



DOSSIER SPECIAL

Les méthodes agiles : de la souplesse dans les contrats informatiques

Par Thomas Beaugrand, Avocat - Cabinet **Staub** & Associés

Révolution relativement récente dans le monde de l'industrie informatique (leurs premières manifestations sont apparues à la fin des années 90), les « Méthodes Agiles » sont des méthodes organisationnelles et opérationnelles qui gagnent en notoriété depuis quelques années et investissent à présent tous les domaines de l'activité logicielle, depuis la réalisation d'un site internet jusqu'aux plus grands projets d'intégration/déploiement de progiciels de gestion (ERP).

Bien que plusieurs écoles existent (« Xtreme Programming », « Scrum »), ces méthodes renvoient toutes au « Manifeste Agile¹ » rédigé en 2001 par une quinzaine d'experts de l'ingénierie logicielle. A partir du constat selon lequel certains échecs industriels seraient dus à une trop grande rigidité méthodologique et juridique, ces experts soulignent que l'adaptabilité des processus de développement s'avère vitale lorsque l'entreprise se lance dans un projet au long cours - surtout en période de crise économique, et dans des domaines technologiques qui se renouvellent tous les deux ou trois ans.

Si ces experts ont développé des méthodes opératoires pour renforcer les chances de succès des projets informatiques, les juristes en charge de rédiger les contrats IT doivent alors impérativement les appréhender et les traduire en termes juridiques, afin que le contrat ne contredise pas la réalité du terrain : si amélioration il y a, le contrat doit la prendre en compte et lui donner force.

L'objet de l'étude qui suit n'est pas d'explorer en détail les subtiles distinctions entre les nombreuses méthodes de gestion de projet qui foisonnent depuis une vingtaine d'années, mais de souligner les principales innovations induites par les Méthodes Agiles, par rapport aux méthodes traditionnelles que le praticien sait traduire en engagements contractuels. Il est donc question de montrer quelques pistes pour que le droit s'adapte à la réalité opérationnelle du projet.

¹ [http // agilemanifesto org/](http://agilemanifesto.org/)



Thomas Beaugrand
Staub & Associés

1. Les principes de l'Agilité

Les principes fondamentaux des Méthodes Agiles sont les suivants : « des individus et des interactions, plutôt que des processus impersonnels et des outils génériques » ; « des fonctionnalités opérationnelles plutôt qu'une documentation exhaustive » ; « une collaboration poussée avec le client plutôt que la contractualisation stricte du projet » ; « l'acceptation d'un changement bien conçu plutôt que la conformité rigoureuse à des plans parfois caducs ». Il s'agit donc d'une émanation très pragmatique, issue de l'expérience des techniciens plutôt que d'une théorie abstraite. Les Méthodes Agiles s'expriment sous la forme de préceptes proches du mantra : livrer fréquemment des modules opérationnels, pratiquer un contrôle qualité permanent, privilégier la cohésion des équipes, leur contact direct et leur auto-organisation, notamment.

Le premier avantage revendiqué par les promoteurs des Méthodes Agiles consiste à accueillir favorablement le changement en cours de projet, quelle qu'en soit l'origine.

D'où peut survenir le changement ? D'une part, de la mutation des besoins du client ou de son contexte économique, en cours de projet. D'autre part, de l'apparition de difficultés analytiques ou techniques en cours de développement. Les juristes connaissent parfaitement ces deux phénomènes : ils les analysent depuis des années en tant que fautes contractuelles, génératrices de dommages et intérêts. En effet, le client commet une faute contractuelle s'il éloigne ses demandes du périmètre fonctionnel initialement défini, car il déjoue ainsi les prévisions techniques et budgétaires de son cocontractant. En face, le prestataire est responsable d'un défaut de livraison conforme s'il dépasse les délais ou échoue à corriger les anomalies et non-conformités du produit... Et l'évolution jurisprudentielle a renforcé l'intensité des obligations du prestataire, notamment via la théorie des obligations essentielles comme l'illustre par exemple l'affaire Faurecia. Au premier abord, les préceptes de l'Agilité pourraient donc paraître inconsistants, voire dangereux, aux yeux du juriste.

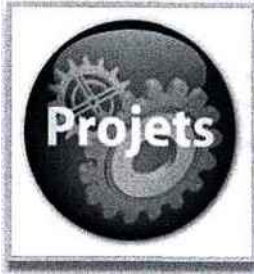
Mais dans les faits, ces changements existent. Sur un projet qui s'étend parfois sur douze ou quinze mois, le besoin du client va très probablement évoluer. Et des complications techniques sont toujours possibles. A partir de ce constat, les Méthodes Agiles ont été pensées pour intégrer le paramètre « changement » dans la feuille de route du projet IT. Le nombre d'entreprises qui y recourent (prestataires comme clients) démontre aujourd'hui leur efficacité. Il s'agit bien d'une alternative concrète aux méthodes habituelles de gestion de projet. Les Méthodes Agiles contredisent donc plusieurs schèmes classiques, et prônent une intensification de la collaboration entre équipes du client et du prestataire, pour un copilotage efficace et réactif, plus axé sur la qualité que sur la conformité.

Le juriste ne doit donc plus appréhender le changement comme cause d'une dérive calendaire, mais comme l'un des paramètres évolutif que les parties ont accepté par contrat. De même, le juriste doit accepter que le contrat ne régit pas tous les processus, mais en pose uniquement les principes opératoires. L'on soulignera toutefois qu'aussi pertinentes qu'elles puissent être sur le terrain, ces méthodes ne peuvent suppléer la signature du contrat, surtout au regard des enjeux financiers. Mais le juriste doit se départir d'un certain nombre de dogmes et de réflexes, afin de mettre son encadrement contractuel au diapason de la pratique adoptée. En effet, si les Méthodes Agiles ne constituent pas automatiquement une garantie de succès, au moins semblent-elles proposer de nouveaux concepts et supprimer des conflits à l'origine de lourds préjudices, ce que le praticien ne peut ignorer.

2. Les dangers du changement de physionomie du projet IT

Dans un contexte classique, le projet d'informatisation (ou plus souvent aujourd'hui, de modernisation du système d'information, mais nous visons également ici tout projet technologique notamment dans les secteurs de l'internet et des communications électroniques) obéit à un corpus de règles relativement figé.

Traditionnellement, les projets IT sont menés en mode cascade, c'est-à-dire par étapes linéaires et successives. Schématiquement, une navette s'instaure entre le client et le prestataire, chacun fournissant à tour de rôle ce qui relève de sa responsabilité : émission des besoins du client, spécifications techniques puis développement et intégration aux soins du prestataire, tests et recette du client, jusqu'au déploiement de la solution cible décrite dès le lancement du projet. Certes les méthodes de travail sont parfois plus complexes que la présentation ici dressée, mais pour le praticien, le contrat IT reflète toujours en dernière analyse le caractère unilatéral des travaux informatiques. Ainsi, le contrat IT comporte souvent une matrice de répartition des tâches, qui va détailler les responsabilités incombant au client (expression de son besoin, fourniture



des contenus spécifiques, exposé de ses contraintes métier, préparation des données à migrer...) et celles incombant au prestataire (analyse préalable, spécifications techniques et fonctionnelles, etc.).

Ainsi, avant de sélectionner son prestataire, le client identifie son besoin et tente de le définir, d'autant plus clairement qu'il dispose d'une Direction Informatique en son sein. Il s'agit là d'une étape critique où réside l'origine de nombreux contentieux : l'expression des besoins est cruciale, qu'il s'agisse de choisir un ERP, d'installer un PABX, de commander le développement d'un logiciel spécifique ou de réaliser un site de commerce en ligne.

Le client connaît son métier, ses besoins et ses contraintes ; le prestataire connaît quant à lui la technologie en vigueur, l'état de l'art, et les qualités de ses produits. C'est donc au client de synthétiser ses besoins et ses pré-requis dans un document techniquement et juridiquement valide, traditionnellement exhaustif, en sollicitant éventuellement l'assistance d'un professionnel (AMOA). Il appartient ensuite au prestataire d'évaluer sa capacité à répondre à ce besoin.

Le contrat IT fait souvent de cette expression initiale des besoins le premier référentiel de conformité, placé en annexe. Le client a donc tendance à la définir le plus précisément possible. Or, il doit surtout penser son projet IT dans une optique dynamique. En effet, trop d'entreprises se contentent d'effectuer un bilan de leurs besoins au moment où elles lancent le projet, et acceptent une proposition commerciale qui y correspond à l'instant T. Si le besoin du client évolue, la proposition du prestataire devient rapidement inadéquate, et ce dernier se retrouve pris entre deux feux : d'une part la nécessité de satisfaire son client, voire de répondre à son engagement de résultat... et d'autre part l'équilibre budgétaire de sa proposition - surtout si celle-ci est forfaitaire. On ne compte plus les projets informatiques au forfait qui tournent à l'intégration continue... et ruinent les capacités du prestataire. Le client et le prestataire s'affrontent alors pour qualifier les demandes, parfois extrêmement nombreuses, entre « corrections dans le périmètre convenu » et « demandes nouvelles hors périmètre », « demandes évolutives », « nouveaux besoins »...

En cas de contentieux, le prestataire dénoncera un manque de définition des besoins, imputable aux choix trop peu prospectifs du client. Celui-ci fustigera l'incompétence technique du prestataire, l'insuffisance de son analyse préalable, ou encore des engagements contractuels irréalistes voire dolosifs. Parfois, les griefs sont fondés, et il revient au praticien de les démontrer devant le juge afin d'obtenir l'indemnisation de son client, qui a consommé ses ressources en vain.

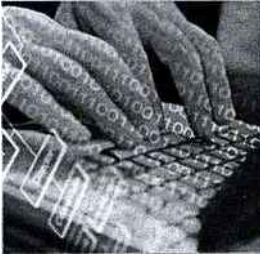
Mais l'entreprise qui définit son projet IT doit impérativement prévoir les hypothèses d'évolution de ses demandes et calquer l'expression de ses besoins informatiques sur ses propres perspectives de développement commercial ou industriel. Un exercice qui s'avère plutôt ardu dans un contexte linéaire classique. Comme on va le voir, les Méthodes Agiles proposent à l'inverse une gestion de projet itérative et collaborative : au lieu de laisser le client décider seul de ses besoins, avec plus ou moins de bonheur, le prestataire participe à ses côtés, dès le début du projet, à un travail de spécification fonctionnelle afin de définir de façon évolutive le besoin réel du client et de faire évoluer le projet en fonction de l'évolution de ces besoins. Dès lors, le praticien comprend qu'il doit abandonner la notion de « matrice de répartition des tâches », car celles-ci seront gérées de façon très différente, conjointement et simultanément.

3. Le développement collaboratif des fonctionnalités

3.1 L'expression des besoins : un backlog plutôt qu'un cahier des charges

Dans un projet en cascade, le cahier des charges classique expose le périmètre fonctionnel, ainsi que la durée du projet et les exigences de performances. Le contrat prévoit généralement que le cahier des charges complète et cristallise l'expression initiale du besoin, et sert de référentiel contractuel en cas de non-conformité constatée après livraison. Or, la précision du cahier des charges contraint souvent le prestataire à s'y tenir quelles que soient les difficultés rencontrées. Même s'il lui substitue ses propres spécifications, des écarts sont possibles entre les attentes du client et la compréhension du prestataire. Et même sans écart, le besoin du client peut évoluer de lui-même. C'est alors que son cahier des charges peut parfois se retourner contre le client, qui en devient prisonnier alors qu'il est en partie caduc.

En contexte Agile, c'est moins le cahier des charges que le backlog, établi conjointement, qui permet de piloter les travaux. Le backlog est une liste fonctionnelle



qui expose les besoins pratiques, objets d'échanges et de repriorisations. Elle présente les fonctionnalités attendues, auxquelles correspondent des unités de valeur et des points de complexité, c'est-à-dire une quantification de la valeur métier du module, et une estimation de sa complexité technique. Le backlog permet ainsi d'assigner et de réassigner les priorités de réalisation en fonction de l'évolution du projet sur le terrain, à partir de ces critères d'appréciation. L'expression des besoins n'est pas figée : le client pourra prioriser telle ou telle fonctionnalité, demander à modifier la physionomie de telle autre, voire abandonner une demande pour y substituer un nouveau besoin. Le backlog traduit ces adaptations, actualisé en temps réel par les équipes du client et du prestataire. Le client exprime donc son besoin non plus en amont des travaux de développement, mais tout au long du projet. Au prestataire de se montrer parfaitement proactif.

Comment viser dans le contrat un backlog qui sera nécessairement évolutif ? Il est possible d'annexer au contrat un premier backlog, à condition de prévoir alors qu'il sera évolutif, et de mettre en place les modalités d'évolution dudit backlog et l'instance qui validera ses orientations successives.

3.2 Le développement des livrables : la fin de la navette

Dans un projet informatique classique (en cascade ou « en V »), le prestataire divise le projet en lots et l'organise en phases, qui sont lancées selon un calendrier (parfois indicatif, souvent impératif) conduisant théoriquement à un résultat prédéterminé. Le prestataire prend en charge la spécification fonctionnelle comme l'exécution technique, ce qui prive souvent le client de la visibilité sur les réalisations intermédiaires.

En contexte Agile, là encore la collaboration des équipes est placée au centre du projet. On voit émerger des notions telles que les users stories, qui procèdent moins d'une définition générique des fonctionnalités attendues que d'une expression concrète de leurs besoins par les utilisateurs eux-mêmes. Les users stories sont actualisées tout au long du projet, et les fonctionnalités afférentes sont contrôlées dès leur mise à disposition, dans le cadre d'un développement sous forme de courtes itérations identiques (time boxing), appelées sprints et releases. Les releases cumulent en leur sein les opérations de spécification fonctionnelle, de développement puis de tests, au lieu de les échelonner dans le temps. En cas de non-conformité, le développement est corrigé avant de passer à la séquence suivante.

Le praticien qui rédige un contrat IT doit donc traduire en droit les notions de courtes itérations de durées identiques, et les options qui s'offrent aux parties au terme de chacune d'elles. Les phases du calendrier, souvent critiques lorsqu'elles ne sont pas respectées, doivent donc être remplacées par la mention d'une unité de mesure, le sprint. Le contrat doit ensuite stipuler le nombre de sprints envisagé pour parvenir à répondre à la demande du client.

Le prestataire propose également une usine logicielle, soit un ensemble d'applicatifs de développement utilisé par les équipes. Celles-ci sont en outre réunies en un même lieu afin de permettre des échanges concrets, plutôt que des échanges parfois pléthoriques d'emails et de synthèses. Cette unité de lieu n'a pas pour fonction de remplacer les comités de pilotage, mais fluidifie considérablement ceux-ci dans la mesure où les équipes bénéficient ipso facto du même niveau d'information. Usine logicielle, unité de lieu... autant d'éléments à faire monter en puissance, au chapitre des modalités de réalisation des prestations du contrat.

Enfin, le calendrier constitue en principe le fil rouge du projet, et fait se succéder les phases du projet à un rythme parfois très délicat. En contexte Agile, le temps est relativisé : il est un paramètre parmi d'autres dans la priorisation des développements. Si la notion de date finale de bascule conserve toute sa pertinence, en revanche les étapes intermédiaires peuvent varier selon les fonctionnalités que les utilisateurs privilégient. Il devient illusoire de les assortir d'une obligation de résultat...

3.3 La recette des prestations : la fin du tunnel

En contexte classique, le client pâtit parfois d'un « effet tunnel », lorsque ses équipes n'interviennent pas dans les opérations de spécification et de développement de la solution. Le client a émis son besoin en amont, et contrôle les livrables en aval par référence aux spécifications issues du cahier des charges. Or, comme plusieurs mois séparent souvent la validation des spécifications par le client et la livraison des développements par le prestataire, les surprises sont parfois nombreuses au moment des opérations de réception ! Non-conformités fonctionnelles, défaut de performances,

Le Monde du Droit est une publication de LegalNews – 7, rue de Clichy 75009 Paris. Directrice de la publication : Aurélie Prigent. Directeur de la rédaction : Arnaud Dumourier, Toute reproduction interdite sauf autorisation.

Renseignements : 01.56.79.89.81
- contact@lemondedudroit.fr

Contributions, communiqués :
redaction@lemondedudroit.fr



dysfonctionnements ou retards graves, les fiches de recette sont parfois extrêmement accablantes pour le prestataire – à moins qu'elles ne révèlent des fautes du client, qui refuse de procéder à un minimum de « conduite du changement » en interne ou qui a insuffisamment documenté son besoin en amont.

Mais à ce stade, le préjudice est consommé, et les partenaires doivent redoubler d'efforts pour parvenir à finaliser un projet qui connaît ses premières dérives. Les dérives s'aggravent parfois pour devenir de véritables accidents industriels, chacun ayant épuisé ses équipes en vain. Bien entendu, certains contentieux permettent de mettre en lumière l'insouciance coupable d'un client qui n'a pas mesuré les implications de son projet, ou à l'inverse, l'incompétence du prestataire dans l'analyse du besoin, voire l'inadéquation de son produit. Mais dans tous les cas, c'est la découverte d'écarts entre les livrables et leurs spécifications initiales qui déclenche la dérive.

En contexte Agile, les risques de mauvaise surprise semblent considérablement réduits. A l'instar des opérations de spécifications, les opérations de tests sont menées de façon collaborative, sur des itérations très courtes. Les écarts sont donc constatés conjointement dans un délai extrêmement réduit, et corrigés de la même façon. Le fil directeur du développement résidant dans l'efficacité de la fonctionnalité, elle sera donc validée même si elle ne répond pas exactement au cahier des charges initial, dès lors que c'est en commun que le client et le prestataire délivrent leur validation. L'effet tunnel est ainsi supprimé puisque les utilisateurs testent les fonctionnalités promises très peu de temps après avoir qualifié leur besoin, et peuvent les mettre en production aussitôt.

Le praticien qui rédige le contrat IT doit donc renoncer au dogme de la recette qui tombe comme un couperet sur les développements en fin de phase : en contexte Agile, ceux-ci ont été spécifiés, puis réalisés, testés et validés (ou corrigés) en commun dans le cadre d'itérations courtes. En revanche, il doit imaginer de nouveaux critères de recette, en conservant à l'esprit qu'eux-mêmes seront susceptibles de varier dans le temps, et qu'il n'est donc pas utile de les « sanctifier » dans le contrat. Ces critères doivent se rattacher non plus au cahier des charges, mais au backlog et aux user stories, tels qu'ils existeront à l'époque de la recette.

4. De nouveaux principes de pilotage et d'évaluation

4.1 Le copilotage Agile du projet

On a vu que le backlog remplace le cahier des charges, et que les sprints et releases combattent l'effet tunnel. En outre, ces notions sont définies conjointement, et demeurent évolutives. L'ensemble des phases du projet traditionnellement juxtaposées (analyse, spécification, développement, recette, correction) est mené de front... Mais d'autres concepts font leur apparition et traduisent une montée en puissance de l'obligation de collaboration dans le contrat : la vélocité caractérise l'efficacité de l'équipe, c'est-à-dire sa faculté à produire un code de qualité et des unités de valeur importantes en fonction du nombre et de la durée des sprints. Dans un contexte où le résultat compte moins que la performance, cette mesure de la performance doit primer le simple respect des délais. Le contrat doit comporter les indices de mesure de cette performance, voire les vecteurs de récompense si elle est satisfaisante (par exemple par des systèmes de bonus / malus).

La collaboration devient centrale : la bonne entente, la répartition des rôles, la motivation commune et le feedback permanent sont très conseillés. Les équipes se réunissent fréquemment (la méthode Scrum insiste sur la réunion quotidienne). Les outils de communication (wiki notamment) acquièrent une grande acuité, et participent de cette communication permanente. Et les équipes du client sont d'autant plus enclines à accepter le changement, qu'elles participent justement à la définition de ce qui change. Ainsi, les prototypes se multiplient : la solution informatique cible ne reste pas longtemps à l'état virtuel, et ses premiers composants sortent rapidement de l'usine logicielle puis sont corrigés au gré des validations. Le contrat doit prendre en compte le sort de ces prototypes, et décider notamment des droits de propriété intellectuelle s'y appliquant.

Les juristes habitués aux contrats IT ne manqueront pas de remarquer que certaines de ces notions Agiles apparaissent déjà dans les contrats IT classiques : l'engagement de reporting, l'obligation de collaboration, la conduite du changement existent depuis longtemps... Mais en contexte Agile, elles gagnent en intensité et doivent focaliser l'attention du praticien. Comme souligné plus haut, il ne sert à rien d'ériger le cahier des charges initial en référentiel de conformité, puisqu'il sera primé par le backlog. Tout au



plus peut-il servir d'indication, mais force doit être donnée aux décisions du product owner, et aux validations et adaptations décidées en comité

Le client doit toutefois s'interroger sur la réversibilité. Certes, l'adaptabilité des équipes aura permis de contourner les éventuelles difficultés techniques rencontrées. Mais le client doit pouvoir faire fonctionner et maintenir le système après le départ du prestataire. La très forte association entre produit et équipe ne doit pas créer de difficulté lorsque l'équipe change ultérieurement côté client. De ce constat émanent parfois certaines critiques relatives au manque de documentation.

4.2 L'évaluation Agile du projet

Les Méthodes Agiles imposent de penser les fonctionnalités attendues par le client non plus seulement en termes de coût ni de délais, mais de valeur. Quelle est la valeur réellement attachée à telle fonctionnalité ? Quels gains (productivité, ergonomique, confort, performances...) la fonctionnalité apporte réellement par rapport aux autres ? Cette notion de valeur implique un changement de culture au sein des métiers habituellement chargés d'établir le cahier des charges. D'où l'importance du product owner, autre illustration de la montée en puissance du capital humain dans les projets IT.

Les Méthodes Agiles procèdent d'une démarche éminemment pragmatique. Le contrat, qui par définition comporte des principes plus abstraits, doit évoluer pour éviter que l'organisation Agile du projet ne soit contredite par un encadrement trop corseté et par voie de conséquence, contre-productif. De la même façon, l'obligation de résultat par rapport à une solution cible définie ab initio se marie mal avec le caractère mouvant du backlog. Le praticien doit donc appréhender ces notions et leur donner la place qui leur revient dans le « contrat Agile ».

Par ailleurs, on ne doit pas négliger que le forfait budgétaire rassure le client. L'engagement forfaitaire du prestataire est très adapté à un cahier des charges initial définissant un périmètre fonctionnel fermé et des contraintes temporelles strictes. En revanche, dans un contexte Agile, les parties adoptent au fur et à mesure des options non initialement envisagées, ce qui paraît incompatible avec l'engagement au forfait. Et aussi souples soient-elles, les Méthodes Agiles ne peuvent suppléer entièrement les garanties que le client exige en termes de périmètre financier. La conciliation entre le goût du client pour le forfait et l'intérêt des Méthodes Agile reste un chantier ouvert, sur lequel travaillent les commerciaux et les juristes.

Les Méthodes Agiles signent-elles la fin du forfait et de l'engagement de résultat dans les contrats informatiques ? Il serait imprudent d'aller aussi loin. Premièrement, les projets IT sont complexes, et les clients peuvent cumuler commandes fermes et périmètre prospectif. Sur la partie ferme, le client peut fort bien exiger une enveloppe forfaitaire et un engagement de résultat, s'il sait que son besoin ne connaîtra aucune fluctuation

Le client qui parie sur l'Agilité souhaitera en outre conserver une sortie de secours, car son angoisse première est de donner un « chèque en blanc » au prestataire. Il s'agit alors d'être imaginatif, et de trouver des moyens d'intéressement commun au succès rapide du projet ou à sa diversification fonctionnelle. Certains prestataires proposent alors de convertir le cahier des charges initial en nombre de jours/hommes total, stipulé dans le contrat et réparti par fonctionnalité. Le forfait contractuel porte alors sur ce nombre de jours/hommes, et plus sur la couverture fonctionnelle du cahier des charges : celle-ci peut alors évoluer sans contredire les bases contractuelles du projet.

De plus, il est souvent proposé de facturer les prestations au terme de chaque release. Le client conserve ainsi une visibilité optimale sur le coût évolutif du projet, et la possibilité d'y mettre un terme quand bon lui semble. Le mécanisme de sortie doit être stipulé au contrat, rendant au client la faculté décisive qui doit rester la sienne : il est la dernière instance et doit pouvoir décider des options fonctionnelles comme des montants investis. Sur le terrain, on constate d'ailleurs que les projets menés en mode Agile n'obéissent pas à une logique unique, mais agrègent plusieurs principes traditionnels aux innovations de l'Agilité. L'engagement au forfait couvre par exemple les fonctionnalités désignées comme indispensables, et des paiements proportionnels ou en régie sont proposés pour les aspects moins contraignants du besoin du client.

Conclusion

Le projet Agile s'analyse moins en une course de fond épuisante qu'en une succession et la qualité du code de la solution qui deviennent l'étalon-mesure du projet, plutôt que la conformité à un cahier des charges initial vieux de plusieurs mois. Les Méthodes Agiles sont donc pragmatiques, mais elles impliquent une évolution des mentalités et de l'économie du contrat.

Cet aggiornamento ne doit donc pas s'arrêter aux seules équipes techniques : le juriste qui encadre le projet, tout comme la Direction Générale qui en décide, doivent être sensibilisés aux Méthodes Agiles, et comprendre que les objectifs (couverture fonctionnelle, sécurité juridique, efficacité économique) peuvent être atteints autrement, à condition de renoncer à certains réflexes. L'aspect protéiforme des Méthodes Agiles permet également aux prestataires informatiques de proposer leur propre « offre Agile » et d'y puiser des avantages concurrentiels. A l'instar des logiciels libres, les Méthodes Agiles sont issues de la pratique, et permettent manifestement de tels gains de valeur qu'elles impliquent désormais une révolution culturelle au sein de l'entreprise. La protection frileuse du code source ou la relation classique « client-prestataire » sont parfois des freins à l'efficacité des projets IT. On constate au contraire que l'innovation, loin d'être menacée par ces nouvelles propositions, est en fait démultipliée.

Quant au juriste, il relève précisément de son office de s'appropriier ces nouveaux concepts pour leur apporter une traduction juridique et contractuelle pertinente. Le contrat ne peut être un monolithe intangible, théâtre du rapport de forces entre le client et son prestataire et checklist des engagements réciproques : il doit aussi s'adapter aux évolutions opérationnelles et accompagner l'émergence de nouveaux concepts pour en assurer l'efficacité.

A leur arrivée en France, les licences Open Source avaient défrayé la chronique. On les disait intransposables en droit français, anti-économiques, voire utopistes... On constate aujourd'hui qu'elles se sont multipliées au point d'innover la plupart des projets informatiques. Plus aucun juriste français ne songe à les combattre, et elles témoignent de ce qu'une autre conception du droit d'auteur est possible et souhaitable. Si le contrat IT leur permet de s'épanouir, peut-être en ira-t-il de même des Méthodes Agiles ?