MARDI
13 OCTOBRE 2020
8H45 / 17H30

MAISON DE L'AVOCAT
NANTES

LEGALFEST

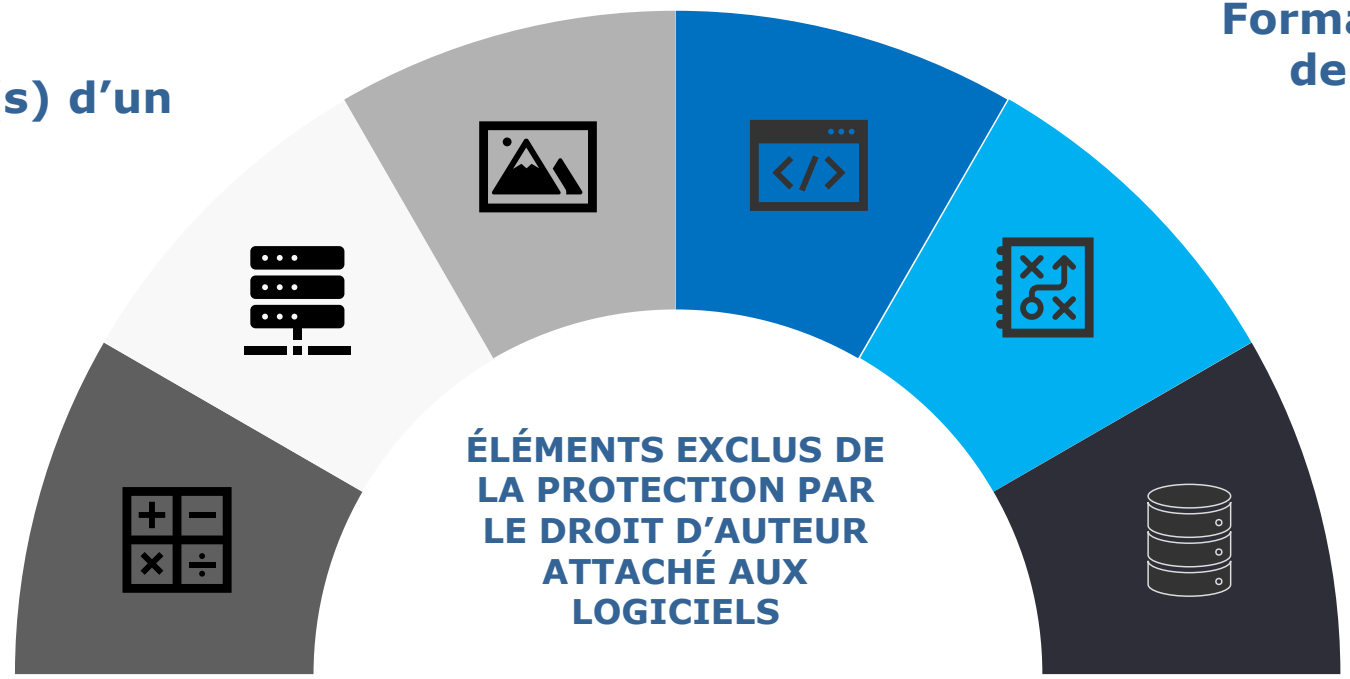
UPDATE JURIDIQUE
PAR LES AVOCATS DU BARREAU DE NANTES

PROPRIÉTÉ INTELLECTUELLE & NUMÉRIQUE #2

Logiciels: la question de l'originalité

Pierre Langlais & Mael Fablet

Qu’est-ce que n’est pas un logiciel?



Qu’est-ce que c’est alors un logiciel?

- Un code source … et un code exécutable …
- Euh oui d’accord ! mais concrètement ?

Qu’est-ce que c’est alors un logiciel?

- **Le code source** est le texte rédigé par un développeur correspondant aux instructions à donner à la machine. Ce texte est le plus souvent annoté afin que le code puisse être compris par d'autres développeurs. Il peut être apparenté à une recette de cuisine.

- Voici un exemple d'une partie de code source, rédigé par un développeur :

```
jeu<-function(){
        print("Entrez votre nom : ")
        nom<-as.factor(readLines(n=1,skipNul=TRUE))
        print("Le jeu consiste Ã  deviner un nombre tirÃ© alÃ©atoirement entre 1 et 100 le plus rapidement
possible")
        i<-1
        nombre<-sample(1:100,1)
        print("Entrez un nombre")
        x<-as.integer(readLines(n=1,skipNul=TRUE))
        while(x!=nombre){
                if(x<nombre){
                        print("C'est plus")
                        i<-i+1
                        print("Entrez Ã  nouveau un nombre")
                        x<-as.integer(readLines(n=1,skipNul=TRUE))
                }else{
                        print("C'est moins")
                        i<-i+1
                        print("Entrez Ã  nouveau un nombre")
                        x<-as.integer(readLines(n=1,skipNul=TRUE))
                }
        }
paste("Bravo",nom,"!","Vous avez trouvÃ© en",i,"coups",sep=" ")
}
```

Qu’est-ce que c’est alors un logiciel?

- **Le code exécutable** correspond au code source compilé, c'est-à-dire traduit en langage binaire, à savoir une suite de chiffres exclusivement compréhensible par la machine (l'ordinateur) qui va l'exécuter.

- Pour passer du code exécutable au code source on parle alors de décompilation.

En résumé :

Le code source correspond au texte d'écriture du logiciel dans un langage compréhensible par l'homme (ie. le développeur) et le code exécutable à sa traduction pour qu'il soit compris par la machine (ex : l'ordinateur) qui va le mettre en œuvre.

Qu’est-ce qu’un logiciel original?

- L’originalité de la création de forme est une notion classique du droit d’auteur. La conception classique considère qu’est originale une œuvre qui porte « *l’empreinte de la personnalité de son auteur* ».

- La protection par le droit d’auteur de créations plus techniques comme le logiciel a contraint cependant les juges à adapter la définition de l’originalité à ces dernières.

- Cass. Ass. Plén.7 mars 1986 (arrêt Pachot) => pour un logiciel, l’originalité se caractérise par "*un effort personnalisé allant au-delà de la simple mise en œuvre d’une logique automatique et contraignante*"

Qu'est-ce qu'un logiciel original ?

⇨ Originalité d'un logiciel: « *suppose que l'auteur l'ait marqué de son apport intellectuel par un apport créatif révélateur de sa personnalité* ***allant au-delà de la simple mise en œuvre d'une logique automatique et contraignante*** *et que la matérialisation de cet effort réside dans une structure individualisée (CA Paris, 24 novembre 2015 n°13/24577).*

⇨ Ces choix sont le résultat d'un effort créatif lorsque **l'auteur démontre qu'ils « *n'auraient pas pu être réalisés aisément par un autre individu* »** (CA Paris, 24 novembre 2015 n°13/24577)

Critères ayant été pris en compte par la jurisprudence pour apprécier l'originalité du logiciel ✓	**Critères non pris en compte par la jurisprudence pour apprécier l'originalité du logiciel** ✗
1. code source (effort personnalisé, absence de logique contraignante) et matériel de conception préparatoire	**1. Eléments ne tenant pas au code source du logiciel ou aux matériels préparatoires** • Caractère innovant du logiciel • Valorisation financière du logiciel • Crédit d'impôts recherche • Dépôt des codes sources à l'APP • Concept du logiciel • Importance des moyens mis en œuvre
2. architecture du programme et enchaînement des instructions	
3. Nouvelles versions : autres langages de programmation	
4. état de la technique existante au moment de sa conception : permet de caractériser l'apport créatif	**2. Mise en œuvre d'une logique automatique et contraintes métiers / techniques :** • Processus métiers • Contraintes techniques • Contraintes opérationnelles • Contraintes techniques induites par la spécificité du service • Solution particulière à une activité donnée • Certaines solutions techniques: API, add-ons, grafcets
5. Existence de spécifications fonctionnelles (mais insuffisant en soi)	
6. Interdépendance des logiciels	

Comment et quand démontrer l'originalité ?

- Démonstration par le(s) développeur(s) eux/lui-même(s) ?
- Démonstration par un expert privé ?
- Démonstration par un expert judiciaire désigné le juge ?
- Au stade l'assignation ou de la requête (en cas de saisie-contrefaçon) ?