

Article publié dans la revue **Expertises des Systèmes d'Information** d'octobre 2012

Logiciel : Les conditions d'un accès réussi au code source

Florence Ivanier, Avocat à la Cour

Les utilisateurs font face à une dépendance technologique croissante, notamment vis-à-vis des composants logiciels des solutions informatiques complexes auxquelles ils doivent désormais avoir recours pour nombre de leurs activités.

Le logiciel est devenu un enjeu critique et stratégique. Critique, car un bogue bloquant et non résolu est susceptible d'immobiliser l'activité de la société. Stratégique, car le logiciel qui constitue l'un des principaux postes de dépense informatique de l'entreprise, représente un investissement lourd dont l'investisseur attend une pérennité qui garantisse sa rentabilité. Or l'un des aspects essentiels de la pérennité d'un programme informatique réside dans la possibilité d'assurer sa maintenance corrective, évolutive et règlementaire. A cette fin, l'utilisateur se doit d'obtenir un accès au code-source du logiciel.

Rappelons que le code-source d'un programme informatique est constitué d'une liste d'instructions écrite dans un langage de programmation évolué, c'est-à-dire intelligible par un professionnel de l'informatique, tandis que le code-objet, ou version exécutable du programme, représente la traduction dans un langage binaire compris par la machine de cette liste d'instructions, après compilation du code-source.

Le code-source constitue l'une des clés de l'indépendance de l'utilisateur. Toutefois, on le verra, pour que l'utilisateur puisse s'estimer autonome vis-à-vis de son fournisseur, sans avoir à se préoccuper de la taille, de l'ancienneté et plus largement de la pérennité économique de celui-ci, l'accès aux sources en tant que telles est loin d'être suffisant. A cet égard, nous nous attacherons ici à retracer les processus techniques de création et de modification des logiciels pour permettre au juriste de mieux appréhender le concept de l'autonomie de l'utilisateur et ainsi, le mettre pleinement en œuvre sur le plan juridique. L'étude de la pratique et le témoignage des hommes de l'art informatique sont d'autant plus précieux que les jurisprudences en matière de codes sources sont rares et que le paysage de la programmation informatique a fait l'objet d'un bouleversement ces dix à vingt dernières années.

Dans la présente étude, seul le cas du progiciel et non celui du logiciel spécifique sera abordé. En effet, lorsqu'un client passe commande d'un logiciel spécifique qui sera développé pour ses besoins propres, il est d'usage que le fournisseur opère à son profit une cession des droits de propriété intellectuelle patrimoniaux accompagnés d'un accès aux sources. Tel n'est pas le cas pour l'utilisateur d'un progiciel, dont les droits sont en général limités à un droit d'utilisation sur le code-objet.

S'agissant du progiciel, pour schématiser, l'on peut distinguer entre le progiciel standard et le progiciel complexe qui, lui nécessite une intégration et un déploiement chez le client.

Concernant le progiciel standard, pour l'utilisation duquel le client adhère à un contrat pré-établi, il n'est pas question pour l'utilisateur d'aménager contractuellement un accès aux

sources. Tout au plus peut-il espérer bénéficier d'un programme de séquestre commun à l'ensemble des utilisateurs, si celui-ci figure au contrat d'adhésion. Nous nous intéresserons donc plutôt au progiciel complexe, pour lequel on peut séparer la partie « socle », sur laquelle l'utilisateur dispose d'un droit d'usage limité à sa version exécutable et, d'autre part, la partie « développements spécifiques », sur laquelle l'utilisateur bénéficie en général de droits plus étendus, y compris des droits de modification et de correction avec transmission des codes-sources qui y sont associés. L'éditeur du progiciel aura à cœur de protéger son savoir-faire et sa propriété intellectuelle qui résident en grande partie dans sa partie « socle ». C'est donc sur le code-source du socle que se cristallisera la négociation.

L'aménagement d'un droit d'accès aux sources par le biais d'un dépôt auprès d'un tiers séquestre constitue la meilleure manière pour l'utilisateur de se prémunir efficacement contre la défaillance du fournisseur (I). L'efficacité d'un tel aménagement sera d'autant plus grande que l'on aura pris soin de définir précisément les cas d'accès aux sources (II) ainsi que les éléments devant impérativement figurer dans le dépôt (III).

I. L'aménagement de l'accès aux sources par le biais d'un dépôt auprès d'un tiers séquestre

A/ Il n'existe pas de droit implicite de l'utilisateur sur le code-source.

On le sait, le législateur n'organise pas l'accès au code-source. Certes, l'article L 122-6-1 IV du Code de la propriété intellectuelle dispense l'utilisateur de recueillir l'autorisation de l'auteur du logiciel pour le décompiler, afin d'obtenir des informations nécessaires à l'interopérabilité avec d'autres logiciels.

Toutefois, la décompilation – ou ingénierie inverse – est une opération complexe ne permettant de générer le code-source à partir du code-objet qu'à condition d'avoir accès à un grand nombre d'informations dont l'utilisateur n'est pas en possession, telles que, par exemple, le compilateur, à savoir le logiciel de compilation, utilisé. Au demeurant, cette autorisation n'est prévue qu'aux seules fins d'interopérabilité et sous réserve que certaines conditions soient remplies.

Rappelons que l'interopérabilité est définie, aux termes de la directive 2009/24/CE du 23 avril 2009, comme « la capacité d'échanger des informations et d'utiliser mutuellement les informations échangées ». Même si, de manière contestée, la Cour de Cassation a récemment élargi la notion d'interopérabilité et donné une lecture extensive de l'exception de décompilation, en considérant que « les opérations de migrations de données (...) pour récupérer les fichiers du logiciel, s'inscrivent dans les strictes nécessités de l'interopérabilité », la correction d'erreurs ne correspond nullement aux fins d'interopérabilité, qu'il s'agisse d'échange d'informations ou d'opérations de migration de données. Les dispositions de l'article L 122-6-1 IV ne servent donc en rien les intérêts de l'utilisateur s'agissant d'assurer la maintenance.

De même, l'article L 122-6-1 du même Code prévoit-il en substance qu'en l'absence de dispositions conventionnelles contraires, les actes de reproduction et de modification du logiciel ne sont pas soumis à l'autorisation de l'auteur pour corriger des erreurs. Mais en pratique, ces dispositions ne prévoient aucune modalité afin de se procurer le code-source qui est nécessaire pour ce faire.

Enfin, la jurisprudence n'est d'aucun secours à l'utilisateur puisqu'en l'absence de disposition conventionnelle, elle tend à refuser l'accès au code-source à l'utilisateur, en particulier

lorsque celui-ci ne dispose que d'un droit d'usage, sans faculté de modifier ou de corriger le progiciel .

L'utilisateur devra donc se ménager l'accès au code-source par le biais d'une clause figurant au contrat lui concédant la licence du progiciel.

B/ Les différents types d'aménagement contractuel de l'accès aux sources

Certains éditeurs de progiciels, soucieux par exemple de se soustraire aux contraintes liées à la mise en place d'un contrat de séquestre, qu'elles soient d'ordre financier, technique ou juridique, proposent à leurs clients de s'engager contractuellement à leur remettre les codes-sources, le moment venu, pour pallier, le cas échéant, leur propre défaillance.

Les limites de ce type d'aménagement contractuel sont manifestes. En particulier, aucun contrôle n'étant exercé par un tiers sur le fait que l'éditeur prend effectivement soin de conserver un exemplaire à jour et exploitable des sources, qu'advient-il en cas de défaillance de ce dernier? Et dans l'hypothèse où celui-ci fait l'objet d'une liquidation judiciaire, il restera à l'utilisateur à faire valoir auprès du mandataire judiciaire son droit d'accès aux sources en vertu de ladite clause. Or , même si le tiers séquestre est également tenu d'obtenir l'autorisation du mandataire judiciaire, il est plus délicat pour l'utilisateur de faire valoir ses droits auprès d'un tiers séquestre que d'un mandataire car, le tiers séquestre veille avant tout à la bonne application du contrat d'entiercement, tandis que le mandataire judiciaire cherche avant tout à préserver le patrimoine intellectuel du débiteur et se montrera toujours réticent à dévoiler le code-source, constitutif de l'un des composants majeurs du patrimoine.

La meilleure garantie sera donc la mise en place d'un contrat d'entiercement, par lequel le fournisseur dépose les sources auprès d'un tiers chargé de veiller à leur conservation et de les remettre à l'utilisateur bénéficiaire de ladite convention, dans certains cas définis précisément et qui caractérisent la défaillance du fournisseur.

En pratique, deux types de montages contractuels sont possibles, soit la conclusion d'un contrat d'entiercement tripartite, soit l'insertion dans le contrat de licence d'une clause d'accès au code-source portant stipulation d'un dépôt du code-source par l'éditeur du progiciel et la conclusion en parallèle dudit contrat de dépôt entre l'éditeur et le tiers séquestre, au bénéfice de l'utilisateur.

Les parties sont libres de choisir comme séquestre tout tiers de confiance. Toutefois, l'entiercement n'aura d'intérêt que si le séquestre est équipé de manière à conserver le code-source de manière sécurisée et s'il dispose des compétences nécessaires pour contrôler la conformité du code-source qui lui est remis.

Les cas où un utilisateur exerce son droit d'accès pour se retrouver face à un code-source inutilisable, voire à une enveloppe vide, ne sont pas des cas d'école. C'est pourquoi les prestataires spécialisés proposent, au-delà du simple dépôt, des prestations plus complètes comprenant :

- la vérification des éléments du dépôt initial, notamment la vérification de la correspondance entre le code-source déposé et le code-objet utilisé par l'utilisateur, ainsi que de la complétude du code-source ;
- puis la vérification, au moins une fois par an, du dépôt d'une mise à jour ou d'une nouvelle version du code-source et de ses éléments associés.

L'entiercement sera d'autant plus efficace que seront encadrées contractuellement les conditions dans lesquelles l'utilisateur pourra exercer son droit d'accès.

II. La nécessité de définir précisément le(s) cas d'accès, à savoir ce que l'on entend par défaillance

Il faut garder à l'esprit que l'entiercement ne constitue pas un mécanisme de garantie à première demande. Le séquestre vérifie donc la réalisation des conditions prévues au contrat de dépôt avant de délivrer une copie des éléments du dépôt à l'utilisateur.

L'utilisateur veillera à ce que les cas de défaillance prévus ouvrant droit à l'accès incluent en particulier:

- i) la survenance d'une procédure collective dont l'éditeur du progiciel ferait l'objet. Il s'agit là de la principale menace pesant sur l'utilisateur acquéreur d'un progiciel. Toutefois, l'ouverture d'une procédure collective prononcée contre le fournisseur ne constitue pas une condition suffisante pour accéder au code-source. Celle-ci ne constitue un cas d'ouverture que si elle se traduit par un arrêt effectif de la maintenance. Il convient de distinguer entre les cas de redressement et de liquidation judiciaires. En cas de redressement judiciaire, seul le cas où l'administrateur décide de ne pas poursuivre les contrats de maintenance en cours ouvre la possibilité pour l'utilisateur de requérir du séquestre l'accès au code-source et même dans ce cas, ainsi que cela a précédemment été indiqué, le séquestre est tenu au préalable d'obtenir l'autorisation de l'administrateur judiciaire. En cas de liquidation judiciaire, seul le cas où il n'y a pas de reprise des engagements par un tiers repreneur constitue un cas d'ouverture. Il sera bon de préciser dans quel délai, à compter du jugement d'ouverture de la procédure collective l'accès aux sources est acquis à l'utilisateur s'il n'y a pas de reprise des engagements par un tiers, afin d'éviter que le client ne soit contraint de rester dans l'expectative durant un temps déraisonnable. Il est à noter qu'un contrat prévoyant un accès aux sources pourrait théoriquement être annulé s'il a été passé durant la période suspecte précédant le jugement d'ouverture de la procédure collective.
- ii) l'arrêt décidé unilatéralement par le prestataire de poursuivre la maintenance du progiciel, en dehors de toute procédure collective. Toutefois, lors de la négociation portant sur les cas d'ouverture, le fournisseur pourra légitimement faire valoir son droit de cesser de maintenir son progiciel, au-delà d'une certaine durée, par exemple de quelques années, afin de se ménager la possibilité ne plus le commercialiser et de le remplacer par un autre progiciel plus évolué.
- iii) la défaillance entendue comme la non-résolution d'une anomalie bloquante ou encore la non-évolution pour répondre à une contrainte législative impérative. Il convient dans ce cas de spécifier après quel délai – en nombre de jours ou tout au plus de semaines- l'utilisateur sera en droit de bénéficier de l'accès aux sources. De même que l'on aura pris de définir précisément au contrat les cas d'accès aux sources, il convient de spécifier contractuellement quels sont les livrables attendus en cas d'accès.

III. Les éléments devant figurer dans le contenu du dépôt auquel l'utilisateur a accès

Le juriste ne pourra faire l'économie d'envisager les aspects techniques de la création de logiciel s'il souhaite traduire contractuellement la préoccupation de son client de pérenniser son investissement. Il lui faut se demander quels sont les éléments qui doivent figurer dans le

dépôt afin que l'utilisateur soit pleinement autonome pour faire maintenir et évoluer le progiciel après avoir accédé aux éléments déposés.

A/ Les évolutions techniques impactant la négociation des clauses d'accès aux sources

Il faut savoir que l'on ne peut corriger ou faire évoluer un programme que si l'on comprend comment il a été écrit. Pour remédier à une anomalie et corriger un programme défaillant, l'homme de l'art doit rééditer entièrement le programme comme l'on retire un négatif de photo. Il faut donc être capable de rééditer la version objet du progiciel. Or si le code-source suffit à lui seul à reproduire le code-objet (à condition de disposer du compilateur adéquat) il ne permet pas au professionnel de l'informatique d'en saisir l'esprit, ce qui rend impossible toute démarche de modification.

Idéalement, l'informaticien qui accède aux sources devrait être mis en mesure de refaire le parcours du processus intellectuel qu'a emprunté par le programmeur pour créer le programme. En pratique, le juriste est contraint d'adopter une approche non exhaustive mais néanmoins rigoureuse de la liste des éléments devant faire l'objet du dépôt.

De prime abord, il convient de stipuler que le fournisseur devra déposer le programme source ainsi que la documentation utilisateur associée, puis ultérieurement et au moins annuellement, les mises à jour ou nouvelles versions du programme source ainsi que la documentation utilisateur mise à jour.

Il faudra encore accéder aux choix de programmation qui ont présidé à l'élaboration du progiciel ainsi qu'à l'ensemble des informations qui caractérisent l'environnement de développement, assimilables au matériel de conception préparatoire, qualifié d'œuvre de l'esprit et objet en tant que tel d'une protection par le droit d'auteur, aux termes de l'article L 112-2, 13 du Code de la Propriété Intellectuelle.

En pratique, le matériel de conception préparatoire a fait l'objet de bouleversements qu'il convient de prendre en compte au plan juridique. Deux évolutions concomitantes ont marqué la nouvelle ère de la programmation. D'une part, les langages propriétaires (c'est-à-dire les langages de programmation propres à telle ou telle société) n'ont plus cours et les développeurs utilisent, à de rares exceptions près, des langages notoires. D'autre part, l'essor du logiciel libre a modifié les modes de programmation. Ceux-ci se caractérisent par l'utilisation quasi-systématique de composants provenant de l'« open source ». Autrement dit, il n'existe plus de programme qui soit écrit à proprement parler de bout en bout, les développeurs utilisant en partie des éléments redistribuables en les téléchargeant. La difficulté réside dans la possibilité d'identifier et de retrouver ces composants lorsque l'on accède au code-source, afin d'avoir accès à l'intégralité du matériel de conception préparatoire précité. Les éléments téléchargés à un moment T ne pourront plus l'être lorsque l'on accède au code-source parfois des années plus tard, c'est pourquoi ces éléments doivent impérativement faire l'objet d'une documentation systématique.

Du point de vue formel, la documentation de développement figure le plus souvent aujourd'hui sous forme de commentaires insérés par le développeur dans le code-source lui-même. De sorte que le juriste réclamera le dépôt du code-source « documenté » ainsi que le dépôt de la documentation associée de programmation, qui contient d'autres éléments que ceux qui figurent sous chaque ligne de codes à titre de commentaires. On réclamera donc, outre le programme source documenté, la documentation de conception, les dossiers d'architecture identifiant les choix et les environnements de développement, la description des structures de programmation (incluant les algorithmes, c'est-à-dire les formules mathématiques révélant la logique du programme ainsi que les organigrammes), de même que

la description des formats de fichiers (qui sont indispensables à l'interfaçage) et des bases de données utilisées.

Il est à noter que les éditeurs qui externalisent tout ou partie de leur programmation, par exemple en off-shore, sont susceptibles paradoxalement de bénéficier de dossiers de documentation très complets, puisqu'ils prennent eux-mêmes soin, en général, de réclamer une documentation exhaustive des codes sources que leur livrent leurs sous-traitants afin de réduire leur propre dépendance à l'égard de ces derniers. Les éditeurs qui développent l'intégralité de leurs programmes en interne n'auront pas toujours pris le soin d'exiger de leurs programmeurs de tenir des dossiers de documentation aussi complets. Il appartient en tout état de cause au client, ou au tiers séquestre si le client lui en confie la charge, de veiller à ce que le matériel de conception préparatoire fasse partie des livrables attendus lors du dépôt.

B/ Les conséquences en termes de propriété intellectuelle dans le contrat

Le formalisme est, comme on le sait, de mise en matière de propriété intellectuelle et l'accès au code-source n'est pas à lui seul n'est pas translatif de propriété. L'utilisateur devra donc se réserver contractuellement, au cas où les conditions d'accès aux sources sont réunies, le droit de reproduire, modifier, corriger et adapter le progiciel afin d'effectuer les opérations de maintenance corrective ou évolutive nécessaires. Bien entendu, ces droits ne seront concédés que dans le respect du périmètre du contrat de licence. Chacun des droits concédés devra faire l'objet contractuellement d'une mention distincte et le domaine d'exploitation des droits devra être délimité quant à son étendue, sa destination, son lieu et sa durée.

L'on veillera notamment à ménager à l'utilisateur le droit de créer une œuvre dérivée, car le progiciel qu'il aura corrigé ou amélioré constituera une œuvre composite sur laquelle le client ne bénéficiera de droits que sous réserve des droits de l'auteur de l'œuvre première, à savoir le fournisseur du progiciel.

Il apparaît que même si l'utilisateur d'un progiciel a pris l'ensemble des précautions nécessaires en matière d'aménagement contractuel, accéder au code-source ne lui permet que de réduire sa dépendance à l'égard de son fournisseur, sans, le plus souvent, la faire disparaître totalement.

Certains utilisateurs l'ont bien compris, notamment ceux dont la taille et l'envergure économique sont bien supérieure à celle des fournisseurs de progiciels auxquels ils font appel. Ils portent une attention toute particulière à la trésorerie de leurs fournisseurs et régleront leurs redevances sans retard afin d'éviter toute défaillance économique qui les mettrait eux-mêmes techniquement en péril!

D'autres sont parfois contraints, en cas de défaillance de leur fournisseur, de recourir au débauchage de l'un des membres de l'équipe de programmation de l'éditeur, quitte à régler le montant de l'indemnité qui sanctionne la violation de la clause de non-débauchage. Autant dire que la rédaction de cette clause et la fixation de ce montant doivent, aux aussi faire l'objet d'une négociation attentive.