

DESIGN WEB

Préface de Kaelig **Deloumeau-Prigent**

Jérémie **Patonnier**

Rudy **Rigot**

Projet Responsive Web Design

DU RECUEIL DES BESOINS À LA MISE EN LIGNE

© Groupe Eyrolles, 2013
ISBN : 978-2-212-13713-2

EYROLLES

PRÉFACE

Lorsque j'ai commencé à bricoler des pages web dans le bureau de la maison familiale, je n'avais pas de connexion Internet. Pour mettre ces pages en ligne, l'opération était un poil compliquée : il s'agissait de compresser au maximum les fichiers (et notamment les images) pour les stocker sur une disquette (1,44 Mo). Celle-ci étant parfois capricieuse, chaque fichier était dupliqué une ou deux fois, histoire de s'assurer qu'il serait récupérable. Je me rendais ensuite à la médiathèque (équipée d'ordinateurs connectés à Internet) pour téléverser les mises à jour via FTP depuis les disquettes.

Quel rapport entre cette anecdote et la conception de sites web aujourd'hui ? La démocratisation des connexions haut débit a décomplexé les designers web, dont les pages ont enflé jusqu'à peser des poids hier délirants (plusieurs mégaoctets). Or, retour de bâton inattendu, cette inflation fait à nouveau obstacle aux performances des sites web et l'on revient à une recherche forcenée de l'optimisation des pages. La rapidité d'accès à l'information est devenue cruciale, alors que les conditions de navigation sur mobile sont aussi variables qu'imprévisibles. La spectaculaire pénétration du marché par les smartphones et les tablettes a bouleversé notre modèle de conception, chose que personne n'aurait soupçonnée avant l'arrivée de l'iPhone en 2007. Tandis que nous produisions des produits destinés à des plates-formes vieillissantes, les consoles de jeu portables se sont mises à intégrer des navigateurs web dignes de ce nom, puis les consoles et platines de salon ont suivi la tendance. Demain, il sera compliqué de trouver un téléviseur qui ne soit pas équipé d'un navigateur web. Au fur et à mesure que le marché se standardisait (résolutions d'écran plus uniformes, connexions ADSL...), nous nous sommes confortablement habitués à une conception en 1 024 × 768 pixels – toujours considérée comme la norme dans de nombreuses agences – sans nous soucier de la multiplicité des périphériques inhérente au *World Wide Web*. Nous sommes désormais mis devant cette évidence : nous avons cassé le Web.

La question qui se pose est celle de la livraison de nos contenus à ce panel grandissant de périphériques et d'usages. Nous pouvons nous évertuer à produire des sites spécifiques à chaque type de plate-forme et des applications dédiées, mais à long terme, cette approche n'est viable ni financièrement ni sur un plan pratique.

Les expérimentations récentes autour du trio des langages du Web que sont HTML, CSS et JavaScript montrent qu'il est possible de produire des sites qui s'adaptent intelligemment aux plates-formes de nos utilisateurs, quel que soit le contexte dans lequel ils accèdent à l'information. Le concept de base derrière cette idée, qu'Ethan Marcotte a nommé *responsive web design* - ou *RWD* pour les intimes - a inspiré des méthodes qui en sont encore à l'état embryonnaire, mais qui semblent tellement prometteuses qu'elle sont sans aucun doute le futur de la chaîne de conception web. Adopter cette approche permettra d'architecturer des produits qui fonctionnent sur les navigateurs d'hier et d'aujourd'hui, tout en restant accessibles sur les périphériques du futur.

En creusant le sujet, on se tournera vers l'*adaptive web design* (design adaptatif), grâce auquel on saura proposer des déclinaisons du même document pour qu'il se transforme de manière adéquate sur des périphériques aussi disparates que des lunettes connectées, des interfaces gestuelles, des consoles de jeu, une montre intelligente...

Le livre que vous tenez entre les mains est le témoin de l'intérêt que la communauté des professionnels du Web porte au responsive web design, mais il ne s'agit pas ici de suivre un effet de mode passager. Sensibiliser le client, former l'équipe, garder un cap précis tout en étant flexible en cours de projet... Tant de défis techniques et humains auxquels ont été confrontés les auteurs en conditions réelles. Merci à eux pour ce précieux partage.

Kaelig Deloumeau-Prigent

Intégrateur et auteur du blog *Le ministère de l'intégration*
Auteur du livre *CSS maintenables avec Sass et Compass*,
aux éditions Eyrolles
<http://kaelig.fr>

TABLE DES MATIÈRES

1		Avant-propos
5		Remerciements
		CHAPITRE 1
7		Quelques concepts clés pour bien démarrer
8		Responsive ? Adaptatif ?
11		Amélioration progressive et dégradation élégante
14		Le lâcher-prise du Web
15		Rappel sur les méthodologies de gestion de projet
		CHAPITRE 2
23		Recueillir les besoins
24		La prise de contact : l'étape fondatrice
27		Faire parler le client et l'écouter
30		Le projet est-il manifestement responsive ?
		CHAPITRE 3
37		Rédiger le cahier des charges
38		Les principales contraintes d'un projet
40		Définir le périmètre du projet
		CHAPITRE 4
45		Monter l'équipe projet
46		Comment orchestrer les rôles projet ?
48		L'évolution des rôles projet typiques dans un contexte responsive
56		Les changements en cours dans les métiers du Web
60		L'amélioration continue des expertises

	CHAPITRE 5
63	La pré-conception
63	Les approches de conception
67	Gérer les gabarits
74	Les outils nécessaires à la conception
	CHAPITRE 6
77	La conception
77	La répartition des rôles
79	Le prototypage
	CHAPITRE 7
87	Le design graphique
87	Organisation et chiffrage
91	Les livrables graphiques
	CHAPITRE 8
97	L'intégration
98	La gestion des images
104	Définir une stratégie de tests
106	Documenter le projet
	CHAPITRE 9
109	Le développement back-end
110	Gérer les risques à l'intérieur du périmètre intégré
115	Gérer les risques à l'extérieur du périmètre intégré
121	Gérer les images fluides côté serveur
	CHAPITRE 10
127	Rôle et formation des contributeurs éditoriaux
128	La contribution d'images

131	Texte : de nouvelles contraintes
135	Tenir compte de la mise en page
	CHAPITRE 11
139	Effectuer la recette applicative
140	Se préparer pour une recette efficace
145	Pendant la recette applicative
	CHAPITRE 12
149	Après la mise en production
151	Le dilemme habituel : trouver le bon consultant de TMA
153	Une difficulté : la continuité du travail d'intégration
155	Après-propos : le mot de la fin
157	À propos des deux auteurs
159	Index

AVANT-PROPOS

Chaque événement de progrès ressemble un peu à un galet qui tombe dans un étang : au début, beaucoup de bruit vient de la surface, et chacun y va de son opinion sur la direction qu'il prendra. Puis, le galet tombe lentement, l'eau et les autres galets se positionnent progressivement pour lui permettre d'avancer à son rythme, prenant le temps de la réflexion à chaque goutte. Alors qu'il heurte le fond et prend sa position finale, chacun s'étonne de tout le chemin parcouru et reste un peu en émoi sur ce qui a été construit, ce qui a changé. Enfin, progressivement, le galet se fait oublier... Il fait maintenant partie d'un tout, il est maintenant une partie du monde et plus personne ne s'en étonne.

C'est ce qui s'est historiquement passé pour chaque innovation s'étant avérée un progrès : quelle est la dernière fois que vous vous êtes enthousiasmé du confort de vie que vous offre votre voiture ou votre transport en commun ? Quand votre grille-pain vous a-t-il fasciné pour la dernière fois ?

Alors que le concept de mobilité se consolide enfin et que nous en sommes déjà à l'étape de l'émoi devant le chemin parcouru, la sphère du Web (à défaut du grand public) est aujourd'hui en train d'observer l'avancée progressive que réalise l'innovation du *responsive web design*, qui consiste à concevoir un site pour qu'il s'adapte à toutes les largeurs d'écrans possibles. Les premières observations empiriques de sites refondus selon ces techniques témoignent d'un véritable progrès : le taux de rebond mobile a baissé de 26 % depuis la refonte du site du magazine *Time*, le chiffre d'affaires du site O'Neill a augmenté de 101,2 % pour les visiteurs sur iPhone/iPod et de 591,4 % sur Android, le chiffre d'affaires du site Skinny Ties a augmenté de 377,6 % pour les visiteurs sur iPhone et de 42 % tous visiteurs confondus, les inscriptions en ligne ont augmenté de 63 % sur le site du Regent College...¹ Certes, ce n'est sans doute pas le

1. Source : <http://www.lukew.com/ff/entry.asp?1691>

responsive web design en soi qui est la cause de ces chiffres enthousiasmants, mais plus largement la prise en compte des terminaux modernes, notamment par le design.

Grâce à ces premiers retours, parmi les clameurs que nous avons entendues lorsque le concept a frappé la surface, nous savons aujourd'hui que celles qui le décrivaient comme un effet de mode passerager semblent s'être trompées. Le responsive web design est bel et bien censé rejoindre le fond de l'étang, celui où toutes les innovations qui restent dans le temps reposent sous nos yeux dans leur état final.

Pourtant, aujourd'hui, il en est encore à un état d'avancement assez jeune de son industrialisation. Charge à nous de pousser chaque goutte d'eau de manière réfléchie.

Que trouver dans ce livre ?

Si vous avez précédemment manipulé l'outil technique que sont les requêtes de média CSS (*CSS media queries*), vous êtes sans doute tombé comme tout le monde en admiration devant la simplicité d'utilisation de cet outil, qui apporte pourtant des possibilités incroyables. Vos premières mises en œuvre (votre site personnel, peut-être ?) ont été tellement simples à réaliser que vous avez peut-être imaginé, au premier abord, être devant un concept à la courbe d'apprentissage très favorable.

Peut-être avez-vous ensuite eu l'occasion de mettre en œuvre des designs *responsive* sur des projets en équipe. Si c'est le cas, c'est là que vous avez dû rencontrer vos premiers véritables problèmes : designers ne sachant pas comment structurer leurs livrables, intégrateurs levant des alertes rouges d'infaisabilités techniques et, évidemment, tout le retard qui s'accumule à force de devoir invalider du travail à refaire...

Et si vous avez eu ensuite l'occasion de participer à un projet responsive d'envergure, vous vous êtes alors aperçu que la

complexité de mise en œuvre de ce type de projets grimpe en flèche en même temps que la complexité du projet.

En prenant un peu de recul, vous vous êtes alors certainement fait cette réflexion, que nous nous sommes faite nous-mêmes : ce n'est pas le design qui se complique, ce sont nos habitudes et nos outils, que nous avons forgés autour des pages à largeur unique, que nous avons de plus en plus de mal à tordre pour ce nouveau contexte.

Dans ce livre, nous avons essayé d'aider à résoudre tous les problèmes typiques que nous avons fréquemment rencontrés sur des projets d'envergure (et de taille moyenne), en comparaison avec les projets dont nous avions l'habitude précédemment. Vous, lecteur, qui vous posez des questions sur la manière dont vous devriez aborder votre prochain projet qui vous inquiète tant, nous avons essayé de vous prendre par la main pour dérouler les étapes et réflexions importantes qui devront jaloner son déroulement.

Ce livre ne contient absolument pas de recettes toutes faites à appliquer à tout projet ; nous considérons l'industrialisation de ce type de projets comme étant un travail en cours de découverte et n'avons pas toutes les réponses, pour la simple raison qu'aujourd'hui, personne ne les a. En revanche, nous avons nos réponses à nous, notre vision des problématiques et, dans la mesure du possible, nous avons essayé de les accompagner d'un maximum de pistes de résolution à adapter à votre situation.

Un guide pratique, mais pour qui ?

Ce livre s'adresse à toute personne étant intervenue, sur le point d'intervenir ou souhaitant intervenir à terme sur des projets utilisant le responsive web design et désireuse de perfectionner sa réflexion sur la meilleure manière d'y participer.

Ce livre aura une résonance dans son intégralité pour les chefs de projet, puisqu'ils sont les témoins de toutes les étapes du projet. Il sera globalement intéressant pour les autres profils, mais ils y trouveront des chapitres qui leur seront davantage destinés que d'autres, qu'ils aient le rôle du client (ou *product owner*), du designer, du commercial, du développeur, etc.

Afin de rendre plus efficace son objectif de guide pratique pour tous ces profils, nous avons décidé de le construire dans l'ordre chronologique du déroulement d'un projet web typique. Nous vous conseillons fortement de lire le premier chapitre, qui vous servira de base de connaissance théorique pour vous lancer sereinement dans les autres sujets ; mais n'hésitez pas, ensuite, à choisir entre lire le reste dans l'ordre chronologique ou privilégier les chapitres qui vous intéressent le plus. Il a été construit de sorte à rendre ce mode de lecture possible.

[Le blog du livre !](#)

Retrouvez les auteurs et échangez avec eux sur :
<http://projetresponsive.fr>

Quelques concepts clés pour bien démarrer

Le Web est une science relativement neuve, chargée de bon sens et de bonnes pratiques qui s'utilisent au fur et à mesure qu'elles se découvrent, et qu'il faudra découvrir au fur et à mesure qu'elles évoluent naturellement. Ce bon sens se déduit de valeurs simples qui s'énoncent clairement, comme l'accessibilité (universalité d'accès des contenus et services à tous les utilisateurs, quel que soit leur matériel, handicap...), les performances (qu'elles soient du côté des technologies serveur ou client), l'optimisation pour les moteurs de recherche (SEO, *Search Engine Optimization*), etc. Dans le contexte du responsive web design, le bon sens de notre réflexion viendra de plusieurs notions qu'il nous faudra également énoncer clairement, pour être sûr de bien les comprendre avant de nous lancer la tête la première.

RESPONSIVE ? ADAPTATIF ?

Qu'est-ce que le responsive web design ?

Avant de décortiquer finement un concept tel que le *responsive web design*, il serait bienvenu de le définir. Un site ayant été conçu selon ce principe est un site dont le design (fonctionnel ou graphique) change en fonction de la taille de l'écran sur lequel il est consulté (ou de la fenêtre du navigateur, selon votre choix de conception). En français, la traduction « design réactif » serait la plus juste, mais force est de constater que c'est le terme anglais qui s'est imposé.

L'amalgame est facile, mais ce concept est différent de celui des *media queries* ; ces dernières sont un outil technique de la spécification CSS3 qui permet d'exécuter des règles CSS en fonction de certaines conditions (la taille de l'écran n'est qu'une partie des 13 règles existantes dans la recommandation, à l'état final depuis le 19 juin 2012).



Media Queries

W3C Recommendation 19 June 2012

This Version:

<http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/>

Latest Version:

<http://www.w3.org/TR/css3-mediaqueries/>

Latest Editor's Draft:

<http://dev.w3.org/csswg/css3-mediaqueries/>

Previous Version:

<http://www.w3.org/TR/2012/PR-css3-mediaqueries-20120426/>

Disposition of Comments:

<http://www.w3.org/Style/2012/MediaQueriesDisposalOfComments.html>

Editors:

Florian Rivoal <florianr@opera.com>

Previous Editors:

Håkon Wium Lie <howcome@opera.com>

Tantek Çelik <tantek@cs.stanford.edu>

Daniel Glazman <daniel.glazman@disruptive-innovations.com>

Anne van Kesteren <annevk@opera.com>

Please refer to the [errata](#) for this document which may include some normative corrections.

FIG. 1-1 : Capture d'écran de la recommandation finale CSS3 media queries publiée par le W3C

<http://xav.cc/css2mq>¹

L'évolution de la définition du terme « responsive web design »

Lorsqu'Ethan Marcotte a initialement écrit l'article « *Responsive Web Design* » sur le blog A List Apart, puis le livre du même nom (initialement aux éditions A Book Apart, puis traduit en français et publié aux éditions Eyrolles), la définition qu'il proposait n'était pas exactement celle que nous venons de poser. Il le définissait comme l'alliance de plusieurs technologies (dont les *CSS3 media queries*, mais aussi les grilles fluides et les images fluides) pour permettre une nouvelle approche moins contraignante (d'un point de vue design) de la fabrication des pages flexibles.

<http://xav.cc/rwd-article>²

 E. Marcotte, *Responsive Web Design*, A Book Apart, Eyrolles, 2011

Seulement voilà, la découverte du responsive web design et sa publication ont été réalisées sous l'égide d'un certain Jeffrey Zeldman (fondateur du magazine A List Apart et de la collection A Book Apart) et, quelques mois plus tard, Zeldman remettait gravement en question la définition qu'ils avaient précédemment choisie pour correspondre à cette appellation, dans un article sur son blog : « Responsive design, je ne pense pas que ce mot signifie ce que vous pensez qu'il signifie ».

Il y déclare : « Je pense aussi qu'il s'agit sans doute d'une plus grande idée que ce que nous imaginions, une idée trop large pour être limitée à une unique approche technique pour résoudre le problème de multiples environnements de

1. <http://www.w3.org/TR/css3-mediaqueries/>

2. <http://alistapart.com/article/responsive-web-design>

consultation très segmentés. (...) Notre compréhension du design responsive devrait être élargie pour contenir toute approche mise en œuvre dans l'intention de proposer une expérience visuelle élégante quelle que soit la taille de l'écran de l'utilisateur. » Voici pourquoi il s'agit de la définition que nous préférons proposer ici.

<http://xav.cc/rwd-zeldman>³

Et le design adaptatif ?

Le concept de responsive web design est très souvent injustement confondu avec celui d'*adaptive web design* ou design adaptatif ; c'est excusable, car tous deux partagent davantage que des sonorités voisines. Tout comme le responsive web design, l'*adaptive web design* est présenté par son propre livre, écrit par Aaron Gustafson (Jeffrey Zeldman en signe la préface). Alors que le *responsive* consiste à gérer uniquement les différentes tailles d'écran, l'*adaptive* s'applique plutôt à une catégorie de sites dont le design (fonctionnel ou graphique) change en fonction des capacités du navigateur (Est-il capable d'exécuter du JavaScript ? Est-il capable de stocker de la donnée côté client ? Est-il capable de représenter des blocs avec des dégradés en CSS ? Etc.) ou du système d'exploitation.

On y retrouve la même idée d'adaptation, mais une différence notable est que, pour un contexte client donné, le design adaptatif apportera par définition une seule réponse (puisque'il s'adapte aux capacités du navigateur et du système d'exploitation, qui ne bougent pas pour un même contexte), tandis que le responsive web design apportera souvent plusieurs réponses : en effet, la taille peut changer soit parce qu'on retaille la fenêtre du navigateur (pour un système d'exploitation multifenêtre),

3. <http://www.zeldman.com/2011/07/06/responsive-design-i-dont-think-that-word-means-what-you-think-it-means/>

soit parce qu'on change l'orientation du terminal (s'il a un gyroscope).

Si beaucoup des conseils que vous trouverez dans ce livre sont applicables pour adapter un design aux capacités du terminal (surtout lorsque l'on vous incitera à prévoir et documenter des alternatives différentes), ce ne sera volontairement pas le centre de nos préoccupations : en ce qui concerne la conception et la gestion de projet, c'est l'arrivée du responsive web design, et non du design adaptatif, qui a fortement rendu nos manières de faire obsolètes. C'est donc sur la gestion des tailles d'écrans que nous nous concentrerons ici, même si nous nous autorisons d'occasionnels commentaires concernant le design adaptatif.

AMÉLIORATION PROGRESSIVE ET DÉGRADATION ÉLÉGANTE

Ces concepts sont pertinents pour notre sujet car il s'agit également d'adapter un design à un contexte.

La dégradation élégante

La dégradation élégante s'utilise sur les langages dits tolérants aux erreurs : c'est le cas de HTML et CSS, dont les spécifications prévoient les cas où l'interpréteur ne comprend pas l'une des instructions, de sorte à la rendre transparente et à laisser les autres s'exécuter quand même. Cela permet d'utiliser les technologies les plus avancées sur les navigateurs les plus à jour, tout en les laissant se dégrader seules et élégamment sur les autres.

Un exemple typique est la propriété `border-radius` en CSS : lorsqu'elle est appliquée à un élément, les navigateurs modernes donnent à ce dernier des bords arrondis. Toutefois, les navigateurs plus anciens et incapables de traiter cette propriété

laissent des bords angulaires classiques à l'élément. À moindre effort, tout reste accessible, seul l'aspect esthétique est légèrement affecté ; voilà qui est élégant !

Un exemple typique côté HTML est la multitude d'éléments de formulaires qui ont fait leur apparition dans les spécifications HTML5. En effet, un champ de type `color`, par exemple, va être interprété par un *colorpicker* (sélecteur de couleur) dans les navigateurs modernes, mais se dégradera en champ texte simple pour les autres navigateurs, dans lequel l'utilisateur aura toute possibilité de saisir le code d'une couleur. Évidemment, cet exemple de dégradation technique simple entraîne une dégradation ergonomique forte (il est souvent beaucoup plus simple pour un utilisateur de sélectionner une couleur dans un colorpicker que de saisir son code hexadécimal). Cependant, certaines dégradations sont plus élégantes fonctionnellement : *datepicker* natif se dégradant en champ texte où saisir la date, *slider* natif se dégradant en champ texte où saisir la valeur, etc.

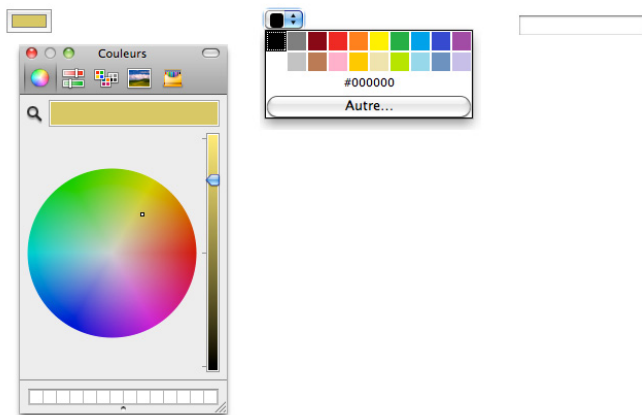


FIG. 1-2 : Le colorpicker, tel qu'interprété à gauche par le navigateur Chrome, au milieu par Opera et, à droite, par un navigateur ancien

L'amélioration progressive

L'amélioration progressive, au contraire, consiste à enrichir le site en détectant proactivement les fonctionnalités acceptées par le navigateur. Ceci a notamment du sens en JavaScript, car ce n'est pas un langage tolérant à l'erreur : une instruction invalide en JavaScript parce que la méthode ou l'objet utilisé n'existe pas dans un navigateur peut totalement interrompre l'exécution des autres instructions.

La solution consiste à s'assurer que vous avez prévu un comportement alternatif pertinent pour toutes les fonctionnalités les plus susceptibles de ne pas être implémentées sur les navigateurs de vos utilisateurs.

L'exemple typique fréquemment utilisé est la géolocalisation : on peut imaginer un bouton **Me localiser**, dont le comportement serait adapté pour les navigateurs qui ne l'implémentent pas. On peut citer aussi un cas qui devient de plus en plus pertinent aujourd'hui : un comportement différent si le terminal s'utilise avec un écran tactile. En effet, bien qu'on trouve chez certains développeurs une volonté de considérer les écrans tactiles comme « mobiles », cette approche est de moins en moins satisfaisante, surtout depuis que les tablettes font, en mode paysage, une largeur équivalente à certains écrans d'ordinateur ; c'est d'autant plus vrai depuis que Microsoft pousse le modèle de l'écran tactile sur ordinateur avec son Windows 8. La solution consiste donc à prévoir un comportement à manipuler avec la souris dans un premier temps (qui puisse être géré par les navigateurs anciens), puis de l'améliorer lorsque la capacité à interagir avec un écran tactile est détectée, en remplaçant le comportement par un autre ou en l'enrichissant.

Vous l'aurez compris, la dégradation élégante et l'amélioration progressive sont deux approches différentes (et compatibles) permettant d'adapter un design aux capacités du navigateur ou du système d'exploitation. Vous aurez peut-être reconnu la

définition : il s'agit en fait de deux solutions techniques pour mettre en place des stratégies de design adaptatif.

LE LÂCHER-PRISE DU WEB

Le blog qui a vu naître le concept de design responsive, A List Apart, fait partie des références les plus anciennes en ce qui concerne le design web. Aussi serez-vous peu surpris en apprenant que ses plus anciens billets datent de 1998 et que l'idée qu'il mettait en avant de réaliser des designs à l'aide de standards du Web était très avant-gardiste à l'époque. Toutefois, si l'on vous dit que l'un de ses plus anciens articles, publié le 7 avril 2000, semble toujours à la pointe de la modernité aujourd'hui, vous serez sans doute plus intrigué.

En signant « *A Dao of Web Design* », John Allsopp expose un message de fond incitant à ne pas tenter de faire des réalisations qui se veulent maîtrisées de bout en bout et absolument identiques dans tous les contextes d'utilisation. À l'époque, il est en guerre contre la perfection au pixel près héritée des médias imprimés et pousse à renoncer aux tentatives d'approvisionnement excessif du Web. Il ne s'agit pas ici de perdre en qualité, mais de se concentrer sur la garantie d'une cohérence seulement là où elle est nécessaire, pour justement gagner en qualité globale ; notamment, il pouvait s'agir, à l'époque, de préférer utiliser un texte (accessible et non maîtrisable esthétiquement) à une image contenant un texte (inaccessible mais fortement maîtrisable) car, n'en déplaise aux graphistes travaillant pour l'imprimé, l'accessibilité était plus importante qu'une esthétique parfaite.

À lire : *A Dao of Web design*

<http://xav.cc/dao>⁴

4. <http://alistapart.com/article/dao>, traduction en français disponible à <http://www.pompage.net/traduction/dao>

Plus d'une décennie plus tard, ce message résonne encore beaucoup alors que les terminaux ayant accès au Web se multiplient, et avec eux, les moyens d'interagir avec le Web. Du navigateur très limité sur un *feature phone* (téléphone à fonctionnalités, de la génération ayant précédé les smartphones, dont l'utilisation est très prédominante dans certains pays, notamment en Afrique) au dernier Chrome sur un ordinateur tout juste sorti, les contextes du côté client se sont tellement diversifiés que vouloir tous les maîtriser est une aventure impossible.

On peut alors reprendre ce lâcher-prise du Web que John Allsopp nous présentait et concevoir une expérience optimale dans les meilleurs cas, mais qui s'adaptera au mieux dans tous les autres cas, en n'hésitant pas à mettre en œuvre la dégradation élégante et l'amélioration progressive.

Ce lâcher-prise du Web est une notion dangereuse, car elle pourrait servir d'excuse à une certaine paresse (« La police d'écriture n'était pas celle prévue en design ? C'est bon, lâche prise ! ») Il faut donc toujours s'assurer de l'utiliser dans le but de tirer la qualité vers le haut pour la globalité des contextes client, à l'exception tolérable des clients les plus à la pointe.

Nous reparlerons très souvent de cette notion au cours du livre.

RAPPEL SUR LES MÉTHODOLOGIES DE GESTION DE PROJET

S'il y a quelque chose de relativement similaire d'un projet web à l'autre, c'est sans doute l'ordre des étapes qui le composeront (bien qu'il y ait des exceptions notables). Pour ne garder que les étapes quasi inévitables, on commencera toujours par lister les charges, puis on rédigera les spécifications, on réalisera le design fonctionnel et graphique, l'intégration, le développement *back-end*, puis la mise en ligne. C'est d'ailleurs la raison

pour laquelle nous avons découpé ce livre suivant les étapes du projet qui nous semblent les plus immuables.

Toutefois, la méthodologie pour gérer les contraintes à l'intérieur des phases, ainsi que l'enchaînement des phases, diffère selon les contextes (que les raisons soient financières, liées à la confiance du client ou à une souplesse exceptionnelle de l'équipe projet, etc.).

Bien que les informations que vous trouverez dans ce livre restent pertinentes quelle que soit la méthodologie que vous choisirez, nous avons jugé utile, pour un livre qui parle de gestion de projet, de rappeler rapidement les principales méthodologies ; nous pourrions alors utiliser librement ces concepts dans les chapitres suivants.

Le modèle en cascade

Le modèle en cascade (vous l'entendrez aussi appelé *waterfall* par les amateurs de français) est le plus basique et il ne connaît qu'une contrainte : la fin d'une phase projet doit être sanctionnée d'une validation pour déclencher la phase suivante. Cela signifie aussi que chaque phase de projet doit se conclure par un livrable à valider. À partir de ces hypothèses, le planning est simple à réaliser, même s'il est idéaliste et prend rarement les risques en compte.

Hérité du BTP (« Les fondations sont bien en place ? On pose les murs ! »), ce modèle est celui utilisé le plus souvent dans les cas réels. Cependant, il arrive souvent, sur un projet web, qu'on ne valide une phase que partiellement pour pouvoir déclencher une partie de la suivante ; bien évidemment, nous ne recommandons pas cette pratique, bien qu'elle puisse se révéler indispensable pour rattraper des glissements de planning avec un minimum de perte de qualité.

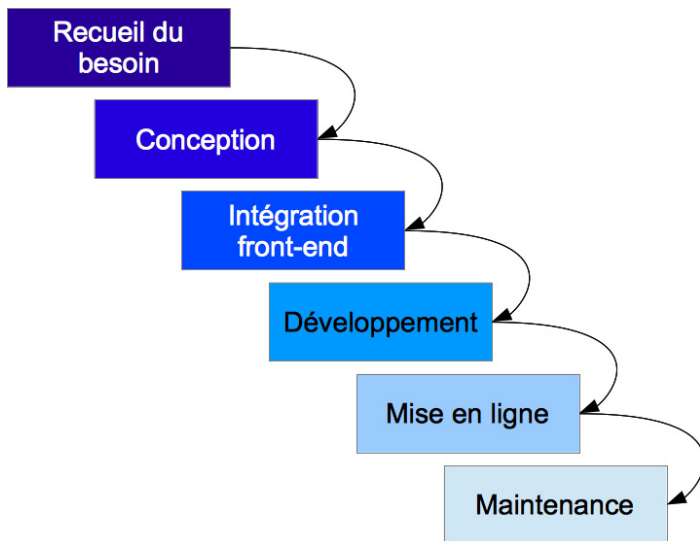


FIG. 1-3 : Un exemple simplifié de modèle en cascade pour un projet web typique

Le cycle en V

Le cycle en V considère que le travail fait en premier sur un projet doit aussi être le dernier à être validé, une fois toutes les autres phases projet passées. Il permet de s'assurer que l'accumulation de complexité au fil du déroulement du projet ne détourne pas le produit fini des conceptions qui ont été réalisées en amont.

En Web, un cycle en V typique va ressembler au schéma représenté sur la **FIG. 1-4**, page suivante.

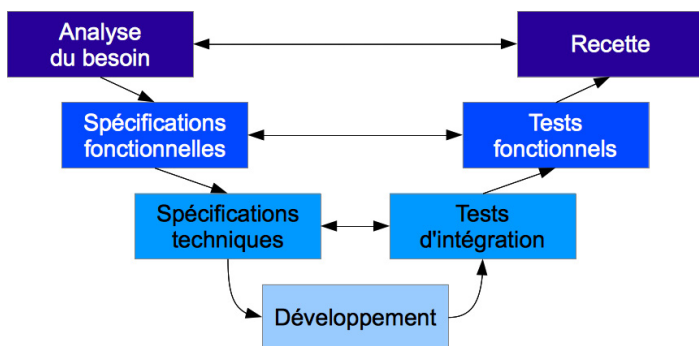


FIG. 1-4 : Un exemple simplifié de cycle en V pour un projet web typique

En régie ou en forfait ?

Que vous fonctionniez en cascade ou en V, en spirale, ou encore selon l'une des nombreuses méthodologies projet aux formes géométriques plus incongrues les unes que les autres, s'il se trouve que vous travaillez dans le service, les modes de contractualisation avec le client sont nombreux ; mais deux d'entre eux sortent du lot comme étant les plus fréquemment rencontrés.

La **régie** correspond à un engagement de moyens : une ressource (dans notre cas, un consultant web) est mise à disposition du client pendant un temps défini et interviendra sur les projets comme le client le décidera. Comme aucun engagement n'est pris par personne sur le travail qui est produit, chacun des partis peut s'avérer perdant : le client, si le consultant n'est pas aussi doué que son commercial l'avait promis, et le prestataire, si le client décide d'arrêter de travailler avec lui du jour au lendemain.

Le **forfait** correspond à un engagement de résultats : suffisamment d'informations sont obtenues en début de projet par le prestataire pour lui permettre de s'engager sur l'intégralité de la suite des événements. Le prestataire émet alors une proposition

commerciale avec une charge prévisionnelle, un planning et un tarif figé pour l'ensemble du projet. Évidemment, tout ceci ne tient aucun compte des risques réels (qui sont alors inconnus) et, dès que les imprévus s'accumulent, la situation est idéale pour que chacun se renvoie la faute : le client rappelle au prestataire qu'il s'est engagé à un résultat pour un tarif figé et qu'il ne paiera pas plus, le prestataire cherche tout écart de conduite du client pour justifier le retard (validation plus longue que prévue, recette mal faite...) et demander une rallonge budgétaire, tout en essayant au maximum de réduire les coûts (et donc la qualité) pour sauver sa chemise. Le client pense partir gagnant car il maîtrise son budget, mais il s'agit aussi du type de contractualisation qui finit le plus souvent très mal...

Il faut modérer ce constat, malgré tout : dans une entente cordiale entre un client et un prestataire, il arrive quand même très souvent que la négociation en cas d'imprévus se passe bien et que l'un des deux (ou les deux conjointement) assume le décalage. Notons que ce type d'entente nécessite une souplesse d'action du côté du client, ce qui n'est pas toujours possible, notamment dans le cadre des marchés publics ; nous reviendrons sur ce cas très particulier.

Les méthodes agiles

L'approche *agile* tente d'apporter simultanément une réponse viable en tant que méthodologie de projet (au même sens que le modèle en cascade ou le cycle en V) mais aussi en tant que cadre de contractualisation (au même titre que la régie ou le forfait). Elle apporte une philosophie qui a le vent en poupe, car elle tente de défavoriser le moins possible l'ensemble des intervenants au projet (côté client comme prestataire).

Tout projet (web ou non) contient toujours trois contraintes principales à surveiller : les délais, le budget et la couverture du chantier (dans notre cas, la couverture fonctionnelle). Tout débordement de l'un des axes (couverture fonctionnelle qui se

révèle plus large que prévue, pertes de temps...) devrait naturellement nécessiter du mouvement sur les autres axes aussi ; mais le modèle du forfait fixe ces trois axes en début de projet, sans possibilité de mouvement.

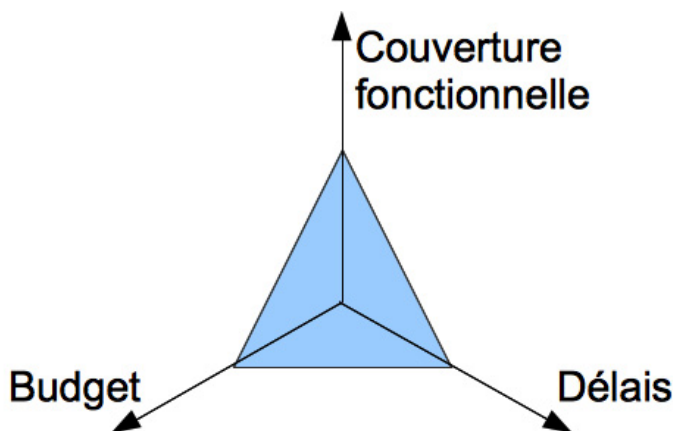


FIG. 1-5 : Les trois contraintes principales de la gestion de projet : tirez sur un des angles du triangle et c'est l'ensemble du triangle qui est censé grandir, sur les deux autres axes également.

Dans le cadre des méthodes agiles, le prestataire et le client vont convenir d'un budget fixe et souvent de délais fixes ; mais ils vont accepter que le troisième axe, la couverture fonctionnelle, ne soit pas fixe.

Pour atteindre cet objectif, le projet va être découpé en fonctionnalités, regroupées par ce que l'on appelle des *sprints*, périodes fixes au terme desquelles le produit est toujours stable et testable, mais incomplet fonctionnellement. Les premiers sprints contiennent les fonctionnalités indispensables et les derniers contiennent celles dont il est possible de se passer.

Au terme de chaque sprint, on pourra effectuer la recette du produit, stable, et toute fonctionnalité ayant un problème sera repositionnée dans le sprint suivant (pour être à nouveau testée

par la suite). Ce principe des méthodes agiles est d'ailleurs fondateur : celui que l'on appelle le *product owner* (responsable de la vision globale du produit) sera plus fortement présent tout le long du projet et aura un rôle actif dans sa construction. Dans le cadre d'une relation client-prestataire, le *product owner* sera le client ; mais un éditeur de logiciel ne pourra pas se passer d'un *product owner* pour autant, alors qu'il n'est pas un « client » à proprement parler ; pour cette raison, nous utiliserons préférentiellement le mot « *product owner* » à « client » dans le reste de ce livre.

Enfin, puisque les méthodes agiles sous-entendent des phases de recueil des besoins puis de conception qui ne sont pas incluses dans ce fonctionnement, on parle de cycle « semi-itératif ».

La **FIG. 1-6** donne une représentation graphique d'un projet suivant les méthodes agiles : après la phase de conception, on itère avec autant de sprints qu'il a été défini lors du cadrage du projet. Chaque sprint livre un produit stable, mais non complet ; le dernier sprint est sanctionné par un produit stable, auquel il manque les fonctionnalités secondaires ne rentrant pas dans le budget initialement prévu.

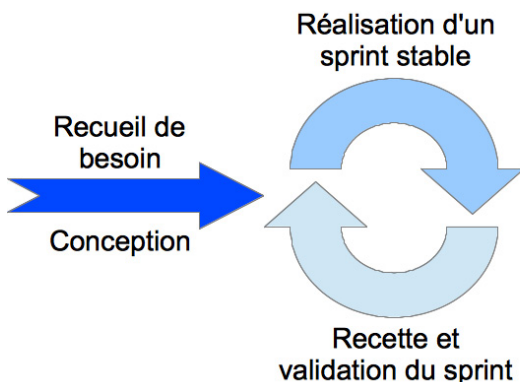


FIG. 1-6 : Représentation graphique d'un projet suivant les méthodes agiles

Au pays des méthodes agiles, il y a plusieurs écoles, apportant chacune ses outils et ses procédés : Scrum, le Kanban, Crystal... En voici une présentation générale, comparée aux méthodologies traditionnelles de gestion de projet :

 V. Messenger, *Gestion de projet agile*, Eyrolles, 2013

Si vous souhaitez mettre en œuvre la méthode Scrum (la plus complète et la plus populaire), nous vous conseillons d'approfondir le sujet avec le livre :

 A. Vannieuwenhuyze, *Scrum, une méthode agile pour votre projets*, ENI, 2013.

À PROPOS DES DEUX AUTEURS



Jérémie Patonnier

Au sortir d'études d'imprimerie et d'arts appliqués, Jérémie se lance dans le Web en tant que designer web. Il découvre alors le développement informatique, trouve ça rigolo et décide de devenir développeur web. Comme ça ne lui suffit toujours pas, il décide de passer au management en devenant responsable d'équipe web. Après toutes ces tribulations, il est maintenant consultant web pour l'agence parisienne du cabinet d'expertises Clever Age, où il fait bénéficier ses clients de son expertise aussi bien côté serveur que côté client.

À côté de ça, il mène plusieurs projets de front, qui ont tous pour objectif de donner aux designers et développeurs web toutes les connaissances nécessaires pour bien faire leur travail. Cofondateur des sites typographisme.net et letrainde13h37.fr, il est également contributeur au projet Mozilla, où il participe à la documentation des technologies du Web ouvert sur le *Mozilla Developer Network* (MDN). Il intervient régulièrement en tant qu'orateur à des conférences, aussi bien en France (Sud Web, Paris Web) qu'à l'étranger (The Graphical Web).



Rudy Rigot

Rudy est un passionné du Web sous toutes ses approches : tombé dedans par la technique (en gagnant ses premiers galons au cœur du développement Java/JEE), il a ensuite trouvé son bonheur dans d'autres contrées, du design fonctionnel au webmarketing, et surtout, dans la jonction de tous ces métiers. Toujours à l'affût de nouvelles choses à apprendre, mais aussi soucieux de partager ses découvertes, il intervient comme orateur dans diverses conférences et comme auteur dans divers magazines professionnels. Il est aussi l'un des membres fondateurs des conférences Sud Web, qui ont lieu depuis 2011.

Lorsque le responsive web design a émergé, il a été parmi les premiers à mettre en garde quant aux impacts sur les processus habituels de conception et de gestion de projet (notamment lors de son intervention aux conférences Front Row à Cracovie, en octobre 2011, ou dans un article publié en avril 2012 par le magazine en ligne *Dev.Opera*).

Il doit son apprentissage et ses travaux de recherche sur le responsive web design à la collaboration efficace avec ses collègues de l'agence Clever Age, pendant quatre ans. Depuis, il a dirigé le produit Steerious (fondé par Clever Age, et aidant les organisations à piloter leur stratégie numérique) et fait maintenant partie du *think-tank* d'innovation web Zenexity (dont le produit phare est le framework Play).