

# Contrats et méthodes Agile : comment mieux gérer les risques liés aux projets informatiques

Alexandre CRUQUENAIRE<sup>1</sup>

## TABLE DES MATIÈRES

I.	Éléments caractéristiques des méthodes Agile	42
A.	La méthode Scrum	42
B.	Comparaison avec l'approche traditionnelle (dite <i>Waterfall</i> )	44
II.	Impact des méthodes Agile sur la gestion contractuelle des projets	45
A.	La phase précontractuelle	45
B.	La délimitation du périmètre contractuel	46
C.	L'exécution du contrat Agile	50
D.	La fin du contrat	54
III.	Conclusion	55

1. Les projets informatiques ont besoin d'un cadre contractuel clair et précis. Raison pour laquelle l'on enseigne traditionnellement l'importance de la phase préparatoire au contrat, car elle permet de fixer les spécifications et les objectifs. Ces spécifications sont alors traduites dans un cahier de charges, qui vient définir, au moment de la conclusion du contrat, les balises qui guideront l'exécution du projet<sup>2</sup>. Le périmètre contractuel est donc, en principe, figé une fois le contrat conclu. Cette approche présente le grand avantage d'offrir une sécurité juridique, par une définition claire des attentes et objectifs.

2. De l'idée abstraite à sa mise en pratique, la situation se révèle plus complexe. Il est en effet rarement possible d'enfermer les projets informatiques d'une certaine ampleur dans une clôture de barbelés dissuadant toute déviation du chemin initialement tracé.

Le besoin de flexibilité dans l'exécution des projets informatiques se heurte alors à la rigidité juridique. Le principe de la convention-loi contient l'exécution dans les limites du cahier de charges initial, sauf accord des deux parties pour mettre en œuvre des modifications.

3. En réponse à ces difficultés opérationnelles, de nombreux gestionnaires informatiques ont plaidé pour une approche plus flexible et collaborative de la

relation client-prestataire. C'est dans ce contexte que les méthodes dites Agile ont vu le jour. Le manifeste Agile repose sur quatre valeurs et douze principes, qui traduisent cette volonté de changer de paradigme dans les projets informatiques<sup>3</sup>.

Cette approche différente est en outre également suivie dans de nombreux autres domaines. Si les méthodes Agile ont vu le jour dans le contexte informatique, leurs bénéfices sont exploitables dans tous les domaines. Cette vision différente de la conduite de projets pose toutefois des questions juridiques sur la gestion contractuelle de la relation client-prestataire.

4. La présente contribution a donc pour objet d'analyser en quoi les méthodes de gestion de projets inspirées du manifeste Agile appellent à certaines nuances dans la mise en œuvre des règles propres au droit des contrats (II.). Pour ce faire, il convient tout d'abord de brièvement expliquer les éléments caractéristiques des méthodes Agile, et plus spécialement de la méthode *Scrum* (I.).

## I. Éléments caractéristiques des méthodes Agile

Selon les quatre valeurs du manifeste Agile, la priorité doit être donnée aux interactions entre les individus, ainsi qu'à la collaboration client-prestataire, dans une approche visant à assurer l'adaptation au changement plutôt que la poursuite aveugle d'un plan initialement convenu<sup>4</sup>. La méthode *Scrum* est sans doute la plus connue des déclinaisons du manifeste Agile<sup>5</sup>. Nous nous y limiterons donc dans le cadre de la présente contribution.

### A. La méthode Scrum

5. Afin que la dynamique collaborative induite par la méthode *Scrum* soit le moteur du projet, les

1. Avocat (Partner, Lexing Belgium) et chargé de cours invité (UNamur – CRIDS), Certified Scrum Master.

2. E. MONTERO, *Les contrats de l'informatique et de l'internet*, tiré à part du *Rép. not.*, Bruxelles, Larcier, 2005, pp. 49-51, n° 7.

3. A propos du manifeste Agile, voir not. Wikipédia, verbo « manifeste agile » : [https://fr.wikipedia.org/wiki/Manifeste\\_agile#:~:text=Le%20Manifeste%20pour%20le%20d%C3%A9veloppement,de%20plusieurs%20m%C3%A9thodes%20dites%20agiles](https://fr.wikipedia.org/wiki/Manifeste_agile#:~:text=Le%20Manifeste%20pour%20le%20d%C3%A9veloppement,de%20plusieurs%20m%C3%A9thodes%20dites%20agiles).

4. Scrum Alliance, *Core Scrum*, disponible sur le site officiel de Scrum Alliance, à l'adresse <https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/Learn%20About%20Scrum/Core-Scrum.pdf>, pp. 2-3.

5. La description qui est faite dans la présente section de la méthode Scrum n'est nullement complète et vise simplement à donner un aperçu de ses caractéristiques essentielles. Pour une description plus précise, le lecteur est invité à consulter la documentation éditée par la Scrum Alliance, à laquelle il est renvoyé en notes.

interactions entre les parties sont balisées par une routine de procédures spécifiques entre les acteurs clés.

Logiquement, les processus collaboratifs sont menés par un représentant du client (le *Product Owner*), qui a pour rôle d'être la courroie de transmission d'informations avec les futurs utilisateurs métiers et de veiller à ce que la direction du client adhère aux options choisies. Le *Product Owner* veille à exprimer les attentes du client. Il doit définir la vision du produit<sup>6</sup>, qui va servir de boussole pour ensuite permettre de valider les choix opérés par l'ensemble de l'équipe du projet.

L'équipe de développement est le deuxième acteur majeur, celui qui doit mettre en œuvre le projet et fournir les services et livrables.

Le troisième acteur principal est le *Scrum Master*. Indépendant du client et du prestataire, il doit veiller au suivi des procédures de la méthode *Scrum*, afin de s'assurer que le projet demeure sur les bons rails.

**6.** L'exécution du projet en *Scrum* s'appuie sur un document de référence, dont le caractère évolutif est de l'essence même de la méthode : le *Product Backlog*. Il s'agit d'une liste d'idées, de fonctionnalités, et/ou exigences attendues du produit/livrable final. Pour chaque élément de cette liste (*item*), le *Product Owner* définit un niveau de priorité et évalue l'effort nécessaire à sa réalisation<sup>7</sup>. Le *Product Owner* est chargé de maintenir cette liste à jour tout au long du processus, avec le support de l'équipe de développement<sup>8</sup>.

**7.** La méthode *Scrum* balise l'exécution du projet en la découpant en séquences courtes (*sprints*). Selon l'importance du projet et des ressources mises à disposition, le nombre ou la durée des sprints peut varier. Avant chaque sprint, une réunion de planification (*sprint planning*) réunit l'ensemble de l'équipe de projet<sup>9</sup>. Cette réunion vise à déterminer les points du *Product Backlog* qui seront à réaliser au cours du sprint qui va démarrer. Le choix se fait sur la base des priorités définies dans le *Product Backlog*, de l'effort requis pour les *items* souhaités, ainsi que la vitesse de l'équipe de développement. En fonction de ces différents paramètres, l'équipe du projet opère des choix et définit l'objectif du sprint (*Sprint Backlog*).

Une fois l'objectif défini, il revient à l'équipe de développement de réaliser les points prévus dans le *Sprint Backlog*. La méthode *Scrum* repose sur une évaluation incrémentale du produit, ce qui suppose qu'à l'issue de chaque sprint, l'équipe de développement soit en mesure de présenter les éléments développés<sup>10</sup> (fonctionnalités, partie de logiciel, composant du produit final, ...). Le processus de développement se clôture donc logiquement par une réunion de démonstration et d'évaluation des résultats du sprint (*Sprint review*)<sup>11</sup>. Les résultats sont évalués au regard des attentes du *Product Owner*, par référence aux spécifications du *Product Backlog* et du *Sprint Backlog*. A l'issue de cette réunion, le *Product Backlog* est mis à jour, afin d'identifier les points validés<sup>12</sup>, revoir les priorités<sup>13</sup>, ajouter de nouveaux *items* (ou en supprimer), réévaluer la charge de travail estimée pour certains *items*, ... La méthode vise à coller à la réalité d'exécution et non à maintenir artificiellement des exigences ou des estimations qui s'avèrent manifestement irréalistes. De la sorte, l'on évite le risque d'un décalage trop important entre les attentes du client et les résultats atteints ou les objectifs atteignables.

L'équipe *Scrum* peut alors définir les objectifs du sprint suivant, et ainsi de suite, jusqu'au dernier sprint du processus<sup>14</sup>. A l'issue du dernier sprint, c'est le produit complet qui doit être testé et évalué, avec tous les incréments provenant des sprints antérieurs.

**8.** Par cette répétition de routines collaboratives, la méthode *Scrum* induit une dynamique collective inclusive, impliquant les équipes du client et du prestataire, qui contribue à une meilleure compréhension réciproque. La réévaluation régulière des objectifs et attentes, ainsi que le découpage du projet en petites étapes validées et discutées entre les parties permet de définir des horizons de travail rapprochés et de rapidement détecter les écarts, incompréhensions ou difficultés. La flexibilité de la méthode permet alors de rectifier les erreurs et donc de limiter les risques d'un crash final du projet. L'implication régulière du client dans la définition des objectifs de chaque sprint, dans la validation de leurs résultats, permet en outre de mieux exprimer les attentes concrètes du client et de les mettre à jour au regard de l'avancement du projet.

6. *Product Vision*.

7. Un nombre de points est ainsi attribué à chaque élément, en fonction de son ampleur et de sa difficulté.

8. Scrum Alliance, *Core Scrum*, op. cit., p. 7.

9. Le *Product Owner*, l'équipe de développement et le *Scrum Master*.

10. *Product increment*.

11. Scrum Alliance, *Core Scrum*, op. cit., p. 12.

12. Conformément à la définition des exigences convenues entre les parties pour considérer un élément comme « fini » (*definition of done*). La définition du « fini » a elle-même vocation à évoluer au fil du projet, au fur et à mesure que l'on s'approche d'un produit final. Scrum Alliance, *Core Scrum*, op. cit., p. 11.

13. L'idée étant qu'en cours de processus, le *Product Owner* peut faire part de nouvelles considérations mises en avant par les utilisateurs (notamment suite aux démos des éléments de produits lors des *sprint reviews* antérieures).

14. Le nombre de sprints variant selon l'importance du projet et les performances de l'équipe de développement (en fonction de sa taille et de l'expérience de ses membres).

## B. Comparaison avec l'approche traditionnelle (dite *Waterfall*)

9. Dans une conception traditionnelle des projets de développement, la formation et l'exécution du contrat se déroulent en trois phases. Tout d'abord, le client et le prestataire définissent les spécifications de la solution à fournir. Les bonnes pratiques recommandent de détailler ces éléments dans un cahier de charges qui sera annexé au contrat<sup>15</sup>. Le prestataire passe alors à la phase d'exécution proprement dite et réalise les prestations nécessaires pour finalement délivrer le produit convenu. L'exécution se clôture alors par la phase d'agrément du travail, au cours de laquelle le client vérifie que le produit livré est conforme aux spécifications du cahier des charges. En cas de validation, le produit peut alors être mis en production<sup>16</sup>.

Cette approche est qualifiée de « *waterfall* » car tout le processus d'exécution se déroule en cascade, d'une manière linéaire, sans aller-retour entre les parties.

10. L'approche *Waterfall* présente l'avantage d'une définition *ex ante* des spécifications, ce qui offre une sécurité juridique quant au périmètre contractuel.

La pratique des projets informatiques a toutefois démontré que cette sécurité juridique est toute relative<sup>17</sup>. En effet, dans un domaine aussi évolutif que les technologies de l'information, il est très fréquent qu'un projet évolue dans ses objectifs ou besoins en cours d'exécution. L'apparente sécurité juridique offerte par un périmètre clair et bien délimité est alors fragilisée par la nécessité de procéder à des ajustements, sous peine d'aboutir à un résultat final ne permettant pas de rencontrer les attentes réelles du client. La gestion contractuelle du projet informatique est alors contrariée par le ballet des fameux « *change requests* » introduits par le client afin d'adapter les objectifs à l'évolution de ses besoins. Cela se traduit par des avenants contractuels, qui complexifient le cadre juridique du projet et peuvent être source de contradictions ou incohérences avec le contrat initial. Autre inconvénient majeur de cette approche d'actualisation des spécifications contractuelles : les retards d'exécution et surcoûts sont souvent importants et difficiles à maîtriser pour le client, qui se trouve à la merci du prestataire puisqu'il est lié par le contrat initial et dépend donc d'un accord du prestataire pour toute évolution des objectifs contractuels.

A l'inverse l'approche Agile se base sur un catalogue de spécifications (*Product Backlog*) délibérément ouvert, qui peut être revu par le client tant dans son contenu que dans les degrés de priorité assignés à chaque élément dudit catalogue. En effet, le *Product Backlog* est mis à jour d'une manière continue et à tout le moins à l'issue de chaque *sprint*. Le processus Agile contient donc un mécanisme de révision qui évite les avenants contractuels.

11. En figeant le périmètre contractuel dans le contrat, l'approche traditionnelle permet de fixer le prix<sup>18</sup> et donc éviter toute dérive budgétaire. C'est vrai, en théorie. La pratique montre que le verrou budgétaire n'est pas à toute épreuve et qu'il ne résiste pas aux modifications nécessaires en cours d'exécution. En effet, la négociation d'un *change request* est souvent l'opportunité pour le prestataire de récupérer une partie des marges qu'il aurait laissé tomber lors de la négociation initiale du contrat. La sécurité budgétaire de l'approche *waterfall* est donc à relativiser dans les faits. En outre, la négociation d'un avenant expose au risque de compromettre les délais d'exécution, car le prestataire se trouve également en position de force pour solliciter un allongement des délais initialement convenus. Notons qu'il est tout à fait possible de maîtriser ces risques dans l'approche *waterfall*, en prévoyant des clauses de gestion des changements protégeant les aspects budgétaire et calendrier. Il convient toutefois de garder à l'esprit qu'une fois de telles clauses activées durant l'exécution du projet, il sera difficile de cantonner la discussion aux seuls points touchés par les changements proposés. Dans ce cas, en effet, le prestataire sollicitera logiquement des assouplissements permettant de rencontrer d'éventuelles difficultés sur d'autres aspects du projet. Le caractère figé du périmètre contractuel deviendra alors très théorique.

Si les processus Agile permettent de rencontrer le besoin de changement d'une manière précoce, grâce aux interactions permanentes entre les parties, cela ne va pas sans poser certaines questions juridiques. Le risque de dérive budgétaire est souvent pointé à l'encontre des méthodes Agile, qui sont naturellement associées à une démarche de travail en régie. A cet égard, il est fréquent dans les contrats Agile de prévoir une enveloppe budgétaire fermée, ce qui rencontre l'impératif de contrôle financier. Par contre, le caractère fermé de l'enveloppe se traduira par un volume de ressources (au sein de l'équipe de

15. J.-P. TRIAILLE et R. ROBERT, « Les contrats informatiques », in *Traité pratique de droit commercial, Tome I – Principes et contrats fondamentaux*, 2<sup>e</sup> éd., Bruxelles, Kluwer, 2009, pp. 1090-1091 ; D. GOBERT et E. MONTERO, « Les obligations de conformité et de garantie des vices cachés en matière informatique : le contrat au secours des incertitudes légales et jurisprudentielles », *R.D.T.I.*, 2002/11, p. 12.

16. En fonction de la nature du projet et des rôles respectifs des parties, la mise en production constitue une tâche dévolue au prestataire ou gérée par le client lui-même. Lorsque le produit livré ne constitue qu'une pièce d'un ensemble plus large, un autre prestataire (intégrateur) peut être sollicité afin de combiner le produit livré avec les autres composants en vue de constituer une solution plus globale. Dans ce cas, la validation peut s'opérer en plusieurs étapes, car des tests d'intégration sont nécessaires pour contrôler que le produit est non seulement capable de fonctionner correctement seul, mais également en interaction avec les autres composants de la solution globale.

17. Sur les difficultés pratiques induites par la nécessité de figer les spécifications au début du projet *Waterfall*, voir la saga judiciaire relative au projet Maif : E. VARET, « IBM c/Maif : la Maif contre-attaque », *R.L.D.I.*, 2015/115, n° 3747.

18. Pour autant que le contrat n'ait pas été conclu en régie.

développement) limité en conséquence. Dès lors, il conviendra pour le client de faire des concessions en termes d'étendue des développements prévus. La possibilité pour les intervenants au projet Agile de prendre des décisions ayant un impact potentiel aussi important pose la question des pouvoirs conférés aux acteurs du projet, et en particulier au *Product Owner*, qui représente le client. Le contrat Agile doit rencontrer ces risques, afin que la méthode Agile et ses implications pratiques soient bien maîtrisées.

**12.** La phase d'agrément du travail est également différente dans les deux approches comparées. L'agrément se fait traditionnellement par une phase de test en fin de développement dans l'approche *waterfall*. Par contre, la répétition de cycles courts dans le processus Agile implique une agrément de chaque incrément du produit final. L'avantage est de permettre une évaluation et réorientation plus rapide. Le saucissonnage de l'agrément en différentes tranches pose cependant la question de la valeur juridique des agréments partiels successives.

## II. Impact des méthodes Agile sur la gestion contractuelle des projets

**13.** Si le manifeste Agile repose notamment sur une valeur énonçant que l'approche collaborative doit primer l'aspect contractuel<sup>19</sup>, les acteurs spécialisés se sont rapidement aperçus qu'il était risqué d'ignorer les aspects contractuels<sup>20</sup>. La contractualisation d'un projet selon la méthode Agile impose de la souplesse sur le plan juridique aussi. L'approche de type Agile soulève une série de questions quant à l'application des principes du droit des contrats. Pour la clarté du propos, la suite de l'exposé envisagera les questions ainsi posées selon un plan chronologique de la vie du contrat : phase précontractuelle (A), délimitation du périmètre contractuel (B), exécution du contrat (C), et fin du contrat (D).

### A. La phase précontractuelle

**14.** Compte tenu de la technicité des projets informatiques et de l'obligation de bonne foi qui doit présider à la formation du contrat, la jurisprudence accorde une importance particulière à la phase de négociation et a dégagé plusieurs obligations propres à cette phase de construction du contrat<sup>21</sup>. Il s'agit principalement

des obligations d'information et de conseil du prestataire informatique<sup>22</sup>.

En vertu du devoir de conseil, le prestataire informatique est tenu non seulement de communiquer au futur client tous les éléments d'information dont il sait (ou doit savoir) qu'ils peuvent présenter un intérêt pour sa décision de contracter, mais aussi d'informer le client sur les avantages et inconvénients de la solution discutée. L'exigence n'est plus seulement de partager de l'information, mais, au-delà, de véritablement orienter le choix du client en fonction de ses besoins et objectifs<sup>23</sup>.

La jurisprudence considère, par exemple, que le prestataire doit participer activement à l'évaluation des besoins du client afin de lui fournir la solution la plus adéquate<sup>24</sup>.

**15.** Comment concilier l'exigence de conseil, préalable à la conclusion du contrat, avec l'approche Agile, qui implique que l'objet contractuel soit délibérément défini d'une manière non détaillée et sujet à révision tout au long du processus d'exécution ?

Le choix de la méthode Agile devrait se traduire par une atténuation de l'obligation préalable de conseil, en tout cas selon la définition classiquement donnée à cette obligation. En effet, durant la phase de négociation, l'objet contractuel ne sera pas défini avec le même degré de précision et la marge de souplesse qui en découle est inhérente au processus Agile.

Il est dès lors logique que l'on diminue l'exigence de précision dans les informations et conseils préalables sur les contours de la solution faisant l'objet du contrat. Par contre, les devoirs de conseil du prestataire pourraient se trouver alourdis à un double égard. D'une part, quant aux implications de la méthode de pilotage de projet choisie. D'autre part, lorsque le processus Agile implique une prise de décision entre les parties sur les choix à opérer dans l'orientation du projet.

Sur le second aspect évoqué, l'on opérerait ainsi un transfert partiel de la portée du devoir de conseil vers le champ contractuel. C'est sans doute en ce sens qu'il convient de lire les quelques décisions de jurisprudence qui soulignent que le choix d'une méthode Agile ne permet pas d'évacuer l'obligation de conseil du prestataire, « *puisque cette obligation avait été*

19. « *Customer collaboration over contract negotiation* ».

20. T. BEAUGRAND, J.-B. BELIN, « Logiciel. Le contrat de développement logiciel en méthode Agile », *Expertises*, 2013, p. 416.

21. Pour un aperçu général de ces obligations, voir not. : M. COIPEL et Y. POULLET, « Introduction aux concepts juridiques », in *Le droit des contrats informatiques – Principes – Applications*, Bruxelles-Namur, Larcier-Société d'études morales, sociales et juridiques, 1983, pp. 29-143 ; E. MONTERO, *Les contrats de l'informatique et de l'internet*, tiré à part du *Rép. not.*, Bruxelles, Larcier, 2005, pp. 57-58.

22. Sur la portée de l'obligation de conseil, voir ainsi : jak18, pp. 11-12 ; B. DOCQUIR, « Droit du numérique », *R.P.D.B.*, Bruxelles, Bruylant, 2018, pp. 24-31.

23. E. MONTERO, *Les contrats de l'informatique et de l'internet*, tiré à part du *Rép. not.*, Bruxelles, Larcier, 2005, p. 62 ; Y. POULLET et al., *Droit de l'informatique et des technologies de l'information. Chronique de jurisprudence (1995-2001)*, Les dossiers du Journal des Tribunaux, n° 41, Bruxelles, Larcier, 2003, p. 12.

24. Bruxelles, 29 juin 2017, *R.G.D.C.*, 2017, p. 574.

contractuellement définie »<sup>25</sup>. Plutôt qu'une atténuation du devoir de conseil du prestataire<sup>26</sup>, il nous paraît préférable d'y voir une reformulation et un déplacement liés au contexte particulier des contrats Agile et au processus collaboratif continu que la méthode induit.

Un tel constat n'a rien de surprenant. En effet, le fondement de l'obligation d'information réside dans l'asymétrie informationnelle entre les parties au moment où la partie réputée moins informée doit prendre une décision juridiquement engageante<sup>27</sup>. C'est en raison de ce déséquilibre que le principe de bonne foi commande à la partie plus informée de délivrer un conseil avisé à son partenaire contractuel<sup>28</sup>. Selon l'approche classique, puisque l'objet du contrat est figé au moment de l'établissement des documents contractuels<sup>29</sup>, il est logique que la jurisprudence ait accordé une attention particulière à la période de négociation, avant que le client marque son consentement sur cette délimitation de l'objet contractuel. Dans un contrat Agile, le périmètre étant souple et évoluant au fil des sprints et des réunions successives entre les parties, il est logique que l'intensité du devoir de conseil soit moindre dans la phase préalable à la conclusion du contrat, mais plus importante dans la phase d'exécution du contrat, lorsque les parties vont réévaluer les contours de l'objet contractuel, le préciser, le réorienter.

## B. La délimitation du périmètre contractuel

**16.** L'usage des méthodes Agile peut avoir un impact sur le périmètre contractuel, à plusieurs égards. L'on peut tout d'abord s'interroger sur les documents à intégrer dans le contrat Agile, afin de sécuriser la portée des engagements (1). La nature variable de l'objet contractuel au fil de la vie du contrat mérite une attention particulière (2). Enfin, la définition du budget et son contrôle constituent des points sensibles (3).

### 1) Les documents clés d'un contrat Agile

**17.** La délimitation des documents faisant partie du cadre contractuel est importante. Distinguer ce qui relève de simples discussions préalables de ce qui devient un engagement juridique n'est pas toujours un exercice aisé lorsque le contrat est l'aboutissement

d'un long processus de négociation. La plupart des contrats informatiques comportent d'ailleurs une clause d'intégralité visant à écarter du champ contractuel les documents auxquels le contrat ne fait pas explicitement référence<sup>30</sup>.

**18.** Dans un projet Agile, le fait que l'objet contractuel ne soit pas complètement figé au moment de la conclusion du contrat rend la question un peu moins cruciale. Les imprécisions, lacunes ou contradictions peuvent en effet être corrigées en cours d'exécution, en particulier lorsque les parties mettent à jour le *Product Backlog*. Si ce dernier a vocation à jouer le rôle de cahier de charges dans la méthode Agile, il s'en distingue par son caractère évolutif et ouvert<sup>31</sup>.

Doit-on en conclure qu'il n'est pas nécessaire d'annexer le *Product Backlog* au contrat ? Il nous semble que la réponse est négative. Il est en effet recommandé de joindre au contrat le *Product Backlog* dans son état initial, ou à tout le moins un modèle de ce document lorsqu'il n'existe aucun catalogue de spécifications au moment de la signature du contrat. Il n'est pas rare que le projet Agile démarre par un sprint initial (*kick-off sprint*), un peu plus long, au cours duquel les parties vont précisément s'accorder sur la version 1 du *Product Backlog*. Dans cette perspective, il n'est naturellement pas possible d'annexer le *Product Backlog* au contrat. Par contre, il est utile et conseillé de joindre au contrat un modèle, de sorte qu'à tout le moins la structure et le format du document clé du contrat soient bien approuvés par les parties. Dans ce cas, il est également conseillé de prévoir une faculté de résiliation unilatérale du contrat à l'issue de ce sprint initial, afin que le client ne soit pas lié au prestataire s'il s'avère que la définition initiale du *Product Backlog* révèle des divergences problématiques entre les parties<sup>32</sup>.

**19.** Outre le *Product Backlog*, un autre document important du contrat Agile est le *Product Vision*, qui décrit les attentes et objectifs généraux du projet. Dans la mesure où ces objectifs essentiels ne sont pas appelés à varier en cours d'exécution, le *Product Vision* constitue un élément important du projet. Il est dès lors recommandé d'annexer au contrat un document décrivant ces objectifs fondamentaux qui doivent guider la définition d'objectifs plus précis et l'exécution du

25. Paris, 3 juillet 2015, *Expertises*, 2015, n° 408, p. 444.

26. Interprétant *a contrario* l'arrêt de la Cour d'appel de Paris en ce sens, voir toutefois A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 434.

27. Sur le fondement du devoir de conseil dans le déséquilibre informationnel, voir : M. COIPEL et Y. POULLET, « Introduction aux concepts juridiques », in *Le droit des contrats informatiques – Principes – Applications*, Bruxelles-Namur, Larcier-Société d'études morales, sociales et juridiques, 1983, p. 73 ; E. MONTERO, « Les obligations d'information, de renseignement, de mise en garde et de conseil des fabricants et vendeurs professionnels », in *Les obligations d'information, de renseignement, de mise en garde et de conseil*, CUP, n° 86, Bruxelles, Larcier, 2006, pp. 320-323 ; A. CRUQUENAIRE et J.-F. HENROTTE, « Le devoir de conseil dans le Règlement Général sur la Protection des Données : bis repetita placent ? », in *Droit, normes et libertés dans le cybermonde – Liber Amicorum Yves Pouillet*, Bruxelles, 2018, p. 602.

28. Ou de négociation dans la phase précontractuelle.

29. En ce compris le cahier de charges.

30. On parle souvent de clause « des quatre coins » en la matière. Sur la portée pratique d'une telle clause, voir A. CRUQUENAIRE, *L'interprétation des contrats en droit d'auteur*, Bruxelles, Larcier, 2007, p. 186, n° 246 ; E. MONTERO, *Les contrats de l'informatique et de l'internet*, tiré à part du *Rép. not.*, Bruxelles, Larcier, 2005, pp. 67-68.

31. A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 433.

32. En ce qui concerne la fin du contrat Agile et ses conséquences, voir ci-après, n° 44 et s.

projet<sup>33</sup>. De cette manière, le client détermine des limites à la flexibilité induite par la méthode Agile, en fixant des balises générales sur les objectifs essentiels à atteindre.

**20.** Enfin, il convient de ne pas négliger la hiérarchie entre les différents documents qui seraient incorporés dans le champ contractuel<sup>34</sup>. En cas d'incohérence ou de contradiction, le contrat doit déterminer quel document prime. Les parties doivent donc définir dans une clause spécifique l'ordre de priorité des documents contractuels.

## 2) Un objet contractuel à contours souples

**21.** L'objet du contrat se définit comme ce à quoi s'engage le débiteur, la prestation promise<sup>35</sup>. Les méthodes Agile se caractérisent par une évolutivité de l'objet des prestations, au gré de l'actualisation du *Product Backlog*. Cet aspect fondamental interroge les principes du droit des obligations, à plusieurs égards.

L'exigence de déterminabilité de l'objet est-elle rencontrée lorsque l'on définit l'objet du contrat avec une telle souplesse, voire lorsqu'on s'abstient de le définir précisément en laissant même la définition initiale à un kick-off sprint ?

Ensuite, si les méthodes Agile prévoient une variabilité de l'objet des prestations, les modifications de l'objet qui en découlent sont-elles juridiquement contraignantes pour les parties contractantes ?

La faculté d'opérer des ajustements potentiellement fondamentaux sur l'objet contractuel en cours d'exécution met aussi au premier plan la traditionnelle obligation de conseil du prestataire informatique.

Enfin, cette nature ouverte de l'objet des prestations n'ouvre-t-elle pas la voie à une sortie plus aisée du contrat Agile ?

**22.** L'exigence de déterminabilité de l'objet signifie que l'objet contractuel doit être déterminé ou à tout le moins déterminable lors de la conclusion du contrat. A cet égard, il suffit que le contrat comporte l'indication des éléments objectifs qui permettront de fixer l'objet de l'obligation sans qu'un nouvel accord de volonté soit requis<sup>36</sup>. La justification de cette condition

est de préserver chaque partie contre l'arbitraire de son cocontractant, auquel l'une des parties pourrait être exposée si le contrat ne permet pas de déterminer la nature et l'étendue des prestations<sup>37</sup>. Cette exigence est-elle rencontrée dans le cadre d'un contrat Agile pour lequel l'objet des services est laissé à une détermination ultérieure et incrémentale ?

La réponse à la question devrait être positive de notre point de vue, pour autant que la nature des services et les objectifs du projet soient définis dans le contrat. En effet, il nous semble que l'intention des parties au contrat Agile est de confier au prestataire la fourniture de services visant à atteindre les objectifs du projet, selon une méthode de travail qui implique la mise en œuvre de processus collaboratifs spécifiques. Dans ce contexte, l'intention des parties quant à la délimitation de l'objet du contrat nous semble claire et l'objet est déterminable, en dépit du fait que le résultat final attendu à l'issue des prestations<sup>38</sup> n'est pas encore défini en détail, compte tenu du choix de méthode opéré par les parties<sup>39</sup>. Le fait que la procédure d'exécution du contrat requiert des accords entre les parties quant à la précision de l'objet (au sens de livrable final) attendu des services ne doit pas être mal compris. Si le livrable final est encore imprécisément défini au stade de la formation du contrat, l'objet du contrat est quant à lui bien déterminé, à savoir : la fourniture de services en vue d'atteindre l'objectif défini au contrat selon une méthode Agile dont les modalités sont elles aussi définies au contrat. L'exigence de déterminabilité de l'objet ne devrait donc pas constituer un obstacle juridique à l'usage des méthodes Agile<sup>40</sup>. La modification dans les attentes précises et les contours du livrable final devrait davantage s'analyser sous l'angle d'une adaptation du contrat conformément à ses clauses, dans la mesure où c'est le choix d'exécuter les prestations selon la méthode Agile qui implique cette variabilité dans l'exécution du contrat<sup>41</sup>.

**23.** La variabilité dans l'objet des prestations pose en outre la question de la valeur juridique des multiples décisions qui sont adoptées par les parties dans le cours d'un projet Agile. La méthodologie *Scrum* offre ainsi de nombreuses opportunités de mettre à jour les attentes et les spécifications. Ainsi, le *Product Backlog* peut être mis à jour en continu, et en particulier lors de la réunion de validation de chaque sprint (*sprint review meeting*). En cas de désaccord ultérieur,

33. T. HOEREN and S. PINELLI, « Agile programming – Introduction and current legal challenges », *Computer Law and Security Review*, 2018, p. 1134.

34. O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 47.

35. P.A. FORIERS, « L'objet et la cause du contrat », in *Les obligations contractuelles*, Bruxelles, Editions du Jeune Barreau, 1984, p. 102.

36. P.A. FORIERS, « L'objet », in *Les obligations en droit français et en droit belge. Convergences et divergences*, Bruxelles, Bruylant, Paris, Dalloz, 1994, p. 67.

37. P. WÉRY, *Droit des obligations, Volume 1 – Théorie générale du contrat*, Bruxelles, Larcier, 2010, n° 290.

38. Et, par ricochet, l'objet précis des prestations elles-mêmes.

39. Sur la délimitation de la portée de l'objet contractuel, voir A. CRUQUENAIRE, « Le paradoxe de l'œuf et de la poule : réflexions sur les liens entre l'objet et l'interprétation du contrat », in *Entre tradition et pragmatisme. Liber Amicorum Paul Alain Foriers*, Bruxelles, Larcier, 2021, pp. 302-305.

40. Confirmant la validité d'un contrat Agile, voir not. Trib. Com. Paris, 7 octobre 2020, <https://www.legalis.net/jurisprudences/tribunal-de-commerce-de-paris-8eme-ch-jugement-du-7-octobre-2020/> (soulignant la conclusion d'un contrat portant sur d'applications mobiles et de site internet en l'absence de cahier de charges).

41. Sur la distinction entre novation par changement d'objet, convention modificative et adaptation du contrat en application de l'une de ses clauses, voir les développements éclairants de P. WÉRY, *Droit des obligations, Volume 1 – Théorie générale du contrat*, Bruxelles, Larcier, 2010, pp. 801-805.

les décisions prises lors des réunions de projet pourraient-elles être remises en cause ?

A ce propos, il convient tout d'abord d'observer que les processus *Scrum* doivent faire partie du champ contractuel<sup>42</sup>. Tel sera le cas si le contrat Agile s'y réfère expressément et/ou contient des dispositions spécifiques décrivant les processus collaboratifs à mettre en œuvre. Il est important que les personnes habilitées à représenter et engager les parties soient clairement identifiées dans ces clauses de gouvernance contractuelle, afin qu'elles disposent d'un pouvoir de prendre des décisions liant les parties. Si de telles clauses pourraient être vues comme conférant un mandat aux représentants identifiés, cette qualification n'évacue pas toutes les difficultés, notamment en raison du fait que les représentants ne sont la plupart du temps pas signataires du contrat qui les désigne<sup>43</sup>.

Dans les rapports entre parties contractantes, il nous semble toutefois que différents éléments devraient conduire à considérer les décisions prises par leurs représentants comme juridiquement contraignantes pour les parties. Tout d'abord, le principe de l'exécution de bonne foi des conventions impose notamment aux contractants de s'abstenir de tout comportement qui, dans l'exécution de la convention, serait de nature à induire leur partenaire en erreur sur l'étendue de ses droits<sup>44</sup> ou serait de nature à priver le partenaire du bénéfice normal de ses droits<sup>45</sup>. Lorsque les parties ont convenu de collaborer selon la méthode Agile et que ce procédé est entré dans le champ contractuel, il est évident que, de la volonté même des parties, les décisions prises dans les réunions de projet doivent être considérées comme juridiquement contraignantes. Toute lecture du contrat contestant le caractère contraignant de ces décisions viendrait ruiner les attentes légitimes du cocontractant. En outre, une telle interprétation du contrat contrarierait frontalement la commune intention des parties quant à l'un des éléments fondamentaux de

la relation contractuelle propre à un contrat Agile. Il y aurait alors selon nous une violation du principe contenu à l'article 1156 du Code civil<sup>46</sup>. L'exécution de bonne foi du contrat, conformément à l'intention des parties, impose donc de respecter les décisions prises selon les procédures convenues. Enfin, compte tenu du contexte contractuel propre à un projet Agile, l'on pourrait aussi considérer qu'il y a un mandat apparent donné aux représentants des parties afin de prendre en leur nom des décisions contraignantes sur l'exécution du contrat Agile<sup>47</sup>. Une partie au contrat Agile devrait donc pouvoir se prévaloir de l'apparence de mandat donné par son cocontractant aux représentants qu'il a désignés en vue des réunions de gestion du projet Agile<sup>48</sup>.

Si des arguments juridiques permettent donc de fonder le caractère contraignant des décisions prises dans le cadre des processus Agile<sup>49</sup>, il est toutefois nettement recommandé de prévoir des dispositions contractuelles particulières afin d'apporter une sécurité juridique totale. Des clauses aménageant un régime de preuve des décisions juridiquement contraignantes constitueront la réponse idéale à ce besoin important lié au recours à des méthodes de gouvernance particulières<sup>50</sup>. Ce type de clause doit organiser les différents aspects liés à ces prises de décision<sup>51</sup>.

**24.** Puisque l'objet précis des prestations et les spécifications détaillées des livrables à fournir sont appelés à évoluer au gré des décisions prises lors des réunions de projet, le devoir de conseil du prestataire devrait trouver ici un terrain d'application particulièrement fertile. En effet, comme déjà souligné, la justification du devoir de conseil réside dans une asymétrie dans l'accès à l'information pertinente pour les parties au contrat<sup>52</sup>. La doctrine s'interroge parfois sur une possible dilution du devoir de conseil dans les méthodes Agile<sup>53</sup>, lisant dans certaines décisions de jurisprudence une exigence de formalisation du devoir de conseil afin qu'il demeure applicable dans

42. Attirant l'attention sur la nécessité d'incorporer les méthodes Agile appliquées dans le contrat, voir not. T. BEAUGRAND, J.-B. BELIN, « Logiciel. Le contrat de développement logiciel en méthode Agile », *Expertises*, 2013, p. 417.

43. Sur les critères de qualification en contrat de mandat, voir A. CRUQUENAIRE, C. DELFORGE, I. DURANT et P. WÉRY, *Précis des contrats spéciaux*, Bruxelles, Kluwer, 2015, pp. 666 et s.

44. En ce sens, voir J.-L. FAGNART, « L'exécution de bonne foi des conventions : un principe en expansion », *R.C.J.B.*, 1986, p. 294.

45. J. F. ROMAIN, *Théorie critique du principe général de bonne foi en droit privé*, Bruxelles, Bruylant, 2000, p. 870.

46. Prescrivant la primauté de la commune intention des parties dans l'interprétation du contrat.

47. Les conditions propres à la théorie de l'apparence sont : (1) il doit exister une situation apparente non conforme à la réalité, (2) la partie contractante contre laquelle on l'invoque doit, par son comportement, avoir au moins contribué au développement de cette situation apparente, et (3) la croyance de l'autre partie à la réalité de cette situation apparente doit être légitime. Dans le cas d'un mandat apparent lié à la mise en œuvre des processus Agile, il nous semble indiscutable que les parties contribuent à créer (sans faute) une apparence de mandat. En outre, il est légitime de se fier à l'existence d'un mandat dans la mesure où le recours aux procédés Agile est un élément essentiel du contrat et que les personnes représentant les parties dans l'exécution du projet sont clairement identifiées au contrat. Pour plus de détails sur les conditions de la théorie de l'apparence, voir A. CRUQUENAIRE, *L'interprétation des contrats en droit d'auteur*, Bruxelles, Larcier, 2007, pp. 299-301.

48. Nous n'aborderons pas ici la question d'un éventuel dépassement de mandat, lorsque le représentant prendrait une décision non approuvée par la partie qu'il représente. En effet, cette hypothèse ne devrait pas remettre en cause le caractère contraignant des décisions prises, mais plutôt concerner la possible responsabilité du représentant à l'égard de la partie qui l'avait mandaté.

49. La jurisprudence a d'ailleurs confirmé ce caractère contraignant. Voir ainsi : C.A. Paris, 3 juillet 2015, n° 13/06963, *Expertises*, 2015, n° 408, p. 444 (concernant des PV envoyés, non contestés, mais pas signés).

50. En ce sens, voir O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48.

51. Par exemple, la question de l'établissement et la validation des comptes-rendus de réunions doit être réglée. Sur l'importance de soigneusement organiser la gouvernance dans ce type de contrat, voir not. J.-P. BEARDWOOD, « Agile Approach Case Study of CIS General Insurance v IBM UK. More lessons learned from another failed Agile project », *Computer Law Review International*, 2018/6, p. 188.

52. Voir *supra*, n° 16.

53. Voir not. A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 435.

un tel contexte. Cette lecture de la jurisprudence ne nous semble pas totalement correcte. Dans un arrêt souvent cité, la Cour d'appel de Paris énonce certes que « [le prestataire] ne peut s'exonérer de son obligation de conseil, en se prévalant d'un copilotage du contrat en mode Agile, puisque cette obligation avait été contractuellement définie ». Nous ne sommes toutefois pas convaincus par une interprétation *a contrario* de cet arrêt, consistant à faire de la formalisation contractuelle une condition de maintien du devoir de conseil en mode Agile. En effet, si la méthode Agile exige une étroite implication du client, l'asymétrie informationnelle entre les parties ne disparaît pas du seul fait que l'on adopte un mode de pilotage particulier pour le projet. Dès lors, le simple fait de suivre les méthodes Agile ne permet pas d'évacuer le devoir de conseil.

L'implication plus grande du client pourrait toutefois conduire à nuancer la portée du devoir de conseil<sup>54</sup>. Opter pour les méthodes Agile requiert une vigilance accrue du client, puisqu'il sait (ou ne peut raisonnablement ignorer) l'importance particulière des réunions de suivi du projet dans une telle approche. Dans ce contexte, un devoir de conseil quelque peu amoindri dans le chef du prestataire pourrait se concevoir. Une exonération complète du prestataire serait, par contre, excessive<sup>55</sup>.

A l'inverse, l'on pourrait soutenir que l'importance particulière de certaines décisions prises en réunions Agile devrait renforcer le devoir de conseil du prestataire<sup>56</sup>. Plus le processus est collaboratif et inclusif du client, plus celui-ci devrait pouvoir se reposer sur l'expertise du prestataire pour prendre les bonnes décisions. Le devoir de conseil glisserait donc pour partie du domaine précontractuel vers la phase d'exécution du contrat, mais ne s'en trouverait pas affaibli. Le principe d'exécution de bonne foi impose en effet de fixer les limites du devoir de conseil au regard de l'importance de l'information contractuelle pour l'autre partie<sup>57</sup>. Il semble donc hasardeux de conclure à une véritable atténuation du devoir de conseil en matière de contrats Agile<sup>58</sup>. Au contraire, il serait plus conforme à l'approche collaborative de renforcer ce devoir de conseil.

**25.** Le fait de ne pas déterminer un résultat précis à atteindre ou des spécifications détaillées et fixes dès le départ est parfois invoqué afin de justifier un saucissonnage du contrat Agile. Ainsi, dans une espèce caractérisée par l'absence de contrat écrit, le Tribunal de commerce de Paris a jugé que le contrat Agile constituait une forme de contrat de « *développements informatiques à la pièce* ». Et d'en conclure que le client pouvait dès lors mettre fin à ce contrat à tout moment, sans indemnité, car l'objet de ce type de contrat présentait un fort degré d'aléa, supposant un renouvellement régulier de la relation contractuelle<sup>59</sup>. La position du tribunal nous semble très sévère et s'explique en grande partie par l'absence de convention écrite, qui empêche de déterminer aisément l'intention des parties. Cette jurisprudence doit en tout cas inciter à la prudence et démontre la nécessité de formaliser le contrat Agile dans un document spécifique et adapté à ce type d'approche. En présence d'un contrat détaillant d'une manière précise la manière dont les parties définissent les obligations réciproques et les éventuelles facultés de résiliation unilatérale, il ne sera plus possible pour un tribunal de considérer que le contrat peut être rompu à l'issue de chaque sprint sans aucune forme d'indemnité<sup>60</sup>.

**26.** Le caractère variable des spécifications et attentes contractuelles se traduit par une qualification plus ou moins naturelle des obligations du prestataire Agile en obligations de moyens<sup>61</sup>. Mais, ici encore, tout est question de circonstances et la méthode Agile n'empêche nullement, par principe, l'existence d'obligations de résultat<sup>62</sup>. Il convient en effet de ne pas perdre de vue que la qualification des obligations contractuelles doit être fondée sur la commune intention des parties<sup>63</sup>.

Le contrat permet donc de clarifier la nature des obligations et, lorsque cela s'impose, d'identifier des obligations de résultat.

### 3) La définition du budget

**27.** S'il y a un mythe autour des contrats Agile, c'est bien celui du dérapage budgétaire que ces méthodes induiraient. La facturation en régie est souvent

54. Sur les limites de l'obligation de conseil, voir E. MONTERO, *Les contrats de l'informatique et de l'internet*, tiré à part du *Rép. not.*, Bruxelles, Larcier, 2005, p. 63.

55. En ce sens, voir O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 46.

56. En faveur d'une intensification du devoir de conseil, voir T. BEAUGRAND, J.-B. BELIN, « Logiciel. Le contrat de développement logiciel en méthode Agile », *Expertises*, 2013, p. 419.

57. Voir ainsi E. MONTERO, « Les obligations d'information, de renseignement, de mise en garde et de conseil des fabricants et vendeurs professionnels », in *Les obligations d'information, de renseignement, de mise en garde et de conseil*, CUP, n° 86, Bruxelles, Larcier, 2006, pp. 325-326.

58. Soulignant l'importance du devoir de conseil au regard de la méthodologie Agile, voir O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 47.

59. Trib. Com. Paris, 9 mars 2015, <https://www.legalis.net/jurisprudences/tribunal-de-commerce-de-paris-13eme-chambre-jugement-du-9-mars-2015/>.

60. Soulignant ce risque, voir S. LERICHE et E. VARET, « Contrats informatiques. Méthodologie Agile et contrats de développement. Révolution ou adaptation ? », *Expertises*, 2012/5, p. 179.

61. En ce sens, voir not. : O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48 ; S. LERICHE et E. VARET, « Contrats informatiques. Méthodologie Agile et contrats de développement. Révolution ou adaptation ? », *Expertises*, 2012/5, p. 178.

62. O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48.

63. P. WÉRY, *Droit des obligations, Volume 1 – Théorie générale du contrat*, Bruxelles, Larcier, 2010, n° 549.

associée à la méthode Agile et il n'en faut pas plus pour y voir un risque fatal pour les finances des entreprises clientes.

Il convient toutefois de fortement relativiser ce risque. En effet, la pratique des méthodes Agile apporte des réponses permettant si pas de l'évacuer, de le contrôler efficacement.

**28.** Un premier moyen contractuel de sécuriser le contrôle budgétaire sur un projet Agile consiste à fixer une enveloppe globale maximale pour le projet. Souvent, cela se fait par la détermination d'un nombre de sprints, dont le coût est déterminé par avance. Davantage qu'un contrat en régie, l'on se trouve alors en présence d'une succession de phases de développement au forfait, ou d'une forme de régie forfaitisée<sup>64</sup>. La variabilité propre à la méthode Agile se traduit alors dans l'étendue des développements réalisés, la quantité d'éléments du *Product Backlog* dépendant de la vitesse de l'équipe de développement et du bon déroulement de la collaboration. Tout risque de dérive budgétaire n'est pas pour autant évité. En effet, si l'équipe de développement n'est pas suffisamment rapide, l'estimation des ressources nécessaires pour la réalisation du travail estimé initialement devrait être actualisée, avec la probable nécessité d'ajouter un certain nombre de sprints à la fin du contrat initial. A titre de comparaison, un tel cas de figure n'est pas moins fréquent en pilotage de projet *waterfall*, car les changements nécessiteront un avenant au contrat, avec négociation de suppléments à la clé.

**29.** Un deuxième outil de contrôle budgétaire fréquemment utilisé dans les projets Agile consiste à lier le budget initialement convenu à l'achèvement de certaines réalisations minimales au projet<sup>65</sup>. L'idée est ici de définir dans le *Product Backlog* initial des composants qui, dans tous les cas, devront être fournis à la fin de l'exécution du contrat. De la sorte, la contrepartie de l'enveloppe budgétaire sera au moins la livraison de ces composants. Cela permet au client de sécuriser la livraison d'une solution qui remplisse un catalogue minimal sous le seuil duquel il estime que le contrat n'aurait pas été correctement exécuté. Pour le surplus, la méthode Agile et sa variabilité d'objet permettra d'ajuster les éléments complémentaires livrés aux besoins tels qu'ils auront été affinés en cours de projet.

**30.** Un autre outil utilisé afin de dissiper les craintes de certaines entreprises sur le « retour sur investissement » d'un projet Agile réside dans la définition d'un plan de pénalités/bonus<sup>66</sup>. Les parties se mettent d'accord sur des objectifs minimaux et sur les conséquences liées à toute déviation par rapport à ces objectifs. Si l'équipe de développement est performante, elle pourra bénéficier de bonus, si elle ne répond pas aux attentes, des pénalités seront appliquées. Dans ce cas, il convient de clarifier les liens entre la sanction de la pénalité et les autres sanctions possibles en cas d'inexécution du contrat par le prestataire<sup>67</sup>. Si la pénalité est stipulée comme seule sanction, elle devient en effet une clause de limitation de responsabilité déguisée<sup>68</sup>, ce qui n'est pas nécessairement avantageux pour le client. Il convient donc d'être prudent dans la rédaction de ces clauses du contrat.

**31.** Enfin, bien qu'il ne s'agisse pas à proprement parler d'une clause financière, un outil de protection efficace contre les risques de dérive budgétaire peut être une faculté de résiliation unilatérale. En effet, si le client dispose de facultés spécifiques de résiliation, liées à la survenance de certains événements propres à un projet Agile qui ne fonctionne pas correctement<sup>69</sup>, il peut limiter le dommage causé par une mauvaise exécution sans compromettre tout son budget. Par exemple, une faculté de résiliation sans faute liée au fait qu'un objectif déterminé à l'origine n'est pas atteint pour une date butoir convenue ou après un nombre de sprints fixé au début du projet. Dans de tels cas, le client dispose d'instruments de contrôle de la bonne exécution, et, par corollaire, de protection de son budget. Lorsqu'il active de telles clauses, le client sera en outre dispensé de démontrer une faute, ce qui évite le risque de litige face à un prestataire qui contesterait ses fautes. Afin que de telles clauses atteignent leur objectif de prévention, encore faut-il qu'elles soient liées à des événements ou constats objectifs non susceptibles de contestation<sup>70</sup>.

## C. L'exécution du contrat Agile

Le pilotage d'un projet en mode Agile implique une collaboration renforcée et continue entre les parties. Il est donc logique que les obligations tirées du droit commun soient renforcées à cet égard. Sur le plan de l'appréciation de la conformité, de l'agrément du travail fourni et de la garantie post-agrément, les contrats Agile se distinguent des contrats classiques.

64. En ce sens, voir A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 434.

65. T. HOEREN and S. PINELLI, « Agile programming – Introduction and current legal challenges », *Computer Law and Security Review*, 2018, p. 1134.

66. O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48.

67. L'éventuelle pénalité devant idéalement être définie « sans préjudice » des autres formes de sanctions.

68. En ce sens, voir B. DUBUISSON, « Les clauses limitatives ou exonératoires de responsabilité ou de garantie en droit belge », in *Les clauses applicables en cas d'inexécution des obligations contractuelles*, Bruxelles, La Chartre, 2001, p. 35, n° 5 ; E. MONTERO, « Les clauses limitatives ou exonératoires de responsabilité », in *Les sanctions de l'inexécution des obligations contractuelles*, Bruxelles-Paris, Bruylant-L.G.D.J., 2001, p. 398, n° 8.

69. A titre d'exemples, citons : refus à plusieurs reprises des livrables présentés lors de la réunion de *Sprint Review*, plusieurs sprints qui n'aboutissent pas aux résultats prévus dans le *Sprint Backlog*, désaccords persistants dans la définition des *Sprint Backlog* ou l'évaluation des ressources nécessaires pour la réalisation des éléments listés dans le *Product Backlog*, ...

70. Il est ainsi recommandé de lier la faculté de résiliation à la survenance d'une date plutôt qu'à une appréciation forcément subjective sur la qualité d'un livrable.

## 1) L'obligation de collaboration

**32.** La première exigence à rencontrer est celle du caractère contraignant de la méthode de pilotage du projet. Le recours aux méthodes Agile doit être effectif et non un simple argument de marketing. Dès lors, il convient que le contrat intègre les processus Agile dans les modalités d'exécution du projet, afin de les faire entrer dans le champ contractuel<sup>71</sup>. Cela permet de rendre le suivi du processus Agile obligatoire pour les parties.

Cette intégration peut prendre la forme d'un simple renvoi vers des documents externes<sup>72</sup>. Il est également possible d'aménager des modalités s'écartant de certains aspects des procédures *Scrum*, afin de les accommoder aux besoins particuliers du projet. Dans ce cas, il est évidemment crucial de soigneusement décrire les processus dans le contrat<sup>73</sup> et de clarifier la prévalence de ces dispositions sur les éventuels documents *Scrum* externes auxquels il serait par ailleurs renvoyé.

**33.** L'exécution du contrat Agile nécessite une implication accrue du client dans la phase de développement. En effet, le processus est rythmé par des réunions régulières entre l'équipe de développement et le client (*Product Owner*), afin de déterminer les points à réaliser lors d'un sprint, évaluer les résultats atteints à l'issue du sprint, mettre à jour le *Product Backlog*, définir les points à développer dans le sprint suivant... Afin que le projet avance correctement, il est indispensable que le client mette à disposition du projet les ressources humaines suffisantes<sup>74</sup>. L'absence de suivi efficace par le client entravera le bon avancement des travaux. Il est donc recommandé de prévoir des engagements clairs des parties en termes de disponibilité, de réactivité et de remplacement en cas d'absence.

Il va donc de soi qu'un client qui serait en défaut d'apporter sa pleine contribution aux processus Agile sera dans une position délicate pour formuler des griefs à l'encontre de la qualité du travail effectué par le prestataire. En effet, dans la mesure où le choix de ces méthodes repose sur une forte interaction entre les parties, le devoir d'implication du client constitue une obligation essentielle dans un contrat Agile.

Si elles s'apparentent parfois à des clauses de style dans les contrats informatiques classiques, les clauses qui écarteraient la responsabilité du prestataire en cas de défaut de collaboration du client sont davantage susceptibles d'être invoquées avec succès dans le contexte d'un contrat Agile. L'importance particulière que revêt la contribution du client l'impose et l'exonération ou la limitation de responsabilité qui pourrait en découler se justifie au regard de la commune intention des parties. Lorsqu'elles ont fait le choix d'inscrire leur projet dans une démarche de ce type, les parties ont souhaité conférer une importance particulière à l'aspect collaboratif du projet. Il est donc logique que cela se reflète également dans les responsabilités respectives des parties quant à la bonne exécution du contrat.

La collaboration insuffisante du client pourrait même être qualifiée de manquement fautif aux obligations découlant du contrat<sup>75</sup>. La jurisprudence consacre cette obligation du client d'une manière générale<sup>76</sup>, mais elle revêt une importance accrue dans le cadre d'un contrat Agile.

**34.** A l'inverse, le fait que le processus Agile se caractérise par une prise de décision différée sur les spécifications finales du produit à livrer devrait se traduire par un renforcement de l'obligation de conseil du prestataire en cours d'exécution du contrat. Le caractère collaboratif de l'approche accroît également le besoin de conseil du client, ce qui devrait conduire à renforcer l'obligation du prestataire à cet égard.

Comme déjà souligné, afin d'assurer la cohérence entre les obligations des parties et le mode de pilotage du projet, davantage qu'un allègement de l'obligation précontractuelle de conseil du prestataire, c'est un transfert de cette obligation vers la phase d'exécution qui nous semble devoir s'opérer<sup>77</sup>. La qualité de spécialiste des méthodes Agile renforcera d'autant l'obligation de conseil du prestataire lorsque le projet requiert des décisions cruciales pour son aboutissement<sup>78</sup>. Au contraire, l'implication insuffisante du client dans le suivi Agile du projet constituera un élément important susceptible d'atténuer l'intensité du devoir de conseil du prestataire<sup>79</sup>.

71. T. BEAUGRAND, J.-B. BELIN, « Logiciel. Le contrat de développement logiciel en méthode Agile », *Expertises*, 2013, p. 416.

72. Documents officiels de la *Scrum Alliance*, par exemple. Afin de lever toute équivoque sur les versions des documents de référence, il est recommandé de les annexer au contrat.

73. Le cas échéant dans une annexe.

74. O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 47 ; T. BEAUGRAND, J.-B. BELIN, « Logiciel. Le contrat de développement logiciel en méthode Agile », *Expertises*, 2013, p. 419.

75. D. GOBERT et E. MONTERO, « Les obligations de conformité et de garantie des vices cachés en matière informatique : le contrat au secours des incertitudes légales et jurisprudentielles », *R.D.T.I.*, 2002/11, p. 16-17.

76. Voir ainsi Cass. Fr., 2 octobre 2001, Ballaric c. SPI-AGF, *Expertises*, 2002/256, pp. 72-73.

77. Voir *supra*, n° 24.

78. E. MONTERO, « Les obligations d'information, de renseignement, de mise en garde et de conseil des fabricants et vendeurs professionnels », in *Les obligations d'information, de renseignement, de mise en garde et de conseil*, CUP, n° 86, Bruxelles, Larcier, 2006, pp. 332-333.

79. *Ibid.*, pp. 338-339.

Il découle de ce glissement et de cette intensification une double nécessité de contractualiser et de documenter la mise en œuvre du devoir de conseil dans le déroulement d'un projet Agile.

La contractualisation permet de consacrer des obligations spécifiques. Ainsi, lorsqu'un projet doit franchir certaines étapes afin de permettre son aboutissement, il est important de faire de ces étapes de véritables jalons, et que, par corollaire, le devoir de conseil découlant du principe d'exécution de bonne foi y trouve un terrain d'application particulier. Lorsque, par exemple, la mise en production d'une nouvelle solution informatique ne peut se faire qu'après vérification de sa conformité avec un cadre réglementaire spécifique, il est important d'intégrer cet élément dans les étapes d'exécution, afin qu'il ne puisse être contesté que cela est entré dans le champ contractuel et que le devoir de conseil s'y applique par conséquent.

Il est en outre nécessaire de documenter le processus Agile en cours d'exécution, afin notamment de démontrer la teneur des échanges entre les parties et ainsi d'éclairer les raisons des décisions prises. De la sorte, l'on peut établir le (non-)respect du devoir de conseil du prestataire.

**35.** Afin de se prémunir contre un possible blocage du processus Agile lié à une réactivité insuffisante du client, il est tentant de prévoir des mécanismes d'acceptation ou de validation tacite de certaines propositions de décisions en cas d'absence de réponse du client dans un certain délai. Si cela permet d'éviter temporairement un blocage, le procédé n'est pas sans risque, car l'absence de collaboration du client se traduira tôt ou tard par une remise en cause des options prises avec son accord tacite. Il nous semble donc dangereux de mettre en œuvre ce type d'outil. La collaboration étant de l'essence même de la démarche, il nous semble qu'elle peut difficilement être remplacée par des acceptations tacites. Le choix de la méthode Agile suppose que les moyens nécessaires à une approche collaborative soient mis en œuvre...

**36.** Les clauses de gouvernance permettent de clarifier les rôles respectifs des parties, organiser la prise de décision, modaliser la mise en œuvre du devoir de conseil, régler les éventuels blocages, ... Il est donc essentiel de les rédiger soigneusement, selon les attentes respectives des parties.

## 2) La notion de conformité

**37.** Classiquement, la conformité s'apprécie par rapport aux qualités et spécifications attendues de la prestation, telles que définies et entrées dans le champ contractuel<sup>80</sup>. Afin de bien délimiter les obligations contractuelles, il est de pratique courante de décrire ces spécifications dans un cahier de charges, qui constituera le document de référence tout au long de l'exécution du contrat<sup>81</sup>. C'est donc par rapport au cahier des charges que la conformité des prestations fournies et/ou des livrables fournis<sup>82</sup> sera évaluée.

Le critère de référence pour apprécier la conformité des livrables est moins évident à identifier dans un contrat Agile. A défaut de disposition particulière dans le contrat, l'on pourrait considérer que le prestataire a rempli son obligation de délivrance lorsqu'il a fourni des livrables répondant au nombre de points correspondant aux éléments du *Product Backlog* qui ont été inclus dans les différents sprints du projet<sup>83</sup>. Une telle solution présente toutefois des dangers pour le client, en raison du caractère vague du niveau d'exigences à atteindre. Il est dès lors vivement conseillé de clarifier les critères de conformité, afin que la procédure d'agrément soit mieux balisée.

**38.** Il convient d'identifier la source de référence pour l'appréciation de la conformité. Compte tenu de la variabilité des spécifications contractuelles durant l'exécution du contrat, si le *Product Backlog* constitue l'outil naturel de vérification, c'est sa version au jour du démarrage d'un sprint qui devrait en principe être utilisée pour vérifier la conformité des éléments développés au cours dudit sprint.

Lorsque le projet nécessite d'intégrer des exigences particulières, il convient de veiller à ce qu'elles soient correctement traduites dans les spécifications du *Product Backlog*. Une alternative possible consiste à définir des étapes particulières à franchir lors de l'exécution du projet<sup>84</sup>. Dans ce cas, la conformité sera évaluée au regard du respect du *Product Backlog* et de la réalisation des étapes concernées.

## 3) L'agrément du travail fourni

**39.** L'agrément consiste pour le client à vérifier la qualité des livrables, et donc à approuver le travail accompli par le prestataire<sup>85</sup>. Pour ce faire, le client contrôle que les livrables correspondent aux

80. Y. POULLET et al., *Droit de l'informatique et des technologies de l'information. Chronique de jurisprudence (1995-2001)*, Les dossiers du Journal des Tribunaux, n° 41, Bruxelles, Larcier, 2003, pp. 24-25, n° 13.

81. D. GOBERT et E. MONTERO, « Les obligations de conformité et de garantie des vices cachés en matière informatique : le contrat au secours des incertitudes légales et jurisprudentielles », *R.D.T.I.*, 2002/11, p. 12.

82. M. VIVANT (dir.), *Le Lamy - Droit du numérique*, Paris, Kluwer, 2020, n° 1588.

83. En ce sens, voir O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48 ; A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 433. Voir aussi M. VIVANT (dir.), *Le Lamy - Droit du numérique*, Paris, Kluwer, 2020, n° 1473.

84. Mise en service dans un environnement particulier, combinaison réussie avec d'autres composants (voire des éléments externes aux développements du prestataire).

85. B. KOHL, « Le contrat d'entreprise », *R.P.D.B.*, Bruxelles, Bruylant, 2016, pp. 404-405.

spécifications du *Product Backlog*. En fonction de la nature des livrables et de la définition des spécifications, des modalités de contrôle particulières pourraient être nécessaires<sup>86</sup>. Il est alors essentiel que ces modalités soient clairement stipulées dans le contrat Agile, afin d'éviter toute discussion<sup>87</sup>. La clarté à ce sujet permet aussi au prestataire de mieux anticiper la nature du contrôle et, dès lors, d'organiser son propre contrôle interne de qualité en amont de la phase d'agrégation. Cela permet naturellement de réduire les incidents d'agrégation. Les rôles respectifs dans la procédure d'agrégation doivent être bien définis<sup>88</sup>.

Parmi les éléments clés de la procédure d'agrégation, l'on insistera sur les critères d'acceptation du travail. Il s'agit des critères qui permettent de considérer qu'un livrable a été fourni avec une qualité suffisamment proche de celle attendue pour que le travail soit accepté, le cas échéant sous réserve de la correction des défauts mineurs qui auraient été identifiés. Dans la procédure Agile, il s'agit de la « *definition of done* »<sup>89</sup>, qui peut varier en cours de projet. Une définition précise permet de sécuriser la procédure d'agrégation et d'éviter les difficultés de mise en œuvre<sup>90</sup>.

**40.** Dans le cadre d'un projet Agile, le découpage du processus en sprints pose la question de la portée que l'on doit conférer à la validation des livrables fournis lors d'un sprint.

Il convient de distinguer l'agrégation de la solution livrée en fin de processus de l'agrégation individuelle de chaque sprint.

Dans l'esprit propre à la démarche Agile, les résultats partiels atteints et validés ne doivent pas être perçus comme définitivement figés. Tout doit rester flexible et ouvert à la modification. Il est conseillé de clairement distinguer dans le contrat Agile l'agrégation des résultats de chaque sprint de l'agrégation du livrable final consolidé. En effet, si séparément les pièces du puzzle peuvent répondre aux attentes, encore faut-il qu'elles s'assemblent et interagissent correctement

entre elles. Il y a donc lieu de distinguer l'agrégation globale des agrégations de chaque sprint<sup>91</sup>. L'agrégation finale devrait idéalement permettre de revenir sur les agrégations partielles, dans la mesure où l'exigence d'interopérabilité entre les éléments de la solution constitue une condition nécessaire sans laquelle les composants ne sont d'aucune utilité pour le client<sup>92</sup>. Le champ large de l'agrégation finale s'impose également afin de prémunir le client contre tout risque de régression fonctionnelle<sup>93</sup>. Il est donc important de lever toute incertitude sur la portée de chacune des agrégations par des clauses contractuelles appropriées.

#### 4) La garantie après agrégation

**41.** Une fois le travail agréé par le client, il subsiste la garantie sur les défauts cachés et la garantie contractuelle éventuelle.

**42.** La responsabilité pour défauts cachés<sup>94</sup> suppose que le livrable final soit affecté d'un défaut qui en altère le caractère opérationnel et qui soit imputable à une faute du prestataire. La comparaison sera effectuée par rapport aux spécifications finales du *Product Backlog*. Sauf disposition contractuelle spécifique, les sanctions sont celles du droit commun de l'inexécution contractuelle<sup>95</sup>. Si les parties souhaitent appliquer un régime spécifique de prise en charge des défauts cachés, il convient de l'aménager contractuellement.

L'exigence d'un caractère caché implique que tout défaut qui aurait été décelable lors de l'agrégation est couvert par celle-ci, et, par conséquent, exclu de la responsabilité pour défauts cachés<sup>96</sup>.

**43.** Comme dans tout projet de développement, la portée de la garantie contractuelle doit être définie dans le contrat Agile. L'on soulignera toutefois que l'obligation du prestataire devrait logiquement être de rétablir les fonctionnalités du produit livré conformément à la définition donnée dans la version finale du *Product Backlog*. Le contrat devrait donc énoncer

86. Des modalités de test spécifiques, l'usage d'un environnement de test avec une configuration bien précise, etc.

87. Voir Trib. Com. Paris, 7 oct. 2020, <https://www.legalis.net/jurisprudences/tribunal-de-commerce-de-paris-8eme-ch-jugement-du-7-octobre-2020/> (rejetant le grief d'absence de test avant agrégation dans la mesure où la procédure de test n'était pas expressément stipulée dans le contrat). Sur les modalités d'agrégation, voir not. J.-P. TRIAILLE et R. ROBERT, « Les contrats informatiques », in *Traité pratique de droit commercial, Tome I – Principes et contrats fondamentaux*, 2<sup>e</sup> éd., Bruxelles, Kluwer, 2009, pp. 1097-1099 ; D. GOBERT et E. MONTERO, « Les obligations de conformité et de garantie des vices cachés en matière informatique : le contrat au secours des incertitudes légales et jurisprudentielles », *R.D.T.I.*, 2002/11, pp. 24-29.

88. On soulignera spécialement la question de l'élaboration et de la validation des scénarios de test qui sont généralement exécutés lors de la phase d'agrégation.

89. Scrum Alliance, *Core Scrum*, op. cit., p. 11.

90. Sur les risques liés à une définition imprécise des attentes, voir M. VIVANT (dir.), *Le Lamy – Droit du numérique*, Paris, Kluwer, 2020, n° 1483. Voir aussi B. DOCQUIR, « Droit du numérique », *R.P.D.B.*, Bruxelles, Bruylant, 2018, n° 111.

91. En ce sens, voir T. HOEREN and S. PINELLI, « Agile programming – Introduction and current legal challenges », *Computer Law and Security Review*, 2018, p. 1135.

92. *Contra*, voir O. DORCHIES, « Pratique contractuelle. Les méthodes agiles dans les contrats informatiques », *Communication Commerce Electronique*, 2020/10, p. 48.

93. Si un composant est validé lors d'un sprint, puis que lors d'un sprint ultérieur ce même composant se voit modifié pour y adjoindre une nouvelle fonctionnalité, il y a un risque que la nouvelle fonctionnalité développée altère le fonctionnement de la première qui avait été précédemment validée. Une batterie complète de test de l'intégralité des fonctionnalités, en fin de processus Agile permet de constater l'absence de régression fonctionnelle sur les éléments ayant été validés au préalable.

94. Il ne s'agit à proprement parler pas d'une garantie, puisqu'elle est, selon la jurisprudence, conditionnée à l'établissement d'une faute du prestataire. En ce sens, voir B. KOHL, « Le contrat d'entreprise », *R.P.D.B.*, Bruxelles, Bruylant, 2016, n° 217.

95. A. CRUQUENAIRE, C. DELFORGE, I. DURANT et P. WÉRY, *Précis des contrats spéciaux*, Bruxelles, Kluwer, 2015, pp. 607-615.

96. Voir not. Trib. Com. Paris, 7 oct. 2020, <https://www.legalis.net/jurisprudences/tribunal-de-commerce-de-paris-8eme-ch-jugement-du-7-octobre-2020/>.

le type de défaut pris en charge, la durée de la garantie et le niveau de service offert dans le cadre de la garantie<sup>97</sup>.

## D. La fin du contrat

**44.** À l'instar de tout projet informatique, la sortie du contrat Agile expose les parties à certains risques. En outre, le pilotage du projet selon ces méthodes nécessite certaines adaptations.

### 1) Les risques classiques liés à la sortie d'un contrat

**45.** Lorsqu'un projet informatique prend fin, différentes questions peuvent surgir et, dans certains cas, être source de risque, voire de litige entre les contractants.

La titularité des droits de propriété intellectuelle sur les livrables constitue un point clé, car il va déterminer la marge de manœuvre des parties pour la suite de leurs projets. L'accès au code source est une autre question sensible, étroitement liée à la première. Enfin, d'une manière plus large, les modalités de transition lors de la sortie du contrat peuvent avoir une incidence pratique importante sur la capacité du client à poursuivre efficacement son projet avec un autre partenaire.

**46.** La titularité des droits d'auteur sur les livrables est le plus souvent réglée contractuellement. Si tel est le cas, encore faut-il que les clauses de propriété intellectuelle envisagent le scénario de fin anticipée du contrat. Si tel n'est pas le cas, la titularité des droits d'auteur sur les livrables déjà reçus et sur ceux en cours de développement dépendra des modalités de fin de contrat. En effet, en principe et sauf disposition contractuelle contraire, la résolution et les conditions résolutoires opèrent avec effet rétroactif, ce qui ne devrait pas permettre au client de conserver la jouissance des réalisations déjà livrées (et encore moins de celles en cours d'élaboration)<sup>98</sup>. Par contre, en cas de fin de contrat suite à une résiliation unilatérale, l'absence d'effet rétroactif permet à tout le moins de préserver les droits acquis sur les livrables déjà reçus<sup>99</sup>. Afin de sécuriser la position du client, il est donc vivement conseillé d'organiser la dévolution des droits intellectuels lorsque le contrat prend fin d'une manière anticipée.

**47.** Régler la titularité des droits d'auteur, y compris en cas de fin anticipée du contrat, n'est pas suffisant. Il convient également d'organiser les moyens de leur exercice. A cet égard, une question particulièrement cruciale est celle de l'accès aux codes sources ou matrices de livrables. En effet, afin de pouvoir exploiter les droits de propriété intellectuelle, et en particulier la faculté de faire évoluer les livrables, il convient d'avoir accès à la documentation et à leurs codes sources.

Dans la pratique, les conditions d'accès varient en fonction de la nature des développements (sur mesure ou basés sur des modules standards)<sup>100</sup>. La question est importante, car elle conditionne la maîtrise du client sur les livrables et leur exploitation ultérieure. Il convient donc d'assurer une symétrie entre les clauses de dévolution des droits de propriété intellectuelle et celles réglant l'accès aux codes sources. Lorsque les livrables sont basés sur des composants standards, l'accès au code source se fait le plus souvent au travers d'un contrat d'*escrow*, les conditions permettant la libération des sources au profit du client étant plus ou moins restrictives selon les circonstances<sup>101</sup>. Dans ce contexte, il convient de ne pas perdre de vue que le code source seul n'est généralement pas exploitable par le client s'il n'est pas accompagné d'une documentation appropriée. Il faut en effet permettre à un développeur tiers de comprendre la logique de programmation, ce qui sera nettement plus compliqué sans une documentation de qualité. Les clauses de documentation ne doivent donc pas être négligées, afin notamment de préciser des standards de qualité, des normes de présentation, qui assureront un transfert de connaissances plus aisé lors de la fin du contrat (ou simplement de la livraison finale du travail). Dans le cadre d'un contrat Agile, cette exigence de documentation est sans doute encore plus cruciale compte tenu du fait que l'activité de développement déployée dans l'exécution du contrat ne repose pas sur un catalogue de spécifications clairement défini en début de contrat, mais repose sur un *Product Backlog* évolutif. Le contrat Agile devrait donc contenir des exigences de documentation du *Product Backlog* et des réflexions relatives à son évolution, afin qu'un développeur tiers puisse comprendre la logique de développement propre au projet et à ses différents stades de progression.

97. Pour des facilités de gestion, lorsqu'un contrat de maintenance et support est conclu par ailleurs, la garantie contractuelle liée aux prestations de développement s'inscrit dans le niveau de services (SLA) du contrat de maintenance et support, avec une réduction de prix applicable pendant la période de garantie contractuelle. À propos des contours de la garantie conventionnelle sur les développements livrés, voir B. DOCQUIR, « Droit du numérique », *R.P.D.B.*, Bruxelles, Bruylant, 2018, pp. 115-119 ; D. GOBERT et E. MONTERO, « Les obligations de conformité et de garantie des vices cachés en matière informatique : le contrat au secours des incertitudes légales et jurisprudentielles », *R.D.T.I.*, 2002/11, pp. 29-32.

98. Sur les hypothèses permettant de s'écarter de ces principes, voir A. CRUQUENAIRE, « Le sort des droits d'auteur en cas de fin de contrat », in *Les obligations contractuelles en pratique. Questions choisies*, Limal, Anthémis, 2013, pp. 170-171 (résolution) et p. 176 (condition résolutoire).

99. *Ibid.*, pp. 173-174.

100. B. DOCQUIR, « Droit du numérique », *R.P.D.B.*, Bruxelles, Bruylant, 2018, pp. 152-155 ; J.-P. TRIAILLE et R. ROBERT, « Les contrats informatiques », in *Traité pratique de droit commercial, Tome I – Principes et contrats fondamentaux*, 2<sup>e</sup> éd., Bruxelles, Kluwer, 2009, n° 1445.

101. La position économique des parties détermine le plus souvent l'amplitude du droit d'accès aux sources, les grands éditeurs de logiciels écartant naturellement toute possibilité, tandis que les plus petits prestataires se montrent généralement plus ouverts à la négociation.

**48.** Dans tout projet d'une certaine ampleur ou durée, il convient d'envisager les modalités de sortie, par l'organisation d'une procédure de transition<sup>102</sup>. L'objectif est principalement de déterminer les rôles respectifs des parties dans ce processus de sortie et de préciser les attentes du client quant à la collaboration du prestataire. Dans le contexte d'un projet Agile, ces exigences devront intégrer la documentation propre aux processus Agile, afin de permettre la poursuite du projet si le client le souhaite.

## 2) Quelques modalités de sortie propres aux contrats Agile

**49.** L'exécution d'un projet selon la méthode Agile nécessite une dynamique collaborative dont l'importance est parfois sous-estimée par le client. Afin de gérer ce risque spécifique aux clients inexpérimentés en la matière, il est important d'aménager des portes de sortie préservant les intérêts des deux parties.

**50.** Lorsque le contrat Agile démarre sans *Product Backlog* et que la confection de celui-ci constitue l'objet du *kick-off sprint*, il convient de prévoir une faculté de résiliation, ouverte aux deux parties, à l'issue de ce sprint initial. En raison de son objet particulier, ce sprint constitue en effet un bon test quant à l'aptitude des deux parties à adopter une approche collaborative de travail. Si des difficultés surgissent dès ce sprint initial, soit les parties décident de corriger ce qui doit l'être, soit l'une d'elles devrait pouvoir décider sans contrainte de se libérer du contrat dans le moule duquel elle ne se sent pas à l'aise.

Lorsqu'une telle option de résiliation est offerte aux parties, il est important de régler la question des droits de propriété intellectuelle sur le *Product Backlog*, afin que le client puisse l'exploiter avec un autre prestataire ou le réutiliser dans le cadre de l'élaboration d'un cahier de charges permettant de poursuivre le projet selon une méthodologie de travail plus classique.

**51.** La dynamique collaborative est nécessaire à la mise en œuvre de l'ensemble des processus Agile. Des divergences persistantes entre les parties peuvent dès lors compromettre la bonne exécution du contrat. Il est donc utile de prévoir des facultés de résiliation liées à certains événements propres au déroulement des processus Agile, et de fixer contractuellement les conséquences qui en découlent<sup>103</sup>.

En fonction de la nature du projet, les parties pourraient, par exemple, attacher une faculté de résiliation à des désaccords persistants sur la mise à jour du *Product Backlog*<sup>104</sup>, au rejet de livrables lors de plusieurs sprints, à la non-réalisation du *Sprint Backlog* à plusieurs reprises.

**52.** D'une manière plus large, la flexibilité induite par les méthodes Agile peut être balisée par la définition d'étapes clés (*milestones*) dans l'exécution du projet. Lorsque l'une de ces étapes n'est pas réalisée pour la date-butoir prévue, une faculté de résiliation spécifique peut être ouverte au bénéficiaire du client. Cela permet au client de se libérer du contrat s'il n'est effectivement pas satisfait, ou à tout le moins de mettre la pression sur le prestataire. Poser de telles balises sur le tracé du projet permet en outre d'indiquer les éléments clés du projet selon l'intention des parties, ce qui peut toujours être utile pour l'interprétation de la portée des engagements.

**53.** Dans la mesure où la jurisprudence expose le prestataire au risque de qualification du contrat Agile en contrat « à la pièce »<sup>105</sup>, il convient de définir les modalités d'exercice de la résiliation unilatérale du contrat de services. A défaut de disposition particulière, rappelons que le principe est que le client peut résilier à tout moment, sans préavis, mais avec une indemnité pour le travail accompli, le bénéfice manqué et les dépenses exposées par le prestataire<sup>106</sup>. Dans le contexte particulier des contrats Agile, il est donc prudent de régler contractuellement les modalités de résiliation<sup>107</sup>, voire d'écarter la simple résiliation sans motif pour y préférer des facultés de résiliation calquées sur la mise en œuvre des processus Agile.

## III. Conclusion

**54.** Ce n'est pas par hasard que les méthodes Agile voient leur domaine d'utilisation s'élargir. Ces méthodes permettent de contourner les difficultés et frustrations liées à certaines rigidités des méthodes classiques de pilotage de projets. Elles permettent aussi de mieux exploiter l'expertise technique du prestataire en la confrontant d'une manière beaucoup plus systématique et encadrée à l'expertise métier du client. Généralement, il en ressort un résultat plus en adéquation avec les besoins réels du client, grâce à la complémentarité des expertises respectives.

102. J.-P. TRIAILLE et R. ROBERT, « Les contrats informatiques », in *Traité pratique de droit commercial, Tome I – Principes et contrats fondamentaux*, 2<sup>e</sup> éd., Bruxelles, Kluwer, 2009, pp. 1125-1126.

103. En faveur de cette approche, voir S. LERICHE et E. VARET, « Contrats informatiques. Méthodologie Agile et contrats de développement. Révolution ou adaptation ? », *Expertises*, 2012/5, pp. 178-179.

104. Évaluation des points de difficulté pour certaines tâches, par exemple. Cette estimation est une question délicate, car elle détermine l'ampleur du travail fourni lors de chaque sprint.

105. C'est-à-dire limitant l'engagement du client à un seul sprint, l'engagement n'étant renouvelé qu'en cas d'accord des parties sur l'objet du sprint suivant. En ce sens, voir Trib. Com. Paris, 9 mars 2015, <https://www.legalis.net/jurisprudences/tribunal-de-commerce-de-paris-13eme-chambre-jugement-du-9-mars-2015/> (hypothesis dans laquelle les parties n'avaient toutefois pas conclu de contrat écrit).

106. A. CRUQUENAIRE, C. DELFORGE, I. DURANT et P. WÉRY, *Précis des contrats spéciaux*, Bruxelles, Kluwer, 2015, pp. 621-625.

107. En ce sens, voir not. A. LEFÈVRE, « Contrat informatique. Le recours à la méthode Agile à l'épreuve des tribunaux », *Expertises*, 2015, p. 435.

**55.** Il convient toutefois de ne pas verser dans l'angélisme.

Tout d'abord, la dynamique collaborative propre aux méthodes Agile interroge certains principes du droit des contrats. En outre, les méthodes Agile ne constituent pas une solution miracle et comportent, elles aussi, leurs inconvénients.

Afin de se prémunir contre de possibles interprétations jurisprudentielles défavorables, il est donc nécessaire de s'appuyer sur un contrat solide. Cette combinaison de l'outil contractuel et des méthodes Agile permet de mieux gérer les risques. Pour ce faire, la rédaction du contrat Agile doit s'inspirer des besoins et enjeux du projet concerné, et donc reposer sur un travail d'équipe entre le juriste et le gestionnaire de projet. Une approche collaborative du travail d'élaboration du cadre contractuel, donc.