

## Chapitre 4-1

# Spécificités du Web

### 1. Introduction

Le Web, en tant que média, dispose d'un certain nombre de spécificités par rapport à un programme distribué ou client-serveur. C'est un média jeune, en pleine évolution, dont la principale caractéristique est la richesse de ses interfaces.

Cette particularité implique de faire appel à des ressources que l'on trouve rarement dans un service technique, les directeurs artistiques.

### 2. Création graphique et méthodes agiles

Historiquement, dans les projets gérés par un cycle en cascade, la création graphique était confiée à une agence de communication pilotée par la direction de la Communication, tandis que le reste était à la charge de la direction informatique qui coordonnait les intervenants des différents métiers impliqués en s'appuyant éventuellement sur un partenaire pour renforcer ses équipes internes ou sous-traitait tout ou partie des travaux.

# 212 \_\_\_\_\_ Diriger un projet web Agile

Utilisez la dynamique des groupes pour décupler Scrum

L'agence faisait valider par la Communication des écrans représentant le futur site Internet, l'agencement des différentes fonctionnalités, les parcours utilisateurs, bref, elle décrivait ce que serait l'expérience utilisateur, c'est-à-dire l'émotion qu'il éprouve en parcourant le site Internet et par là même, l'association positive ou négative qu'il réalise entre cette émotion et la marque. Ce faisant, elle fixait des contraintes pour la réalisation technique ultérieure. Ces dernières étaient plus ou moins fortes selon la maîtrise qu'a cette agence du média Internet.

Casser ce modèle pour le rendre agile représente parfois un véritable casse-tête. Remplacer l'approche séquentielle par une approche itérative implique de résoudre plusieurs dilemmes.

## 2.1 Premier dilemme

Il n'est pas envisageable de réaliser une création graphique sans avoir une vision exhaustive des fonctionnalités. Comment savoir ce qui doit être mis en avant sur la page d'accueil si le périmètre fonctionnel doit pouvoir évoluer librement au cours de la production ?

D'un autre côté, donner libre cours aux créatifs contribue à fixer le périmètre fonctionnel de la page d'accueil, ce qui remet en cause l'approche agile dans son essence.

Se contenter de poser une charte graphique, puis laisser le champ libre à des ingénieurs pour la réalisation, revient à risquer de sacrifier l'expérience utilisateur.

Ce dilemme est récurrent dans les équipes s'appropriant une méthode agile et ce n'est que la partie émergée de l'iceberg. Il ne porte en effet que sur l'analyse du présent, mais ne considère pas du tout un autre aspect de la production web, la dépréciation liée à l'évolution technologique induisant une évolution des usages.

## 2.2 Second dilemme

Si autrefois le *designer* d'une solution web devait prendre en compte des contraintes de navigateur et de résolution d'écran représentant une quinzaine de configurations différentes, aujourd'hui il doit élargir sa réflexion aux tablettes et téléphones mobiles, démultipliant à tel point les configurations qu'il est impossible de toutes les tester. L'émergence de nouveaux supports et de nouveaux modes de consommation de l'information n'étant qu'à ses débuts, il n'est même pas possible d'envisager les configurations possibles à court terme. Qui peut prévoir de quoi seront capables les futures montres, lunettes ou télévisions connectées ? Une solution courante consiste à décliner des interfaces spécifiques pour chaque média en utilisant différentes feuilles de style, mais cette solution risque de n'être que temporaire.

Si la multiplication des supports implique d'adapter les modèles de production par un découplage fort entre le code et la présentation, la multiplication des interfaces risque de bouleverser un paradigme déjà ébranlé. L'apparition des écrans tactiles, de la réalité virtuelle ou augmentée, ainsi que des fonctions de commande vocale, introduites notamment par Google, représentent des enjeux considérables en termes d'expérience utilisateur.

Alors que la multiplication des feuilles de style destinées à adapter le contenu aux différentes interfaces complexifie la maintenance évolutive, la prise en compte de nouvelles interfaces risque de rendre toute solution impossible à maintenir. Les entreprises qui n'auront pas anticipé ces opportunités risquent de se voir confrontées à des choix pénibles : maintenir des solutions en double, ou bien faire l'impasse sur de nouveaux modes de consommation de l'information, au risque de prendre un retard considérable sur leurs concurrents. Toutes les sociétés françaises ont déjà vécu cette expérience lorsqu'Internet a supplanté l'historique Minitel. Il n'est pas sûr qu'elles souhaitent revivre cette expérience.

Faire du shopping depuis son téléviseur 3D dans une réalité augmentée, puis valider et payer sa commande directement depuis son téléphone n'est pas une prospective à dix ans, c'est réalisable aujourd'hui.

# 214 \_\_\_\_\_ Diriger un projet web Agile

Utilisez la dynamique des groupes pour décupler Scrum

Nous avons donc affaire à un double dilemme, organiser le mode de production de solution web en fonction de nouvelles méthodes mais également prendre en compte le fait que les interfaces évoluent aussi vite que les processus métier, voire plus vite. Tout ceci, sans compter les évolutions des serveurs d'applications, les dépréciations technologiques et les risques de sécurisation des applications.

## 2.3 Commençons par ce qu'il ne faut pas faire

Il ne faut pas rester sur l'ancien modèle et considérer que la création graphique représente une contrainte à respecter, et qu'il sera éventuellement possible de l'adapter à *la marge*, avec l'accord de l'artiste l'ayant réalisée. Cette approche est à l'opposé de l'agilité. Elle est même à l'opposé du bon sens. Faire de l'agilité en acceptant que la créativité soit une contrainte, plutôt qu'une source d'innovation relève quasiment de la dissociation mentale.

Tenter un mix revient aussi à s'égarer de la philosophie de base de l'agilité. Ainsi, créer des ateliers ou des réunions spécifiques avec des directeurs artistiques, ergonomes, spécialistes de l'UX (expérience utilisateur), chefs de produits auxquels on convie le Product Owner et éventuellement le Scrum Master, réduira la créativité de l'équipe et engendrera des tensions.

Généralement, ces ateliers servent à concevoir un concept graphique à partir de la charte graphique et de différentes pistes préalablement retenues (études concurrentielles ou stratégie d'émergence).

Le problème est double. Ce concept graphique pose des contraintes qui réfrèneront la créativité de l'équipe au lieu de la nourrir. Elles sont validées par le Product Owner et les artistes, mais bien souvent le Scrum Master n'est là que pour apporter une garantie de faisabilité. Second effet néfaste, l'équipe étant exclue de la conception, elle se sent au mieux dévalorisée, au pire, trahie par le Scrum Master.

À l'opposé, concevoir la solution sans contrainte, puis faire appliquer la charte graphique sur le résultat une fois que celui-ci est parfaitement fonctionnel peut amener à une impasse. Ceux qui ont testé cette expérience, croyant que ce serait une bonne idée, vous déconseilleront vraiment d'essayer le design non contraint.

## 2.4 Utiliser pleinement Scrum

Plutôt que de conduire des ateliers, puis des réunions pour rendre compte de ces ateliers et organiser des comités pour entériner les décisions, utilisons pleinement la revue de Sprint Scrum.

Lors de cette réunion, le Product Owner et l'équipe Scrum présentent à l'ensemble des parties prenantes ce qu'ils ont réalisé dans le Sprint en cours et le Product Owner indique ce qu'il lui semble le plus judicieux de réaliser lors des prochains Sprints. Lorsque le résultat présenté correspond aux attentes de chacun, il est possible de poursuivre le plan de release tel qu'envisagé.

Il sera toujours plus facile de faire réagir les utilisateurs en leur montrant un produit en cours d'élaboration et en les faisant participer à l'élaboration, directement en présence des développeurs, que de conduire de multiples réunions.

Il ne reste plus qu'à faire travailler en bonne intelligence des designers et des techniciens.

### 2.4.1 Première solution : découplage du front office

Il s'agit de mettre en place une équipe qui sera exclusivement dédiée à la conception et aux développements des interfaces. On parle alors de développements front office, que l'on retrouve souvent sous l'acronyme DFO. Cette équipe est composée de spécialistes des interfaces maîtrisant le responsive web design (RWD), de développeurs maîtrisant des frameworks, tel Angular, ou un langage spécifique tel Sass (langage informatique permettant de générer des feuilles de style, acronyme de *SyntacticallyAwesome Style Sheets*), d'un directeur artistique, d'infographistes et éventuellement d'un ergonomiste. La présence d'un ingénieur peut renforcer le savoir-faire technique et faciliter les échanges avec les autres équipes.

Cette approche offre à la fois une opportunité et une contrainte. Elle implique un découpage produit particulier avec des lots dédiés aux interfaces et un fort découplage applicatif. Un travail d'architecture est essentiel en amont et lors de la conduite du projet. En revanche, elle permet de produire une solution qui sera très adaptable aux évolutions technologiques, notamment en termes d'interfaces. Elle permet aussi de modifier les interfaces indépendamment des règles métier (attention, l'inverse n'est pas avéré).

# 216 \_\_\_\_\_ Diriger un projet web Agile

Utilisez la dynamique des groupes pour décupler Scrum

Elle oblige à avoir deux Product Owners distincts pour un même produit, ou bien un Product Owner qui fait le va-et-vient entre plusieurs équipes.

Enfin, elle implique une planification des lots et une organisation drastique. Dans une organisation ayant adopté l'agilité à l'échelle tel que Scrum2Scrum ou SAFe, cette organisation sera facilitée.

## 2.4.2 Seconde solution : le brassage culturel

Cette approche permet de bénéficier pleinement des avantages de la gestion de projet agile, mais également de ceux de la dynamique de groupe.

Elle consiste à mixer les compétences artistiques et techniques au sein d'une même équipe, ingénieurs et créatifs travaillant de concert à élaborer une solution complète, s'appuyant sur les idées et compétences ainsi réunies au sein d'une même équipe.

Elle offre par ailleurs des bénéfices à court terme mais également à long terme.

À court terme, l'équipe apporte ensemble une solution à la *user story* relatée par le Product Owner. Elle exprime magistralement le concept du  $1+1=3$ . Là où l'ingénieur aurait disposé une suite de boîtes à remplir pour réaliser un formulaire fastidieux, le créatif proposera des sliders, une commande tactile ou vocale, chacun rebondira sur l'idée de l'autre pour arriver à une solution efficace et élégante, prenant en compte l'expérience utilisateur tout en respectant les contraintes techniques. On constate alors que le rôle du Product Owner a tendance à s'inverser. Lui qui poussait l'équipe à trouver des solutions élégantes, aura tendance maintenant à refréner sa créativité sur les fonctions non critiques.

Le bénéfice à long terme le plus évident est la compétence des équipes. Ce brassage produit un effet formateur exceptionnel, car l'information est véhiculée non plus par une autorité détentrice d'un savoir, mais par un membre de sa propre tribu. Le partage et l'appropriation sont beaucoup plus profonds. L'ingénieur commence à penser en créatif et vice versa.

En revanche, elle implique d'avoir pour chaque sprint des tâches de design pour occuper le plus efficacement possible les ressources créatives, ou d'avoir des ressources fortement polyvalentes, acceptant de sortir du cadre strict de leur mission afin d'atteindre l'objectif d'un Sprint.

### 3. Découplage des interfaces

Dans la catégorie des bibliothèques graphiques permettant de construire des interfaces, si Swing est considéré comme un composant léger, comparé à AWT, alors les interfaces web font figure de poids plume. L'usage des feuilles de style et du DHTML permettent nativement un couplage très faible entre les règles métier et les interfaces.

Cependant, certains langages permettent une architecture brouillon, voire pas d'architecture du tout. On retrouvera alors mêlés au sein d'un même script des règles métier, des interfaces et des traitements de surface. L'effet Flaccid Scrum décrit par Martin Fowler peut alors prendre des proportions dantesques. À vouloir produire rapidement un produit utilisable, dans l'objectif de valider des hypothèses, les développeurs ont parfois tendance à prendre des raccourcis qui se traduisent par une dette technique colossale.

Cette démarche a tendance à se produire lorsque la pression sur les délais est très forte et lorsque l'équipe est dans un esprit de défiance, bien souvent parce qu'elle se sent en difficulté.

L'approche que nous venons d'évoquer, consistant à former une équipe DFO (développements front office), permet de minimiser cet effet et force un découplage de manière systémique. Mais pourquoi insister autant sur le découplage et particulièrement celui des interfaces ?

Premièrement, parce que lors du processus itératif, les demandes de modification des utilisateurs portent principalement sur les interfaces. À moins que les règles métier aient été mal exprimées dans les User Stories, elles ne doivent évoluer qu'à la marge lors des itérations successives. Les sprints de remboursements de dette technique sont généralement plus importants que ceux consistant à intégrer des évolutions fonctionnelles.

Si les interfaces sont fortement découplées, le risque de régression sur les processus métier est alors très faible, voire inexistant.