



Figure 30 - L'étude des interfaces

IV.1.2 TecDoc

La partie TecDoc est celle qui a nécessité le plus de **recherche** dans ce projet. En effet, il a fallut d'abord comprendre la logique métier du catalogue produit (il s'agissait en fait de répondre à la question : *comment fonctionnent les pièces auto?*) avant de s'attaquer au fonctionnement des web services. Je vais exposer ici l'étude de la **logique métier** et du **web service**, mais aussi la **méthodologie** employée dans le cadre de cette étude.



Figure 31 - Logo
TecDoc

IV.1.2.1 Le catalogue produit TecDoc

La base de données TECDOC recense l'ensemble des véhicules ainsi que les pièces automobiles de chaque fournisseur. Elle permet également de faire le lien entre les types des véhicules et les pièces des fournisseurs.

Les véhicules sont classés par marque / modèle / type. Chaque marque a un logo dans le référentiel. Une marque commercialise un ou plusieurs modèles de véhicules dont chacun a des dates de début et de fin de production. Par exemple, Audi (marque) construit des A3 Sportback (modèle) de 2004 à 2008. Chaque modèle est également décliné en plusieurs types. Les types apportent notamment l'ensemble des informations sur les puissances des véhicules. Par exemple, le type 1.9L TDI 105CV pour l'Audi A3 Sportback.

En France, il existe également une donnée appelée « type mine » pour chaque type de véhicule. Un type de véhicule peut avoir plusieurs numéros de « type mine ». Par exemple, pour la Clio 1.1 RL 5 Portes, les types mines associés sont les types mines n° B57105 ou B57104.

Par le biais de ses partenariats avec le fournisseur de pièces, TECDOC recense également l'ensemble des pièces automobiles ainsi que les possibilités d'adaptation des pièces sur les types de véhicules – une pièce étant adaptable sur plusieurs types de véhicules. Un fournisseur peut produire différents types de pièces (ou familles de produits, ex : balais d'essuie glace). De la même manière, un type de pièce peut être produit par plusieurs fournisseurs.

Les pièces sont identifiées par un numéro de référence et un numéro de fournisseur de pièce. Par exemple, le filtre de référence LS867B chez le constructeur Purflux.

Une pièce est fabriquée par un et un seul fournisseur. Elle peut jouer plusieurs rôles en fonction du type de véhicule. Par exemple, un moteur d'essuie glace pour une A3 Sportback 1.9L TDI 105CV peut jouer le rôle d'un moteur de lève vitre sur une Clio 1.1 RL 5 Portes. Les rôles des pièces sont classés par catégorie et sous catégorie (exemple : Moteur / Allumage / Bougies). Ces rôles et catégories sont également enregistrés dans la base de données TECDOC.

Les familles de produits TECDOC peuvent être soumises à une consigne. La consigne permet aux membres effectuant des commandes dont le panier comporte des produits d'une famille spécifique de renvoyer leurs anciens produits et d'être remboursés. Par exemple, si le membre achète un alternateur, il peut renvoyer son ancien alternateur et récupérer une consigne. Le montant d'une consigne est une valeur fixe pour l'ensemble des produits d'une famille de produits.

Les pièces ont deux sortes de descriptions. Une description « physique » qui concerne toutes les informations propres à la pièce. Par exemple, ses tailles ou poids. Mais il est également possible d'associer des critères spécifiques pour un type de véhicule et une pièce donnée. Par exemple, la pièce LS867B de Purflux fonctionne sur une Renault Clio 1.1 RL 5 Portes de 1998 à 1999.

En plus d'avoir une référence chez le fournisseur de la pièce, la pièce peut également avoir une référence au sein d'un constructeur automobile. Le filtre LS867B peut ainsi avoir une autre référence chez Renault ou chez Citroën.

IV.1.2.2 Les Web Services

Une fois que nous avions compris la logique métier, il nous fallait prendre en main le service web Tecdoc. Le web service Tecdoc est basé sur la technologie SOAP. L'appel et la récupération des résultats furent donc assez triviaux.

Par contre, l'enchaînement des appels nous a posé quelques problèmes. En effet, plusieurs cheminements étaient possibles pour l'exploration du catalogue produit et cela nous a un peu perturbé. Finalement la solution principale choisie, et celle qui correspondait le mieux à notre site, était la suivante :

1. Sélection de la marque du véhicule
2. Sélection du modèle du véhicule
3. Sélection de la motorisation du véhicule
4. Choix des catégories de produits sur 3 niveaux
5. Choix de l'article générique (la famille de produit)
6. Affichage de la liste des produits
7. Choix d'un produit et affichage du détail du produit pour le véhicule sélectionné

IV.1.2.3 L'étude

L'étude a été réalisée en deux temps. J'ai affecté cette tâche à Louis Saunders. Elle a duré environ un mois.

Tout d'abord, pour comprendre **la logique métier**, je lui ai demandé de travailler avec un catalogue produit sous forme de « texte à plat » qu'il a intégré dans une base de données MySQL avec un script d'import. Cette première phase a été la plus complexe puisqu'il a fallu comprendre les règles métier énoncées dans la première partie avec pour seule et unique documentation la base de données ainsi qu'une description en anglais des tables et champs.

La seconde phase a été **l'étude du service web**. Cette seconde étape était plus rapide car nous avions bien compris la structure de l'information. Il nous fallait simplement trouver quelles étaient les méthodes du web service à utiliser pour arriver à notre fin. J'ai demandé ensuite à Louis de nous réaliser un prototype afin de le présenter au client.

J'ai constaté que le web service était trop lent pour l'utiliser tel quel. J'ai donc pris la décision d'intégrer un système de cache des résultats en base de données.

Au départ, je pensais effectuer des appels successifs afin de récupérer l'ensemble des pages (produits, catégories etc.) et n'utiliser ensuite plus que notre base de données. Cette solution n'était pas viable car le traitement aurait été trop long et il nous aurait fallut des mois entiers pour récupérer toute la base Tecdoc. De plus, on perdait ainsi tout l'intérêt du web service qui était d'avoir des produits constamment à jour sur le site.

Le principe finalement adopté a été de stocker en cache dans notre base de données le résultat de chaque appel au web service pendant trois mois dans notre base de données. Passé ce délai, la requête serait une nouvelle fois envoyée au serveur Tecdoc.

De cette manière, j'avais trouvé une solution intermédiaire entre la version sans cache (trop lente pour notre site) et la solution initialement imaginée (impossible à mettre en œuvre).

Etape 1 : choix de la marque
ACURA

Etape 2 : choix du modèle
INTEGRA 3/5 portes

Etape 3 : choix du type
1.6 (CV)

Etape 4 : choix de l'assembly
Filtre à air

Etape 5 : choix de l'assembly de niveau 2 (puis 3, etc.)
ASHUKI 0396-6004

Etape 6 : choix de l'article générique
Filtre à air

Etape 7 : Affichage d'une liste de produits

Figure 32 - Prototype TecDoc

IV.1.2.4 Récapitulatif des interfaces

Le tableau ci-dessous dresse la liste des systèmes d'information avec lesquels nous devions interfaçer et les solutions techniques que nous avons mis en œuvre.

S.I.	Solution technique	Type de traitement
La poste	Expeditor Inet	Génération de fichier et upload serveur distant
Chronopost	Chronopost Scrut	Génération de fichier et upload serveur distant
Adipa	-	Génération de fichier et upload serveur distant
Cegid	-	Génération de fichier et download sur poste client
BNP Paribas	ATOS SIPS (Mercanet)	Execution d'un binaire sur le serveur
Tecdoc	Tecdoc Webservices	Web services SOAP et sauvegarde en base

Table 1 - Interactions avec les systèmes d'information des partenaires

Je n'ai volontairement pas intégré les SI suivants dans le mémoire, car leur interfaçage n'a pas été réalisé pour la livraison du site, il s'agissait en fait d'évolutions : Paybox Payment Services, logistique Doyen, plateforme SMS OVH, livraison en point Ici Relais – Exapaq.

IV.2 La gestion de la qualité

La « gestion de la qualité » est une composante importante dans **la gestion de projet informatique**. La nature abstraite du projet informatique, sa pluridisciplinarité ainsi que la quantité importante des acteurs sont des facteurs qui rendent le projet informatique **complexe**. Il est alors primordial de définir des règles pour garantir (ou au moins essayer de garantir) la livraison d'un produit conforme aux attentes du client : c'est la « gestion de la qualité ».

Tout ceci reste cependant très abstrait. En tant que chef de projet, j'ai du répondre à des questions comme : **Comment se traduit la « gestion de la qualité » dans le cadre d'un projet e-commerce tel que Pièces Avenue.com** ? Quelles sont les règles, actions à mener, organisation à définir, méthodes à employer pour garantir la qualité d'un site e-commerce tel que Pièces Avenue ? Que dois-je faire pour garantir à mon client que son projet va être une réussite ?

Ce chapitre du mémoire va décrire la démarche mise en œuvre pour répondre à ces questions dans le plan d'assurance qualité. Dans un premier temps, je vais exposer mon retour d'expérience sur **la rédaction du PAQ**. Le second point que j'aborderais ici relève plus de l'aspect « communication » et concerne **la gestion des évènements**, ainsi que les flux entre le client et l'équipe de développement. Enfin, je souhaite également mettre en avant un concept que j'estime fondamental dans la gestion de projet et qui est **la gestion de la configuration** avant de finir par un **bilan** de la qualité sur Pièces Avenue.

IV.2.1 Le plan d'assurance qualité

Le plan d'assurance qualité a été le premier livrable soumis à notre client. Il s'agit d'un dossier qui a pour but de spécifier l'ensemble des dispositions spécifiques ainsi que les règles de travail à suivre dans le cadre du projet.

Le premier objectif de cette démarche est **d'assurer la qualité du produit** ou du service fourni. Cet objectif est souvent mis en exergue car il s'agit de l'argument principal pour la validation du PAQ auprès du client. En effet, le client doit y trouver son compte. Il doit voir **un intérêt à adopter le plan de qualité** et à faire l'effort de suivre les règles qui y sont définies. Généralement, convaincre le client de l'importance du plan de qualité n'est pas une chose aisée. Mais dans mon cas, mon client avait aussi une expérience dans le domaine de l'informatique et le message est passé assez facilement.

Le second objectif est de **rassurer la direction** de la maîtrise d'œuvre. Les méthodes et processus mis en place permettent de formaliser, de définir et de respecter un cadre ainsi que des règles de travail qui ont un **effet positif indéniable sur le projet** et notamment sur le respect des délais et donc des coûts de production. Enfin, cela permet aux développeurs de **travailler sereinement** et d'être ainsi beaucoup plus productifs.

Le PAQ est donc un document important pour tous les acteurs du projet. Mais quelles sections doit-il contenir ? J'ai choisi de le décomposer en plusieurs parties :

- La démarche du projet
- Les livrables

- L'organisation
- Les livraisons et installations
- Le bilan projet
- La gestion de la configuration
- La gestion des événements

IV.2.1.1 La démarche du projet

Dans cette partie, j'ai essayé d'expliquer au client les différentes **étapes du projet** dans l'ordre chronologique. J'ai pensé que le client devait pouvoir se situer par rapport à l'avancement du projet. Il devait avoir une vision des étapes accomplies et de celles restantes. C'est donc une vision à très haut niveau. Cela permet surtout de former et d'informer le client sur le déroulement d'un projet web.

IV.2.1.2 Les livrables

Cette section du plan d'assurance qualité a consisté à **définir exhaustivement la liste des livrables qui vont être soumis** au client ainsi que le contenu de chacun d'entre eux. J'y explique par exemple ce qu'est un zoning, une arborescence ou encore un cahier des charges. Encore une fois, il s'agit d'une section qui a pour objectif de former et informer le client.

A la relecture du document, je me rends compte que j'ai oublié de spécifier dans le plan d'assurance qualité un livrable absolument crucial qui est le planning détaillé. Car un planning a été effectivement livré au client malgré qu'il ne soit pas mentionné dans le PAQ...

IV.2.1.3 L'organisation

L'objectif de cette section était de **définir l'organisation du projet**. En effet j'ai pensé qu'il était important pour le client et pour l'équipe de développement de savoir comment nous allions nous organiser et travailler dans le cadre du projet.

Cette section contient un organigramme général ainsi qu'une description des rôles et des coordonnées des personnes figurant sur cet organigramme.

J'ai considéré que trois de ces rôles étaient particulièrement importants et devaient être décrits. Le premier rôle est celui de **la maîtrise d'ouvrage**, qui est représentée par Stéfan EDET. Initialement, le rôle de chef de projet de la maîtrise d'oeuvre devait être scindé en deux : il devait y avoir un **responsable de la direction artistique** et un responsable pour le **reste**. L'expert DA devait prendre mon relais pour les phases d'étude du zoning, de la création des maquettes et du logotype.

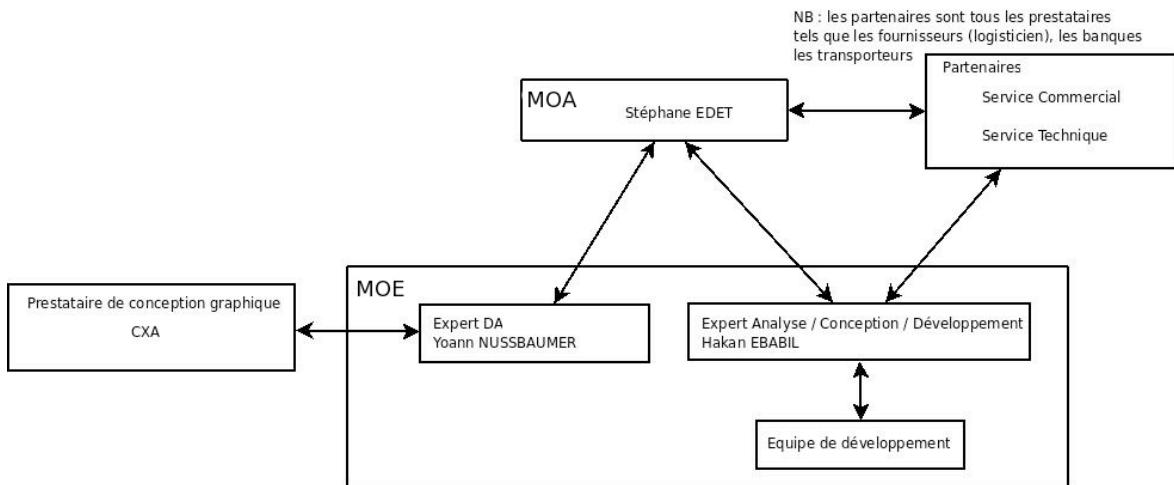


Figure 33 - Organisation de l'équipe projet

A la base, nous avions convenu que Yoann devait se charger de la direction artistique du projet (l'aspect graphique) car il avait beaucoup plus d'expérience et une bien meilleure sensibilité graphique que moi. Cependant, j'ai finalement endossé ce rôle car il n'avait pas le temps de s'en charger. Il m'a quand même beaucoup conseillé et aidé lors de ces phases. Ce fut très instructif pour moi qui n'avais eu que très peu de cours à ce sujet à l'IUT ou au CNAM. J'étayerai mon retour d'expérience dans les parties correspondantes du mémoire.

Dans ce diagramme, les flèches représentent les axes de **communications** possibles entre les acteurs. Ces communications ainsi que les **canaux** à utiliser ont été décrites dans le PAQ. J'ai spécifié deux types de communications : une communication **interne** au sein de la maîtrise d'oeuvre et une communication que j'ai qualifiée **d'externe** et qui s'opère entre la maîtrise d'oeuvre et la maîtrise d'ouvrage. J'ai volontairement omis de décrire les communications avec les autres partenaires car je pensais qu'il serait vraiment difficile de formaliser celles-ci et de les faire accepter par des sociétés telles que Chronopost. J'ai pensé que c'était à moi de m'adapter au mode de travail de chacun de nos partenaires. La seule règle que je m'étais fixée était que je souhaitais n'avoir qu'un seul interlocuteur par partenaire.

Cette section aborde enfin le **fonctionnement de l'équipe projet** et en particulier les réunions hebdomadaires de l'équipe de développement, les réunions de suivi avec le client tous les quinze jours et enfin les formations. Je me suis rendu compte que le rôle de chef de projet implique la rédaction d'un nombre très élevé de rapport : comptes rendus de réunions mais aussi notes, dossiers, etc. Ce n'est d'ailleurs pas toujours évident de trouver le temps pour leur rédaction...

Je pense avec le recul que cette partie aurait pu être améliorée avec une **approche plus systémique** et **mieux formalisée**. Il aurait aussi fallu décrire l'organisation ainsi que les moyens de communication étape par étape. En clair, pour chaque phase du projet (réécriture du cahier des charges, arborescence,...) j'aurais pu spécifier des diagrammes de *use-case* ou des modèles conceptuels de traitements pour décrire l'organisation...

IV.2.1.4 Les livraisons et installations

Cette section du PAQ traite de la livraison et de l'installation des fichiers sur les serveurs du client. Tout d'abord, j'ai fait la **distinction entre environnement de développement et environnement de production**. Pour chacun de ces deux environnements, j'ai décrit les règles à respecter pour la mise en ligne.

Par exemple, la mise en ligne sur l'environnement de production devait s'accompagner d'un ordre de mise en ligne signé par la maîtrise d'ouvrage. Ce document devait contenir la liste des modifications qui allaient être mise en ligne. Une autre règle que j'ai défini était que les mises en production ne pouvaient avoir lieu que les mercredis et les jeudis. Il s'agit en fait d'une clause courante, notamment au sein des banques pour la mise en ligne des paiements en ligne. Il faut en effet éviter de mettre en ligne la veille d'un jour non travaillé, car en cas de problème personne ne peut intervenir pour effectuer une correction.

IV.2.1.5 Le bilan projet

L'amélioration continue est au centre de toutes les démarches qualité. J'ai rédigé le PAQ en gardant ce principe en tête.

En plus d'un **brainstorming interne** à l'équipe de développement, j'ai mentionné dans le plan d'assurance qualité qu'il serait possible de réaliser des **interviews avec la maîtrise d'ouvrage**, dans le but d'identifier les points qui peuvent être améliorés pour les prochains projets. Cependant, si une réunion interne de débriefing a bien eu lieu, je n'ai pas eu le retour de notre client par manque de temps mais surtout d'implication.

IV.2.2 La gestion des évènements

Un autre aspect du plan d'assurance qualité que je souhaite développer ici est ce que l'on appelle la « gestion des évènements ». Mais avant d'entrer dans le vif du sujet, il me semble important de définir ce qu'est la « gestion des évènements » et ce que l'on entend par le terme « évènement ».

En réalité, les évènements que je vais mentionner dans ce chapitre sont les **risques du projet**. On pourrait donc tout aussi bien parler de gestion des risques. Mais quels sont ces risques ? J'en ai identifié trois grands, mais il en existe probablement beaucoup plus. L'idée était de me limiter à ceux dont la probabilité d'occurrence était la plus élevée. Les trois risques majeurs que j'ai identifié sont donc : les **incidents** logiciels et matériels, les demandes **d'évolution** et enfin, les **retards**.

L'objectif de cette section du PAQ est donc de décrire les actions à mener dans le cas d'un incident logiciel (ex : **que fait-on lorsqu'on découvre un bogue ?**), d'un incident matériel (ex : **que fait-on lorsqu'un disque dur tombe en panne ?**), en cas de demande de modification ou d'évolution et enfin, elle décrit ce qu'il faut faire si l'équipe détecte et anticipe un retard.

Je voudrais mettre l'accent ici sur la gestion des incidents et des évolutions car c'est (malheureusement) un aspect que je considère comme commun dans les projets informatiques. Très souvent, il existe pour chaque projet un référentiel où l'on recense les

bogues trouvés. Les utilisateurs / testeurs renseignent cette base et les développeurs prennent ou non en compte les demandes de correction.

Toutefois, pour Pièces Avenue, j'ai imaginé une autre manière de fonctionner, avec non pas un, mais deux outils de *tracking* de bogues. Le premier outil était une interface web (l'application *Active Collab*) que l'on a mis à disposition de notre client afin qu'il puisse renseigner les bogues qu'il trouve ainsi que ses demandes d'évolution. Seul le client et moi avions accès à cet outil en ligne. Le second outil était l'interface Trac et était à destination des développeurs. En tant que chef de projet, je me chargeais d'analyser les demandes du client sur *Active Collab*, puis de faire le lien dans Trac.

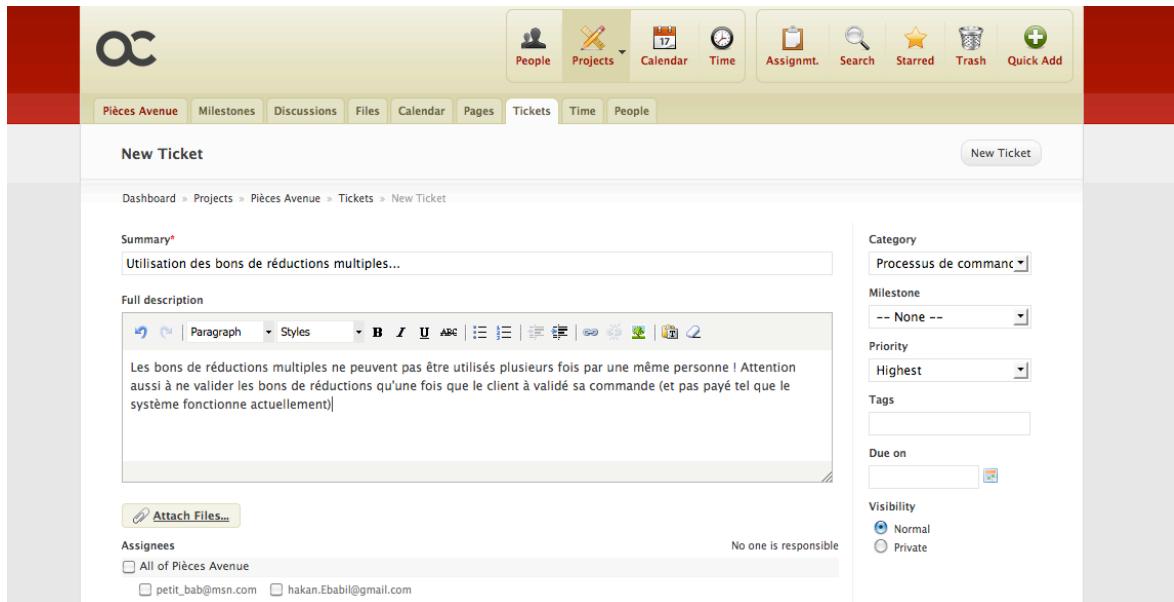


Figure 34 - Interface de l'outil Active Collab

Pourquoi avoir séparé les deux ? A priori on pourrait imaginer que c'est une perte de temps ! Mais en fait il y a plusieurs avantages à fonctionner de cette manière...

Le premier est qu'il permet de **filtrer les demandes** du client. En effet, il se peut très bien par exemple que le client considère une fonctionnalité comme étant un bogue alors qu'il s'agit d'une demande d'évolution qui ne figure pas dans le cahier des charges. Il se peut aussi qu'en fait ce bogue n'en soit pas un. Dans d'autres cas, la requête du client peut être contradictoire avec une autre clause du cahier des charges. Au final, il est possible de se retrouver dans des situations qui n'ont absolument aucun sens. Cette séparation permet d'ignorer ce type de demandes ou essayer d'en savoir plus avec le client.

Mon second objectif était de **maîtriser le travail de l'équipe de développement**. En choisissant ce que j'intègre depuis *Active Collab* dans Trac, je maîtrisais les demandes de modification ou de correction. Car avant le projet Pièces Avenue, nous avions réalisé plusieurs sites où le client était en relation directe avec le développeur et nous avions de grandes difficultés à contrôler et maîtriser le temps de travail des développeurs. Souvent, les délais n'étaient plus respectés à cause des abus. Ici, il n'y a qu'un seul interlocuteur, le chef de projet et il décide de ce qui doit être fait ou non, et de qui doit le faire.

Le troisième avantage est qu'il permet de **traduire la demande** du client. En effet, celle-ci n'est pas toujours claire et il faut donc la retranscrire dans un langage plus compréhensible et proche de celui utilisé par les développeurs. L'idée ici est avant tout de faire gagner du temps à l'équipe de développement en « déchiffrant » (le terme employé ici n'est vraiment pas exagéré...) les demandes du client.

Enfin, le dernier aspect positif de cette séparation est de pouvoir proposer à chaque acteur une **interface qui lui est adaptée** :

Active Collab est un outil intéressant pour le client car son ergonomie, son interface graphique ainsi que ses fonctionnalités sont étudiés pour lui. Nous verrons d'ailleurs plus loin qu'il y a certaines autres fonctionnalités d'*Active Collab* que nous avons utilisé pour communiquer et travailler en collaboration avec notre client.

Trac est un outil beaucoup plus adapté aux développeurs. En effet, il intègre par exemple un explorateur du *repository* Subversion du projet. Il permet aussi de créer des tickets ou d'écrire des commentaires en rapport avec une révision spécifique. Enfin, il intègre aussi un wiki, très utile par exemple pour communiquer entre développeurs.

Ticket	Résumé	Composant	Version	Jalon	Type	Sévérité	Propriétaire	Status	Created
#10144	make it easier to switch off commit messages in timeline	general		next-major-0.1X	defect	trivial		new	25 avr. 2011
#9567	Script to update Trac copyright date in license header of source files	project		not applicable	task	critical		new	14 août 2010
#4431	wiki_to_wikidom	wiki system	0.10.3	0.14-wikiengine	enhancement	critical	cboos	assigned	20 déc. 2006
#886	Add support for Master tickets	ticket system	devel	next-major-0.1X	enhancement	major	nkantrowitz	reopened	5 nov. 2004
#10485	Genshi UnicodeDecodeError error while rendering template (unknown template location) while rendering a ticket web page	web frontend	0.12-stable	0.12.3	defect	major	cboos	new	1 déc. 2011
#10458	Commithook is aborted when trac cache warning appears	ticket system	0.11.5		defect	major		new	11 nov. 2011
#10267	Version controle diff of large text file	version	0.12.2	next-major-0.1Y	defect	major		new	15 juill.

Figure 35 - Interface de l'outil Trac

IV.2.3 La gestion de configuration

Le dernier aspect du plan d'assurance qualité que je vais traiter dans ce mémoire est la gestion de configuration. Dans un premier temps, il me semble important de **définir** ce qu'est la « gestion de configuration ». Puis je détaillerais les différents éléments qui ont fait **l'objet de cette activité** ainsi que les raisons qui m'ont poussé à les choisir plutôt que d'autres. Enfin j'exposerai les **avantages** de cette activité, en particulier dans le cas du projet comme Pièces Avenue.

Mais tout d'abord, qu'est-ce que la « gestion de configuration » ? Le principe de cette activité est de **gérer l'historique des versions et des modifications d'un système**. En clair et adapté au projet Pièces Avenue il s'agit de mettre en œuvre des outils et des

méthodes afin de pouvoir tracer l'ensemble des modifications apportés au site web. Mais pas uniquement. Il me semblait évidemment nécessaire d'historier le **code** du site. Mais j'ai également voulu garder un historique des **documents du projet** et des **éléments graphiques**.

IV.2.3.1 Les documents du projet

Il est utile de garder une trace des modifications des documents du projet pour plusieurs raisons. La première et la plus importante est qu'elle permet de **résoudre les conflits** en remontant à la source des décisions. En effet, il m'était déjà arrivé qu'un client fasse preuve de mauvaise foi (par exemple, en refusant d'admettre qu'il avait oublié telle ou telle fonctionnalité dans le cahier des charges) et cela engendre des tensions qui ont un impact négatif sur le projet. Je souhaitais avoir une sécurité à ce niveau, même si nous n'avons rien vécu de tel avec Pièces Avenue car il y avait entre le client et le prestataire un vrai climat de confiance.

Validations			
Nom	Date	Validation (O/N)	Commentaire
Edet Stéfan	08-01-09	N	Voir mail retour.

Modifications			
Version	Date	Etat	Description
1.0		Rédaction	Document initial
1.1	06-01-09	Modification	Correction de la navigation produit TECDOC
1.2	09-01-09	Modification	Ajout des produits « <u>options</u> », formatage <u>références commandes</u> , calculs de prix de produits, traitement des n° de suivis de colis, devis personnalisés, modes de livraison: ajout du retrait en point de vente, modification relances, modification des lignes de factures.
1.3	26-01-09	Modification	Suppression des petites annonces,

Figure 36 - Historique des modifications du cahier des charges

Parmi les documents du projet, on distingue notamment les comptes rendus de réunions, les notes et dossiers, les ordres de mise en ligne mais aussi l'ensemble des documents « livrables » du projet tel que le cahier des charges, l'arborescence, etc. J'ai également défini une **règle de nommage** pour ces documents afin de les classer et de les retrouver plus facilement. Celle ci était composée de la manière suivante :

[type de document]_[date au format aaaa-mm-jj]_[version majeure]_[version mineure]_[première lettre du prénom][nom].[extension du fichier]

Le document nommé **CDC_20111226_1_1_HEBABIL.odt** était donc le cahier des charges du projet à la date du 26 décembre 2011, dans sa version 1.1 et corrigée par H. EBABIL. Avec le recul, je pense que dernière partie du nom du fichier (le nom et prénom) n'était pas une indication pertinente car, en tant que chef de projet, j'étais le seul à effectuer des modifications sur les documents du projet.

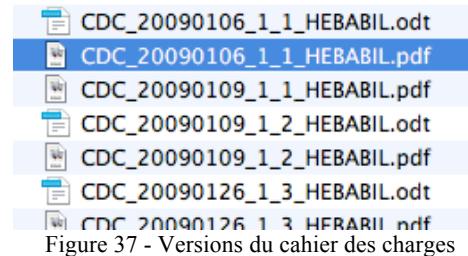


Figure 37 - Versions du cahier des charges

Les fichiers étaient ensuite téléchargés dans notre outil *Active Collab* afin de les partager avec le client pour validation.



Figure 38 - Logo Active Collab

IV.2.3.2 Les éléments graphiques

J'ai souhaité garder une trace des modifications sur l'ensemble des éléments graphiques. Il s'agissait d'avoir un historique des maquettes et des logos. J'ai laissé notre partenaire CXA s'occuper de cette tâche.

IV.2.3.3 Le code source de l'application

Le code source du site Pièces Avenue a été géré à l'aide de l'outil Subversion. J'avais déjà utilisé Subversion dans le cadre d'autres projets, donc je le connaissais déjà bien et l'équipe était également familiarisée à cet outil.

Le premier avantage à utiliser un outil de contrôle de code source tel que Subversion est qu'il permet de retracer l'historique des modifications apportées aux fichiers. Lorsque l'on travaille à plusieurs sur un projet, il arrive aussi qu'un développeur écrase par erreur un fichier. Cela est synonyme de perte de temps puisqu'il faut réécrire le code érasé. SVN permet de faire face à ces situations et de revenir très rapidement à une ancienne révision.

J'utilise aujourd'hui encore le même dépôt subversion pour la mise à jour du site car même si je travaille seul sur le projet, Subversion me donne la possibilité de revenir sur une ancienne révision en cas de soucis lors d'un archivage. C'est pour moi une garantie.

Une des particularités que j'ai souhaité mettre en place était la **mise à jour des serveurs de développement et de production à l'aide de l'opération SVN « update »**. Ces opérations étaient effectuées via une interface web SVN qui est encore utilisée aujourd'hui.



Figure 39 - Interface Web SVN

De cette manière, non seulement j'avais un **historique des fichiers** et des modifications apportées, mais il m'était aussi possible de **savoir à tout moment quelle révision était installée** sur les serveurs de développement et de production... Le fonctionnement était le suivant :

L'équipe de développement disposait d'un accès en lecture et écriture sur le dépôt Subversion. Une fois leurs travaux terminés, les développeurs avaient la possibilité d'archiver les modifications via l'opération SVN « commit ».

Ce commit effectué, les mises à jour sur les serveurs se faisaient en **trois étapes** :

1. Se rendre sur l'interface web SVN,
2. Sélectionner le projet souhaité,
3. Cliquer sur le bouton « update ».

Dans un premier temps, nous mettions à jour le serveur de développement, pour que le client puisse visualiser et valider les modifications.

Ensuite, une fois l'aval du client obtenu, la mise à jour du serveur de production était effectuée sur le même principe. Cette méthode a grandement simplifié les mises en ligne puisqu'elle nous a notamment évité les transferts manuels de fichiers, une opération qui à mon avis introduit un risque qui n'est pas négligeable et qui peut être évité dans ce *process*. D'ailleurs aucun développeur n'avait d'accès FTP au serveur. A mon sens, cette méthode

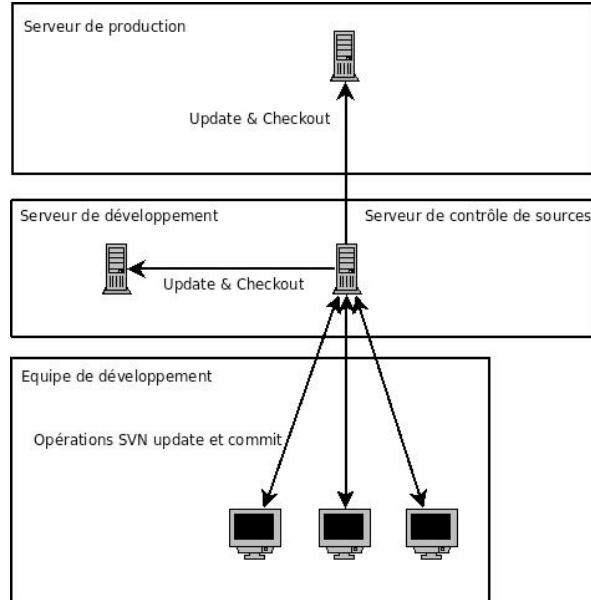


Figure 40 – Architecture des serveurs de développement et de production

est vraiment un gage de sécurité et de qualité et aujourd'hui je ne souhaiterais pas travailler autrement.

IV.2.4 Bilan

Je me suis rendu compte que le respect du plan d'assurance qualité nécessite une **implication de tous les participants** du projet et que cette implication doit être **constante**, tout au long du projet. Il s'agit vraiment de faire un **effort supplémentaire** par rapport au travail normalement fourni. C'est pourquoi il n'est pas évident de convaincre tous les acteurs, et en particulier le client. En effet, il faut que le PAQ soit le moins intrusif possible pour lui, surtout s'il n'est pas sensibilisé à l'informatique et à la gestion de projet.

En fait, la qualité est un aspect que j'aurais aimé approfondir, mais malheureusement le budget ainsi que les délais ne m'ont pas permis d'y passer plus de temps. Le PAQ du projet Pièces Avenue est **mon premier plan d'assurance qualité**. C'est probablement la raison pour laquelle il y manque beaucoup de choses. J'ai notamment oublié certains livrables, certaines étapes du projet... Cependant, je pense que j'ai été sur la bonne voie et que le contenu, même non complet, était cohérent.

D'une manière générale, nous avons suivi le plan d'assurance qualité. Les développeurs ont parfaitement joué le jeu avec l'utilisation de l'outil Trac. Le client a aussi utilisé *Active Collab* pour visualiser ses fichiers, spécifier des bogues, etc. Cet aspect de « **gestion de l'évolution et des corrections** » ainsi que les méthodes de **mises à jour** des serveurs étaient vraiment **LES deux points positifs de la gestion de la qualité**.

Mais je ne peux pas nier non plus qu'il me soit arrivé de faire quelques **entorses** au plan de qualité, principalement dû à un manque de temps. Par exemple, il m'est arrivé « d'oublier » d'écrire certains compte rendu de réunion. J'ai également envoyé et fait valider très peu d'ordres de mise en ligne... Au lieu de ça, le client validait oralement la mise en ligne ou en production. Parmi tous les projets auxquels j'ai participé, Pièces Avenue est vraiment celui qui m'a permis de me rendre compte de la difficulté qu'il peut y avoir à faire adhérer le client aux règles du plan de qualité et à lui imposer la rigueur nécessaire au bon déroulement du projet.

Au lancement du site, nous n'avons eu à déplorer **aucun bogue critique**, aucun gros souci et le lancement s'était réalisé dans le calme. Je pense sincèrement que sans ce document et sans l'effort consentit par l'équipe pour le respecter, nous n'aurions pas vécu le même lancement.

Comparé aux autres petites agences web, Skalpel a donc fait des efforts sur la qualité du produit livré et sur les processus de travail. Ce qui est assez paradoxal, c'est que la qualité devrait être un levier pour la réduction des coûts. Car comme cela est enseigné au CNAM, la non-qualité a bel et bien un impact économique. Mais à court terme, la qualité se traduit aussi par un prix plus élevé que celui de nos concurrents car elle implique des efforts supplémentaires à l'ensemble des acteurs du projet. Et si cette différence est légitime, elle est difficilement acceptée par le client.

Je pense que c'est une des raisons pour laquelle nous avions des difficultés à obtenir des projets. D'autant plus en période de crise où le facteur « coût » est logiquement plus important pour le client. Cependant, je n'ai vraiment aucun regret et aujourd'hui encore

j'estime que l'instauration d'une gestion de la qualité était absolument nécessaire, surtout pour un projet comme Pièces Avenue.

IV.3 La rédaction du cahier des charges fonctionnel

Le cahier des charges est le second livrable du projet après le PAQ. Il a pour objectif de définir les **spécifications** de base d'un produit ou d'un service. Dans le cadre du projet Pièces Avenue.com, il s'agissait donc de rédiger un document qui regroupe les caractéristiques du site Internet à réaliser. Encore une fois et avant toute chose, il est absolument primordial de comprendre ce que l'on entend par « caractéristique » et par « spécification ». En particulier quelle est la signification de ces deux termes dans le cadre d'un projet web e-commerce.

Dans cette partie du mémoire, je vais donc détailler le **contenu** du cahier des charges de Pièces Avenue ainsi que les choix que j'ai fais dans le cadre de sa rédaction. Je vais aussi décrire la **méthodologie** employée pour son écriture et mon **retour d'expérience**.

Nous verrons qu'il s'agit d'un document très important dans un projet puisqu'il revêt, au même titre que l'ensemble des autres documents de conception (arborescence, zoning, etc.), un **caractère contractuel** entre le prestataire et le client. De plus, c'est un document qui est diffusé à l'ensemble des acteurs du projet. Il joue donc un rôle de **référentiel** pour tous ces acteurs.

L'échec des projets informatiques est très souvent imputé à **l'absence** ou à la **mauvaise spécification** du cahier des charges. Il peut aussi arriver que ce dernier **évolue** au cours du temps ou encore qu'il soit **mal interprété**, ce qui aura probablement aussi pour conséquence un échec du projet. C'est pourquoi il faut soigner son écriture, ne pas laisser d'ambiguité.

IV.3.1 Le contenu

Que doit contenir précisément le cahier des charges ? Cette question, qui est à première vue anodine, est la première que je me suis posé avant d'entamer la rédaction du cahier des charges. Une sorte de remise en question dans la mesure où j'avais déjà suivi des cours du CNAM à ce sujet.

La première chose que j'ai fait a été de rechercher sur Internet et d'essayer de trouver un plan type pour mon cahier des charges. Il existe en effet sur le réseau Internet des exemples très détaillés. Toutefois, je me suis très vite rendu compte que chaque projet était différent et donc impliquait une approche particulière. Par exemple, un projet de site web est différent d'un projet pour application embarquée et donc le contenu et le plan du cahier des charges diffère forcément. D'autant plus que je n'étais pas du tout convaincu par certains de ces exemples.

Il me fallait donc un plan plus abstrait. Je me suis alors tourné vers le plan type proposé par l'AFNOR. Mais ce dernier n'était pas satisfaisant car il incluait trop d'aspects que je ne jugeait pas utile, qui avaient déjà été traités auparavant (comme par exemple, les études économiques ou celles de rentabilité) ou qui n'avaient tout simplement pas leur place ici.



Figure 41 - Logo AFNOR

J'ai donc décidé **moi même** du contenu du cahier des charges. J'ai estimé qu'il était important de mettre l'accent sur deux aspects. D'une part, **la description du métier** du

client et d'autre part **les spécifications fonctionnelles**. La description du métier est ce qui va permettre de concevoir les **structures de données** du projet. En effet, cette description intègre les règles métier propre au projet du client, mais aussi les concepts ainsi que les liaisons qui peuvent exister entre ceux-ci. Les spécifications fonctionnelles permettent quant à elles de définir ce qui va être fait de ces données (**les traitements**) via différents outils : algorithmes, diagrammes de cas d'utilisation, logigrammes, grafcet etc.

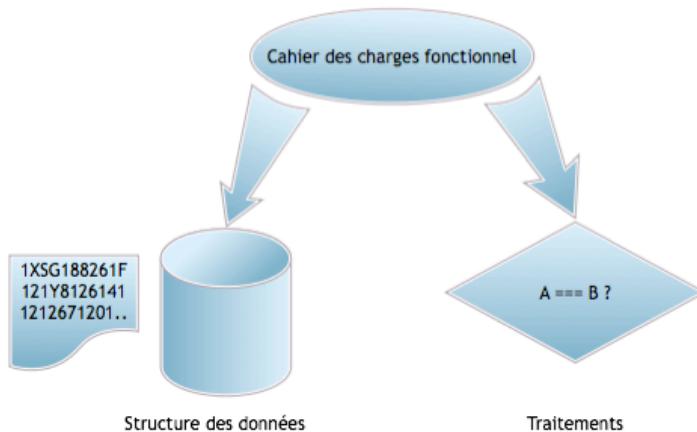


Figure 42 - Contenu du cahier des charges

Certains sites (et même l'AFNOR) indiquent que le cahier des charges doit proposer des solutions à la problématique. Je ne suis pas d'accord. Pour moi, le cahier des charges fonctionnel doit rester à un certain niveau d'abstraction par rapport au problème.

A titre d'exemple, je suis contre les cahiers des charges qui intègrent des captures d'écran. Mon point de vue est que ce document doit se limiter (*autant que possible*) à la définition des fonctionnalités attendues et une capture d'écran n'a donc pas sa place ici. De même le référencement, l'ergonomie, le graphisme ne sont pas des fonctionnalités.

Pour illustrer mon point de vue, je vous propose un extrait du cahier des charges ci-après:

Inscription au site

L'inscription des internautes est unique et valable pour l'ensemble du site. A l'inscription, l'internaute spécifie uniquement son adresse e-mail. A la validation de son inscription, l'internaute se voit attribuer un mot de passe généré automatiquement. Il reçoit ce mot de passe par courrier électronique.

Perte de mot de passe

Si l'internaute perd son mot de passe, il peut en recevoir un nouveau par e-mail en entrant son login. Cette opération modifie son mot de passe et lui en affecte un nouveau généré de manière aléatoire.

Dans l'exemple de la page précédente, vous remarquerez que j'ai uniquement décrit la fonctionnalité, sans mentionner ni l'ergonomie (« clic sur tel ou tel bouton »), ni le graphisme (« bouton rouge avec un effet d'ombre »). A la limite, il est presque possible d'adapter ce cahier des charges à une application autre que web (c'est ce que j'appelle l'abstraction). Je pense que tout cela vient bien après la rédaction du cahier des charges. Il ne s'agit pas non plus d'aller trop loin dans la fonctionnalité. En effet, le cahier des charges ne doit pas être trop précis, car le temps de rédaction s'en trouverait allongé.

Dans tous les cas, j'ai l'intime conviction que la conception d'un site web est une activité qui doit se faire par **étapes**. Chaque étape permet d'apporter sa pierre à l'édifice. Le cahier des charges fonctionnel décrit donc les fonctionnalités, sans entrer dans leurs implémentations et sans aborder les autres aspects du projet.

IV.3.2 La méthodologie d'écriture

Puisque nous savons ce que doit contenir le cahier des charges, je vais maintenant expliquer comment je l'ai écrits. Deux chapitres importants vont composer cette section. D'une part, j'évoquerai **l'approche descendante**, qui est la clef de voute du cahier des charges. D'autre part nous verrons quels ont été les **langages et outils** utilisés dans le cadre de sa rédaction.

IV.3.2.1 L'analyse descendante

J'ai en fait utilisé cette méthode sans le savoir. C'était un peu comme une découverte pour moi, même si d'autres y avaient pensé bien avant. Mais qu'est ce donc que cette méthode de « **l'analyse descendante** » ?

De tous temps, les Hommes ont essayé de construire des choses complexes en partant de choses simples. C'est l'analyse montante.

L'analyse descendante est le raisonnement complètement inverse dans laquelle on part d'un système complexe que l'on **décompose progressivement en sous-systèmes** moins complexes. Par exemple, on devait trouver sur le site Internet Pièces Avenue, une foire aux questions ainsi qu'une boutique en ligne. Puis, en creusant un peu plus le sous système boutique en ligne, on se rend compte qu'il doit y avoir un catalogue produit et ainsi de suite.

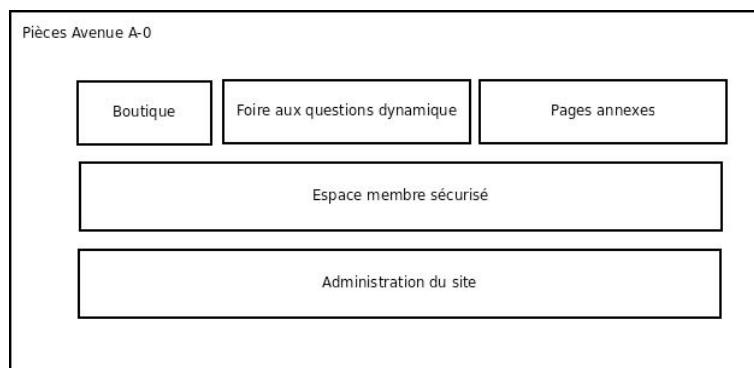


Figure 43 - Analyse descendante, niveau A-0

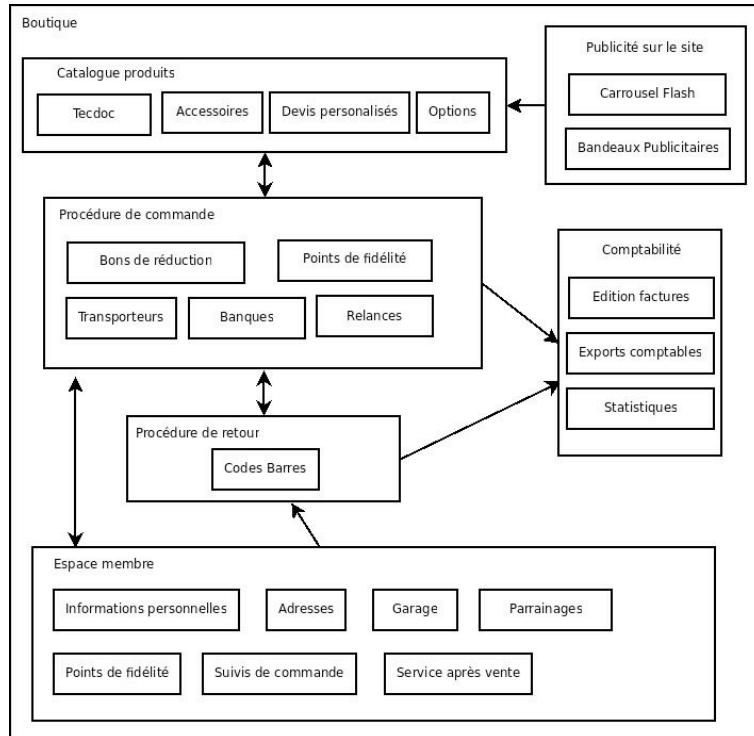


Figure 44 – Analyse descendante, niveau A-1

Cette méthode est efficace car elle permet de ne rien oublier malgré la complexité du système. Elle garantit la **complétude** du cahier des charges. De plus, cette structure m'a aidé à ne pas faire de redites. Chaque fonctionnalité était spécifiée dans un seul et unique chapitre du cahier des charges. Encore une fois, pour ne pas inutilement allonger les délais, il faut éviter de descendre de manière trop profonde dans cette étude. Enfin, dernière remarque : Le **plan** du cahier des charges fonctionnel est basé sur les différents **blocs de fonctions** que l'on retrouve sur ces schémas.

IV.3.2.2 Le langage et les outils

Les deux précédentes parties abordent le contenu ainsi que la structure du cahier des charges. Un seul point reste encore à éclaircir : de quel manière ce contenu est-il rédigé et quels sont les outils sur lesquels je me suis basé dans le cadre de sa rédaction.

Sans entrer dans l'aspect abstrait de la définition d'un langage (syntaxe, sémantique, etc.), je voudrais simplement signaler ici qu'il y a plusieurs types de langages à la disposition du rédacteur. Chacun ayant ses avantages et inconvénients.

Par exemple, les langages **formels ou mathématiques** permettent d'éviter les mauvaises interprétations et permettent de spécifier plus simplement les tests, mais ils sont aussi beaucoup plus lourds à lire. D'autant plus que le cahier des charges est un document lu par des acteurs qui n'ont pas nécessairement les connaissances requises pour comprendre ces langages.

Il y a aussi les langages **naturels** comme le français. Ces langages sont évidemment plus simples à comprendre pour le lecteur et toucheront un lectorat plus vaste. Cependant, ils peuvent aussi être source de mauvaise interprétation. De plus, il y a des risques que le cahier des charges ne soit pas cohérent. Entre ces deux extrêmes, il y a les **langages naturels structurés**. Dans ce type de langage, les phrases sont construites sur la base de gabarits. C'est évidemment plus pénible à lire, mais ce langage est un compromis entre les deux précédents. Par exemple : « l'objet X a N référence à l'objet Y. L'objet Y peut ou non avoir une référence à l'objet X ».

Dans le cas de Pièces Avenue, j'ai principalement utilisé le langage naturel. Par petites touches j'ai aussi intégré le langage naturel structuré. Mais en plus de ces langages, je me suis aussi aidé de schémas.

Le premier type de schéma utilisé est le **diagramme de cas d'utilisation** ou « use-case » issus de la méthodologie UML. Les use cases ont principalement été utilisé pour spécifier les droits que peuvent avoir les internautes sur le front office ou des administrateurs dans le back office.

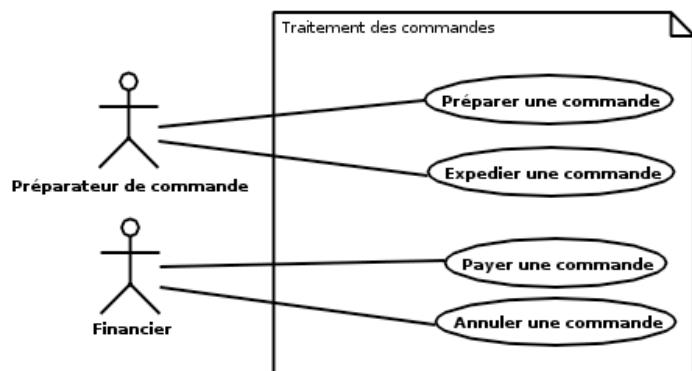


Figure 45 - Diagramme de cas d'utilisation

Le second type de schéma utilisé est le **logigramme**. Ce type de schéma permet de spécifier les enchainements dans les actions, ainsi que les différents états dans lequel peut se trouver le système.

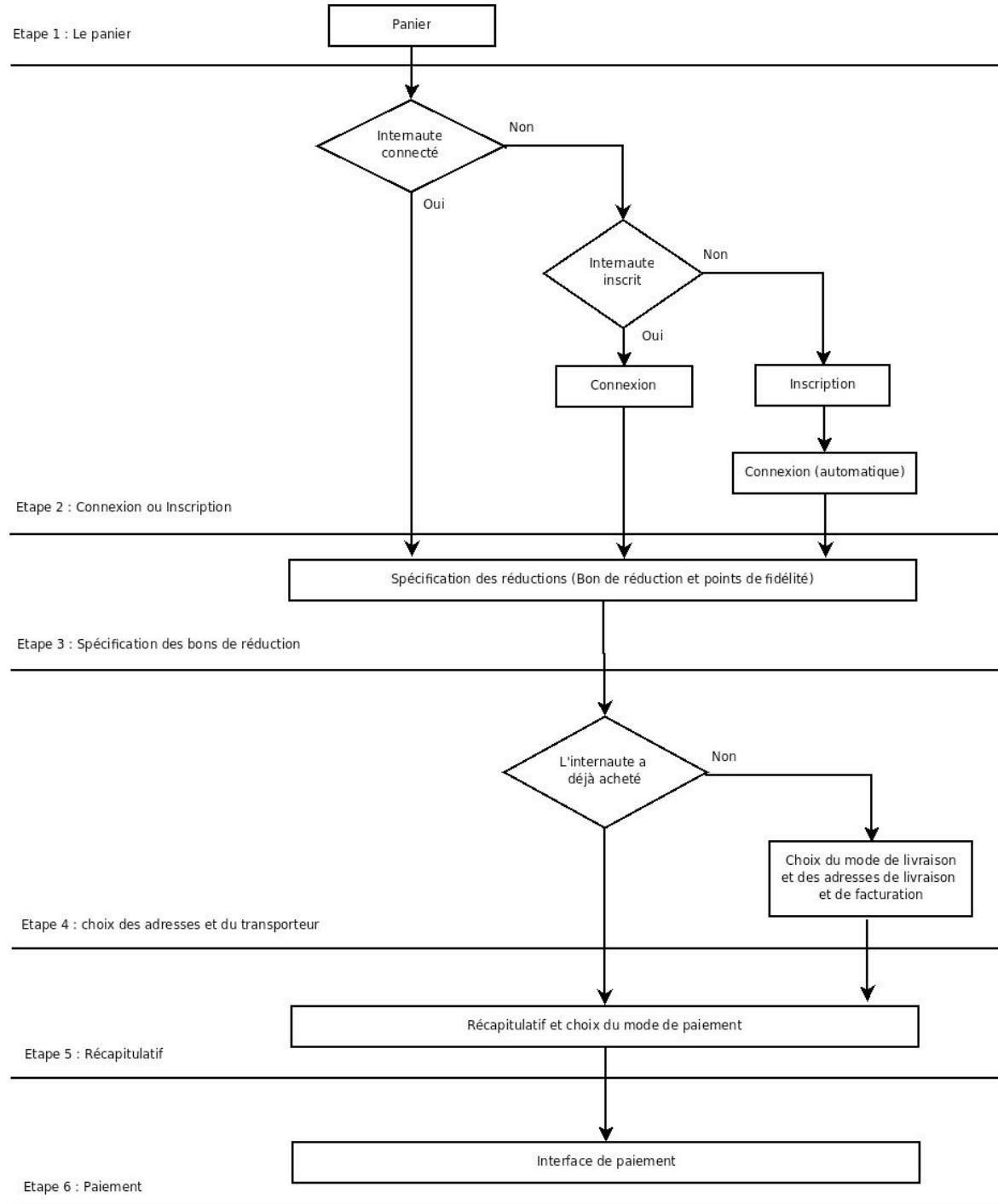


Figure 46 – Logigramme

IV.3.2.3 La communication

La communication joue un rôle central dans la rédaction du cahier des charges. En tant que chef de projet, la première chose que j'ai eu à faire a été de comprendre le métier de mon client. Puis, il fallait analyser la demande, comprendre les besoins qui ont été formulés par Stéfan et faire un *feedback* pour s'assurer que l'on a bien compris.

Quelques fois, j'ai même du aider le client à définir ses propres processus métier. Par exemple, le processus de retour des produits a été étudié en collaboration avec Stéfan. Il m'a tout d'abord expliqué son besoin. Ensuite, j'ai proposé un fonctionnement qui répond à ce besoin et nous l'avons validé ensemble.

L'étape suivante est de formaliser et de décrire dans le cahier des charges ces besoins. Lors de la rédaction du cahier des charges, il faut bien avoir à l'esprit que l'objectif principal est de transmettre un message au lecteur.

L'essentiel ici est donc de **s'adapter au lecteur** pour que celui-ci comprenne ce message et qu'il n'y ait aucune ambiguïté, quelque soit le plan du cahier des charges, son contenu, son style d'écriture ou le nombre de pages du document. Il faut donc prendre en considération son lectorat et faire des choix de rédaction en fonction de ce dernier.

Tout ceci a impliqué de nombreux échanges (téléphone, mail, etc.) mais aussi plusieurs itérations pour arriver au cahier des charges définitif.

IV.4 La conception d'une arborescence

L'**arborescence** permet de définir et d'organiser les pages du site Internet, de manière logique et hiérarchisée. Elle découle du cahier des charges fonctionnel en reprenant l'ensemble des fonctionnalités décrites dans celui-ci et en les regroupant sous forme d'un schéma détaillé. L'arborescence du site permet également de lier les pages entre elles et ainsi de créer le maillage interne du site (liens hypertextes entre les pages). La conception de l'arborescence fait partie de la conception « **ergonomique** » du site.

Cette partie du mémoire va présenter l'arborescence sous trois angles : le premier axe est l'énoncé des **facteurs** qui impactent la conception de l'arborescence. L'idée est de visualiser les paramètres que j'ai pris en compte lors de la conception de l'arborescence. Le second chapitre sera une mise en pratique et une **illustration** de l'utilité de l'arborescence sur un exemple qui, à mon sens, est intéressant et qui est le processus de commande. Enfin, je terminerai par mon **retour d'expérience**, ce que j'ai appris et ce que j'aurais pu améliorer dans cette phase du projet.

IV.4.1 Les facteurs clés

Dans le cadre du projet Pièces Avenue, l'ergonomie du site et la hiérarchisation des pages ont été dictés par plusieurs facteurs. Le premier facteur est **l'utilisation des outils** tels que Tecdoc, qui ont imposé un fonctionnement particulier, notamment pour la recherche et l'affichage des produits.

Le second paramètre à prendre en compte est l'aspect « **mercatique** » du site Internet. En effet, le site Internet est avant tout une boutique. Le client doit trouver rapidement son produit et pouvoir facilement commander.

Du point de vue de l'enchaînement des pages et du cheminement de l'internaute sur le site, il faut donc respecter les « **us et coutumes** » des sites de commerce en ligne. L'objectif ici est de ne pas brusquer ou déstabiliser l'internaute dans son achat. En particulier, cela est valable dans le cas du processus de commande où l'internaute a souvent des habitudes qu'il souhaite retrouver sur le site Pièces Avenue.

Ces trois facteurs m'ont guidé dans la conception de l'arborescence, qui est disponible dans le dernier chapitre de cette partie du mémoire.

IV.4.2 Exemple du processus de commande

Je vais développer l'exemple du processus de commande dans ce chapitre car je pense qu'il illustre bien l'intérêt de l'arborescence dans un projet web. Ce processus est décrit de la manière suivante dans le cahier des charges : la première étape est celle du panier. Puis l'internaute doit se connecter ou, s'il n'a pas de compte, s'inscrire. Ensuite il peut définir les bons de réduction et spécifier s'il souhaite ou non utiliser ses points de fidélité. Puis il doit choisir son mode de livraison, définir ses adresses de livraison et enfin choisir son mode de paiement et régler sa commande.

L'objectif de l'arborescence est de spécifier **sur quelle page se trouvera chacune de ces fonctionnalités**. Par exemple, nous pouvons regrouper le choix du mode de livraison et la spécification des adresses sur une seule page (ce qui semble assez logique finalement), ou

choisir d'en faire deux pages web distinctes. Il en est de même pour la spécification du bon de réduction et des points de fidélité. Ici, j'ai choisi de procéder en cinq étapes :

1. Le panier : une page où l'utilisateur peut visualiser et modifier le contenu de son panier,
2. La connexion / inscription : la seconde étape implique que l'internaute s'inscrive ou s'identifie avec ses paramètres de connexion,
3. La spécification des réductions : l'internaute est connecté et peut maintenant spécifier son bon de réduction et choisir ou non d'utiliser ses point de fidélité,
4. Le choix du mode de livraison : la quatrième étape est celle du choix du transporteur et des adresses de livraisons et de facturation,
5. Le récapitulatif : il s'agit de la dernière étape où l'internaute valide sa commande et choisit son mode de règlement.

Avec l'arborescence, nous allons donc un cran plus loin dans la définition du besoin. Le projet devient ainsi moins « abstrait ».

IV.4.3 Retour d'expérience

Mon seul regret concernant cette arborescence est que nous n'avons pas pu, pour des raisons techniques, mettre en place la navigation alternative TecDoc. En effet, à la base il devait y avoir une navigation principale par marque de véhicules et une navigation secondaire par famille de pièces.

La première navigation proposait l'affichage des pièces auto en sélectionnant d'abord un véhicule et en naviguant ensuite dans les familles de pièces. En opposition, la seconde navigation devait permettre de sélectionner d'abord la famille de pièces voulue puis de choisir dans un second temps le véhicule. Nous avons pu implémenter la première recherche sans problème, mais la seconde n'était pas réalisable car aucune méthode du web service TecDoc ne permettait de lister les familles de pièces sans avoir au préalable sélectionné de véhicule.

Je pense toutefois que cette arborescence était conçue de manière **optimale**. En effet, il respecte les contraintes de navigation imposées par les outils tel que Tecdoc, mais aussi les règles préétablies dans le domaine du commerce en ligne. Aujourd'hui encore, si je devais refaire cet arborescence, le résultat serait peu ou prou le même.

IV.4.4 Le résultat

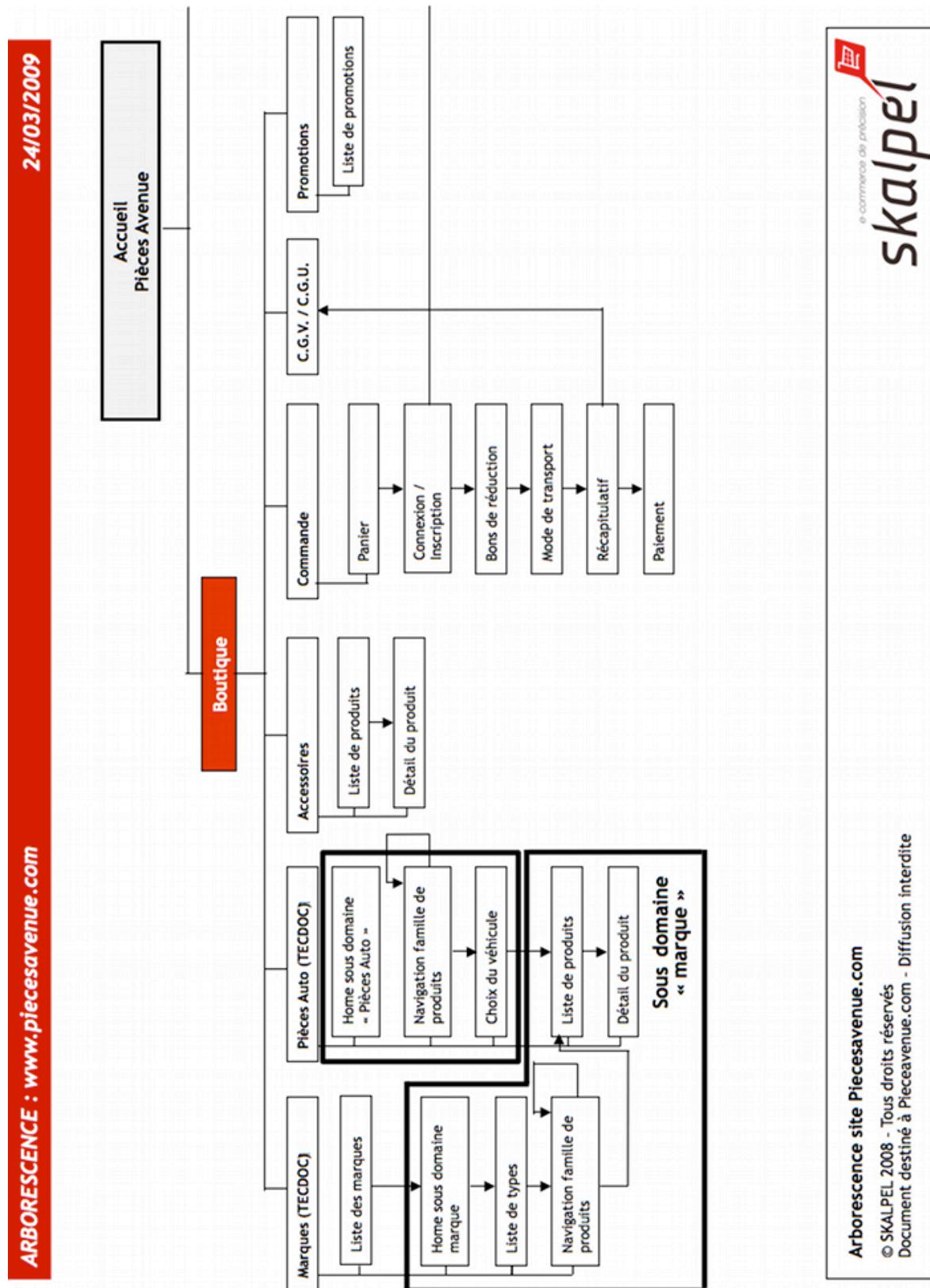


Figure 47 - Arborescence 1/2

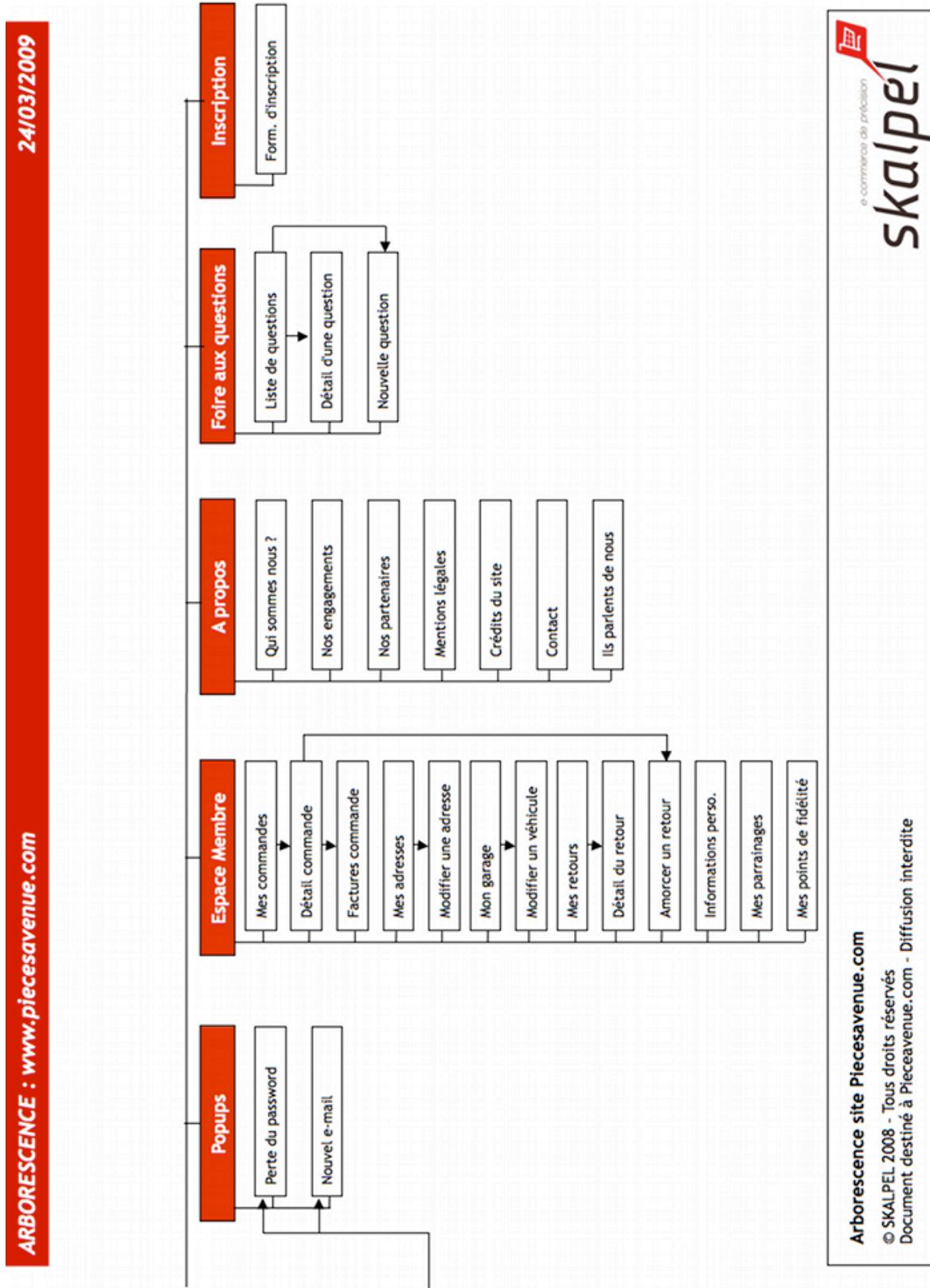


Figure 48 - Arborescence 2/2

Arborescence site Piecesavenue.com
 © SKALPEL 2008 - Tous droits réservés
 Document destiné à Pieceavenue.com - Diffusion interdite

IV.5 L'annexe à l'arborescence

L'annexe à l'arborescence est un document qui complète l'arborescence en décrivant le **contenu de chacune des pages** mentionnées dans l'arborescence. Cette description a deux grands intérêts : le premier est qu'il s'agit d'une étude approfondie portant sur **l'ergonomie du site**. Le second intérêt de l'annexe à l'arborescence est qu'il **sert de guide pour les développeurs**. Je vais développer ces deux axes dans les chapitres suivants du mémoire.

IV.5.1 Une étude ergonomique

L'annexe à l'arborescence est avant tout une étude ergonomique. En effet, il contient notamment les différents éléments graphiques tel que les types de contrôles à utiliser dans les formulaires, la liste des informations qui seront présentes sur chaque page et éventuellement l'agencement général de la page ou les différents styles à maquetter.

Le panier

La page de panier ne comporte pas la colonne de droite présente sur les autres pages. Le panier présente sous la forme d'un tableau à cinq colonnes la liste des produits achetés par l'internaute. Les colonnes sont :

- L'intitulé du produit
- Le prix unitaire
- La quantité achetée (avec la possibilité de la modifier)
- Le prix total
- La possibilité de supprimer le produit du panier

Une zone spécifique devra être prévue pour les produits de type «`option`» sous ce tableau. Les options sont représentées graphiquement sous la forme d'un intitulé, d'un prix et d'une case à cocher. Sous cette zone sera présente le bloc des produits d'impulsion. Enfin, nous afficherons le bouton de passage à l'étape suivante.

Figure 49 - Page du panier dans l'annexe à l'arborescence

En ce sens, l'annexe à l'arborescence joue un **rôle important dans la conception ergonomique** du projet. D'ailleurs, ces informations sont notamment utilisées ensuite lors de la création des maquettes. L'exemple ci-dessus illustre le cas du « panier ».

Comme pour l'arborescence, cette description apporte aux acteurs une vision plus **concrète** du produit final.

IV.5.2 Un guide pour le développement

Mais outre cet aspect ergonomique, l'annexe à l'arborescence contient aussi, sous forme d'un tableau, des informations qui sont destinées plus spécifiquement à la maîtrise d'œuvre, aux développeurs et aux graphistes.

En particulier, il contient les **métadonnées** : contenu de la balise *title*, des mots clés et de la description de la page ainsi que l'url de la page. C'est ce que j'appelle la conception « SEO » du projet. L'objectif ici est de prendre en compte le **référencement** du site dès la phase de conception. Ces informations seront notamment ensuite utilisées par les développeurs lors de la réalisation du front office.

La seconde catégorie d'information contenue dans ce tableau relève encore une fois du « développement ». Il s'agit de la méthode qu'il faudra **coder** dans le *framework Serum*,

ainsi que la nécessité d'être ou non connecté sur le site pour visualiser la page. Ce sont des informations qui guident et facilitent le travail du développeur.

Enfin, la dernière ligne de ce tableau permet de spécifier si oui ou non le zoning ainsi que la maquette de la page doivent être réalisés. En effet, le contrat avec le prestataire graphique CXA portait sur quatorze maquettes. Or il y a sur le site bien plus de quatorze pages. Il faut donc sélectionner les quatorze pages que l'on souhaite faire maquetter. Ci dessous, le tableau pour la page du panier :

URL de la page	Ex: http://www.pieces-avenue.com/commande/
Méthode (Serum)	Orders~Cart:index
Titre	Commandez vos pièces auto sur PiecesAvenue.com, spécialiste de la piece auto sur Internet
Mots clefs	Commande, pieces auto,
Description	Commandez vos pièces auto sur PiecesAvenue.com, spécialiste de la piece auto sur Internet
Connexion nécessaire	NON
Zoning + Maquette	OUI

Table 2 - Métadonnées dans l'annexe à l'arborescence

IV.5.3 La rédaction de l'annexe à l'arborescence

Le choix de la rédaction d'une annexe à l'arborescence est partit d'une **initiative personnelle**. En effet, j'ai pris la décision d'écrire ce document car j'estimais qu'il manquait un support faisant le lien entre l'arborescence, qui est une simple cartographie des pages du site Internet, et les *zonings* qui, nous le verrons plus loin dans ce mémoire, se rapprochent plus des maquettes.

De plus, j'ai voulu faire de ce document un **support** supplémentaire pour les acteurs de la **maîtrise d'œuvre**. Il fournit des informations auxquelles les développeurs n'auront pas besoin de penser lors de l'intégration du site tel que les urls des pages, leurs métadonnées ou les méthodes à coder. Mon expérience en tant que développeur web m'a poussé à préciser ces informations dans l'annexe, car il s'agit de questions récurrentes lors du développement d'un site Internet. J'ai voulu de cette manière préparer le travail de l'équipe de développement

Enfin, il s'agit d'un document qui sera très largement **exploité par les prestataires graphiques** lors de la création des maquettes. Encore une fois, c'est mon expérience qui m'a montré qu'il fallait guider les graphistes dans la réalisation des maquettes car ceux-ci n'arrivent pas toujours à travailler uniquement à partir du cahier des charges et de l'arborescence.

L'annexe à l'arborescence est un document que j'ai rédigé en grande partie **seul**. Je me suis cependant appuyé sur les connaissances de notre expert en référencement Maxime Castanié pour valider mes choix sur les métas et les urls des pages.

IV.6 Le zoning

Les termes « *zoning* » ou encore « *wireframe* » sont souvent évoqués lors des phases de conception ou d'initialisation des sites web. L'objectif de cette partie du mémoire est de **présenter** dans un premier temps ce concept et de montrer en quoi cette étude est fondamentale avant de se lancer dans la réalisation graphique. Le second chapitre va **illustrer ce concept** dans le cas de la page d'accueil et du processus de commande. Enfin, j'exposerai mon **retour d'expérience** à ce sujet.

IV.6.1 Définition du Zoning

La conception du zoning est l'étape qui suit l'analyse de l'arborescence et la rédaction de l'annexe à l'arborescence. Le **zoning** est un document qui permet de **représenter de façon schématique les gabarits des principales pages d'un site**. Comme pour l'arborescence, le zoning est à classer dans la catégorie de « conception ergonomique » du site Internet.

Chaque page est découpée en zones de contenus, ce qui permet d'avoir un aperçu assez précis du site: hiérarchisation des différents textes, description des liens principaux et secondaires, description de la navigation... Le zoning est constitué à partir des pages présentes dans l'arborescence du site et en prenant en compte l'annexe à l'arborescence ainsi que l'ensemble des fonctionnalités définies pour ces pages.

Un des intérêts majeur du zoning est qu'il permet au client d'avoir un bon **aperçu** de son site avant que le projet n'entre dans la phase de réalisation des maquettes. Ainsi, il peut réagir, proposer sa vision et il est possible de **rectifier certains aspects de l'ergonomie**. Cela évite par la suite la multiplication des versions de maquettes et donc, une perte de temps et d'argent.

IV.6.2 Mise en pratique

Durant cette étape de la phase de conception, Yoann Nussbaumer m'a apporté son aide pour l'élaboration du *zoning* de la page d'accueil ainsi que pour ceux du processus de commande. Il s'agit en fait de deux éléments stratégiques pour un site de commerce électronique.

En effet, la « *homepage* » représente la vitrine de la boutique et, de ce fait, il faut y apporter un soin particulier pour que l'internaute soit rassuré dès son arrivée sur le site. Le processus de commande est quant à lui souvent comparé à un entonnoir ou « *tunnel de conversion* ».

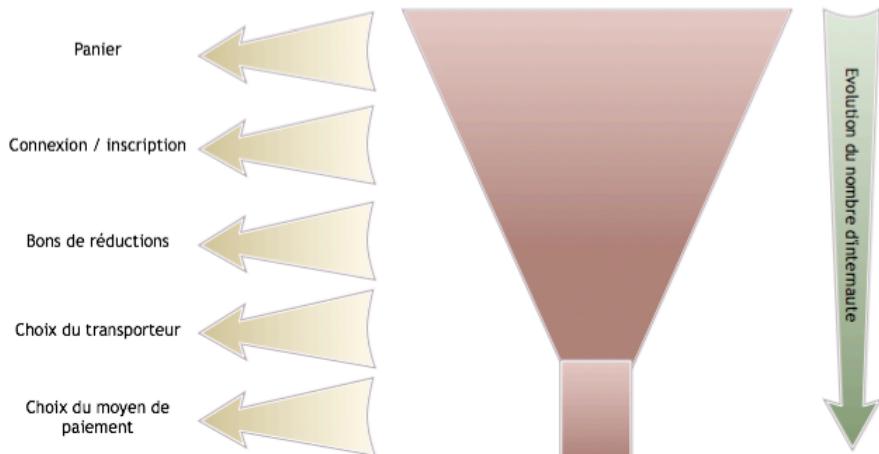


Figure 50 - Entonnoir de conversion

Le nombre d'internautes décroît à chaque étape du processus de commande. Par exemple, si 100 internautes arrivent sur la page du panier, il n'y en aura peut être plus que 70 à s'inscrire et 50 à renseigner ses adresses et choisir un transporteur. Au final, peut être que seul deux ou trois internautes effectueront une commande et la règleront.

Il faut donc bien étudier l'ergonomie de ce processus afin que la base de l'entonnoir soit la plus large possible et qu'il y ait un maximum de commandes. Dans ce domaine, Yoann avait plus d'expérience que moi. Il m'a donc aidé et j'ai beaucoup appris grâce à lui.

La page suivante présente le *zoning* de la page d'accueil dans sa troisième (et dernière) version ainsi qu'une capture d'écran du site prise au moment de la rédaction de ce mémoire. Vous remarquerez les similitudes entre les deux visuels... malgré les modifications apportées depuis plus de deux ans maintenant.

PiecesAvenue.com
Experts en Pièces et Services auto

Assistance téléphonique gratuite
0800 222 222
du lundi au vendredi de 7 à 19h

Zone de promotion des avantages à commander chez Pieces Avenue

Retour produits | Suivi de commande | Besoin d'aide ?
Panier ▾ Terminer ma commande
2 articles **198,00 €**

Accueil Pièces auto Accessoires & Entretien ▾ Promotions Services ▾ Nos engagements

Slideshow promotionnel

1 | 2 | 3 | 4 | 5 | 6

Nos pièces auto par catégories Voir toutes les catégories

Suspensions
Amortisseurs avant ou arrière
Triangle ou bras de suspension
Silent bloc de suspension
Roulement de roue à l'unité
Barre stabilisatrice

Suspensions
Amortisseurs avant ou arrière
Triangle ou bras de suspension
Silent bloc de suspension
Roulement de roue à l'unité
Barre stabilisatrice

Suspensions
Amortisseurs avant ou arrière
Triangle ou bras de suspension
Silent bloc de suspension
Roulement de roue à l'unité
Barre stabilisatrice

Suspensions
Amortisseurs avant ou arrière
Triangle ou bras de suspension
Silent bloc de suspension
Roulement de roue à l'unité
Barre stabilisatrice

Suspensions
Amortisseurs avant ou arrière
Triangle ou bras de suspension
Silent bloc de suspension
Roulement de roue à l'unité
Barre stabilisatrice

Votre garage
Ajoutez votre véhicule et naviguez plus efficacement dans notre catalogue produit

Espace client
Email :
Mot de passe : Valider
Mot de passe perdu ?

Paiement
100% sécurisé

CB débitée à l'expédition

Expédition en 24h à 48h

Satisfait ou Remboursé

Inscrivez-vous pour recevoir nos bons plans par email et profitez d'offres exclusives ! Ok

Zone promotionnelle

Ca se passe sur le blog ...
Un plateau de qualité, un stock de champions
Voir tous les messages du blog

Nos dernières annonces de véhicules d'occasions

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Aston Martin DBS 1 000Km / 2000 € Détail de l'annonce

Déposez gratuitement votre annonce
Consulter toutes les annonces

Nos partenaires

Logos partenaires banque + expédition

Vos Achat
Pièces auto
Accessoires & entretien
Promotions
Nos engagements

Espace client
Suivre ma commande
Retourner un produit
Besoin d'aide ?
Conditions générales de vente

Nos Services
annonces véhicules d'occasion
Questions / réponses mécaniques
Centres de montage
Devis personnalisé

A propos de Pieces Avenue
Qui sommes nous ?
Nos engagements
Nos partenaires
Mentions légales

Votre avis nous intéresse !
Blog
Contactez-nous
Crédits du site

Moto85 / Tops Cars / Skalpel / Moto85 / Tops Cars / Skalpel / ... Plan du site / Flux RSS

Figure 51 - Zoning de la page d'accueil (V3)

<ul style="list-style-type: none"> <input type="checkbox"/> Allumage / Préchauffage <input type="checkbox"/> Courroies / Distribution <input type="checkbox"/> Circuit Carburant <input type="checkbox"/> Combustion / Injection 	<ul style="list-style-type: none"> <input type="checkbox"/> Triangle, Rouleau, Roulement, Moyeu <input type="checkbox"/> Entrainement des roues <input type="checkbox"/> Entrainement des essieux 	<ul style="list-style-type: none"> <input type="checkbox"/> Chauffage - Climatisation <input type="checkbox"/> Chauffage / Ventilation <input type="checkbox"/> Climatisation
Essuyage <ul style="list-style-type: none"> <input type="checkbox"/> Éléments d'essuyage 	Chauffage - Climatisation <ul style="list-style-type: none"> <input type="checkbox"/> Chauffage / Ventilation <input type="checkbox"/> Climatisation 	Embrayage (Kit, Volant) <ul style="list-style-type: none"> <input type="checkbox"/> Embayage
Echappement <ul style="list-style-type: none"> <input type="checkbox"/> Echappement 	Freinage <ul style="list-style-type: none"> <input type="checkbox"/> Systèmes de Freinage 	Eclairage <ul style="list-style-type: none"> <input type="checkbox"/> Eclairage
Électricité <ul style="list-style-type: none"> <input type="checkbox"/> Pièces Électriques 		

Sélectionnez votre véhicule...

Sélectionnez votre véhicule ci-dessous pour afficher l'ensemble de ses pièces auto.

Marque	<input type="text" value="Tous les éléments"/>
Modèle	<input type="text" value="Tous les éléments"/>
Type	<input type="text" value="Tous les éléments"/>

[Afficher la liste des pièces pour mon auto](#)

[Cliquez ici pour définir votre véhicule à l'aide de votre carte grise](#)

... ou recherchez une pièce auto par sa référence

Entrez la référence ou un équivalent de la pièce recherchée pour l'afficher.

<input type="text" value="Référence"/>	
--	--

[Afficher la liste des pièces pour mon auto](#)



Kit embrayage jusqu'à
-50%

Vos pièces auto sur PiècesAvenue.com

[Voir toutes les marques](#)

Mon garage



ALFA ROMEO 159
 Type : 1.9 JTDM 16V (150 CH)

[Changer de véhicule](#)

Livraison

en 48H
7,00€
7,00€ de colis par jour

en 24H
10,00€

Service Technique



Besoin d'un conseil ?
Tél. 02 40 720 720
Numeros non surtaxés

Point de Retrait



Gratuit

Nantes - Treillières

Bons plans

Inscrivez-vous pour recevoir nos bons plans par email et profitez d'offres exclusives !

Sur notre blog

[Kit Embrayage Luk - 4 Pièces](#)

[Visitez notre blog](#)

Votre avis nous intéresse



Aidez-nous à améliorer notre service en nous donnant votre avis !

[Que pensez-vous de PiècesAvenue ?](#)

Nos partenaires





Figure 52 - Capture d'écran de la page d'accueil du site

IV.6.3 Mise en pratique

Hormis ces deux points que nous avons traité ensemble avec Yoann, j'ai élaboré l'ensemble des autres zonings seuls. Je dois dire que ce fut très instructif pour moi. En effet, l'ergonomie est un point que l'on n'aborde pas ou peu à l'Université. Il m'a fallu donc principalement me baser sur l'expérience acquise sur le terrain : en naviguant sur les sites ou au travers des différents sites réalisés.

Je pense que la conception des zonings est une tâche complexe. Premièrement parce qu'elle implique une part de créativité que tout le monde ne possède pas forcément. De plus, cette inventivité doit absolument être maîtrisée. Il faut savoir la doser et apporter une certaine originalité (ce qui permettra de différencier le site de ses concurrents, produire une *plus value*) sans pour autant déstabiliser l'internaute en ignorant ses habitudes ou en omettant les codes déjà établis des sites e-commerce.

IV.7 Le pré-briefing graphique

Le pré-briefing fait partie de la conception « **graphique** » du projet. C'est un document destiné avant tout aux graphistes. En effet, il permet à ces derniers de mieux connaître l'entreprise et de cerner les objectifs du client. De cette manière, il est possible de créer une identité graphique en adéquation avec ces objectifs.

Concrètement, il s'agit d'un **formulaire** composé d'une douzaine de questions auquel le client doit répondre. Ces questions sont les suivantes :

1. Décrivez en quelques phrases ce qu'est votre entreprise et ses valeurs.
2. Décrivez vos **produits et services**.
3. Brossez le portrait de votre **client type** : âge, sexe, catégorie socioprofessionnelle.
4. Quels sont les **axes de communication** définis par votre service marketing ou votre agence ? Si ceux-ci sont actuellement inexistant, vers quels axes voudriez-vous tendre ?
5. Sur quels **supports** allez-vous communiquer : web, papier, tv, et dans quelles proportions (75% web, 25% papier par exemple) ?
6. Quelle sera la **portée de votre communication** : locale, régionale, nationale, internationale, multinationale (déclinaisons différentes par pays ?)
7. Avez-vous des **exemples d'entreprises** dont la communication graphique et/ou logo vous inspirent, ou dont il faudrait au contraire se démarquer ?
8. Dans le cas d'une refonte graphique, énumérez les **points forts et les points faibles** de l'existant.
9. Quelles sont les **couleurs** qui plaisent ?
10. Pour résumer, donnez une liste d'une dizaine de **mots-clés** que vous souhaiteriez retrouver dans votre charte graphique/logo à venir.
11. Quels seront les intitulés exacts à faire figurer sur votre **logo** ?
12. Éventuelles **remarques/constraintes** spécifiques complémentaires à préciser.

Vous remarquerez que certaines de ces questions trouvent déjà leur réponse dans le cahier des charges du projet. Cependant, l'objectif ici est d'avoir la vision originale du client, exprimée dans ses propres termes.

Ces questions ne sont donc pas dénuées d'intérêt. Elles permettent ensuite de **guider** les graphistes dans la phase de réalisation des maquettes et du logo si celui-ci n'existe pas encore (c'était le cas du projet Pièces Avenue). En effet, ces derniers n'adopteront par exemple pas la même **approche** si le site s'adresse à des seniors ou s'il s'adresse au contraire à un public plus jeune. De même, ils essayeront d'utiliser au maximum les tons de couleurs spécifiés dans ce pré-brief graphique ou encore de prendre en compte les remarques et contraintes du client.

3. Brossez le(s) portrait(s) de votre « client type » : sexe, âge, CSP, etc

La cible principale est le grand public désireux d'entretenir son automobile

Cependant il ne faut pas exclure les professionnels (artisans, PME)

Le public est essentiellement masculin et majeur.

Figure 53 - Question n°3 du pré-briefing graphique

Ce formulaire est un formulaire type que nous utilisions quelque soit le projet. Il existait donc déjà avant le projet Pièces Avenue. En tant que chef de projet, j'ai simplement **soumis** ce document à mon client et réceptionné la **version complétée**, que j'ai transféré à notre partenaire graphique.

IV.8 L'estimation et le planning

La dernière étape dans le cadre de la conception du site Pièces Avenue est l'estimation et le planning de la phase de réalisation. L'estimation et le planning se déroule en quatre étapes que je vais détailler dans la suite de ce mémoire.

La première étape est le **découpage des tâches**. Une fois ce découpage effectué, il faut définir **l'enchaînement** logique entre ces tâches, **les estimer et affecter** les ressources. Enfin, la dernière étape consiste à **ajuster** la durée des tâches en fonction des ressources associées.

IV.8.1 Le découpage en tâches

Comme la rédaction du cahier des charges, l'estimation est une activité complexe car la réalisation d'un site Internet implique, comme je l'ai mentionné auparavant, plusieurs corps de métier et donc un nombre élevé et varié de tâches.

Pour simplifier mon travail, j'ai eu l'idée d'utiliser la décomposition en sous système fonctionnel que j'avais réalisé auparavant pour l'écriture du cahier des charges. Ce travail de décomposition m'a beaucoup aidé dans le découpage en tâches car chaque bloc fonctionnel du cahier des charges se transformait en tâche. Le visuel ci-dessous illustre les tâches dans le cas du catalogue produit.

!	▼ Back Office	56d 5h
.1	▼ Catalogue produit	23d 6h
.1.1	▼ TECDOC	13d 6h
.1.1.1	Module consignes	4h
.1.1.2	Module type de promotions	4h
.1.1.3	Module logisticiens	4h
.1.1.4	Analyse Services Web & tests de récupération de données	1d
.1.1.5	Module de marques / Modèles / Types (Webservices)	3d 6h
.1.1.6	Module fournisseurs (Webservices)	2d 4h
.1.1.7	Module familles (Webservices)	2d 4h
.1.1.8	Module produits (Webservices)	2d 4h
.1.2	▼ Accessoires	4d
.1.2.1	Module marque	1d
.1.2.2	Module familles	1d
.1.2.3	Module produits	2d
.1.3	Options (API)	2d
.1.4	Devis personnalisé	4d

Figure 54 - Découpage en tâches

En réalité, dans un premier temps j'ai choisi de décomposer le projet en trois grandes phases, chacune représentant un étape importante dans la chronologie du projet :

Il y a tout d'abord le développement de **l'outil de gestion de contenu (Serum)**. Puis, dans un deuxième temps, le développement du **back office** et enfin celui du **front office**. Une fois ce premier découpage effectué, j'ai intégré l'ensemble des aspects fonctionnels du cahier des charges dans chacune des trois sous parties. En effet, si l'on garde l'exemple du catalogue produit, les tâches relatives au back office (administration des produits, des catégories, etc.) sont très différentes de ce qu'il va falloir faire sur le front office (affichage des listes de produits, requêtes TecDoc etc.) Il est donc important de les séparer car il s'agit bel et bien de tâches distinctes.

Mais le cahier des charges n'est pas le seul document sur lequel je me suis basé. En fait, **tous les documents de conception jouent un rôle dans cette étape d'estimation**. Par exemple, l'arborescence permet de savoir combien d'écrans il va falloir découper en HTML puis intégrer sur le front office, alors que le zoning permet d'avoir une idée sur la difficulté de découpage de chacun de ces écrans.

▼ Front Office	50d 2h
► Zonings + dév maquettes CXA	7d 6h
▼ Découpage XHTML	10d 4h
Découpage XHTML - 30 maquettes	10d 4h

Figure 55 - Utilité de l'arborescence dans le découpage et l'estimation

IV.8.2 L'enchaînement des tâches

Une fois l'ensemble des tâches définies, il faut spécifier leur enchainement.

Pour cela, il y a deux paramètres à prendre en compte. Le premier est l'aspect logique de cet enchainement, qui est imposé par le métier de « réalisation de site Internet ». Par exemple, il semble assez évident qu'on ne peut pas effectuer le découpage HTML d'un écran avant de l'avoir maquetté et conçu graphiquement. De même, on ne peut pas imaginer intégrer cet écran avec le moteur *Dwoo* si on ne l'a pas auparavant découpé.

Le second paramètre est la date de livraison du site, ou *deadline*. Dans le meilleur des mondes, cette date devrait être définie lors de l'estimation et du planning. Or, le domaine du web est très concurrentiel et impose de prendre puis assumer des décisions très rapidement dans la vie du projet. L'ingénieur technico-commercial Julien Lavault a pris cette décision bien avant le planning et il a donc fallu que je respecte cette date. Pour cela, il faut essayer de travailler en parallèle sur plusieurs tâches, ce qui impose aussi d'avoir plus de ressources à disposition.

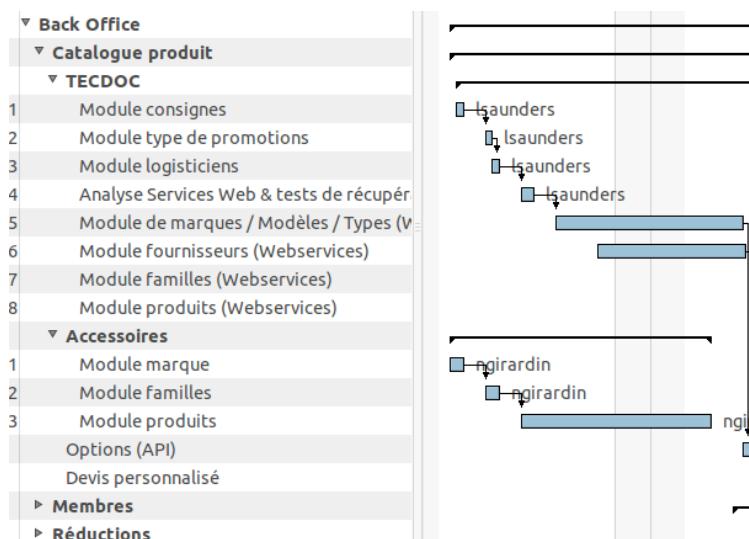


Figure 56 - Travaux menés simultanément

L'exemple ci-dessus montre les travaux menés simultanément sur les différents types de produit du catalogue Pièces Avenue.

IV.8.3 L'estimation

Une fois ces tâches mises bout à bout, le planning commence à prendre forme. Cependant, il manque encore une donnée importante. Il faut maintenant estimer la durée que prendra chacune d'elles. Pour cela, je me suis essentiellement basé sur mon **expérience**. Le fait d'avoir décomposé les tâches m'a une fois de plus beaucoup aidé ici. Je me suis rendu compte que plus la tâche était décomposée et plus l'estimation se révélait être proche de la réalité.

J'ai également utilisé des **méthodes et formules empiriques**, notamment pour l'estimation du découpage HTML des écrans. Une des formules que j'ai pris l'habitude d'utiliser est de dire qu'il faut une journée par type d'écran et deux heures pour chaque déclinaison de cet écran.

Par exemple, sur Pièces Avenue il y a deux types d'écrans : la page de panier, qui fournit un cadre graphique pour toutes les étapes du processus de commande et la home page qui sera ensuite déclinée sur l'ensemble des autres pages du site (puisque seule la zone de contenu va différer d'une page à l'autre). Ces deux types d'écrans étaient déclinés en 28 autres pages sur le site. Au total, pour Pièces Avenue, le découpage prendra donc 2×8 heures + 28×2 heures = 72 heures.

Il faut bien avoir à l'esprit que cette estimation ne prend pas encore en compte les **ressources** que l'on a à disposition. Nous verrons dans le chapitre suivant comment influe ce paramètre sur l'estimation.

IV.8.4 L'affectation des ressources

Enfin, la dernière partie de l'élaboration du planning est le choix et l'affectation des ressources aux tâches. Dans le cas de Pièces Avenue, cette étape s'est apparentée pour moi à un casse tête tellement il y avait de contraintes à prendre en compte.

La première contrainte était que je souhaitais absolument affecter des développeurs qui avaient une bonne expertise pour certaines **fonctionnalités clés**. Les stagiaires pour leur part ont été affectés aux travaux moins « vitaux ». Par exemple, j'ai choisi d'assigner Killian et Hervé pour les retours, le processus de commande et pour les produits TecDoc. Nicolas et Amauri étaient quant à eux affectés aux produits « accessoires » alors que Than Tri et William se sont occupés de la « FAQ » ou de l'espace membre, des tâches à priori moins importantes mais toutefois formatrices et intéressantes.

De plus, dans un souci de **cohérence**, je souhaitais qu'un développeur s'occupe à la fois du back office et du front office d'une fonctionnalité. Par exemple, Nicolas et Amauri ont d'abord travaillé sur Serum pour coder le back office des produits « accessoires » avant de passer aux listings, pages de détail, ajout au panier des accessoires, etc. sur le front office.

Il faut aussi prendre en compte **l'expérience** de chaque développeur lors de l'affectation des tâches. Par exemple, Than Tri avait déjà travaillé sur l'espace membre de Moto 85. Il avait donc déjà une certaine idée de ce que j'allais demander sur Pièces Avenue. Hervé avait pour sa part travaillé sur le processus de commande de Motoshopping avec moi et il savait donc comme moi les points qu'il fallait que nous améliorions et les erreurs à éviter.

Le transfert de **connaissances** est aussi un facteur clé. C'était surtout le cas pour Louis, qui a travaillé seul sur le prototype TecDoc. Au sein de l'équipe de développement, il s'agissait donc du seul développeur qui avait l'expérience de ce référentiel et des *webservices*. Or il s'agit d'un risque potentiel. En effet, il était tout à fait possible qu'il tombe malade ou qu'il ait un autre empêchement et donc cela aurait fortement impacté le projet. J'ai donc choisis de l'associer à Hervé pour certaines tâches relatives à TecDoc. Ainsi, le transfert de connaissances s'est fait très naturellement.

Enfin, les **affinités** de chaque développeur est le dernier paramètre que j'ai pris en compte. J'ai souhaité que l'ensemble de l'équipe de développement s'épanouisse en faisant des choses qu'ils considèrent intéressantes. Si une tâche était pénible, j'ai essayé de compenser au mieux par une autre tâche plus captivante vis à vis du profil de chaque développeur. Ce n'est pas évident à gérer et surtout cela implique de très bien connaître les membres de son équipe.

Connaître son équipe permet aussi **d'affiner** la durée pour chaque tâche. Bien sûr, j'ai demandé à chaque développeur son avis sur le planning et sur les durées que j'avais estimées. D'ailleurs, il était important pour moi de considérer les avis de chaque développeur pour pouvoir les intégrer dans le projet avant la phase de développement et pour faire en sorte qu'ils s'approprient un peu plus celui-ci. Cependant, il est déjà possible d'affiner en amont ces durées en ayant une bonne connaissance des membres de son équipe. Par exemple, Than Tri est plus rapide que la moyenne, donc j'ai pu appliquer un coefficient négatif et ainsi réduire sur la durée des tâches le concernant.

Une fois toutes ces contraintes prises en compte, il est possible d'affecter les ressources sur le planning. Je me suis alors rendu compte qu'il restait encore une dernière problématique à résoudre : certaines ressources étaient utilisées à plus de cent pour cent dans certaines phases du projet. La solution consiste alors à utiliser un outil tel que Microsoft Project. Pour Pièces Avenue, j'ai choisi le logiciel *Planner* sous Linux...

Cette étape fut pour moi réellement délicate à aborder. Je n'avais pas du tout pensé à toutes ces problématiques. L'élaboration du planning a donc pris plus de temps que ce que je prévoyais initialement. Mais finalement ce fut une bonne expérience et ce même malgré que le planning n'ait pas été respecté. En effet, nous avons commencé et terminé le projet avec un mois de retard. L'estimation des tâches était cependant proche de la réalité et donc c'était très positif pour moi.

IV.9 Bilan

Cette phase de conception m'a donné une excellente **vision** du projet. De plus, elle m'a aussi permis de conseiller et ainsi d'aider le client à définir clairement le **périmètre de son besoin**. Je me sentais prêt à attaquer la phase de réalisation après cette longue phase d'étude et d'assistance à la maîtrise d'ouvrage.

A mon sens, cette étude était nécessaire et même **vitale** pour le bon déroulement du projet car elle couvre toutes les activités de la phase de réalisation du site. Ne pas l'avoir faite aurait impliqué de lourdes pertes en terme de temps et donc d'argent dans la réalisation du site. Le tableau ci-dessous liste **l'intérêt de chaque document** rédigé dans le cadre de cette phase de conception :

Conception	Documents
Organisationnelle	Plan de qualité, planning
Fonctionnelle	Cahier des charges
Ergonomique	Arborescence, annexe et <i>zoning</i>
Graphique	Pré-briefing graphique
Référencement	Annexe à l'arborescence et <i>zoning</i>

Table 3 - Utilité des documents de conception

Mais il est aussi possible de voir cette étude sous un autre angle. En effet, en qualité nous utilisons souvent l'outil **QQOQCP** (un acronyme pour Qui, Quoi, Où, Quand, Comment, Pourquoi). Ce dernier permet d'obtenir des informations sur toutes les dimensions d'un problème. Il est possible d'utiliser cette méthode pour la phase de conception d'un site Internet. En effet, les différents documents que j'ai rédigés permettent de répondre à chacune des questions de cette méthodologie.

Le *quoi* et le *pourquoi* trouvent leurs réponses dans le cahier des charges du projet, l'arborescence, son annexe ainsi que le *zoning* et le briefing graphique. Les questions *qui* et *quand* sont quant à elles résolues grâce au planning détaillé. Enfin, le *comment* et, dans une moindre mesure, le *où* sont définis dans le plan de qualité.

Cette phase de conception **représente environ 60 à 70 %** de mon travail sur le projet. J'ai principalement travaillé seul en relation directe avec mon client Stéfan Edet, même si j'ai ponctuellement demandé de l'assistance aux spécialistes (Yoann pour le graphisme et le *zoning* et Maxime pour le référencement). Je pense que **j'ai beaucoup appris**, en particulier dans le domaine du e-commerce qui a des spécificités et contraintes qu'il faut connaître en tant que chef de projet et concepteur. Je considère cette première phase comme un succès car elle a parfaitement joué son rôle d'information auprès des différents acteurs. Toutefois, il y a encore beaucoup de domaines sur lesquels je peux m'améliorer (tel que la qualité) et j'espère en avoir l'occasion sur de futurs projets.

V La réalisation du projet Pièces Avenue

Une fois le périmètre du besoin défini dans la phase précédente, mon rôle de chef de projet s'est orienté vers celui de la « direction » du projet. Malgré ce que l'on pourrait imaginer, mon point de vue est que cette phase n'est pas la plus complexe du projet.

Ce rôle m'est apparu très différent de celui d'assistant de maîtrise d'ouvrage, puisqu'il a impliqué une **dimension supplémentaire** de conduite des équipes de travail chez Skalpel ainsi que chez nos partenaires. Il en résulte que mes tâches étaient moins axées sur la réflexion ou sur l'étude et plus sur la gestion quotidienne des acteurs du projet et du client.

Du fait de cette proximité avec les acteurs, **l'aspect relationnel** a aussi pris une part plus grande dans mon rôle. Tout au long de cette phase, j'ai été confronté à d'autres contraintes, d'autres problématiques. Je me suis notamment rendu compte qu'en tant que chef de projet, il fallait que je gagne le soutien des développeurs.

En effet, il arrive qu'un développeur rechigne à travailler s'il juge par exemple que son supérieur hiérarchique n'est pas assez compétent ou s'il sent que ce dernier n'a pas le contrôle du projet. Dans ma carrière, il m'est arrivé d'avoir des chefs de projet trop distants avec leurs équipes ou qui accordaient une importance plus que nécessaire au respect des plannings ou des coûts... Le résultat était toujours la perte de confiance et surtout de motivation de ses collaborateurs, impactant de manière négative les délais.

Se faire accepter par les développeurs est une tâche moins évidente qu'il n'y paraît, car il ne s'agit pas uniquement d'une question de respect de **hiérarchie** ou de **compétences**. Evidemment, ces facteurs jouent un rôle important. Mais dans mon cas, j'ai pu aussi compter sur une certaine **légitimité** qui était due au fait que j'étais le plus ancien au sein de la structure. Cette expérience a joué un rôle de premier plan, notamment aux yeux des stagiaires.

Un autre facteur essentiel a été pour moi ma volonté **d'amélioration** de la méthodologie et de la gestion des projets. Par voie de conséquence, cette amélioration impactait de manière positive les conditions de travail des collaborateurs. Moins de stress, moins de dépassement d'horaires, des objectifs clairement définis... Ces arguments ont fait que l'équipe de développement m'a soutenu en tant que chef de projet.

Le troisième facteur qui a joué en ma faveur a été mon **soutien** de toute épreuve aux développeurs. En effet, malgré toutes les tensions qui ont pu régner entre les différentes personnalités de l'équipe de développement, j'ai toujours essayé d'être équitable, juste et de protéger au mieux les intérêts de l'équipe, notamment vis à vis des demandes ou des réclamations du client.

Enfin, comme je l'ai mentionné auparavant dans l'étape du planning, j'ai souhaité que les membres s'épanouissent au travail. En prenant en considération leurs attentes et leurs souhaits, j'ai pu créer un climat « gagnant-gagnant » avec les développeurs. Mon but était de rendre le projet **intéressant** pour chacun des membres de l'équipe. Grâce à ce climat positif, j'ai gagné leur confiance et je les ai par la même occasion convaincu de travailler avec moi pour faire avancer le projet.

Mais la gestion des équipes n'est pas la seule nouveauté lors de cette phase. La relation avec le client change elle aussi et n'est plus la même que lors de la phase précédente. En effet, j'ai ressenti ici **un niveau supplémentaire de pression**, car le client est généralement très impatient et souhaite voir rapidement des résultats après la première phase d'étude.

Toutefois pour arriver à ces résultats, il faut respecter scrupuleusement les **étapes** de la réalisation d'un site Internet. Celles-ci sont les suivantes :

- La création d'un logotype
- L'étape d'élaboration des maquettes
- Le *mapping* objet-relationnel
- Le développement du back office
- Le découpage HTML
- L'intégration du front office
- La recette
- La post production et l'optimisation du référencement

Je vais détailler ces étapes dans la suite du mémoire, en expliquant les travaux que j'ai eu à réaliser en tant que chef de projet ainsi que le retour d'expérience pour chacune d'elles.

V.1 La création du logotype

Un **logotype**, plus couramment appelé **logo**, est une représentation graphique qui sert à identifier de manière unique et immédiate les sociétés, associations, institutions, produits, services, événements, ou toute autre sorte d'organisations dans le but de différencier le propriétaire du logotype des autres entités.

Source: Wikipedia

Lorsque nous avons démarré nos travaux pour le projet Pièces Avenue, la société n'avait pas encore de logotype. Dans la mesure où elle souhaitait créer et imposer sa marque, il fallait que nous lui associons un logo afin que les clients puissent rapidement identifier Pièces Avenue de ses concurrents.

Le logotype joue un rôle capital dans la communication d'une entreprise. C'est la raison pour laquelle un soin particulier doit être apporté à sa conception. Il faut entre autres choses prendre en compte l'activité de la société mais aussi, la clientèle ciblée ou encore les couleurs imposées par le client. L'ensemble de ces informations était mis à disposition des graphistes dans le pré-briefing graphique.

Tout en prenant en compte ces contraintes, il faut essayer de faire preuve d'un maximum de créativité. C'est pourquoi l'étape de création du logotype a été confiée à notre partenaire graphique, plus habitué à traiter avec ce genre de problématique. L'enveloppe totale allouée pour la prestation était de 4000 euros, soit environ deux semaines de travail.



Il y avait en réalité deux prestations achetées à cette société. D'une part, la réalisation d'un logotype utilisable sur les canaux « print » (papier, par exemple, les journaux) et « web » dans un format vectoriel. Ce format permet l agrandissement ou la réduction de la taille du logo en évitant les effets de crénage, aussi appelé *aliasing*. La seconde prestation attendue était la création d'un « favicon », petite icône (généralement 16 pixels de large par 16 pixels de haut) affichée dans le navigateur web au niveau de l'url.

L'appréciation d'un logotype revêt, comme pour tout élément graphique ou toute œuvre d'art, un caractère subjectif. C'est pourquoi j'ai presque uniquement joué un rôle de messager dans cette étape. CXA m'envoyait son logotype par e-mail, puis je l'envoyais à Stéfan et attendais ses retours pour effectuer les éventuelles corrections. Tout au plus, nous faisions part de notre ressenti à notre client.

V.2 L'élaboration des maquettes

Une fois le logotype conçu, il faut passer au maquettage. Cette étape consiste à créer les écrans du site à l'aide de logiciels spécialisés tel que le logiciel *Adobe Photoshop* ou *Adobe Fireworks*. Comme pour le logotype, ce travail a une fois de plus été confié à notre partenaire Cube X Agency puisqu'il s'agit là aussi de son domaine d'expertise. Pour cette tâche, la société CXA a détaché un graphiste à temps plein. Il s'agissait de Jimmy Raheriarisoa.

Ce chapitre mettra l'accent sur trois points. Tout d'abord, je vais détailler la **prestation** achetée ainsi que le processus d'élaboration des maquettes. Dans un second temps, je vais exposer **l'organisation** mise en place avec le prestataire ainsi que les méthodologies de travail. Enfin, je terminerai par un point de vue personnel sur cette étape, en mettant en exergue mes difficultés mais aussi l'expérience acquise.

V.2.1 La prestation

La prestation que nous avions acheté à la société CXA incluait la conception du style général (par exemple, le fond ou la typographie) ainsi que quatorze déclinaisons de ce style, chacune représentant un écran spécifique. Par exemple, une des déclinaisons que j'ai choisi était la page de listing de produits TecDoc ou encore celle de détail d'un produit. Le budget total alloué pour cette prestation était de 12 000 euros. Il a été défini au lancement du projet par l'ingénieur technico-commercial.

Pour l'élaboration des maquettes, le graphiste a principalement pris en compte deux documents en entrée. Le premier est le « pré-brief graphique » qui a aussi été utilisé auparavant pour le logotype et qui permet par exemple d'imposer les couleurs à employer sur le site. Le second document utilisé est le *zoning*. Ce document guide le graphiste par rapport aux différents éléments à faire figurer sur chaque page.

Il était important pour moi de bien cadrer les graphistes car dans les précédents projets auxquels j'avais participé ceux-ci avaient tendance à être un peu trop créatifs à mon goût et à inventer des fonctionnalités inexistantes dans le cahier des charges. Enfin, j'ai également proposé à CXA de lire l'annexe à l'arborescence pour compléter ces deux documents.

L'élaboration des maquettes est une activité qui se déroule en deux temps. La première étape consiste à concevoir un style général pour le site au travers de la page d'accueil. Il s'agit de l'étape la plus longue de cette tâche car il faut obtenir la validation du client, ce qui n'est pas toujours aisé. Une fois ce style validé par le client, le graphiste passe aux différentes déclinaisons, qui elles ne nécessitent d'une manière générale pas de validation. Je vais détailler l'organisation que j'ai mise en place pour cette activité dans la partie suivante du mémoire.

V.2.2 L'organisation

Plusieurs personnes sont intervenues dans cette étape de réalisation des maquettes.

La première est, comme je l'ai mentionné auparavant, Jimmy, le graphiste de la société CXA. Notre prestataire l'a **détaché** pour ce projet et il a travaillé directement sous ma direction. En fait c'était un peu comme s'il faisait partie de Skalpel, à la différence près qu'il ne travaillait pas dans les mêmes locaux que nous.

La seconde personne qui est intervenue ici est Yoann Nussbaumer. Yoann m'a aidé à guider Jimmy sur le plan **esthétique du site**. En effet, il a joué un rôle important dans la définition du style général ainsi que dans la conception de la home page. J'aurais peut être été capable de m'en sortir seul, mais le résultat aurait probablement été moins bon qu'avec l'aide de Yoann, qui a une bien meilleure sensibilité graphique que moi.

Mickael Knauer (*l'intégrateur HTML du projet*) m'a quant à lui apporté ses connaissances par rapport à la faisabilité **technique** du découpage. Cette fois ci, il ne s'agit pas d'une question de sensibilité graphique mais bien de compétences dont je manquais. En effet, les enseignements des langages HTML et CSS dispensés à l'IUT informatique ne sont pas suffisants lorsqu'il s'agit de la découpe de maquettes « professionnelles » et je manquais d'expérience pour statuer seul sur cette faisabilité.

Notre fonctionnement a été le suivant. Dans un premier temps, nous avons laissé Jimmy prendre en main les documents de conception et nous faire un premier retour sur sa vision de la page d'accueil. Yoann et moi avons ensuite souhaité qu'il apporte des modifications et un certain nombre d'échanges et de versions de maquettes ont transité entre Jimmy et nous.

Ensuite, une fois que Yoann et moi jugions une version de la maquette « valable » esthétiquement, je la soumettais à Mickael pour qu'il valide la faisabilité technique. Nous nous réunissions tous les deux pour voir s'il était possible de découper la maquette. Le cas échéant je la téléchargeais sur *Active Collab* pour que le client puisse nous faire ses retours. Dans le cas contraire, la maquette était renvoyée à Jimmy pour qu'il effectue les modifications souhaitées par Mickael.

Huit versions ont été nécessaires pour valider le style général et la page d'accueil. Les pages suivantes permettent de voir l'évolution entre la première version proposée par Jimmy et la version finale qui a été adoptée. Elles contiennent respectivement, la première version, la cinquième version ainsi que la huitième.

Une fois la validation de la page d'accueil obtenue, Yoann et Mickael n'avaient plus besoin d'intervenir. Le cadre graphique étant posé, je n'avais plus qu'à guider Jimmy pour les déclinaisons. Il réalisait chacune d'entre elles. Si je jugeais que l'écran était stratégique (par exemple, dans le cas du processus de commande), je demandais conseil à Yoann pour la validation. Sinon, je validais moi même l'écran.

PIECESAVENUE.com
Experts en Pièces et Services auto

Assistance téléphonique gratuite
0800 222 222
du lundi au vendredi de 7 à 19h

Mon panier | Terminer ma commande
2 articles | **194,00 €**

Accueil | Pièces auto | Accessoires & Entretien | **Promotions** | Services | Nos engagements

Votre garage
Ajoutez votre véhicule et naviguez plus efficacement dans notre catalogue produit

Espace client
Email :
Mot de passe :
Mot de passe perdu ? | Valider

Nos pièces auto par catégories | Voir toutes les catégories

- Suspensions**
 - Amortisseurs avant ou arrière
 - Triangle ou bras de suspension
 - Silent bloc de suspension
 - Roulement de roue à l'unité
 - Barre stabilisatrice
- Suspensions**
 - Amortisseurs avant ou arrière
 - Triangle ou bras de suspension
 - Silent bloc de suspension
 - Roulement de roue à l'unité
 - Barre stabilisatrice
- Suspensions**
 - Amortisseurs avant ou arrière
 - Triangle ou bras de suspension
 - Silent bloc de suspension
 - Roulement de roue à l'unité
 - Barre stabilisatrice
- Suspensions**
 - Amortisseurs avant ou arrière
 - Triangle ou bras de suspension
 - Silent bloc de suspension
 - Roulement de roue à l'unité
 - Barre stabilisatrice

Paiement 100% sécurisé

CB débitée à l'expédition

Expédition en 24h à 48h

Satisfait ou Remboursé

Newsletter
Inscrivez-vous pour recevoir nos bons plans par email et profitez d'offres exclusives ! | ok

2 pneus Toyota
215/45 ZR17 R888
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc non magna, more star. Aliquam eu turpis laquilo pelumo.
194,99 € | détails

Véhicules d'occasions | Déposez gratuitement votre annonce | Consultez toutes les annonces

- Aston Martin DBS | 1 000 km / 2 000 € | détails
- Aston Martin DBS | 1 000 km / 2 000 € | détails
- Aston Martin DBS | 1 000 km / 2 000 € | détails

Nos partenaires

Vos Achats
Pièces auto | Accesoires & entretien | Promotions | Nos engagements | Espace client | Suivre ma commande | Retourner un produit | Besoin d'aide ? | Conditions générales de vente

Nos services
Annonces véhicules d'occasion | Questions / réponses mécaniques | Centres de montage | Devis personnalisé | Qui sommes nous ? | Nos engagements | Nos partenaires | Mentions légales | Votre avis nous intéresse ! | Blog | Contactez-nous | Crédits du site

Figure 58 - Première version de la page d'accueil du site

Cette première version était très basique puisqu'elle ne faisait que reprendre le zoning de Yoann, sans apporter de créativité d'un point de vue graphique.

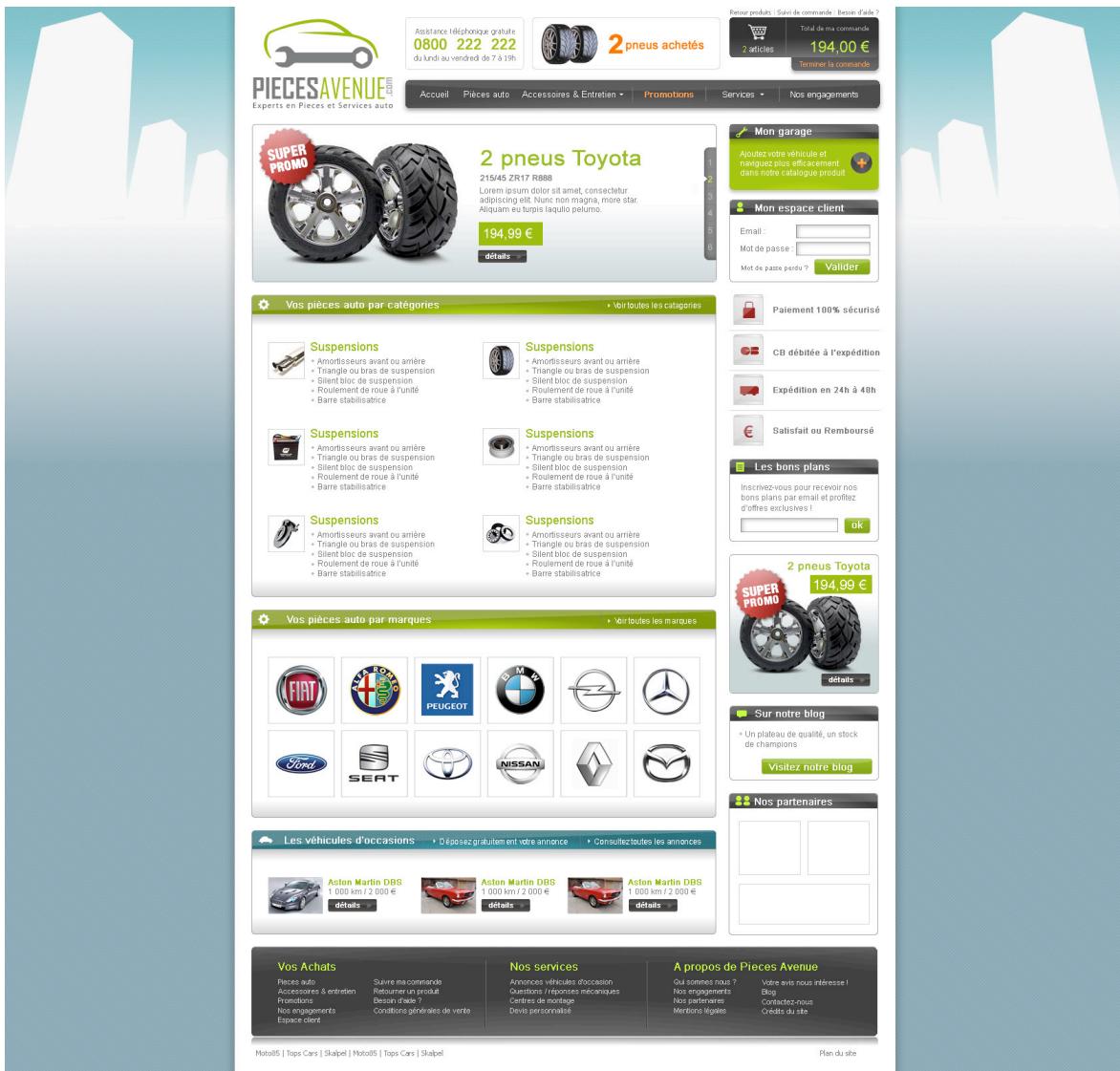


Figure 59 - Cinquième version de la page d'accueil du site

Plusieurs essais ont été faits par rapport au fond du site. Dans cette cinquième version, il s'agit d'un fond uni en bas, avec un dégradé vers le blanc. Les formes géométriques représentent des bâtiments d'une ville. Certaines couleurs (logos) ont été changées dans la colonne de droite.

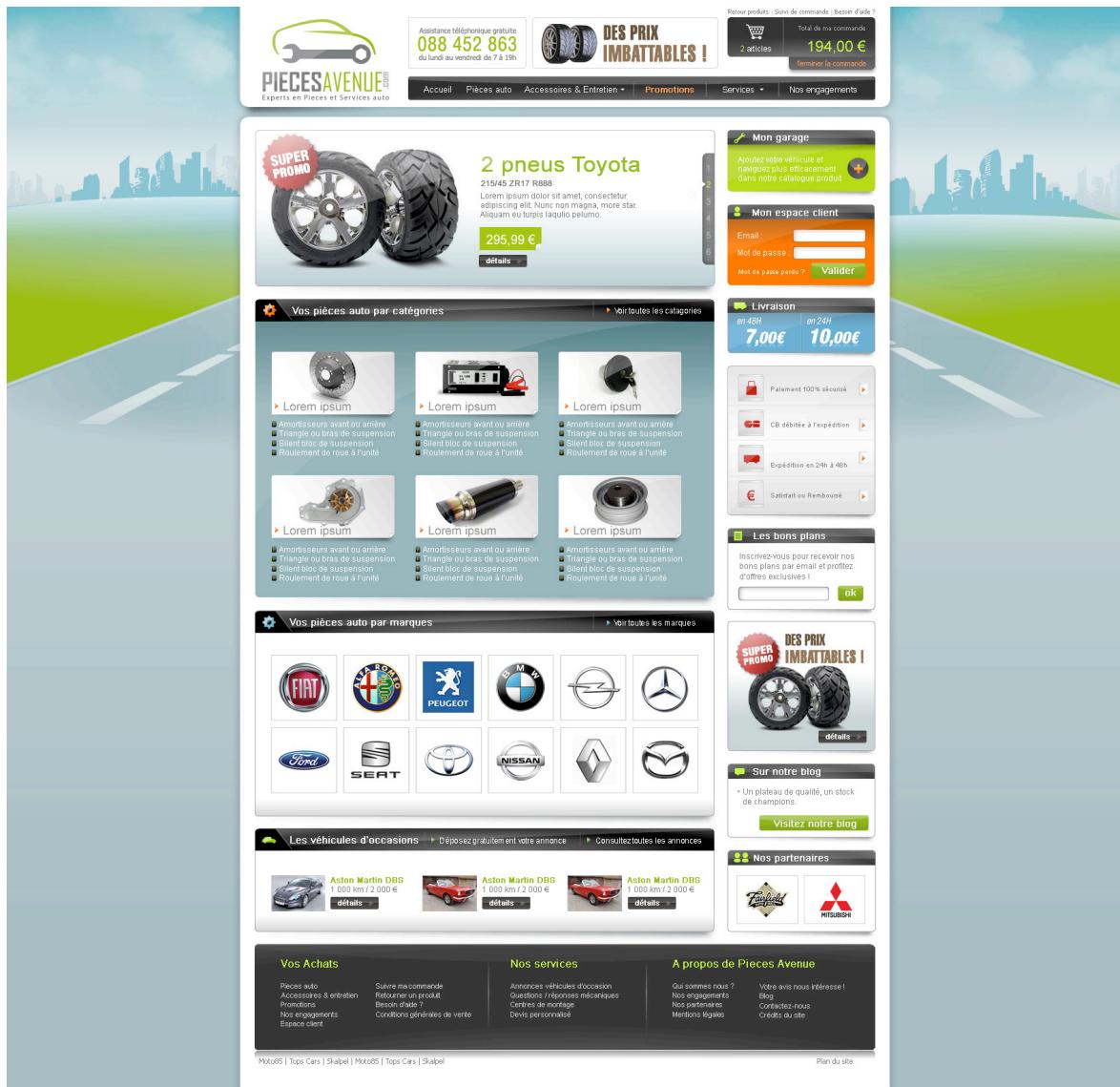


Figure 60 - Huitième et dernière version de la page d'accueil du site

Pour cette dernière version, outre l'évolution du fond du site, un travail supplémentaire a été réalisé sur les différents blocs de contenu. La couleur de fond ainsi que la police de l'entête de ces derniers ont été modifiés. Une ombre a également été ajoutée à chaque bloc, ce qui donne du relief et les fait ressortir. L'aspect « bicolore » du site a été en partie estompé par l'ajout de couleur (orange et bleu).

V.2.3 L'expérience

Durant cette étape d'élaboration des maquettes, j'ai clairement ressentis en moi un manque d'expérience et de compétences dans le domaine du graphisme.

D'un côté, je me disais que la sensibilité graphique était un talent que l'on avait ou pas. Mais d'un autre côté, je me disais aussi que cette situation était logique dans la mesure où j'ai suivi une formation beaucoup plus axée sur l'analyse et le développement et qui accordait moins d'importance à l'aspect graphique.

Peut être qu'un chef de projet issu d'une filière telle que la section « SRC » (*Services et Réseaux de Communication*) ou d'une école plus orientée dans l'art aurait été meilleur que moi à ce niveau et moins bon dans l'analyse ou dans la conceptualisation du problème. Peut être aussi qu'il ne s'agissait que d'une question d'expérience et qu'avec plus de pratique j'aurais pu très bien m'occuper de cette tâche seul.

Quoi qu'il en soit, c'est vraiment dans cette étape que j'ai remarqué qu'un chef de projet ne pouvait pas tout savoir. Ou en tout cas, être bon dans tous les domaines. Je pense que chaque chef de projet a un domaine qu'il apprécie particulièrement ou dans lequel il excelle, qu'il s'agisse de part son expérience, de part ses études ou ses affinités. Mais hors de son domaine de prédilection, j'estime que le chef de projet doit aussi savoir faire appel aux compétences de spécialistes pour avoir un résultat qui soit le meilleur possible.

Ce sentiment était plus marqué ici que lors de la phase de conception car dans cette dernière mes demandes étaient vraiment ponctuelles. Il s'agissait même très souvent uniquement de demandes de validation. Je n'avais donc pas l'impression d'avoir besoin des experts. Mais ici, j'étais plus spectateur qu'acteur, tout du moins dans la première phase d'élaboration du style général de la maquette. Et même si j'ai beaucoup appris, cette étape m'a permis de prendre conscience de l'ampleur des connaissances qu'il me restait à assimiler.

Enfin, j'ai aussi appris à utiliser quelques petites astuces en ma faveur. En effet, du fait que nous avions une limite de quatorze écrans, je ne pouvais pas faire maquetter toutes les pages du site. Qu'à cela ne tienne, j'ai fais maquetter sur le même écran le contenu de plusieurs pages. La zone centrale de la page incorporait par exemple la liste et le détail d'un produit accessoire ce qui me permettait de faire « d'une pierre deux coups ».

V.3 Le mapping objet relationnel

Le *mapping* objet-relationnel est une **technique de programmation** qui consiste à faire correspondre des objets du langage de programmation aux tables d'une base de données. Comme je l'ai mentionné dans la première partie de ce mémoire, le *mapping* objet-relationnel a beaucoup d'avantages tel que l'abstraction du moteur de base de données ou encore la réduction des lignes de codes d'accès aux données.

L'ORM (pour *Object Relational Mapping*) et, par extension, l'élaboration de la base de données, a été la première activité des développeurs lors de cette phase de réalisation. Elle a eu lieu en parallèle avec la production des maquettes. C'est une activité qui consiste à lire et analyser le cahier des charges fonctionnel puis à en retirer les concepts et objets qui seront ensuite utilisés sur le site.

Je fais souvent une analogie de cette activité avec la construction d'une maison. L'ORM est de mon point de vue l'équivalent des fondations de l'ouvrage. Ici, tout va découler des objets et du *mapping* qui va être fait. Si le *mapping* est mal pensé, l'ensemble du programme sera bancal et peut être bogué. C'est donc une activité qui mérite la plus grande attention.

C'est la raison pour laquelle j'ai souhaité avoir **deux approches du problème**. Je vais décrire ces deux approches et les choix que j'ai pris dans une première partie de ce chapitre. Dans une seconde partie, je vais détailler un peu plus le **format YML**, qui a été utilisé pour le *mapping*, avec un rapide exemple montrant concrètement les avantages de l'utilisation d'un outil d'ORM.

V.3.1 Deux visions de la problématique

L'ORM est une technique de programmation relativement **récente**.

Avant l'essor des ORM, les développeurs devaient étudier et élaborer la base de données avec des techniques de conception telle que la méthodologie **MERISE**. Ici, l'idée était de créer un modèle conceptuel de données (MCD), puis de décliner ce dernier en modèle logique (MLD) et enfin en modèle physique de données (MPD). Ensuite, il était possible de générer les scripts pour créer la base de données.

C'est une méthodologie qui m'a été enseignée lors de ma première année de DUT et que je **maitrise** bien. J'ai d'ailleurs eu l'occasion de travailler en tant qu'architecte de base de données après mon expérience chez Skalpel. Louis Saunders avait aussi beaucoup d'affinité pour cette méthode, puisqu'il a développé un logiciel en C# pour concevoir des modèles physiques de données.

Tout au long de l'écriture du cahier des charges fonctionnel, j'avais moi même **forgé ma propre vision** de la base de données en m'aïdant de MERISE. Même sans avoir conçu de MCD (*Modèle Conceptuel de Données*), j'avais donc une première idée de ce qu'allait être la base de données. Cependant, j'ai quand même demandé à Louis de faire le même travail et d'élaborer un modèle conceptuel et physique de données. Mon objectif ici était de valider que l'architecte et moi étions sur la même « longueur d'onde ».

Puis j'ai souhaité que nous analysions le problème sous un autre angle. En effet, j'avais besoin du point de vue « **objet** » pour l'intégration dans l'ORM. Cette étude nécessitait des

connaissances en UML (*Unified Modeling Language*), c'est la raison pour laquelle que je l'ai confiée à Killian qui était le plus à l'aise d'entre nous avec cette méthodologie. Lorsque l'ORM avait créé les tables, j'ai pu comparer et recouper les deux résultats.

Ces deux études m'ont donné **satisfaction** car les résultats obtenus étaient très proches l'un de l'autre. De très légères modifications ont été apportées dans les fichiers YML de l'ORM (notamment pour les produits « options ») mais dans l'ensemble le cahier des charges fonctionnel avait été **bien compris** par les développeurs. Cela a conforté mon sentiment initial de succès de la première phase de conception.

Enfin, je pense qu'en choisissant de travailler ainsi, j'ai fait preuve de **sécurité**. En effet, je ne voulais pas qu'on ait à apporter trop de modifications au niveau de la base ou que celles-ci soient trop lourdes car cela aurait induit des délais supplémentaires et notre planning était déjà très serré. J'ai préféré prendre le temps de bien faire les choses dans cette étape plutôt que de perdre ce temps par la suite.

V.3.2 Le format YML

Les fichiers YML définissent la structure des objets, que nous appelons les **modèles** de l'ORM Doctrine. Ils ont l'avantage d'être très légers et donc leur écriture est rapide. Dans ce chapitre, nous prendrons l'exemple des clients du site pour illustrer très rapidement ce format ainsi que ses avantages.

```
# TABLE CUSTOMER
#####
DbCustomer:
    inheritance:
        extends: DbBase
    tableName: customers_customers
    columns:
        lastname: string(255)
        firstname: string(255)
        phone: string(255)
        mobile: string(255)
        birthdate: date
        login:
            type: string(255)
            unique: true
            notnull: true
            notblank: true
            email: true
        password:
            type: string(255)
            notnull: true
            notblank: true
        civility_id:
            type: integer
        sponsor_id: integer
    offers:
        type: boolean
        default: false
    optin:
        type: boolean
        default: false
    isCustomer:
        type: boolean
```

```

    default: false
indexes:
  idxLogin:
    fields: login
relations:
  DbAddress:
    local: id
    foreign: member_id
    type: many
  DbCivility:
    local: civility_id
    foreign: id
  Sponsor:
    class: DbCustomer
    local: sponsor_id
    foreign: id
  Sponsored:
    class: DbCustomer
    local: id
    foreign: sponsor_id
    type: many

```

Le format YML est très intéressant car il permet de définir des contraintes comme la taille ou le type des champs. Il permet aussi de spécifier les relations (simples ou multiples) entre les objets. Enfin, ce format est relativement proche de la base de données car il donne en outre la possibilité de définir les indexes dans les tables. Ces caractéristiques seront utilisées dans la suite du projet et tout particulièrement dans le développement du back office.

Le fonctionnement de l'ORM est le suivant : lorsque Doctrine analyse ce fichier YML, il crée ou met à jour les tables ainsi que les classes d'accès aux données en fonction du contenu du fichier. Ces classes sont ensuite étendues par les développeurs qui ajoutent uniquement la logique métier, qui est propre à Pièces Avenue. A titre d'information, 91 tables ont été générées par l'ORM Doctrine dans le cadre du projet Pièces Avenue.

V.4 Le développement du back office

Le back office est l'interface d'administration web de la boutique. Comme son nom l'indique, il s'agit de la partie qui se trouve à « l'arrière » et qui n'est pas visible des internautes. Ici, les administrateurs préparent et expédient les commandes, gèrent les internautes ou les produits affichés sur le site.

Le développement du back office a été scindé en deux. Nous avons tout d'abord codé le **framework Serum**, l'outil de gestion en ligne de contenu. Dans un second temps, nous avons implémenté les modules nécessaires et **spécifiques** au projet Pièces Avenue. Les deux prochains chapitres vont détailler ces deux étapes.

V.4.1 Le FrameWork Serum

Le nom de la société, Skalpel, fait référence à l'instrument utilisé par les chirurgiens lors des opérations. Yoann avait choisi ce nom car la chirurgie est un domaine d'activité qui nécessite de l'expérience et de la précision. Ce terme fait d'ailleurs partie de la *baseline* de Skalpel qui était « e-commerce de précision ».

Serum est lui aussi un mot qui fait partie du champ lexical de la médecine. Dans ce domaine, il s'agit d'un liquide qui permet de soigner infections ou maladies. Il s'agit donc de la réponse du médecin face à une problématique, de même que Serum était notre réponse aux problématiques de l'administration web et du e-commerce.

V.4.1.1 Les besoins et contraintes

Lorsque nous avions commencé à développer Serum nous disposions déjà d'un outil de gestion de contenu : le Skalpel Manager. Cet outil n'était pas dépassé en soi, mais il y avait des sujets sur lesquels nous souhaitions nous améliorer. Un des points que nous souhaitions perfectionner était **l'expérience utilisateur** et plus particulièrement l'aspect graphique de l'interface d'administration.

Le second point à améliorer était la réduction des **temps de développement** et l'amélioration de la productivité, qui est un facteur important, surtout dans un domaine très concurrentiel comme le web.

Il fallait aussi prendre en compte **l'internationalisation** des sites web pour ce nouvel outil. En effet, nous avions eu des difficultés à ce sujet dans le cadre du projet Labyrinth Bikes, un site en plusieurs langues. L'internationalisation n'était pas géré nativement par le CMS (*Content Management System*) Skalpel Manager et nous avions appliqué des patches pour répondre à ce besoin. Le résultat n'était pas satisfaisant, notamment à cause des temps de développement induits par les patchs. Une gestion native aurait permis ici aussi des gains de temps.

Enfin, une des dernières problématiques qu'il nous fallait résoudre était la **mise à jour de l'outil**. En effet, la gestion des corrections et des évolutions pour chaque instance du CMS devenait trop complexe. Il existait presque autant de version de Skalpel Manager que de site développé. C'est la raison pour laquelle nous avons voulu intégrer un système de mise à jour qui permettrait à chaque instance du Skalpel Manager de contacter un serveur et de se mettre à jour sans intervention humaine.

Ces améliorations devaient se faire en respectant le principe **modulaire** du Skalpel Manager. En effet, les modules sont sensés être indépendant les uns des autres. Pour répondre à toutes ces problématiques, Yoann et moi avons définis des prérequis pour le nouvel outil de gestion de contenu.

Pour la couche de présentation de l'interface d'administration Yoann a choisi d'utiliser la technologie **javascript** et en particulier le *framework* Ext JS. Celui ci permet de réaliser rapidement des applications dont le look ressemble aux programmes sous Windows, avec notamment la possibilité d'appliquer des effets, d'utiliser le clic droit etc.

Pour améliorer notre productivité, j'ai pris la décision d'inclure dans Serum le *framework* Zend ainsi que Doctrine. Ce dernier a permis entre autre la **génération automatique** des formulaires (d'ajout, modification) et des listes à partir des fichiers YML, une des fonctionnalités que j'avais demandé à mettre en place et qui nous a fait gagner énormément de temps dans le développement du back office. De plus, Doctrine avait également des extensions qui lui permettaient de supporter l'**internationalisation**. J'ai donc pu ainsi résoudre deux problèmes avec un seul outil.

Enfin, le principe de modularité du CMS a pu être conservé grâce à l'adoption **d'une structure de répertoire** bien spécifique.

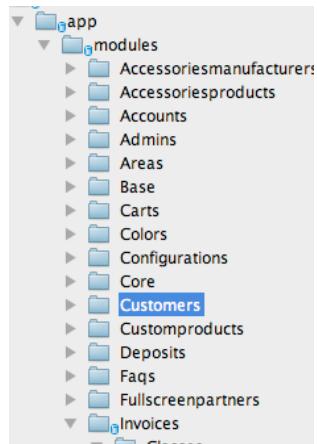


Figure 61 - Structure modulaire

V.4.1.2 Le développement

Suite à mes discussions avec Yoann, le développement de Serum a été **délégué** à Killian. Ce dernier a été en quelque sorte le responsable et chef du projet. Il a en aussi pu compter sur Hervé et Louis pour l'aider dans la réalisation du CMS. Les trois développeurs se sont partagés les travaux en prenant en compte mes demandes et les contraintes imposées.

J'ai été contraint de prendre cette décision faute de temps. C'est aussi ici que je me suis rendu compte qu'en plus de ne pas tout savoir, le chef de projet ne **peut pas tout faire**. Déléguer cette tâche à eu une autre conséquence positive à mon sens : les développeurs ont eu des responsabilités et cela a « *boosté* » leur implication dans le projet.

Ext JS est très différent de l'idée du javascript que peut avoir un débutant dans ce langage. Il est plus structuré que l'on ne peut l'imaginer et permet par exemple l'utilisation des paradigmes objets et en particulier de l'héritage. Cette spécificité nous a permis par exemple de définir les listes et formulaires par défaut dans les modules et de les étendre pour les besoins de Pièces Avenue.

V.4.2 Les spécificités de Pièces Avenue

Avec Serum, nous nous sommes doté d'un cadre ergonomique, graphique et d'une norme pour coder le Back Office de Pièces Avenue et des futurs projets Skalpel. Il a donc fallu intégrer dans ce cadre les spécificités du métier de la vente de pièces automobiles en ligne, comme par exemple le traitement des commandes, des retours de produit etc. Ces derniers ont été définis auparavant dans le cahier des charges du projet.

Ci-après, un extrait du cahier des charges concernant le traitement des commandes dans l'interface d'administration:

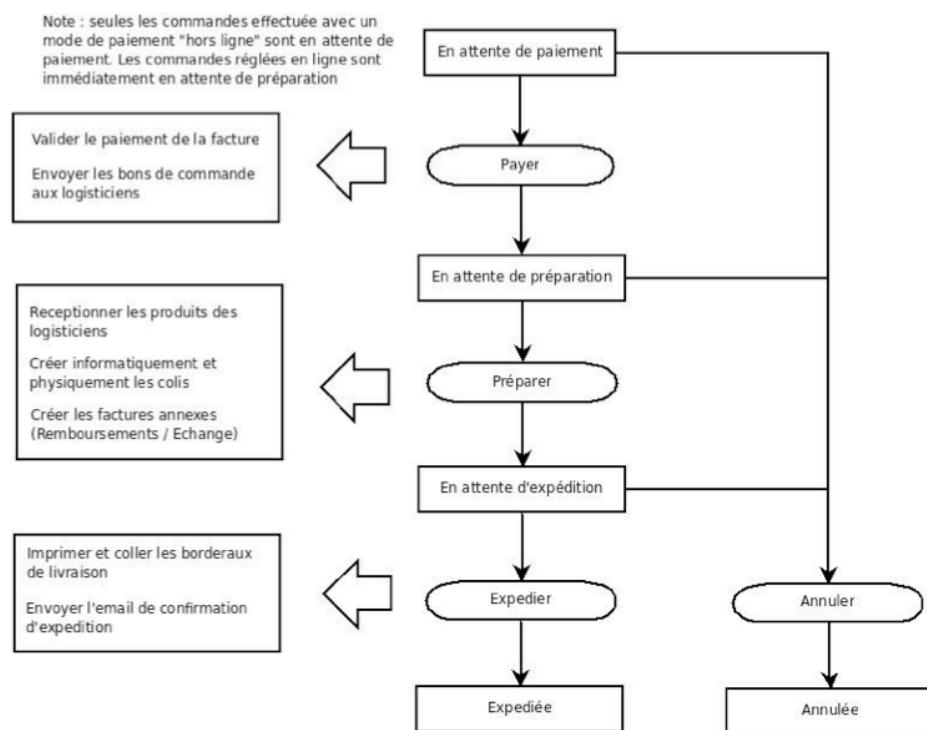


Figure 62 - Processus de commande dans le back office

Tout au long de la réalisation du Back Office, j'ai vraiment trouvé très intéressant de voir se concrétiser l'étude que j'avais faite en amont, mais je dois avouer que **l'étude** en elle-même a été pour moi bien plus **intéressante**.

Qu'il s'agisse de la phase de conception ou de réalisation du Back Office, le maître mot a été ici de se **mettre à la place** du client qui va administrer sa boutique. Cela implique de **connaître les utilisateurs et l'utilisation** qui est faite du Back Office. Par exemple, si un administrateur à souvent recours à une fonctionnalité, il faut essayer de réduire le nombre de clics à effectuer, de texte à entrer ou encore le nombre de cases à cocher. Il est aussi possible par exemple d'intégrer un système de code barre pour accélérer les recherches, etc.

Avec le recul je pense que j'aurais pu **insister sur ces aspects dans les documents de conception**, mais je manquais d'expérience pour me rendre compte de l'importance que cela peut avoir pour les administrateurs et les gains de temps que cela peut engendrer.

Suite au lancement du site, nous avons rectifié le tir sur plusieurs fonctionnalités et par la même occasion réduit drastiquement la durée moyenne de traitement des commandes et des retours. Aujourd'hui encore j'utilise et modifie Serum pour ajouter de nouvelles fonctionnalités ou optimiser celles existantes.

Le Back Office de PiècesAvenue a, entre autres choses, permis l'administration :

- Du catalogue produit (accessoires, Tecdoc, options, produits personnalisés)
- Des paramètres de configuration du site (administrateurs, mots de passes, contenus des pages, constantes)
- Des modes de paiement et de livraison (Chronopost, La Poste, Mercanet, Paybox)
- Des clients (adresses, paniers, coordonnées)
- Des campagnes de publicité (banderoles, carrousel, bons de réduction)
- Des commandes et des retours (factures, commandes client et fournisseur, retours de produits, logistique, etc.)

V.5 Le découpage HTML / CSS

Le découpage HTML est une étape de la phase de réalisation qui consiste à créer des fichiers HTML à partir des maquettes fournies par les graphistes. Ces fichiers ont la particularité de pouvoir être interprétés (lus et affichés) par les navigateurs Internet. Il s'agit ici de reproduire **au pixel près** les écrans conçus par le graphiste et validés par le client.

Cette partie va rapidement décrire les avantages et inconvénients des deux **principales méthodes** disponibles pour le découpage ainsi que celle qui a été **choisie**. Elle mettra aussi l'accent sur mon **retour d'expérience** vis à vis du découpage HTML.

V.5.1 Les techniques de découpage

Il existe **deux grandes méthodes** de découpages. La première est celle basée sur les tableaux (balises HTML 'table', 'tr', 'td', etc.). Dans cette méthode, le contenu de la page est positionné dans les différentes cellules d'un tableau. La démarche consiste à découper la page en zones dans un premier temps, puis d'intégrer les contenus dans chaque zone dans un deuxième temps. Elle n'est plus beaucoup utilisée aujourd'hui car elle a beaucoup d'inconvénients (détournement de l'utilisation des tableaux, pas de séparation de la forme et du contenu, profondeur du DOM (*Document Object Model*) qui impacte de manière négative le référencement etc.)

L'autre méthode consiste à utiliser les CSS, ou *Cascading Style Sheets* (en français, feuilles de styles en cascade). Par rapport à la méthode des tableaux, la démarche CSS est complètement inversée. Ici l'intégrateur HTML doit d'abord partir du contenu pour ensuite travailler sur le positionnement et le design. La philosophie de cette approche est de construire le design en spécifiant des **blocs** de contenu. Les styles sont ensuite appliqués sur chaque bloc dans un fichier CSS. Par exemple, lorsque l'on découpe la page de liste des produits, un des blocs peut être celui d'un produit, qui sera composé d'une image, d'un texte descriptif, d'un lien d'ajout au panier et d'un lien vers la page de détail.

Ce raisonnement permet de séparer le contenu de la forme, le CSS étant indépendant des fichiers HTML. Toutefois, le *CSS positionning* souffre d'un défaut majeur qui est la compatibilité entre navigateurs.

Cette compatibilité est aujourd'hui la principale difficulté pour les intégrateurs. Par exemple, certains navigateurs appliquent des marges ou des bordures alors que la CSS ne les mentionne pas. Ces comportements par défaut peuvent toutefois être corrigés à l'aide de scripts adaptés. Enfin, dernier point mais non des moindres, cette méthode a l'avantage d'être conforme avec les **recommandations** du consortium W3C, dont le but est de définir les standards du réseau Internet.



Figure 63 - Logo W3C

V.5.2 La méthodologie employée

Pour rester fidèle aux recommandations du W3C, j'ai souhaité que le découpage soit basé sur la méthodologie des feuilles de style CSS. Mickael s'est chargé seul du découpage de l'ensemble des écrans. Encore une fois, il s'agissait pour moi de déléguer et de donner des responsabilités au développeur pour que celui-ci soit impliqué dans le projet. Je pensais qu'une plus grande marge de manœuvre aurait un impact positif sur le découpage.

Cependant, le cas de Mickael était un peu différent. En effet, il s'agissait pour lui (et pour Louis) du dernier projet en tant que membre de l'équipe Skalpel. Si globalement le travail produit par Mickael a été très satisfaisant et que les délais ont au final été tenus, je pense que l'excès de liberté a peut être eu l'effet inverse de celui recherché au départ.

En effet, j'ai constaté par exemple que certains blocs étaient **moins bien découpés** qu'à l'habitude. Peut être était-ce dû à un manque d'envie du fait de son départ ? Ou tout simplement était-ce dû à une mauvaise analyse ?

Par exemple, Mickael avait choisi de définir une hauteur et un fond fixe pour le bloc des produits Tecdoc dans la page de listing des produits.

Ce choix permet de simplifier sensiblement le découpage car il implique uniquement la spécification d'une image de fond fixe pour le bloc, quelque soit le contenu de ce dernier. Mon point de vue est donc que le découpage avec une taille fixe est peu judicieux car, dans notre cas, la taille ainsi que le nombre de descriptions (et donc, de la hauteur du bloc) varie d'un produit à l'autre.

Découper le bloc produit en séparant l'ombre du bas en une image distincte a pour grand avantage de donner au bloc produit une hauteur variable. Cela est rendu possible notamment en affichant l'ombre du bas dans un bloc à part, à la fin de la description du produit.

J'ai donc du **intervenir personnellement** pour corriger le découpage de certains blocs comme le montre le visuel précédent. Je savais que le découpage était une activité qui nécessitait de bonnes connaissances en HTML et en CSS. Mais j'ai aussi pu me rendre compte qu'il y avait aussi des aspects auxquels on ne pensait pas instinctivement et qu'il fallait néanmoins prendre en compte lorsqu'on réalisait le découpage d'une maquette. Par exemple le choix des **formats ou des extensions pour les images** ainsi que les contraintes de **poids des pages** à télécharger ne sont pas des points cruciaux à première vue... Ils jouent pourtant un rôle fondamental dans **l'expérience utilisateur**.

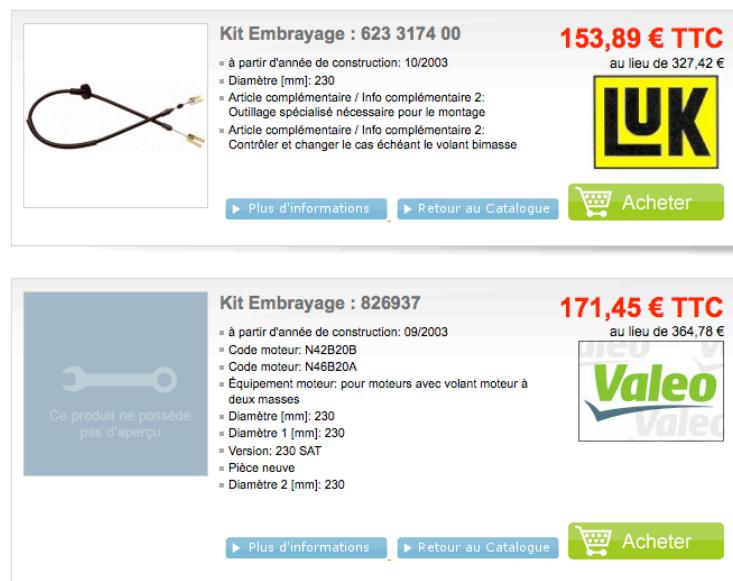


Figure 64 - Découpage HTML de la liste de produit

La formation en HTML et CSS dispensée à l'IUT Informatique n'est à mon sens pas suffisante pour faire un découpage de la complexité de celui de PiècesAvenue. En fait, je pense que beaucoup d'aspects du langage sont purement et simplement éludés. Il faut alors faire des recherches soit même ou être au contact d'experts tel que Mickael pour s'améliorer, comprendre l'intérêt et les subtilités du découpage.

Je me suis aussi rendu compte que c'est un domaine qui nécessite une remise en question constante avec l'évolution des langages (aujourd'hui HTML5) mais aussi des navigateurs. Finalement, mon opinion sur cette activité a grandement changée avant et après le projet ! Je trouvais le découpage fastidieux et trop complexe, mais finalement c'est plutôt ludique...

V.6 L'intégration du Front Office

Une fois la maquette découpée, il est possible de l'afficher dans le navigateur. Mais, le contenu affiché est purement statique et, qui plus est, factice. Il faut donc rendre dynamique cet affichage en prenant en compte les informations présentes en base de données. C'est ce que j'appelle **l'intégration** du Front Office.

Le processus d'intégration se décompose en trois étapes. La première consiste à réaliser une **hiérarchisation des écrans**. Ensuite, ces écrans sont transformés en **gabarits** (ce que l'on appelle aussi des *templates* en anglais). Enfin, pour chaque écran, il faut réaliser les éventuels **traitements et assignations** des variables à afficher dans les gabarits. Nous verrons ces trois étapes dans les sous parties de ce chapitre.

On pourrait penser que l'intégration est l'étape la plus longue de la phase de réalisation du site. Or, ce n'est pas du tout le cas. En effet, si la base de données est correctement conçue, il ne reste ici qu'à récupérer et afficher l'information. C'est une opération très rapide grâce à l'utilisation de l'ORM. A mon sens, deux points ont nécessité plus de temps et d'efforts que les autres.

Pour le premier point, il s'agit de l'intégration d'un système de cache pour le Webservice TecDoc. Ce système a été mis en place afin d'éviter de relancer plusieurs fois les mêmes appels SOAP et réduire ainsi les temps de réponse du site. Le second point sur lequel j'ai souhaité me focaliser était le processus de commande, un élément stratégique du site et qui ne doit comporter absolument aucun bogue.

V.6.1 La hiérarchisation des templates

Avant d'entrer plus en détail dans le sujet de la hiérarchisation des *templates* (ou gabarits), il me semble important de définir ce concept.

En effet, il ne s'agit pas ici de spécifier des répertoires physiques particuliers ou de donner un ordre d'importance pour chaque gabarit. En fait, il faut plutôt voir cette hiérarchisation d'un **point de vue « objet »**, en faisant une analogie avec **l'héritage** dans les langages de programmation orientés objet. En POO, il est en effet possible d'étendre les fonctionnalités d'un objet en le dérivant (héritage) et en complétant la classe ainsi obtenue.

Par exemple, si on prend le concept des « individus » dans le cadre d'un programme pour le CNAM, on pourrait tout à fait avoir une classe mère « personne », étendue par les classes « enseignant » et « étudiant ». Ces trois classes ont des informations communes, tel que le nom et prénom, mais aussi des spécificités (seuls les enseignants ont un salaire, alors que les étudiants passent des épreuves). Le principe de la programmation orientée objet est que les informations communes ne sont traitées qu'une seule fois dans la classe mère tandis que les spécificités sont implémentées dans chaque classe fille.

Pour les gabarits, le principe est similaire. Sur le site Pièces Avenue, certaines données sont récurrentes et affichées sur plusieurs pages. Par exemple, l'entête, le pied de page ainsi que la colonne de droite avec le bloc « espace membre », « garage », « partenaires » etc. sont affichés sur l'ensemble des pages du site. Seul le contenu intérieur des pages change. Donc, le gabarit parent à ces pages va définir les variables à afficher dans l'entête, le pied de page et la colonne de droite et chaque gabarit aura uniquement comme rôle de traiter les données qui lui sont propres. Cela permet de gagner du temps dans l'intégration.

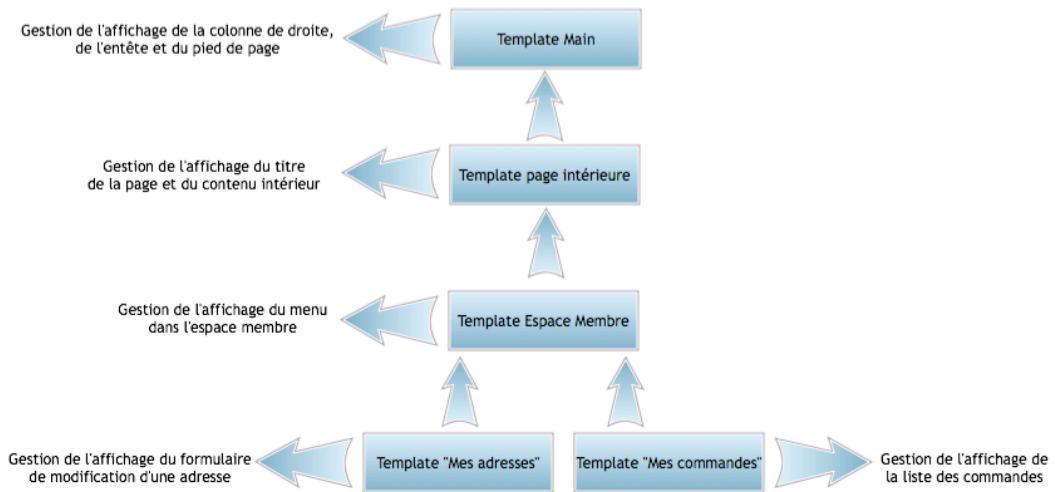


Figure 65 - Héritage des templates

Avant toute chose, il nous a fallu visualiser tous les écrans du site et les développeurs ont du réaliser cet « arbre d'héritage des gabarits ». Cela est rendu possible grâce à l'utilisation du moteur de *template* Dwoo, qui accepte une syntaxe particulière pour l'héritage. Mon rôle dans cette étape a simplement consisté à valider les travaux des développeurs.

V.6.2 La création des gabarits

La création des gabarits est l'étape suivante dans l'intégration du Front Office. Ici, les développeurs vont créer des fichiers ayant l'extension « .tpl ». Ceux-ci sont ensuite affichés par le moteur de *template* Dwoo et contiennent le code HTML / CSS issus du découpage réalisé auparavant par l'intégrateur.

Les héritages définis dans l'étape précédente sont aussi pris en compte dans le développement des *templates* à l'aide de l'instruction Dwoo « extends ».

```

1 [extends "body.tpl"]
2 {block "css"}
3     {$woo.parent}
4     <link href="${$SKIN}css/style
5 {/block}
6

```

Figure 66 - Extension d'un template

A ce stade de l'intégration du Front Office, les *templates* sont toujours statiques et il faudra attendre l'étape suivante pour que les données affichées soient celles issues de la base.

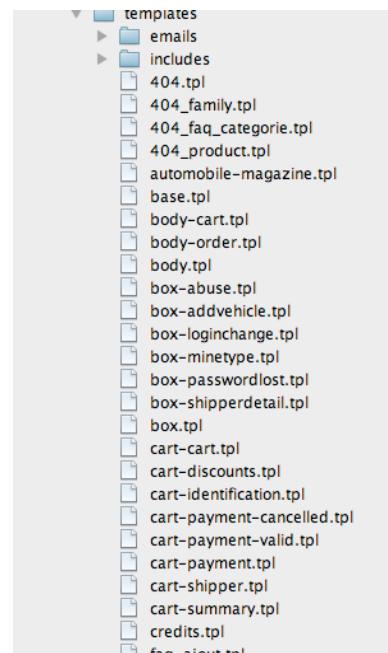


Figure 67 - Liste des templates

V.6.3 Les traitements et assignations

Le codage des traitements et les assignations Dwoo sont les deux dernières étapes de l'intégration du Front Office. Il s'agit ici de relier les données de la base aux *template* en utilisant le langage PHP et en effectuant les assignations des variables.

L'exemple ci-contre présente une requête qui récupère l'ensemble des marques de véhicules actives sous forme d'un tableau et les affecte ensuite à la variable « brands ».

Une fois la variable assignée, il ne reste plus qu'à l'afficher dans le gabarit en prenant en compte cette variable.

Ici aussi, il s'agit d'une opération triviale puisqu'il est possible d'accéder à la variable dans le *template* à l'aide du tag {\$NOM}. L'exemple ci-contre illustre l'utilisation de la variable {\$reasons} dans une boucle « foreach » (exemple tiré du gabarit d'ajout d'un nouveau retour dans l'espace membre).

Les deux exemples présentés ci-dessus sont très simplistes mais ils **permettent de bien comprendre le fonctionnement** du moteur de *template* ainsi que les avantages qui en découlent : séparation de la logique métier et de l'affichage, meilleure organisation du code, possibilités de travailler simultanément sur les gabarits et le code métier, etc.

En tant que chef de projet, mon rôle dans cette étape a surtout consisté à **conseiller les développeurs et leur apporter mon expérience** lorsque cela était nécessaire et notamment dans le cas de certains algorithmes ou vis à vis des particularités du langage. Les développeurs ont quant à eux du intégrer plus d'une cinquantaine de gabarits ainsi que l'ensemble des e-mails envoyés aux membres puisque l'envoi de mail repose sur le même principe que l'affichage d'une page, avec des affectations de variables, etc. A la fin de cette étape, les pages sont donc dynamiques et il est possible de tester le site avec des données réelles.

```
/**  
 * Displays form  
 * @return Core_Controller_Response_Abstract  
 */  
public function addVehicle()  
{  
    $rep = parent::initResponse();  
  
    $brands = Doctrine_Query::create()  
        ->from('DbTecdocbrand b')  
        ->where('b.status = ?', DbBase::ENABLED)  
        ->execute(array(), Doctrine::HYDRATE_ARRAY);  
  
    $rep->data->assign('brands', $brands);  
    return $rep;  
}  
...
```

Figure 68 - Assignation d'une variable

```
<p class="select raisons">  
    <label for="reason">Les raisons :</label>  
    <select name="reason" id="reason">  
        {foreach $reasons reason}  
            <option value="{$reason.id}">{$reason.name}</option>  
        {/foreach}  
    </select>  
</p>
```

Figure 69 - Affichage d'une variable dans un template

V.7 La recette

En informatique, la **recette** est une des phases de développement des projets. C'est celle au cours de laquelle les différents acteurs du projet se rencontrent afin de vérifier que le produit est conforme aux attentes formulées.

Source: Wikipedia

Dans le cas de Pièces Avenue, la phase de recette a été très courte car nous avions commencé le projet avec un mois de retard sur le planning initial et le client souhaitait que son site soit lancé le plus rapidement possible en production pour ne pas être en décalage avec son business plan.

Les tests ont été effectués à deux niveaux. Dans un premier temps, les développeurs effectuaient leurs tests avant de valider leurs modifications (opération commit) dans Subversion. Ensuite, je mettais à jour l'environnement de développement et Yoann, Stéfan et moi vérifiions à notre tour que les spécifications fonctionnelles étaient bien respectées.

Dans nos tests, nous avons particulièrement mis l'accent sur le **processus de commande**. Il fallait que les clients puissent rechercher, lister, visualiser les produits, les ajouter dans le panier et commander (ce qui inclut la connexion, la spécification des bons de réduction, le choix des transporteurs et le paiement) sans rencontrer le moindre accroc, même mineur.

Nous avons essayé de tester tous les scénarii possibles mais, **faute de temps**, je n'ai pas pu rédiger les fiches de tests décrivant en détail ces scénarios, les résultats attendus et ceux obtenus. De ce fait, le cahier de recette du projet est inexistant. Avec le recul, je pense que c'est la seule étape de la phase de réalisation que **nous avons mal négociée**.

J'avoue que nous aurions dû accorder plus de temps et d'importance aux tests (adoption d'une vraie stratégie de tests, écriture systématique de tests unitaires, d'intégration, rédaction des documents de tests, etc.), mais les contraintes du projet (principalement les délais ou le budget) imposent souvent des choix qu'il faut **assumer en tant que chef de projet**.

Je pense que nous avons pris un **risque non négligeable** en mettant le site en ligne sans le recul vis à vis de ses fonctionnalités et en le déployant très rapidement en production avec un trafic et un environnement réel. Cependant, c'est le contexte qui m'imposait cette décision. Finalement, malgré le manque de tests, la mise en ligne s'est déroulée correctement et nous n'avons pas eu à déplorer de bogue, ce qui était un soulagement pour moi.

Enfin, je dirais que si le **processus de qualité** mis en place en amont pour la phase de développement a peut-être joué un rôle dans ce succès, **l'implication et le savoir faire** des acteurs (qu'il s'agisse des développeurs, intégrateurs, graphistes, etc.) est sans aucun doute possible l'élément déterminant dans la réussite du projet.

V.8 La post production et le référencement

A la fin du développement, qui s'accompagne généralement de la signature du procès verbal de recette et du passage en production, le site est officiellement lancé et les internautes peuvent commander leurs produits. Toutefois, je pense que la signature du procès verbal de recette n'est pas forcément synonyme de fin des travaux.

En effet, il est très instructif de suivre le lancement d'un site pour de multiples raisons que je vais évoquer dans les chapitres suivants du mémoire. Je vais aussi aborder le sujet délicat du référencement et essayer d'expliquer en quoi ce domaine est une science empirique. Enfin, je vais exposer quelques indicateurs spécifiques au domaine du commerce électronique.

V.8.1 L'après mise en ligne

Il est très intéressant et instructif de suivre le site dans les premières semaines, voire les premiers mois, de son existence.

Durant cette période, et malgré la mise en production officielle, certains sites n'hésitent pas à employer le terme « béta ». En effet, ce terme est habituellement utilisé en phase de test, avant le lancement du produit. Le concept de la préversion est de permettre aux *béta-testeurs*, un groupe restreint d'utilisateurs sélectionnés, de trouver des bogues et de permettre leurs corrections ou d'améliorer un logiciel avant sa sortie officielle.

Dans le cas du projet Pièces Avenue, il aurait été maladroit de communiquer sur l'aspect « béta » du site. En effet, cela aurait eu un impact négatif puisque les clients auraient pu penser que le site comporte encore des bogues et n'est pas totalement finalisé. C'est d'autant plus important puisqu'il s'agit d'un site e-commerce et que les clients entrent leurs informations bancaires et veulent donc un environnement complètement sécurisé.

Malgré tout, nous pouvons quand même parler d'une phase de test « grandeur nature » pour Pièces Avenue. Les premières semaines qui ont suivi le lancement du site nous ont en effet permis de corriger quelques aspects ergonomiques. L'exemple le plus frappant est celui du bouton « acheter » dans les listes et dans les pages de détail des produits. Ce bouton était à la base de couleur rouge. Or, le rouge est la couleur du danger et cela nous avait complètement échappé lors de la réalisation des maquettes.

Le simple fait de changer la couleur du bouton et de le passer sur un fond vert a permis une augmentation significative du nombre de commandes. Ce qui est intéressant ici, c'est de voir que dans le domaine du commerce électronique, un paramètre à priori insignifiant peut avoir de lourdes conséquences en terme de ventes et de chiffre d'affaire réalisé... D'une manière générale, cette période nous a donné l'occasion de tester certaines idées et de constater les corrélations entre les améliorations apportées et l'évolution des ventes.

V.8.2 L'optimisation du référencement

Une autre activité qui fait suite au lancement d'un site Internet est le *SEO* (pour *Search Engine Optimisation*), en français l'optimisation du référencement. Le référencement consiste à faire apparaître le site dans les moteurs de recherche. Maxime Castanié, notre expert dans ce domaine, s'est occupé de cette activité dans le cadre du projet Pièces Avenue.

Lorsque l'on parle de référencement, on pense tout de suite au moteur de recherche Google. Cette société américaine met à disposition des webmaster et des administrateurs un ensemble d'outils pour analyser et améliorer le référencement de son site Internet. Je vais en présenter quelques uns dans la suite du document.

V.8.2.1 Le sitemap et le fichier robot.txt

Le Google *sitemap* est un fichier au format XML qui permet de spécifier au moteur de recherche les urls des pages à référencer, avec éventuellement quelques options supplémentaires comme les dates et fréquences de dernières modifications ainsi que le poids de chaque page.

Le fichier robots.txt a un objectif opposé puisqu'il sert à bloquer certains moteurs de recherches. Il est surtout utilisé pour éviter le référencement de pages qui n'ont pas d'intérêt pour les internautes...

La génération du *sitemap* et l'écriture du fichier robots.txt ont été confiés à Hervé.

V.8.2.2 Google e-commerce Analytics

L'outil Google *Analytics* permet d'analyser l'audience d'un site Internet. Par exemple, il permet de connaître parmi les internautes naviguant sur le site Internet étudié le pourcentage d'hommes et de femmes ou encore la zone géographique depuis laquelle les internautes se connectent. Avec sa solution pour les boutiques en ligne, cet outil permet d'aller encore plus loin et de recouper ces informations avec les ventes réalisées. Ainsi, il est possible de savoir qui achète quoi et où sont les clients.

Dans la pratique, il s'agit d'un code javascript qui est exécuté lorsque le client valide sa commande et qui envoie un certain nombre d'informations aux serveurs de Google.

Evidemment, des questions d'éthiques se posent. Il faut aussi avoir conscience que Google peut tout à fait exploiter ces données et donc il s'agit d'une décision qui revient au client. Encore une fois,

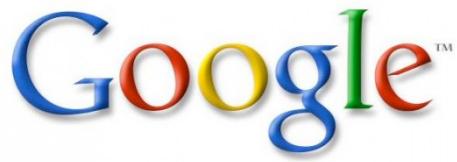


Figure 70 - Logo Google

```
- <urlset xsi:schemaLocation="http://www.sitemaps.org/sch
- <!--
-   created with Free Online Sitemap Generator w
- -->
- <url>
-   <loc>http://www.piecesavenue.com/</loc>
-   <changefreq>weekly</changefreq>
- </url>
- <url>
-   <loc>http://www.piecesavenue.com/marques/</loc>
-   <changefreq>weekly</changefreq>
- </url>
- <url>
-   <loc>
```

Figure 71 - Google sitemap

```
<title>receipt for your clothing purchase from Acme <
<script type="text/javascript">
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-XXXXX-X']);
  _gaq.push(['_trackPageview']);
  _gaq.push(['_addTrans',
    '1234', // order ID - required
    'Acme Clothing', // affiliation or store name
    '11.99', // total - required
    '1.29', // tax
    '5', // shipping
    'San Jose', // city
    'California', // state or province
    'USA' // country
  ]);

```

Figure 72 - Google e-commerce Analytics

l'intégration du code pour Google *Analytics* a été faite par Hervé.

V.8.2.3 Google e-commerce AdWords

Le dernier service que nous avons utilisé est le programme Google *AdWords*. Ce dernier permet aux entreprises d'acheter des mots clés en rapport avec leur activité et ainsi d'apparaître plus facilement dans le moteur de recherche Google.

Julien Lavault, qui était à l'époque ingénieur technico-commercial de la société Skalpel, avait déjà planifié le budget des *AdWords* lors de l'étude du plan d'affaires (*business plan*) et avant même le lancement du site. Il avait fixé un budget mensuel alloué aux *AdWords*. Celui-ci était décroissant pendant les deux premières années du site. En effet, il était prévu que la baisse du budget *AdWords* soit compensée par le développement du référencement naturel (que nous verrons plus loin).

A titre d'information, au lancement du site ce budget était d'environ 10 000€ par mois. Aujourd'hui, il se situe aux alentours de 2 000€.

V.8.2.4 Le référencement naturel

Le référencement naturel est à opposer aux programmes payants tel que celui présenté dans la partie précédente. Il s'agit ici d'apparaître dans les moteurs de recherche grâce à la pertinence du contenu et à certaines techniques de référencement que je vais présenter très rapidement dans ce chapitre.

La première chose qu'il faut avoir à l'esprit, c'est que le référencement est une science empirique dans le sens où personne aujourd'hui ne connaît les algorithmes des moteurs de recherche. Il faut donc modifier son site en tâtonnant et visualiser les résultats obtenus. Quelques fois le positionnement vis à vis d'un mot clé augmente et d'autres fois on obtient un résultat totalement inverse.

Lorsqu'un moteur de recherche va analyser un site, il va prendre en compte un certain nombre de paramètres, tel que :

- La vitesse de génération des pages,
- Les balises HTML « title », « description », « h1 », etc.,
- Le ratio nombre de mots clés / contenu de la page,
- Le nombre et la pertinence des « *back links* » (liens externes vers le site),
- Etc.

Cette liste n'est évidemment pas exhaustive et il y a énormément d'autres paramètres à prendre en compte pour améliorer le référencement, certains étant très complexes. De plus, ces derniers n'ont pas le même poids, la même valeur entre eux. Certains sont en effet plus importants que d'autres. Il faut aussi savoir que ces paramètres et leurs poids varient d'un moteur de recherche à l'autre. Par exemple, Bing n'analyse et ne qualifie pas une page de la même façon que Google. Et donc il est tout à fait possible d'apparaître en première page dans l'un et pas dans l'autre !

Il faut jouer sur chacun de ces facteurs pour améliorer le positionnement dans les moteurs de recherche. Certaines modifications, comme celles sur les balises HTML, sont relativement simples à mettre en œuvre, mais d'autres nécessitent plus de travail. D'autant plus que le référencement nécessite un travail constant car le réseau Internet est en perpétuelle évolution. De nouveaux acteurs, concurrents, produits arrivent et il faut sans cesse adapter son référencement à ce marché.

C'est finalement une activité assez proche de ce que j'ai pu étudier pendant les cours de mercatique dispensés au CNAM : l'entreprise est obligée d'adapter sa stratégie marketing au marché tout comme elle est obligée d'adapter son référencement aux évolutions du réseau Internet.

En tant que chef de projet web, je pense qu'il faut évidemment avoir une bonne vision du fonctionnement du référencement et des moteurs de recherche. Mais, j'estime aussi que ce travail est celui de spécialistes comme Maxime. D'autant plus qu'il s'agit d'une science aléatoire qui nécessite de faire beaucoup d'essais et qui est donc chronophage.

V.8.3 Les indicateurs du e-commerce

Les sites de ventes en ligne ont quelques indicateurs particuliers qui n'existent pas dans les autres types de sites web et qui sont propres au commerce électronique. Parmi ces indicateurs, on distingue **le taux de conversion** et **le panier moyen**.

Le **taux de conversion** est le pourcentage d'internaute effectuant un achat sur le site. Ce taux varie habituellement entre 0 et 3%. D'après la FEVAD (*Fédération de l'E-commerce et de la Vente À Distance*), la moyenne des taux de conversion de l'année 2011 se situe aux alentours de 2,4%. Ce qui veut dire que pour cent internautes naviguant sur le site, 2,4 internautes effectuent réellement un achat. Sur Pièces Avenue, ce chiffre est cependant inférieur à 1%. Cela veut dire qu'il y a encore une marge d'amélioration et que les clients ne vont pas au bout de leur achat. Aujourd'hui, je me focalise sur le gain de cette clientèle.



Figure 73 - Logo de la FEVAD

Le second chiffre est **le panier moyen**. Toujours d'après la FEVAD, le panier moyen sur les sites de ventes en ligne se trouve aux environs de 90€. Dans le domaine de la pièce automobile et dans le cas particulier de Pièces Avenue, ce chiffre est plus élevé puisqu'il atteint les 110€. Ce qui est intéressant avec ces deux chiffres, c'est qu'il est possible, via des calculs simples, d'estimer le chiffre d'affaire en fonction du nombre de visiteurs. C'est d'ailleurs la méthode utilisée pour l'élaboration du business plan.

Par exemple, imaginons que l'audience du site est de **5000** visiteurs uniques par jour. Le nombre de commandes avec un taux de conversion de **1%** est de **50** commandes par jours. Il suffit alors de multiplier ce chiffre par le panier moyen de **110€** pour arriver à un chiffre d'affaire quotidien estimé à **5500€**.

V.9 Bilan

Pendant la conception du site, mon rôle en tant que chef de projet a surtout consisté à analyser la problématique sous l'ensemble de ses angles. C'est une phase durant laquelle j'ai travaillé en grande partie seul. Mais contrairement à cette première phase, la réalisation a impliqué le concours de l'ensemble des acteurs du projet. Ici, le concept de **travail en équipe** prend alors tous son sens.

Notre objectif en tant qu'équipe a donc été de concrétiser toutes les décisions et tous les choix pris en amont. D'un point de vue graphique, nous avons tout d'abord travaillé sur l'élaboration d'un **logo et d'une maquette**. Cette maquette a ensuite été « **découpée** » en HTML avant que les développeurs ne l'intègrent sur le **front office**. Auparavant, ceux-ci avaient travaillé sur le **mapping objet-relationnel**. Enfin, le site a été **testé** et nous avons essayé d'optimiser son **référencement**. Toutes ces activités nécessitent autant de spécialistes et d'experts dans leurs domaines.

Dans ce contexte, je pense que le chef de projet a trois grands rôles que je vais développer dans ce bilan. Le premier est la **gestion du processus** de réalisation du site Internet. Le second concerne la **gestion humaine** et la direction de l'équipe de développement. Enfin, le troisième et dernier rôle du chef de projet est à mon sens la **gestion du client** et des partenaires.

V.9.1 La gestion du processus

L'approche par les processus est un bon moyen de comprendre la réalisation d'un site Internet. Il faut imaginer chacune des activités de la phase de réalisation comme un **processus** qui nécessite des **entrées** et qui produit des éléments en **sortie**. Par exemple, dans le cas du découpage HTML, il y a deux éléments en entrée : la maquette et le logo. Le résultat de ce processus est l'ensemble des fichiers HTML, CSS et Javascript obtenus. Cette sortie sera utilisée ensuite en entrée par les développeurs pour l'intégration du front office.

Il y a donc un certain **cheminement** à respecter pour arriver à un site Internet depuis les documents initiaux. Le premier rôle du chef de projet est à mon sens de **définir** et **coordonner** ces processus entre eux. Ce travail de coordination est primordial car il permet d'avoir une excellente vision de l'état d'avancement du projet. En effet, je savais à tout moment à quelle étape de la phase de réalisation nous étions et donc, ce qu'il restait à faire.

Pour continuer dans cette approche par processus, je pense que le second rôle du chef de projet est de faire en sorte que chacun **ait en entrée** les documents, données, fichiers et ressources nécessaires pour la bonne réalisation du processus. Le chef de projet doit également s'assurer de la **qualité de la sortie** de chacun de ces processus. L'objectif étant d'avoir une base saine, conforme aux spécifications et validée pour le processus suivant.

Enfin, je pense que le chef de projet doit être capable gérer au mieux son temps et celui de son équipe. Il s'agit évidemment de faire **respecter le planning**, mais aussi de savoir **faire face aux aléas**. En effet, en cas de dépassement de délais, de retards à la livraison ou encore d'ajout de fonctionnalités, il doit être capable d'adapter le planning et d'organiser le travail de chacun des membres pour réduire autant que faire se peut les pertes de temps, et donc d'argent.

V.9.2 La gestion de l'équipe

Le second rôle du chef de projet est la gestion de l'équipe et des acteurs du projet. Cet aspect managérial est **déterminant** pour la bonne conduite du projet et nécessite des qualités humaines tel que l'écoute, la compréhension, le respect, la sincérité, l'équité ou encore la tolérance. Je pense que le chef de projet ne doit pas perdre de vue que **l'humain est au cœur du projet informatique**.

Pour moi, le point le plus important est qu'un bon chef de projet doit être capable de **motiver** son équipe et de faire en sorte que chacun des membres de cette équipe soit heureux dans son travail. Car c'est de cette manière uniquement que le chef de projet pourra tirer le meilleur de son équipe et qu'il obtiendra une productivité maximale. Une des anecdotes que je peux partager à ce sujet est le développement de *l'easter egg* (œuf de Pâques, en français) que nous avons intégré à Pièces Avenue.



Figure 74 - Easter egg de l'ajout au panier

Un *easter egg* est une fonctionnalité cachée et amusante intégrée dans un programme informatique. Sur le site Internet Pièces Avenue, il est par exemple possible de jouer avec un camion qui apparaît lors de l'ajout d'un produit dans le panier et qui sert de message de confirmation pour cette opération.

L'*easter egg* permet aux développeurs de sortir (un peu !) du cadre du projet, de faire des choses amusantes et de prendre du plaisir au travail.

Mais la gestion de l'équipe ne consiste pas uniquement à motiver les « troupes ». Il est le guide, le meneur ou leader de l'équipe. Le chef de projet doit être capable de **résoudre tout type de conflits** qui peuvent survenir tout au long du projet. Il doit aussi savoir gérer les problématiques qui concernent les **connaissances et la formation** de ses collaborateurs. Et, évidemment, il doit faire tout cela en impactant au minimum le projet et en prenant au maximum en considération le bien être de son équipe.

Je voudrais terminer en rappelant que le chef de projet **a besoin** du soutien de ses collaborateurs car c'est avant tout grâce à eux que le projet avance. Comme je l'ai mentionné plus haut, ce soutien est obtenu grâce aux qualités humaines du chef de projet, au respect de ses collaborateurs ainsi qu'à l'attention portée à leurs conditions de travail.

Enfin, je pense que le domaine informatique nécessite un haut niveau de compétence et donc que l'approche managériale ne doit pas être trop rigide, envahissante. Peut être est-ce du à ma jeunesse, mais je n'arrive pas à comprendre par exemples les entreprises dans ce secteur qui adoptent des heures de travail fixes. A mon sens, il faut donner de la **liberté** aux collaborateurs afin d'en tirer le **meilleur**. Ceci est clairement valable dans le domaine du graphisme dont le pilier est la créativité, mais je pense que même les développeurs ont besoin d'une certaine forme de liberté, notamment dans le cadre de la conception de leurs programmes ou de certains algorithmes. C'est d'ailleurs ce que l'on voit dans de grandes sociétés (le meilleur exemple étant Google) aux Etats Unis.

J'ai vraiment essayé d'avoir cette démarche dans ma façon de gérer ce projet. Je voulais donner de la liberté sans pour autant être trop laxiste et permettre trop de dérives... Et je pense que cela a été payant pour Pièces Avenue.

V.9.3 La gestion du client

Durant cette phase, la **gestion du client et des partenaires** est le troisième et dernier rôle du chef de projet. Ici plus encore qu'ailleurs, le chef de projet est l'interlocuteur frontal de la maîtrise d'ouvrage. Cette position centrale lui confère plusieurs responsabilités dont les principales sont à mes yeux : le *reporting* et les retours sur l'évolution du projet, le cadrage ainsi que la prise en compte des nouvelles demandes du client, la négociation (qu'il s'agisse de délais, de fonctionnalités, ou du budget).

Avec Stéfan, nous nous réunissions souvent et nous organisions des appels téléphoniques via Skype pour aborder ces trois sujets. Et j'en ai tiré deux grandes leçons. Tout d'abord, j'ai compris qu'il était important **d'historiser tous les échanges** avec les partenaires et les clients. En particulier, il faut enregistrer toutes les demandes de modification et toutes les décisions prises. Cela dans le but d'avoir une garantie et un support servant de preuve en cas de mauvaise foi ou de contestation des clients (cela arrive malheureusement).

Le second enseignement que j'ai tiré au contact du client est qu'il faut constamment essayer de maximiser les pertes de temps à imputer au client et/ou aux partenaires et de minimiser celles causées par l'équipe de développement. Par exemple, notre client avait reçu le kit de paiement bancaire avec beaucoup de retard. Dans ce cas de figure, le chef de projet doit savoir insister sur la perte de temps induite et les complications que cela peut avoir sur le projet.

Avec le client, le chef de projet doit être capable de saisir toutes les **opportunités de grappiller du temps**. Evidemment, ne pas avoir ce kit de paiement était vu par le client comme un blocage et en tant que chef de projet, j'ai insisté sur cela. Or, dans la pratique, cela ne nous a pas beaucoup impacté, car nous connaissions parfaitement le système de paiement développé par Atos Origin et donc nous étions capable de le mettre en place sans même l'avoir reçu.

Enfin, je voudrais terminer ce bilan en rappelant une des caractéristiques que je juge importante pour un chef de projet : il doit être un **bon communiquant**. Et il doit l'être non seulement avec son équipe, mais c'est encore plus le cas avec les partenaires et le client où sa capacité à convaincre et à argumenter est importante, notamment dans les différentes phases de négociations. Il doit également être capable de **tirer avantage** de toutes les situations et aléas du projet. Et pour finir, je pense qu'il doit avancer dans le projet avec **une grande assurance**, qui peut être obtenue avec un historique des choix et des décisions prises tout au long du projet.



Compte rendu de réunion Pièces Avenue

PROJET & PARTICIPANTS

	Code Projet : 099GWD	
Projet:	Pièces Avenue	
Emetteur:	Hakan EBABIL	Tél : + 33 3 88 29 44 56 Mail : hakan@skalpel.fr
Participants :	Stéfan Edet, Pièces Avenue Hakan Ebabil, Skalpel	
Date de réunion:	05/02/09 de 10h15 à 10h45	
Date d'émission:	05/02/09	
Prochaine réunion:	N.C.	

ORDRE DU JOUR

1. Remarques de Pièces Avenue sur les maquettes et demandes de complément d'information
2. Suivi des relations et état d'avancement avec les partenaires de Pièces Avenue

Figure 75 - Compte rendu de réunion

Conclusion

A l'époque où nous avons conçu et réalisé le site Internet Pièces Avenue.com, je n'avais pas encore terminé mes études d'ingénieur, loin de là.

J'avais certes ce sentiment qu'il me restait encore beaucoup de choses à apprendre : non seulement d'un point de vue **théorique** où je n'avais pas encore validé l'ensemble des unités d'enseignement en rapport avec la gestion de projet, mais aussi **sur le terrain**, en mettant en pratique ces acquis. Toutefois, je n'avais pas d'appréhension particulière. Je savais que j'en étais capable.

Tout d'abord, j'avais une totale confiance en mes **capacités techniques**. Je connaissais suffisamment le domaine du web et l'ensemble des activités de ce domaine. De part nos précédents échecs, j'avais aussi une **bonne vision des grandes étapes du projet** ainsi que de la procédure à suivre pour arriver au mieux à notre objectif et en étant le plus productif possible et en évitant nos erreurs. Mais ce n'était évidemment pas suffisant.

Je pense qu'une des grandes chances que j'avais sur ce projet était que je connaissais parfaitement **l'équipe** de développement ainsi que **les individualités** qui la comptaient. En effet, j'avais déjà travaillé plusieurs années avec certains d'entre eux et je savais donc quel type de management je devais adopter, comment je devais communiquer et quel type de rapport je devais entretenir pour obtenir le meilleur de chacun d'eux.

Enfin et par dessus tout, j'avais une grande confiance dans mes **capacités humaines**. Je savais que je serais un élément moteur, que je serais capable de motiver l'équipe, mais aussi d'être juste et impartial, que je pourrais m'adapter aux situations difficiles que l'on peut vivre tout au long d'un projet informatique. C'est peut être du à mon caractère, un peu hyper actif mais toujours souriant !

Dans l'ensemble, j'estime (et j'espère que mes supérieurs et clients partagent eux aussi cet avis) que ce projet était une **réussite d'un point de vue technique et du budget**. Les *dead-lines* ont été quelque peu dépassés mais, globalement la durée du développement était conforme à celle estimée. D'un point de vue personnel, et si l'on prend en compte le contexte, mes connaissances et mon expérience, je pense avoir joué mon rôle de chef de projet au mieux et en donnant **le meilleur de moi même**. Ce fut donc pour moi une expérience très positive.

Mais évidemment, tout n'a pas été rose sur ce projet et on ne peut donc pas parler de succès total pour Pièces Avenue. Avec le recul, je pense qu'il y a bien des choses sur lesquelles je peux encore **m'améliorer**. En particulier, je pense que je dois être plus appliqué au sujet de la **gestion de la qualité** (meilleure gestion documentaire, des tests qui ont été omis, etc.) ainsi que sur **l'aspect financier**. A ce sujet, j'aurai vraiment aimé aider Julien Lavault à estimer et définir le budget du projet.

Ce que j'ai beaucoup apprécié tout au long ce projet, c'est que j'ai instinctivement pris des décisions en tant que chef de projet et que ces décisions se sont révélées être les bonnes, puisque j'ai constaté avec étonnement (et plaisir) que plusieurs enseignants et professeurs

ont confirmés mes choix durant leurs cours. C'était d'ailleurs amusant par moment car j'avais vraiment l'impression d'avoir déjà assisté au cours ! Ce que je retiens aussi, c'est l'importance des unités d'enseignement générales que je pensais à la base peu utiles (tel que le marketing, la communication et même la comptabilité) et qui permettent de comprendre tout l'environnement qui gravite autour de l'aspect technique d'un projet informatique.

Enfin, je voudrais conclure ce mémoire en donnant un point de vue très personnel sur le rôle de chef de projet informatique et pourquoi je l'apprécie.

Tout d'abord, je pense que ce poste est **gratifiant** car il est central dans le projet. Le chef de projet est celui à qui il faut se référer et avec lequel tout le monde doit communiquer. Il implique aussi de grandes **responsabilités** car il faut être constamment au contact du client, des partenaires, des collaborateurs et que chaque choix, chaque décision impacte l'évolution du projet. Il est donc important d'avoir **confiance** en soi, de ne pas avoir peur des responsabilités et de toujours être le moteur du projet.

Ce que j'aime beaucoup dans ce poste, c'est qu'il permet de **s'améliorer** constamment. En effet, je pense que c'est un poste qui permet de rencontrer et de travailler avec des personnes qui n'ont pas le même cursus, les mêmes connaissances et il est possible de **profiter de l'expertise et de l'expérience** de chacun de ces collaborateurs. A ce propos, je pense aussi qu'une autre des qualités (j'en ai déjà cité tellement!) du chef de projet est la **curiosité**.

Enfin, je me suis rendu compte en écrivant ce mémoire que le chef de projet doit avant tout se connaître. Il doit savoir quelles sont ses limites, ses aptitudes, ses connaissances, il doit connaître son caractère ainsi que sa personnalité... Et je pense très sincèrement que ce travail de rédaction a été l'occasion pour moi de faire une rétrospection et de me rapprocher un peu plus de ce rôle de chef de projet puisqu'il m'a lui aussi permis de mieux me connaître à tous les niveaux !

Bibliographie

Ouvrages imprimés

Matheron J-P, 2002. Comprendre MERISE : outils conceptuels et organisationnels. 5^{ième} éd. rev. et augm. Eyrolles, France, 288 p.

Sites web

AFNOR. Portail AFNOR. Disponible sur :

<http://www.afnor.org>

DEVELOPPEZ.COM. Conception et génie logiciel : actualités, forums avec sondages et débats, cours, Faq, blogs, critiques de livres. Disponible sur :

<http://conception.developpez.com/>

FEVAD. Fédération e-commerce et vente à distance. Disponible sur :

<http://www.fevad.com>

W3C. World Wide Web Consortium (W3C). Disponible sur :

<http://www.w3c.org>

WIKIPÉDIA. Portail:Informatique – Wikipédia. Disponible sur :

<http://fr.wikipedia.org/wiki/Portail:Informatique>

Liste des figures

Figure 1 - Logo Skalpel	10
Figure 2 - Logo Infomoto	10
Figure 3 - Capture d'écran du site Moto 85	15
Figure 4 - Capture d'écran du site Motoshopping	16
Figure 5 - Capture d'écran du site Top's Cars.....	17
Figure 6 - Avatar Louis Saunders.....	16
Figure 7 - Avatar Killian Ebel	16
Figure 8 - Avatar Hervé Weltzer	17
Figure 9 - Avatar Mickael Knauer.....	18
Figure 10 - Statistiques VAD en 2010	20
Figure 11 - Produits / Services achetés en ligne.....	20
Figure 12 - Capture d'écran d'une page de détail sur Pièces Avenue	24
Figure 13 - Triangle de la qualité	22
Figure 14 - Logo CXA	22
Figure 15 - Interconnexion des systèmes d'information.....	27
Figure 16 - Architecture Web.....	28
Figure 17 - Logo Celeonet.....	26
Figure 18 - Logo PHP.....	26
Figure 19 - Logo Doctrine	26
Figure 20 - Logo Zend Framework	30
Figure 21 - Logo Dwoo	27
Figure 24 - Capture d'écran de l'interface d'administration	31
Figure 22 - Logo Mootools.....	28
Figure 23 - Logo Ext JS.....	28
Figure 25 - Logo SubVersioN	29
Figure 26 - Logo Ti-Pièces	30
Figure 27 - Les deux phases de la réalisation d'un site Web	34

Figure 28 - Etapes de la phase de conception d'un site Web.....	35
Figure 29 - Etapes de la phase de réalisation d'un site Web.....	36
Figure 30 - L'étude des interfaces.....	35
Figure 31 - Logo TecDoc	36
Figure 32 - Prototype TecDoc	38
Figure 33 - Organisation de l'équipe projet	44
Figure 34 - Interface de l'outil Active Collab.....	46
Figure 35 - Interface de l'outil Trac.....	47
Figure 36 - Historique des modifications du cahier des charges.....	48
Figure 37 - Versions du cahier des charges.....	46
Figure 38 - Logo Active Collab.....	46
Figure 39 - Interface Web SVN.....	50
Figure 40 – Architecture des serveurs de développement et de production.....	47
Figure 41 - Logo AFNOR	50
Figure 42 - Contenu du cahier des charges	54
Figure 43 - Analyse descendante, niveau A-0.....	55
Figure 44 – Analyse descendante, niveau A-1	56
Figure 45 - Diagramme de cas d'utilisation.....	54
Figure 46 – Logigramme	58
Figure 47 - Arborescence 1/2	62
Figure 48 - Arborescence 2/2	63
Figure 49 - Page du panier dans l'annexe à l'arborescence.....	64
Figure 50 - Entonnoir de conversion	67
Figure 51 - Zoning de la page d'accueil (V3).....	68
Figure 52 - Capture d'écran de la page d'accueil du site	69
Figure 53 - Question n°3 du pré-briefing graphique	72
Figure 54 - Découpage en tâches.....	73
Figure 55 - Utilité de l'arborescence dans le découpage et l'estimation	74

Figure 56 - Travaux menés simultanément	74
Figure 57 - Logo Pièces Avenue	77
Figure 58 - Première version de la page d'accueil du site	83
Figure 59 - Cinquième version de la page d'accueil du site	84
Figure 60 - Huitième et dernière version de la page d'accueil du site.....	85
Figure 61 - Structure modulaire	88
Figure 62 - Processus de commande dans le back office	92
Figure 63 - Logo W3C	91
Figure 64 - Découpage HTML de la liste de produit	92
Figure 65 - Héritage des templates	98
Figure 66 - Extension d'un template.....	95
Figure 67 - Liste des templates.....	95
Figure 68 - Assignation d'une variable	96
Figure 69 - Affichage d'une variable dans un template	96
Figure 70 - Logo Google	99
Figure 71 - Google sitemap	99
Figure 72 - Google e-commerce Analytics	99
Figure 73 - Logo de la FEVAD	101
Figure 74 - Easter egg de l'ajout au panier	103
Figure 75 - Compte rendu de réunion.....	108

Liste des tableaux

Table 1 - Interactions avec les systèmes d'information des partenaires	41
Table 2 - Métadonnées dans l'annexe à l'arborescence	65
Table 3 - Utilité des documents de conception	77

La gestion d'un projet web e-commerce : Pièces Avenue

Mémoire d'Ingénieur C.N.A.M., Strasbourg 2012

RESUME

Pièces Avenue est un site Internet e-commerce dans le secteur de la pièce automobile. Ce mémoire présente les rôles ainsi que les tâches d'un chef de projet web tout au long de la conception et de la réalisation d'une boutique en ligne.

La première phase englobe l'étude de la faisabilité, la rédaction du plan d'assurance qualité et de celle du cahier des charges fonctionnel, la conception d'une arborescence ainsi que de son annexe, l'élaboration des zonings, la complétion du briefing graphique et enfin l'estimation et le planning. Durant la seconde phase, il a fallu créer un logo, ainsi qu'une maquette qui est ensuite découpée pour que les navigateurs web puissent l'interpréter. Nous avons aussi travaillé sur la base de données, et plus particulièrement sur le mapping relationnel. L'intégration du back, du front, ainsi que les tests ont été les dernières étapes de cette phase.

Le mémoire relate mon expérience en tant que chef de projet, les difficultés rencontrées ainsi que les solutions apportées avec des choix argumentés. Enfin, un point de vue plus personnel est donné quant aux qualités nécessaires pour ce rôle.

Mots clés: Web, gestion, projet, e-commerce, automobile, VAD, conception, analyse, réalisation, développement.

SUMMARY

Pièces Avenue is an e-commerce website in the automobile sector. This dissertation presents the roles and tasks of a web project manager during the conception and the realisation of an online shop.

The first phase includes the feasibility study, the writing of the quality assurance plan and of the functional specifications, the conception of the page tree and its annex, the elaboration of the wireframes, the filling up of the graphical briefing, and finally the estimation and planning. During the second phase, we had to create a logo and a mockup which will be cutted so that web browsers can interpret it. We also worked on the database, and more specifically on the object relational mapping. The back and front office's integration and the quality control are the last stages of this phase.

This dissertation relates my experience as a project manager, the issues faced and the solutions found with argued choices. Finally, a more personal point of view is given towards the needed qualities for this role.

Keywords: Web, management, project, e-commerce, automobile, distance selling, PPO, conception analysis, realisation, development.