

# L'IPRA (Intellectual Property Rights Analysis) pour renforcer la confiance dans les logiciels issus de la recherche

06/07/10

INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE



Patrick MOREAU  
INRIA, Direction du Transfert et de l'Innovation

En collaboration avec Magali Fitzgibbon et Luc Gâteau,  
INRIA, Direction du Transfert et de l'Innovation

<http://creativecommons.org/licenses/by-nc-nd/2.0/fr/>

# Agenda

Présentation des objectifs de l'IPRA

Définitions communes

Présentation de la méthodologie

Outillage de la méthodologie

Bonnes pratiques sur la prise en compte de l'IPR

Conclusions



# Constat

La dimension du marché mondial des logiciels et des services:

- 550 milliards d'euros en 2007, dont environ 30 milliards d'euros en France,
- Sur ces 30 milliards d'euros en 2007 en France, 1,1 milliard d'euros en 2008 pour le logiciel libre

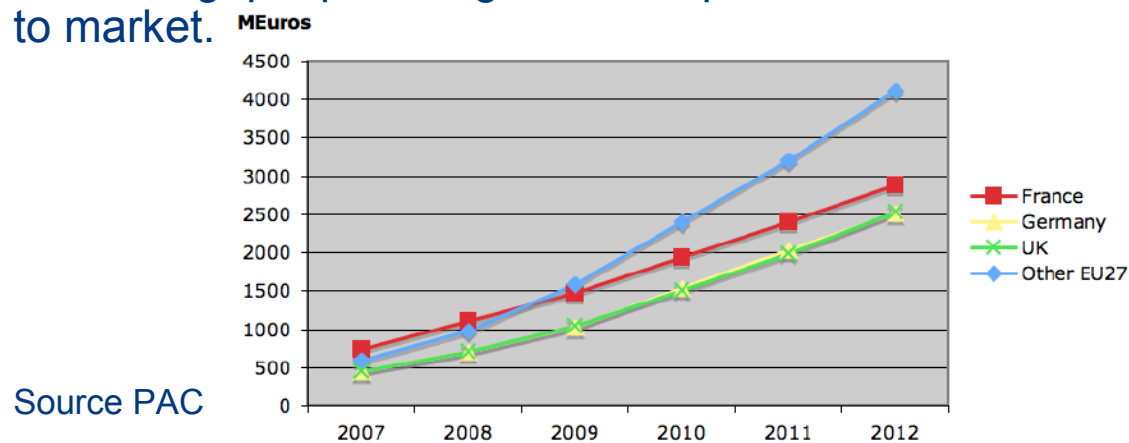
Forte croissance du Logiciel Libre en France:

- Croissance de plus de 33% en 2009 et prévue à 30% en 2010

Cela reste faible en valeur absolue.

- Il existe une certaine frilosité à utiliser du Logiciel Libre

Le Logiciel Libre est un instrument stratégique pour augmenter la productivité des entreprises et améliorer le time to market.



# L'INRIA réaffirme son soutien à la dynamique du logiciel libre

Le Logiciel Libre est devenu un fait incontournable de l'édition logicielle.

Le choix de ce mode de diffusion / transfert par l'INRIA doit être mûrement réfléchi afin de rechercher la stratégie qui maximisera l'impact sur la société (scientifique, sociétal et économique)

La définition de la stratégie d'utilisation du Logiciel Libre comme un outil de transfert conduit à deux axes de réflexions et d'actions:

- internes:
  - Professionnalisation des pratiques d'édition
  - Accroissement de l'impact des logiciels développés et stratégie de sortie
- externes:
  - Contribution à la création de la confiance autour du Logiciel Libre
  - Mise en place d'écosystèmes facilitant le transfert et l'innovation





# Logiciel Libre = objet de Transfert

Le Logiciel Libre est un objet complexe et le choix de ce mode de diffusion / transfert doit être mûrement réfléchi afin de rechercher la stratégie qui maximisera l'impact sur la société (scientifique, sociétal et économique)

Il se peut qu'il y ait parfois une contradiction entre les logiques de diffusion scientifique et de transfert

- utiliser systématiquement la diffusion des résultats sous forme de Logiciel Libre peut obérer le potentiel de transfert de la recherche publique
- le modèle du Logiciel Libre n'a pas toujours l'impact escompté y compris en terme académique

La décision de diffusion se prend avec les chargés de transfert / valorisation



# Confiance juridique dans le Logiciel Libre

Les logiciels deviennent de plus en plus complexes

Le développement collaboratif a pris un essor considérable

Au niveau de l'INRIA, mais vrai pour toute communauté, la gestion de l'IPR est indispensable dès que l'on évoque la diffusion d'un logiciel

- Respect de la loi
- Diminution du risque
- Renforcement de la confiance dans les logiciels issus de l'INRIA



# Présentation des objectifs de l'IPRA

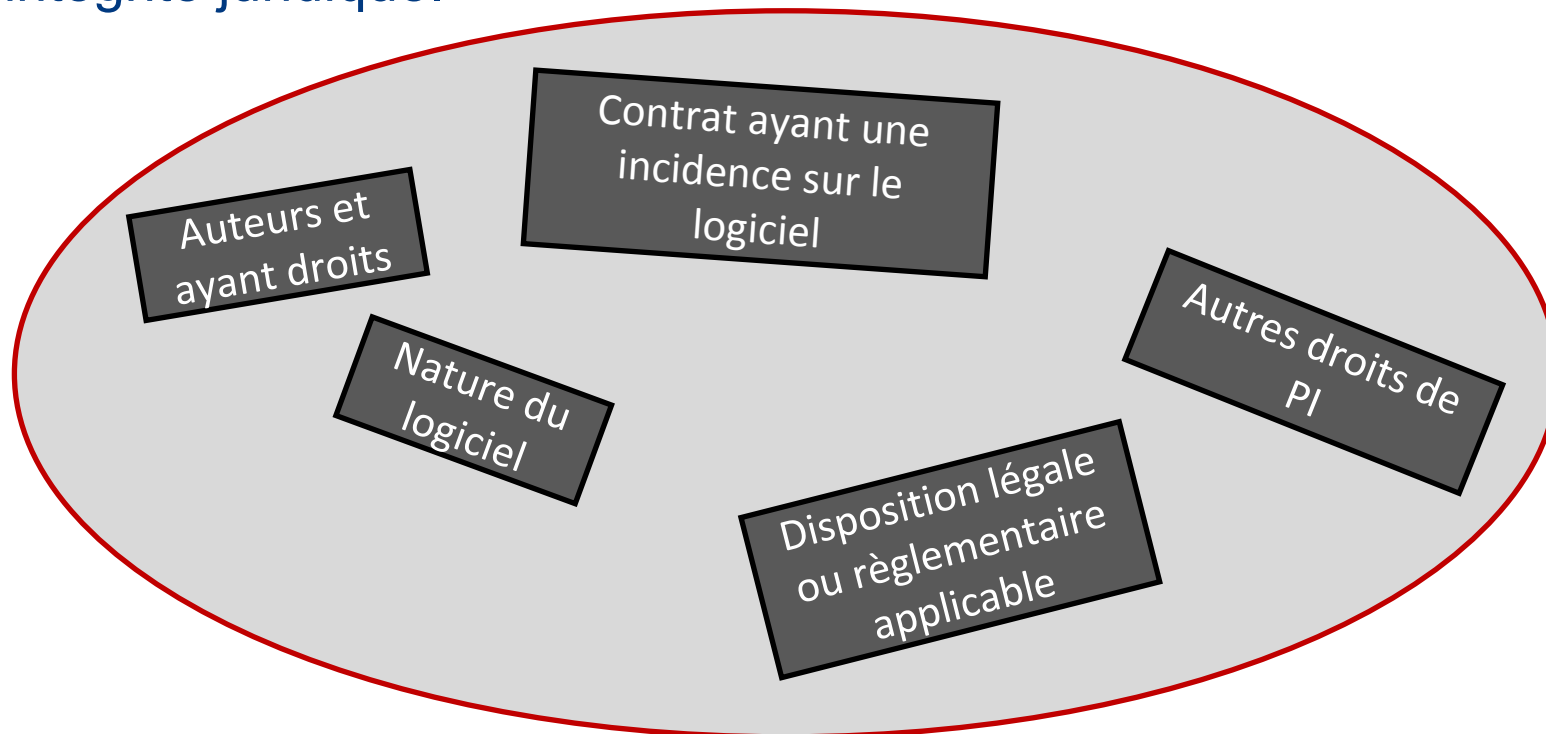


INSTITUT NATIONAL  
DE RECHERCHE  
EN INFORMATIQUE  
ET EN AUTOMATIQUE



# IPRA: Intellectual Property Rights Analysis

L'analyse IPR est une « photo » à un moment de la vie du logiciel visant à qualifier le statut juridique d'un logiciel c'est-à-dire son intégrité juridique.





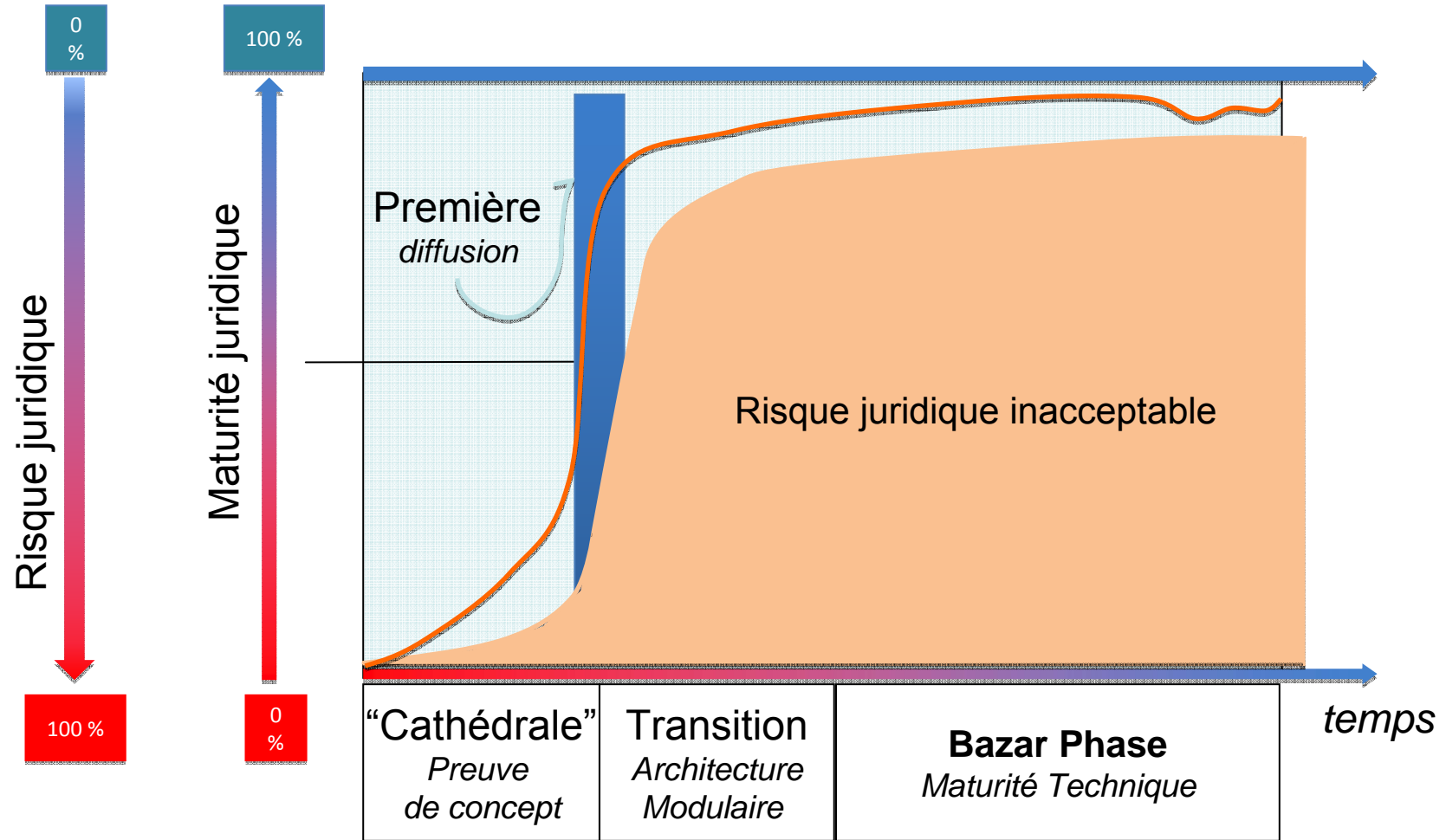
# Les risques

Nous illustrons ce besoin par quelques exemples fortement problématiques:

- une Equipe distribue un logiciel contenant des composants d'autres Equipe , qui eux-mêmes sont distribués sous une autre forme contractuelle
- Une start-up est créée mais découvre au bout d'un certain temps qu'elle n'est pas propriétaire du logiciel qui la fonde
- Un stagiaire revendique des droits de propriété sur un logiciel
- Un contrat de consortium interdit la diffusion d'un logiciel sous une forme ou une autre
- Un composant Open Source intégré au projet restreint fortement les capacités de transfert du logiciel



# L'importance du juridique dans un projet



# Problèmes de compatibilité de licences dans un logiciel et réalité des risques encourus

Idée reçue n°1: il n'y a aucun risque, la preuve, il n'y a jamais de litige!

Faux!

Plusieurs arrêts rendus en France:

- EDUCAFIX c/ CNRS, UJF
- EDU4 c/ AFPA, 21 septembre 2004 (TGI de Bobigny)
- EDU4 c/ AFPA, 16 septembre 2009 (Cour d'appel de Paris)
- Litige lié aux logiciels libres de la FreeBox

Il n'est pas possible de connaître le nombre de litiges liés à l'open source pour lesquels une solution amiable à été trouvée.

Un risque plus grand du fait de l'apparition d'outils permettant de tracer la réutilisation de code dans les logiciels et d'identifier les licences rattachées aux composants d'un logiciel



# Problèmes de compatibilité de licences dans un logiciel et réalité des risques encourus

Idée reçue n°2: il n'y a pas de risques lorsque le logiciel est un projet open source, car c'est du libre et pas du propriétaire!

Faux!

Le risque est, il est vrai, sans doute plus limité mais pas inexistant.

Au-delà du risque juridique se pose la question du risque lié à l'image de l'Institut



# Quelques définitions communes



# C'est quoi un logiciel?

## Réponse juridique

- Le terme « logiciel » désigne tout ensemble de programmes informatiques, que ce soit en code source ou en code objet, qui constitue une œuvre protégeable par le droit d'auteur sur laquelle des droits peuvent être concédés. Les spécifications, documentations font aussi partie de l'œuvre.

## Réponse technique

- « C'est un système à composants (SaC) dont certains peuvent eux - même être d'autres SaC et pouvant être le composant d'un autre SaC plus complexe » (Luc Grateau, INRIA)
- Le Logiciel est l'ensemble des composants, bibliothèques, ... permettant une fois linké et compilé l'exécution sur du matériel
  - La frontière en ce qui concerne les composants faisant partie du logiciel et les autres logiciels indépendants peut être difficile à définir donc sujette à discussion



# Parler un langage commun

## Composants:

- un fichier de code source ou un ensemble de fichiers assurant une fonction cohérente

## Architecture

- Description des composants du logiciel par grands blocs fonctionnels et description des relations entre ces blocs

## Logiciel dérivé

- « modifications »,

## Logiciel composé ou assemblé

## Seront clarifiées les notions suivantes:

- "re-programmation", "re-programmation dans un autre langage", "traduction", « ré-écriture »

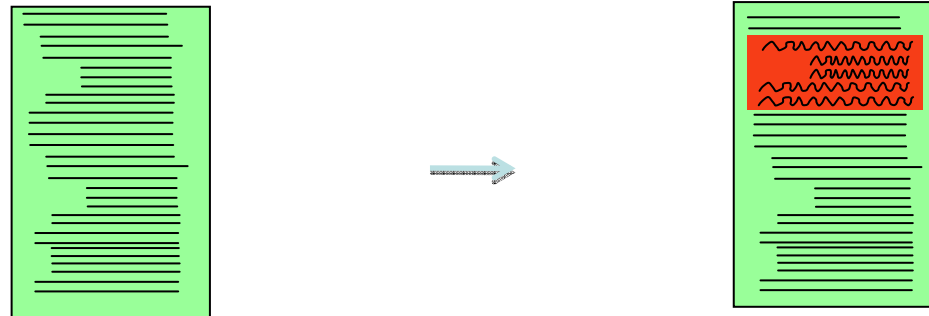
## Un logiciel peut être à la fois composé et dérivé

Ces termes sont utilisés au sens du Logiciel Libre et non à celui du droit des auteurs



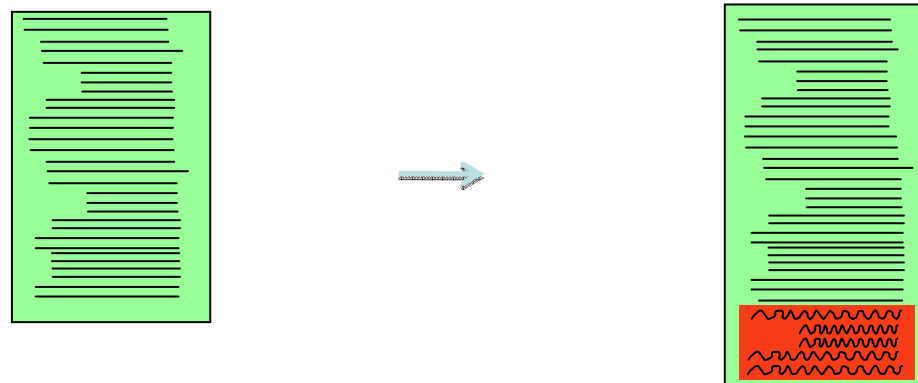
# Dérivation de composants

Modification / amélioration du code source d'au moins un fichier d'un composant



Ajout de code au code source d'au moins un fichier d'un composant

- Créé ex-nihilo





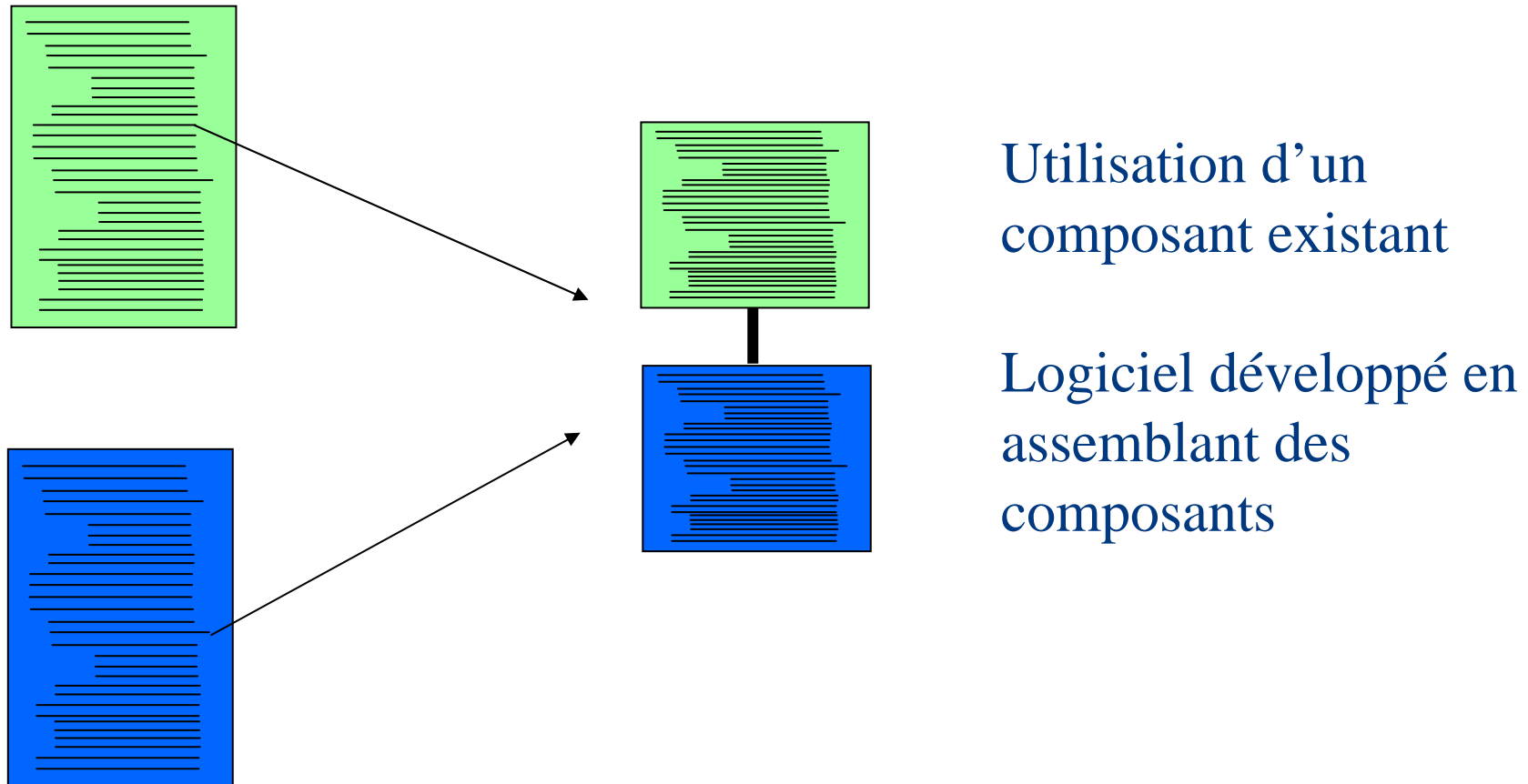
# Dérivation de composants

Ajout de code au code source d'au moins un fichier d'un composant

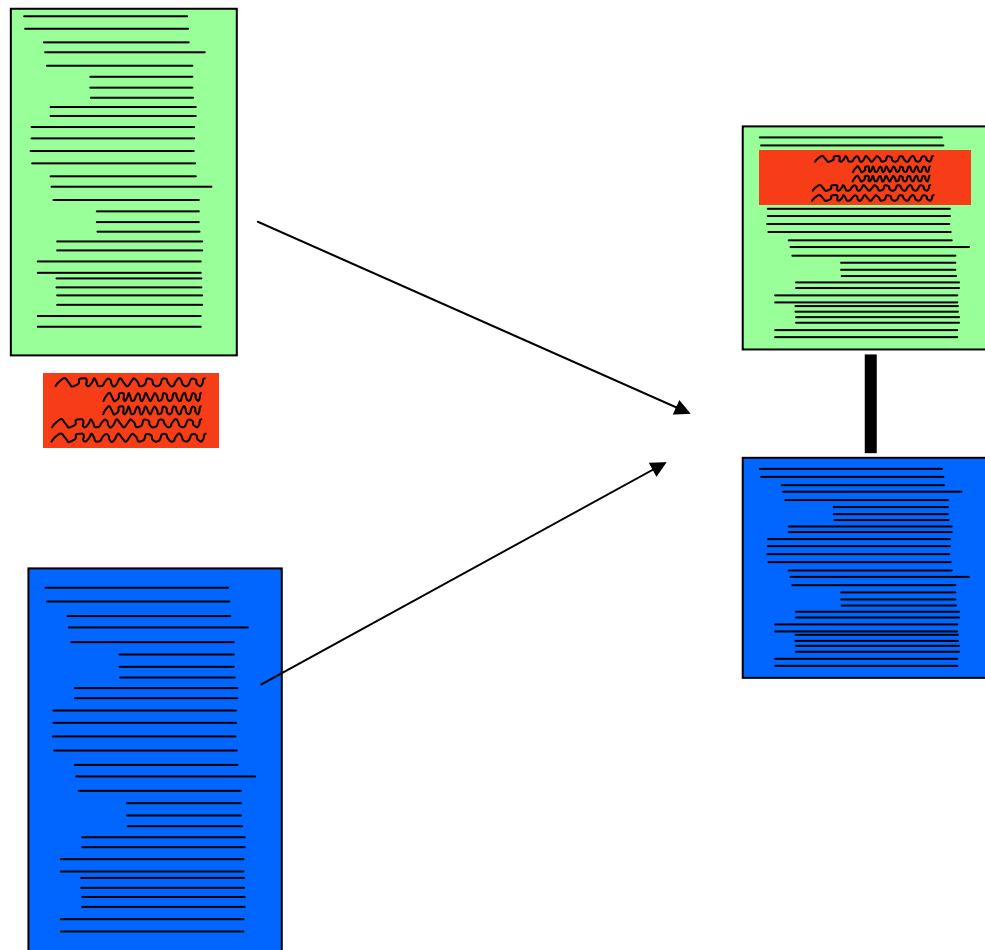
- issu d'un composant pré-existant



# Composition ou assemblage de composants



# Logiciel dérivé et composé



Assemblage possible  
de composants dérivés  
ou non



## Tableau de synthèse

Nature	Catégorie	Exemple de licence	DERIVATION Modification et redistribution ?	COMPOSITION Agrégation et redistribution ?
NON - LIBRE	"Propriétaire"		N'est pas automatique	N'est pas automatique
LIBRE  NON PERMISSIVE	non permissive en dérivation et composition	CeCILL, GNU GPL	sous la même licence	sous la même licence
	non permissive en dérivation, permissive en composition	CeCILL C, GNU LGPL	sous la même licence	OUI, y compris sous une licence différente
LIBRE PERMISSIVE	en dérivation et composition	CeCILL B, BSD	OUI, y compris sous une licence différente	OUI, y compris sous une licence différente

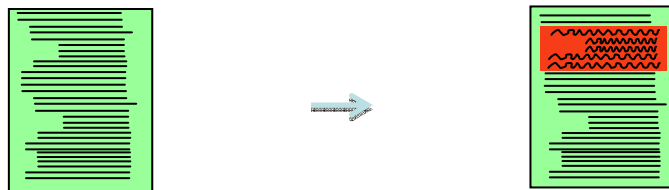
Nous n'utilisons pas le terme « licence contaminante » car peut effrayer un néophyte



# Exemples

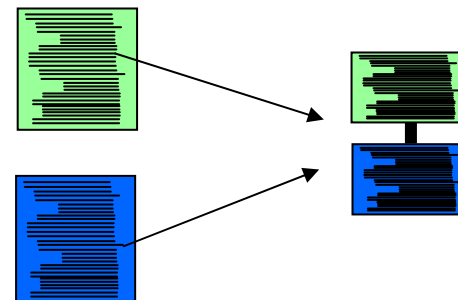
## Dérivation

- Composant initial sous LGPL
- L'intégralité du code source du composant final devra être redistribué avec la licence LGPL



## Assemblage

- Composant initial sous LGPL
- L'assemblage pourra être distribué avec la licence souhaitée
- Le composant initial reste sous LGPL



# La méthodologie



# Principe

La méthodologie est basée sur la responsabilité et la bonne foi des chercheurs / ingénieurs.

- D'où la préférence du terme « analyse » plutôt que celui d' « audit »

C'est un élément essentiel qui devrait aussi permettre une meilleure acceptation de celle-ci.



# La Legal Situation IPR

La Legal Situation(LS) est déterminée par l'identification:

- des auteurs / ayant-droits
- de la nature du logiciel (premier, dérivé, composé)
- des contrats ayant une incidence sur le logiciel (modes contractuels et de financements du développement des différents modules constitutifs du logiciel, licences des composants...)
- d'autre disposition légale ou réglementaire applicable
- des autres droits de PI opposables

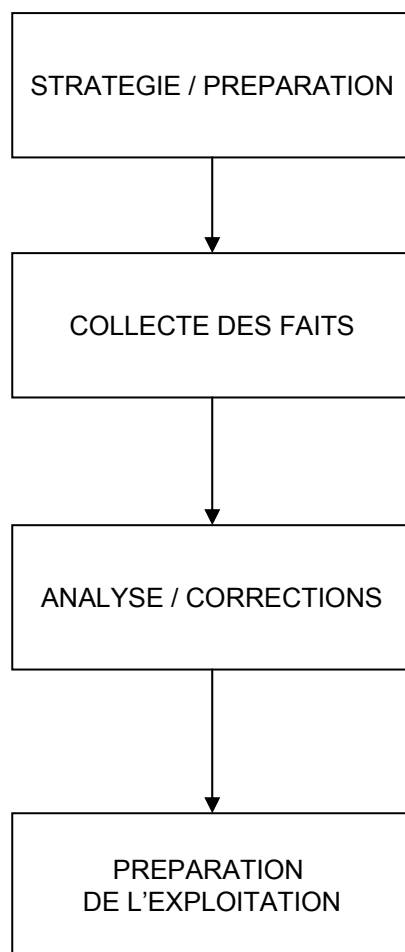
Cette définition donne un ensemble minimum d'éléments, qui ont un impact sur la façon dont le logiciel pourra être utilisé.

Ceci permet de déterminer en conséquence les possibles modes d'utilisation et d'exploitation du logiciel et, de déterminer si ces modes sont compatibles avec les objectifs d'exploitation initialement choisis

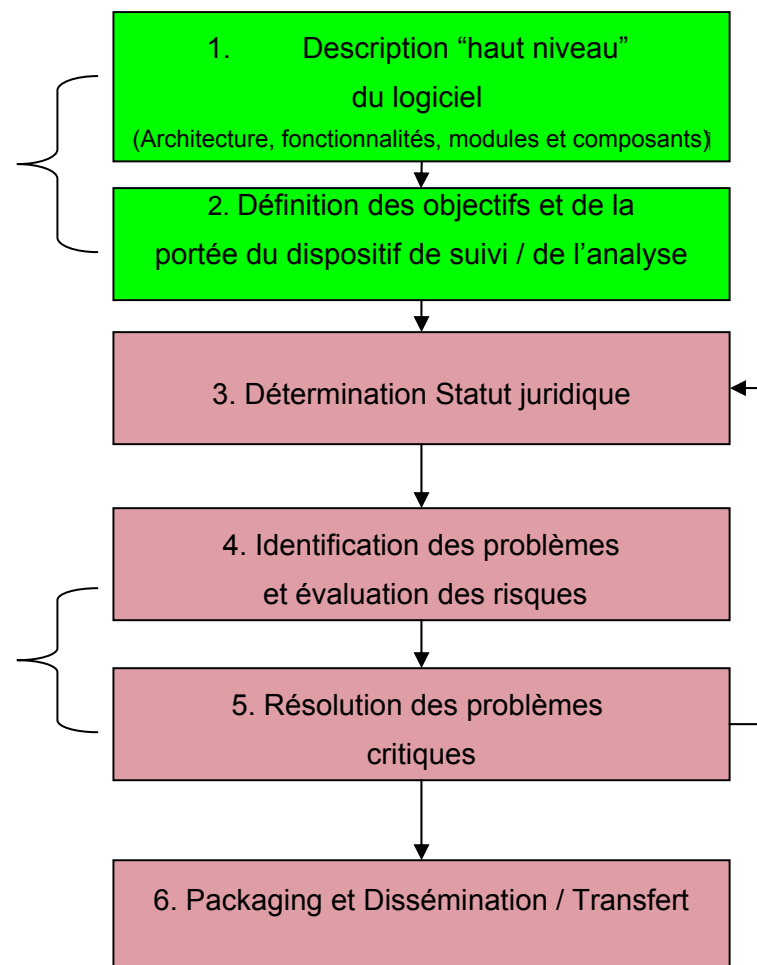




# Méthodologie d'IPRA



## Phases :



# Phase 1: Description haut niveau du logiciel

La phase 1 de la méthodologie de suivi juridique du développement logiciel est le point de départ dont la qualité détermine celle de la suite de la démarche.

C'est à partir de la description haut niveau du logiciel que

- sera défini l'objet de l'analyse
- sera formalisé le travail de définition de la stratégie de transfert.



# Phase 1: Description haut niveau du logiciel

## Quoi?

- Documenter le logiciel et l'architecture du système d'information dans lequel il s'inscrit

## Pourquoi?

- Définir le périmètre de l'analyse
- Permettre d'établir ou formaliser une stratégie de développement et d'exploitation en amont (phase 2) et de définir un processus ad hoc de suivi.
  - Cette stratégie peut être nuancée en fonction des éléments constitutifs. Par exemple, noyau en centralisation des droits, modules connexes sans centralisation (ou l'inverse)
- Permettre la mise en relation les éléments constitutifs du logiciel (composants identifiés en phase 3) avec la description haut niveau du logiciel.



# Phase 1: Description haut niveau du logiciel

## Comment?

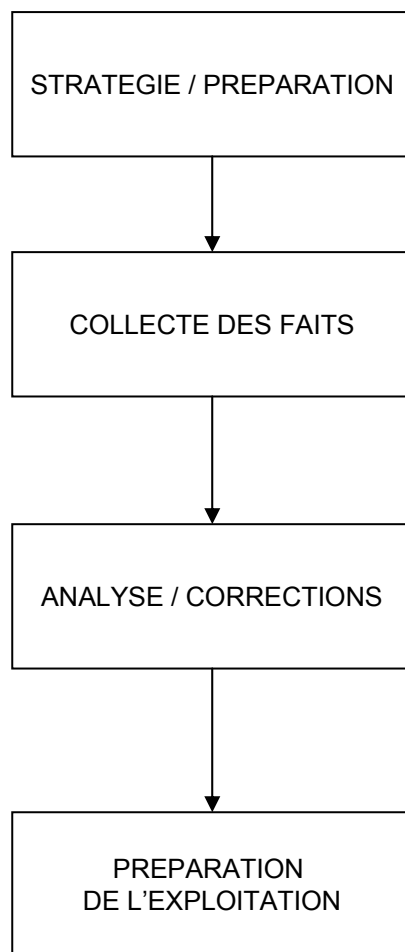
- Il n'y a pas de description générique et universelle.
- La description est spécifique de chaque logiciel.
- Elle consiste en:
  - Une description générale du logiciel à plus grande échelle (facultatif et seulement si pertinent)
  - Une description des différentes zones fonctionnelles du logiciel et leurs interactions entre elles (obligatoire).
  - Une description de chaque zone fonctionnelle détaillant les différents composants qui en font partie, ainsi que leur interaction et les dépendances entre eux (obligatoire)

## Qui?

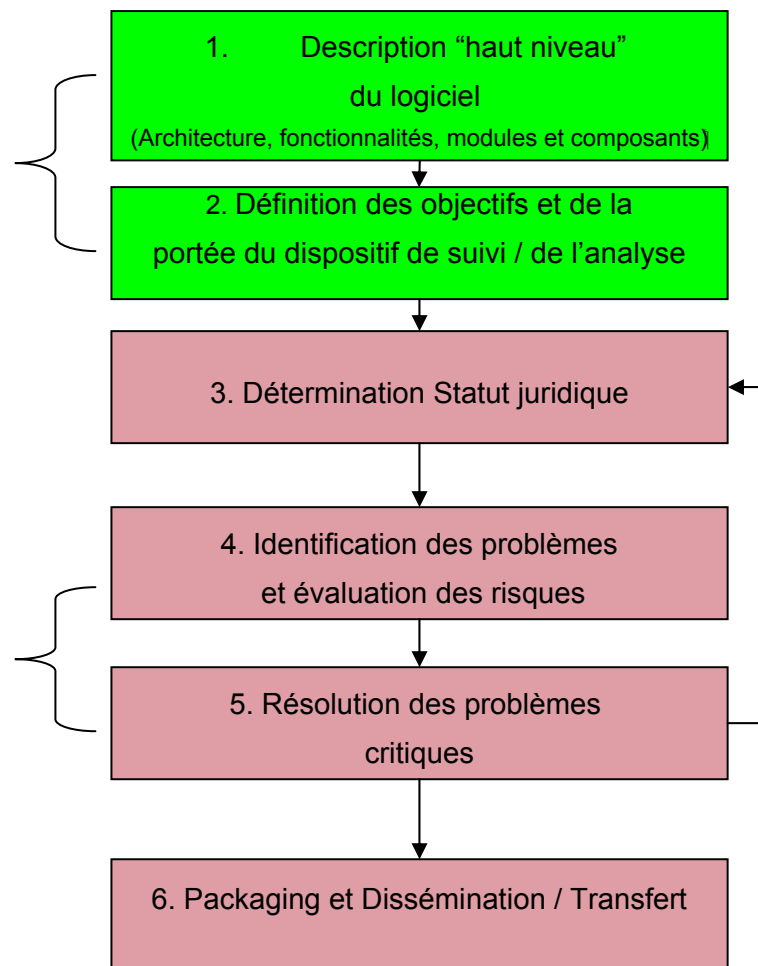
- La description haut niveau du logiciel est définie par l'équipe de développement, son responsable, ou un architecte, selon les cas.



# Méthodologie d'IPRA



## Phases :



## Phase 2: définition des intentions

Si cela n'a pas été fait auparavant par l'équipe et le CPPI (chargé de valorisation) au démarrage dans le cycle de vie du projet, une action préliminaire clé de l'équipe d'analyse est de formaliser le plan d'exploitation.

Si ces buts et objectifs ne sont pas correctement définis, la pertinence de la vérification ne sera pas optimisée

- En effet, une fois que l'analyse est faite et que la situation juridique du logiciel est établie, les buts et objectifs seront utilisés conjointement avec ce statut juridique au cours de la phase d'analyse, afin de déterminer si les deux sont compatibles (4)
- Si ce n'est pas le cas, certaines mesures correctives doivent être prises pour résoudre les problèmes identifiés (5) avant de préparer la version finale (6).

Ces intentions peuvent être nuancées en fonction des différentes parties constitutives du logiciel, telles que décrites dans la phase 1.



## Phase 2: définition des intentions

Capter une partie  
de la valeur

NON	• <b>Free Saas</b>	• <b>Licensing Permissif</b>
	• <b>Licencing Propriétaire (Editeur ou Saas)</b>	• <b>Double Licensing</b> • <b>Licensing non Permissif + Services</b>
OUI	NON	OUI

Licence Libre

## Phase 2: outils

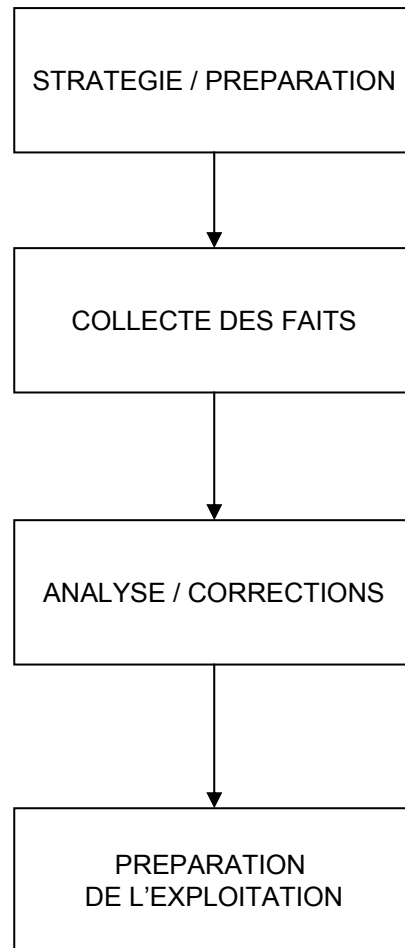
La note, pour l'instant interne, « Cadre de réflexion pour le choix de diffuser le logiciel sous une licence libre » donne un certain nombre de guidelines.

Lorsque le logiciel peut être naturellement séparé en plusieurs parties (par exemple un noyau et des bibliothèques ou dans le cas d'une architecture de composants ou de plug-ins), il pourrait être judicieux d'effectuer la définition des intentions sur chaque partie séparément et par la suite de définir éventuellement des schémas de transfert différents avec des licences différentes selon les parties.

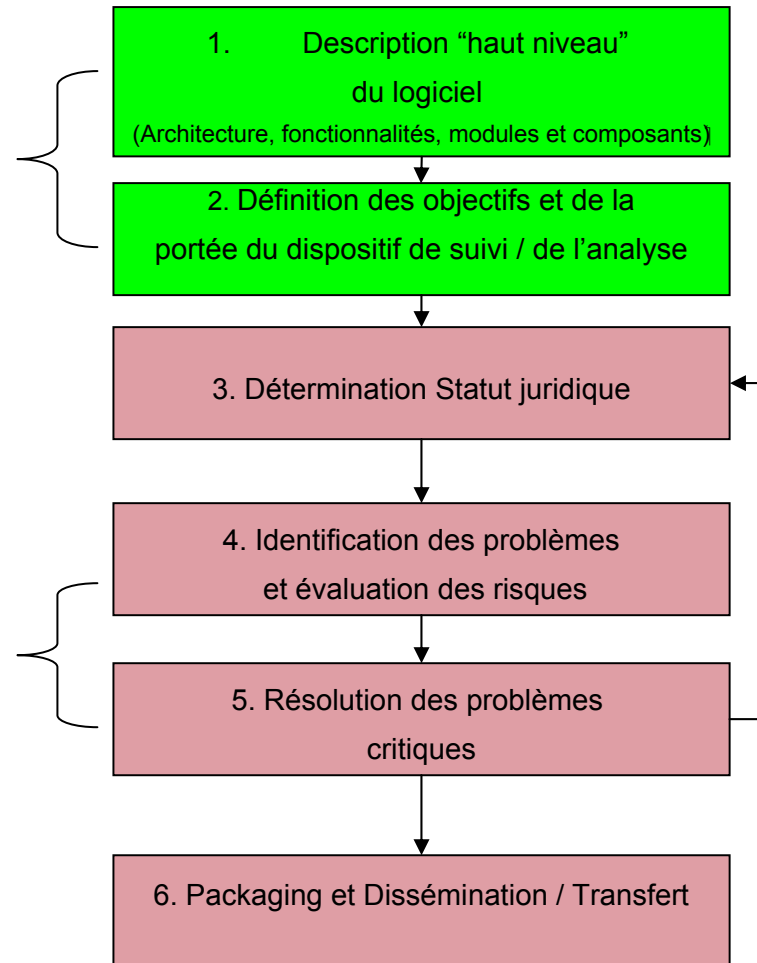




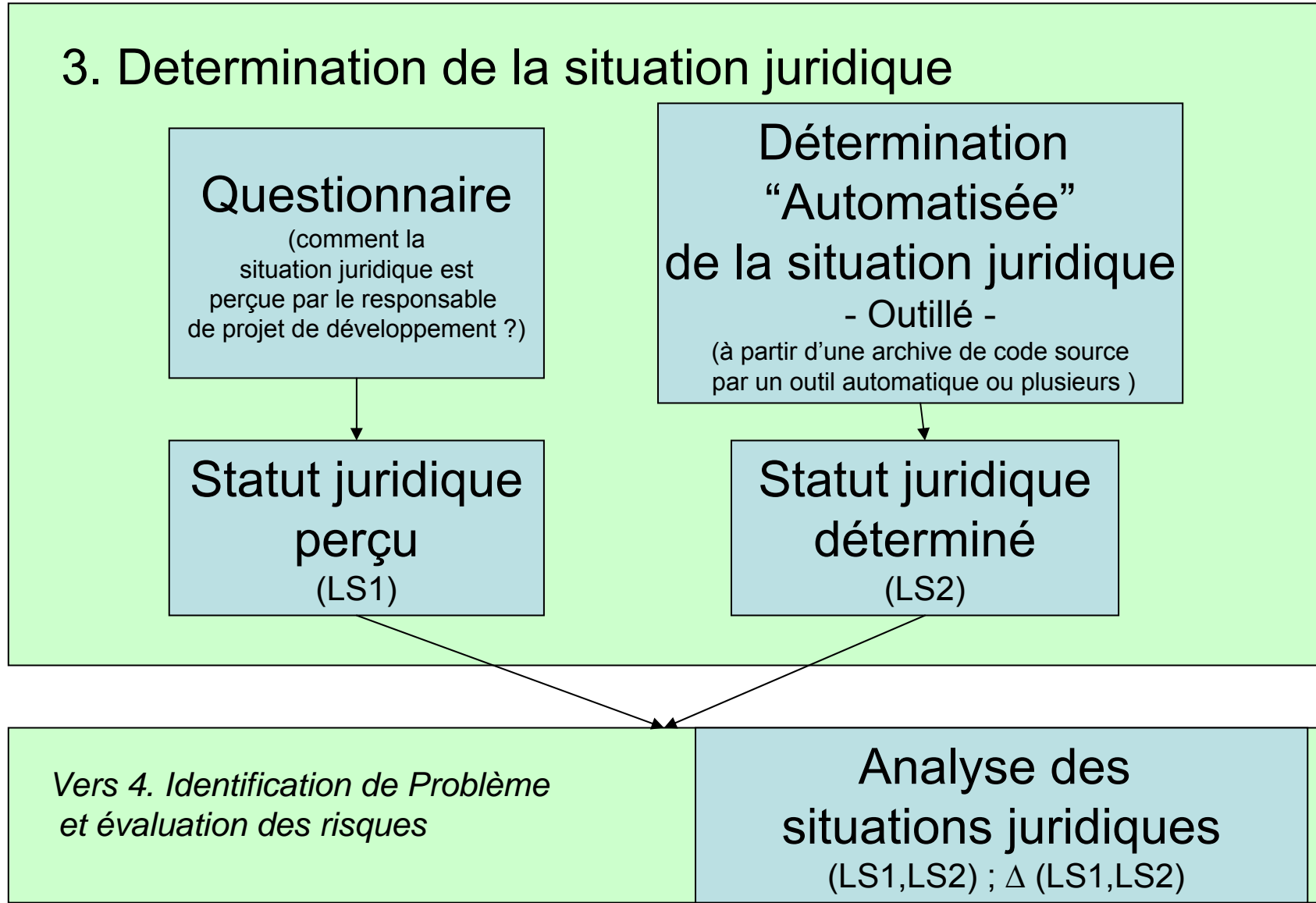
# Méthodologie d'IPRA



## Phases :



# Phase 3: détermination de la situation juridique



## Phase 3: détermination de la situation juridique

Cette étape consiste en l'analyse des différents éléments ayant une incidence sur le statut juridique du logiciel, et par conséquent sur la «marge de manœuvre » de l'INRIA concernant l'exploitation/diffusion du logiciel.

Le statut juridique est déterminée en comparant le statut juridique "perçu" du statut juridique « déterminé ».

Un guideline très utile se trouve en annexe du document Qualipso, « Report on the proposed IPR tracking methodology ».



# Phase 3: détermination de la situation juridique

## Situation perçue:

- Basée sur questionnaire
- Elle consiste en un recueil d'informations
- Double but: responsabilisation et but pédagogique
  - en effet des éléments peuvent être oubliés
- Exemple d'un logiciel
  - L'intégration d'un composant sous GPL avait été oublié, ce qui aurait posé problème...



## Phase 3: actions de l'équipe d'analyse

Dresser une liste des composants du logiciel, tout en précisant **la licence régissant chaque composant**.

A partir de la description haut niveau, indiquer les licences connues et identifiées des composants

Description du mode d'organisation de développement du logiciel.

- historique des contrats de développement autour du logiciel

Préciser, pour chaque composant, s'il a été (à votre connaissance) développé à l'INRIA, s'il s'agit d'un composant préexistant appartenant à tiers ou d'un composant développé à l'INRIA mais intégrant du code préexistant (un membre de l'Equipe développe un composant et y intègre du code exogène). Ces informations peuvent être indiquées dans la cartographie du logiciel par le biais du code couleur suivant :

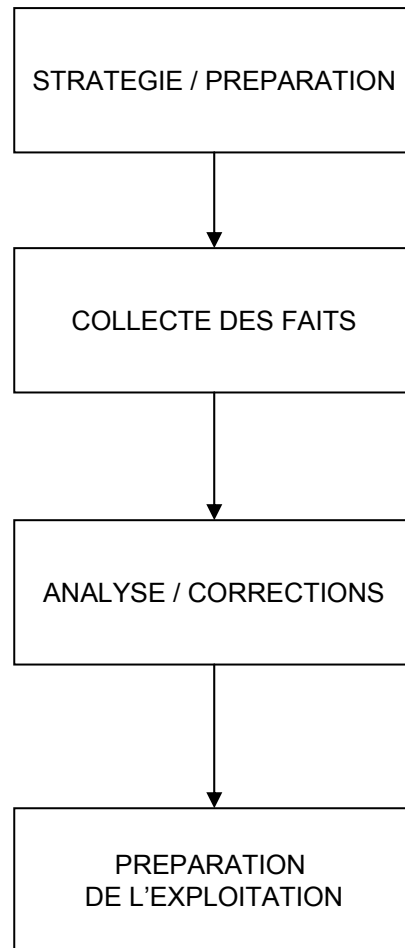
- Composant développé à l'INRIA : **vert**
- Composant développé à l'INRIA mais intégrant du code préexistant : **orange**
- Composant préexistant appartenant à un tiers : **rouge**

Etablir la liste globale des auteurs du logiciel et leur statut au cours du développement du logiciel

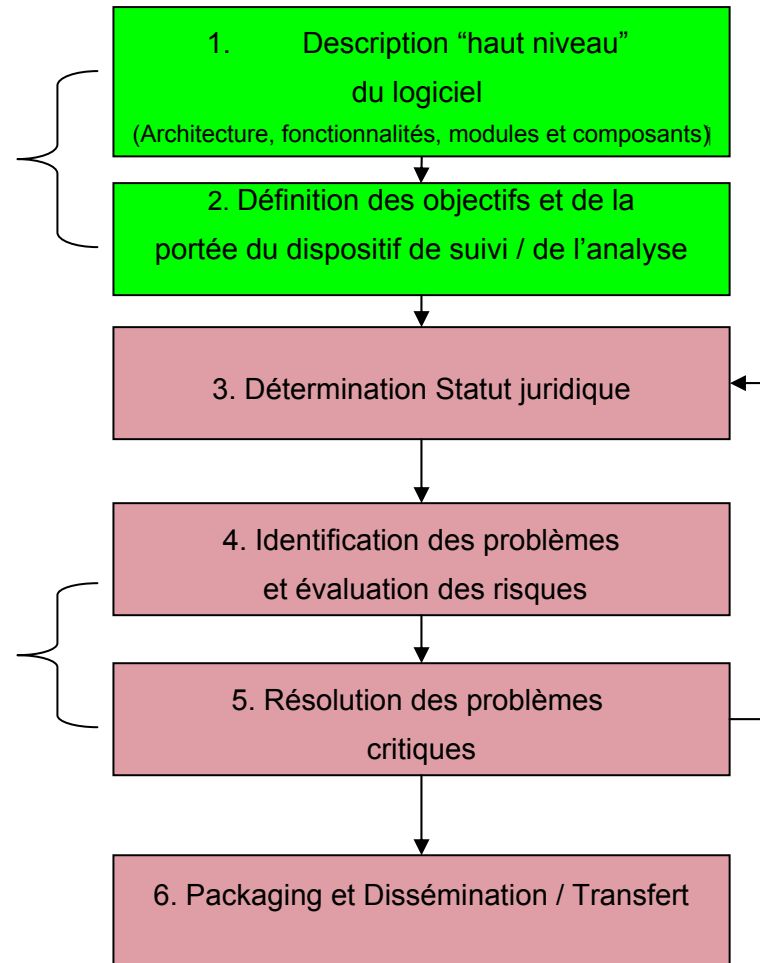
Préciser quels sont les composants qui ne sont pas indispensables au bon fonctionnement du logiciel et qui pourraient éventuellement être retirés.



# Méthodologie d'IPRA



## Phases :



# Phase 4: Contexte de la problématique

## Contexte collaboratif

- Gestion de la PI dans les contrats de partenariats / collaboratifs
- Gestion de la PI de contributeurs externes

Compatibilité de licences IN vers licence OUT



## Phase 4: Contexte de la problématique

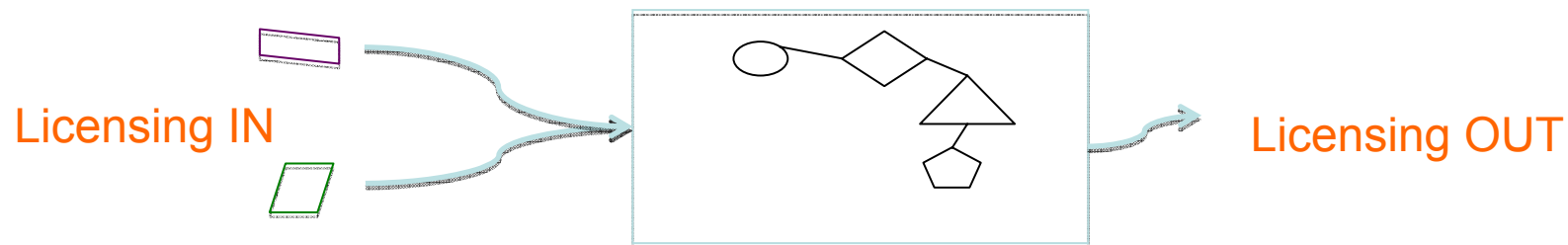
Il existe plusieurs méthodologies de développement

- Partir ex-nihilo (de plus en plus rare)
- Utiliser des composants préexistants (dérivés, composer, combiner/liier)

Certaines licences de composants peuvent être incompatibles avec la licence de diffusion propriétaire ou libre souhaitée

- La stratégie de développement doit être adaptée à la stratégie d'exploitation


Il faut ainsi vérifier les compatibilités des licences des composants du Logiciel (Licensing-in) versus la licence choisie pour diffuser le Logiciel Libre (Licensing-out)





# Matrice de compatibilité entre licences libres

## DANS UN SCHEMA AVEC COMBINAISON DE COMPOSANTS PREEXISTANTS



		Licence de distribution souhaitée (LICENSING OUT)													
		Licenc e 1	Licenc e 2	Licenc e 3	Licenc e 4	Licenc e 5	Licenc e 6	Licenc e 7	Licenc e 8	Licenc e 9	Licenc e 10	Licenc e 11	Licenc e 12	Licenc e 13	Licenc e 14
L I C E N S I N G  I N	Licence 1	Grey	Orange	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 2	Red	Grey	Red	Red	Green	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 3	Green	Green	Grey	Orange	Red	Red	Green	Green	Green	Green	Red	Green	Red	Green
	Licence 4	Red	Green	Green	Grey	Red	Red	Green	Green	Green	Green	Red	Green	Red	Green
	Licence 5	Red	Green	Red	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 6	Green	Green	Red	Red	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 7	Red	Red	Red	Red	Red	Green	Grey	Green	Green	Red	Green	Red	Red	Red
	Licence 8	Red	Red	Green	Green	Red	Red	Green	Grey	Green	Green	Green	Green	Red	Green
	Licence 9	Green	Green	Green	Green	Green	Red	Red	Green	Grey	Green	Green	Green	Green	Green
	Licence 10	Red	Red	Green	Green	Red	Red	Red	Green	Green	Grey	Red	Red	Red	Green
	Licence 11	Red	Green	Green	Green	Green	Red	Red	Green	Green	Green	Grey	Green	Green	Green
	Licence 12	Red	Red	Orange	Orange	Red	Red	Orange	Orange	Orange	Orange	Orange	Grey	Red	Orange
	Licence 13	Green	Red	Red	Red	Red	Green	Red	Red	Red	Red	Red	Red	Grey	Green
	Licence 14	Red	Red	Green	Green	Red	Red	Green	Green	Green	Green	Red	Green	Red	Grey

# Matrice de compatibilité entre licences libres

DANS UN SCHEMA AVEC « DERIVATION AVEC INTEGRATION » DE CODE PREEXISTANT  
DANS LES FICHIERS SOURCES



		Licence de distribution souhaitée (LICENSING OUT)													
		Licenc e 1	Licenc e 2	Licenc e 3	Licenc e 4	Licenc e 5	Licenc e 6	Licenc e 7	Licenc e 8	Licenc e 9	Licenc e 10	Licenc e 11	Licenc e 12	Licenc e 13	Licenc e 14
L I C E N S I N G  I N	Licence 1	Red	Orange	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 2	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 3	Green	Green	Grey	Orange	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 4	Red	Green	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 5	Red	Green	Red	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 6	Green	Green	Red	Red	Red	Grey	Red	Red	Red	Red	Red	Red	Red	Red
	Licence 7	Red	Red	Red	Red	Red	Green	Grey	Green	Green	Red	Green	Red	Red	Red
	Licence 8	Red	Red	Red	Red	Red	Green	Green	Grey	Red	Red	Red	Red	Red	Red
	Licence 9	Green	Green	Green	Green	Green	Green	Green	Green	Grey	Green	Green	Green	Green	Green
	Licence 10	Red	Red	Red	Red	Red	Red	Red	Red	Red	Grey	Red	Red	Red	Red
	Licence 11	Red	Green	Red	Green	Green	Green	Green	Green	Green	Green	Grey	Red	Green	Green
	Licence 12	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Grey	Red	Red
	Licence 13	Green	Red	Red	Red	Red	Green	Red	Red	Red	Red	Red	Red	Grey	Green
	Licence 14	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Grey

## 4. Identification des problèmes et évaluation des risques

Analyse des situations  
(LS1,LS2) ;  $\Delta$  (LS1,LS2)

LS1 :  
Analyse du statut perçu

$\Delta$  (LS1,LS2) :  
Analyse des différences

LS2 :  
Analyse du statut déterminé  
/mesuré par l'(es) outil(s)

### Identification de problème / évaluation des risques

Questions sur les auteurs

Questions sur les propriétaires

Autres questions  
(i.e. composant en doublon)

Questions sur le domaine public

Intégrité du texte de licence attachée au composant /questions relatives aux modifications

Composant sans licence  
Questions sur les en-têtes

Questions Technico - juridiques  
(i.e. étude des liens statiques/dynamiques, etc...)

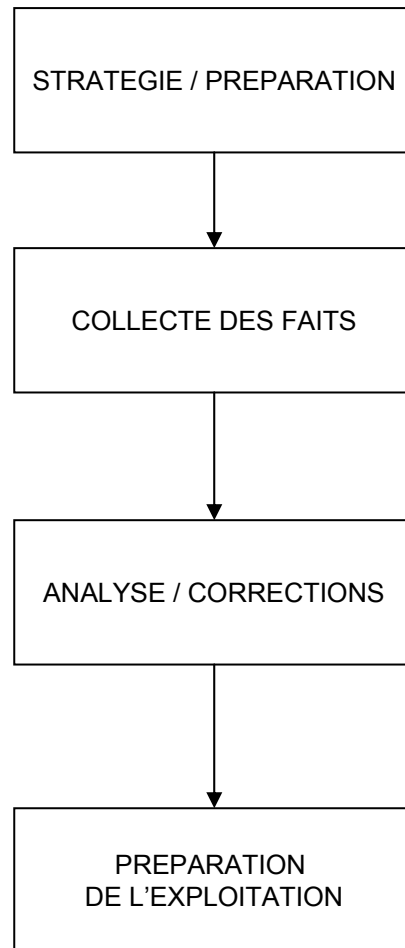
Questions de compatibilité des licences des composants (ascendante et descendante)

Questions sur d'autres obligations liées aux composants (i.e. : citation, etc...)

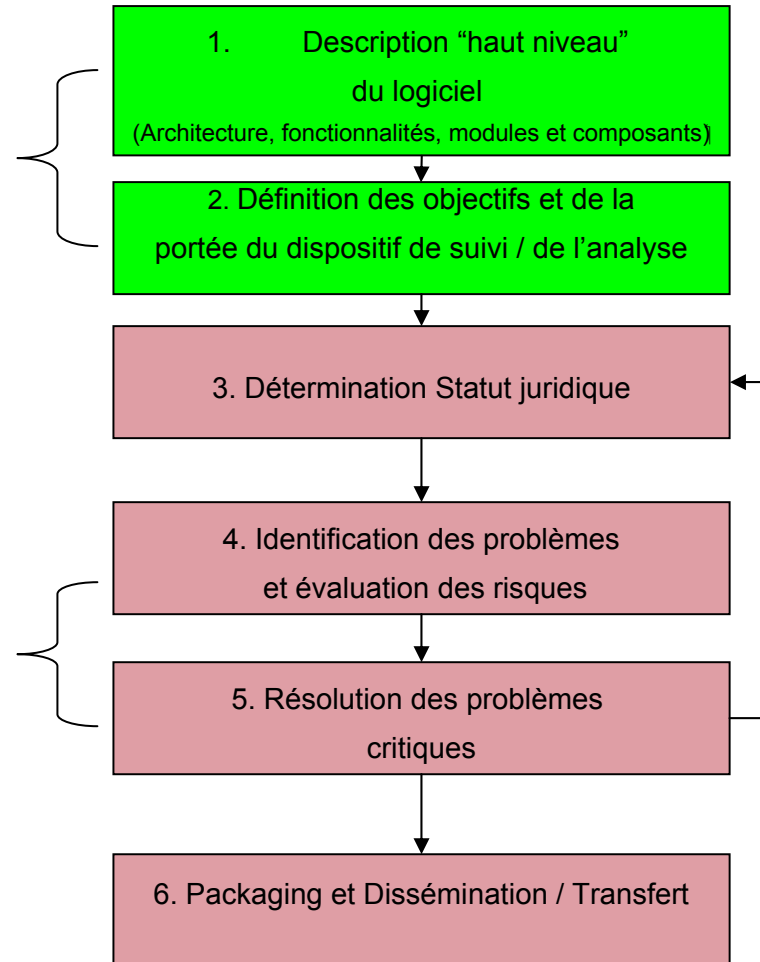
Vers 5. Résolution de problème(s) critique(s)

6. Assurance, Dissémination et traçabilité des IPR

# Méthodologie d'IPRA



## Phases :



## 5. Résolution de problème(s) critique(s)

### Résolution de problème par l'équipe de développement

Re-développement  
d'un composant  
de fonctionnalités  
équivalentes

Substitution  
du composant  
par un composant  
de fonctionnalités  
équivalentes et  
de statut compatible

élimination  
du composant

### Résolution de problème par l'équipe juridique

**Double - licensing**  
Négociation d'un  
autre régime  
de licence concernant  
un composant critique  
qui soit compatible  
avec le schéma de  
développement  
et de licensing de l'INRIA

**acquisition d'IPR**  
(Licensing in)

**Notification  
de situations non résolues**  
à l'équipe de développement

*Vers 6. (Assurance), Dissémination et traçabilité des IPR*

# Synthèse de l'analyse

Sous forme de tableau de bord



Feuille de calcul  
Microsoft Excel



# Les acteurs de la méthodologie



# L'organisation en mode projet

L'analyse d'un logiciel devant tenir compte d'aspects relevant à la fois du domaine scientifique/technique, juridique et de celui du transfert, il est important qu'il soit mené par une **task force** composée de personnes relevant de ces différents profils.

Il s'agit typiquement d'une task force composée :

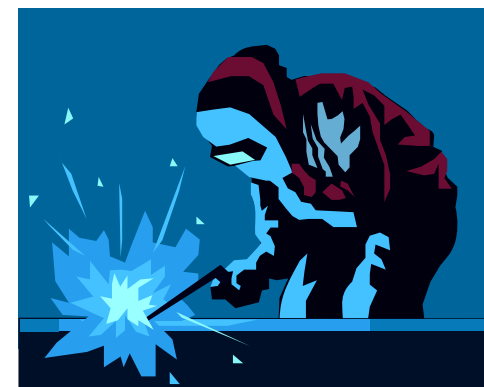
- d'un membre de l'Equipe, éventuellement supporté par le SED (Service Etude et Développement)
- d'un CPPI (chargé de valorisation)
- d'un juriste

Le chef de projet, au sein de cette task force est le CPPI.





# Les principes de base



## Expertise de chaque acteur

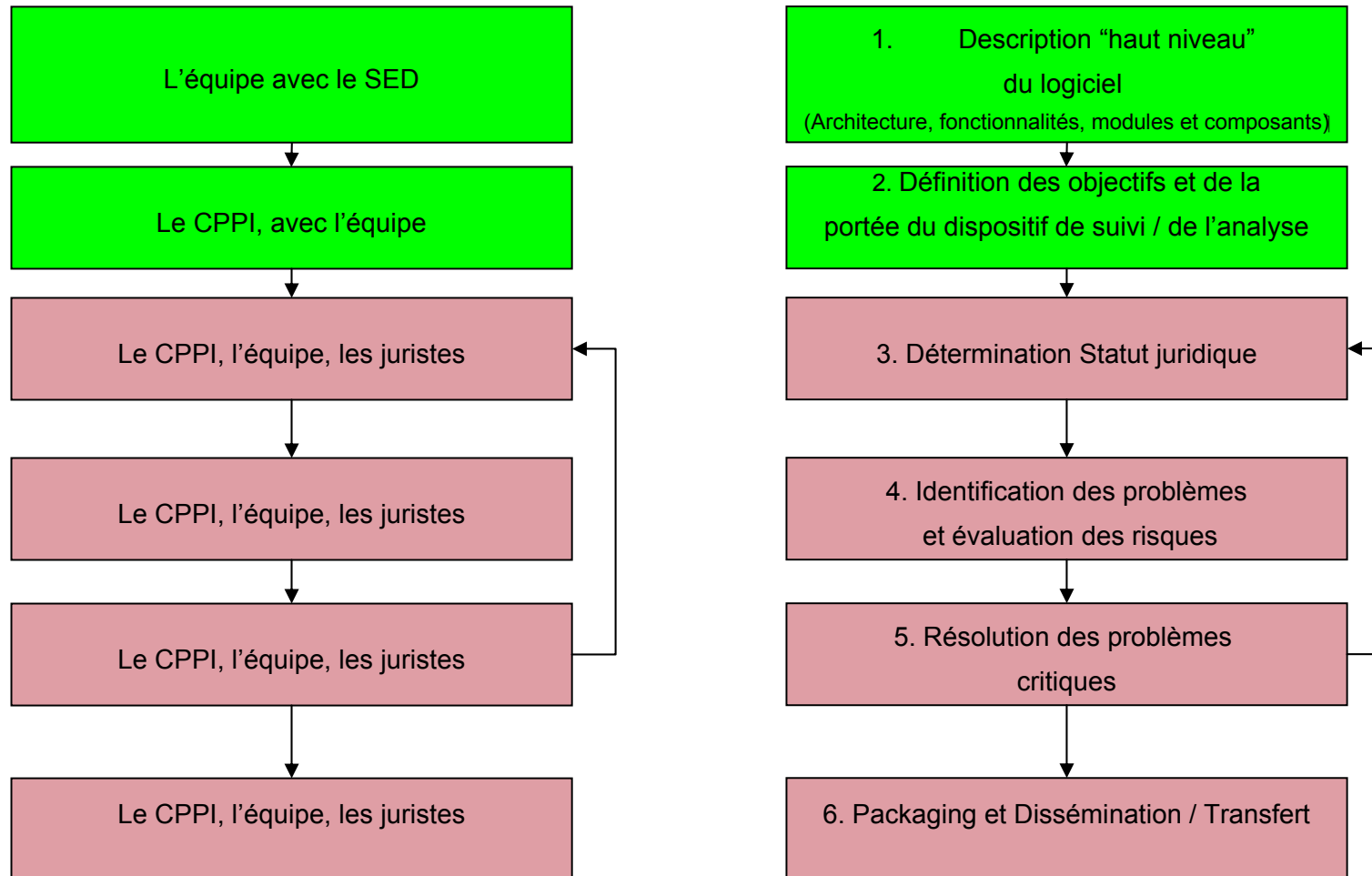
Il est important que le chercheur / ingénieur ait une connaissance de la problématique des licences et des problèmes de compatibilité.

Cependant, les analyses de compatibilité dans le détail requièrent la compétence des juristes formés pour cela.

- Notamment, il faut faire très attention aux clauses particulières des licences.



# Les acteurs par phases



# Outillage de la méthodologie



# Pourquoi des outils

La méthodologie présentée peut être mise en œuvre sans outil

Mais cela peut être très long et fastidieux

- Par exemple, l'analyse des licences sur un gros logiciel développé sur plus de 15 ans a nécessité le travail d'une personne à plein temps pendant 6 mois

Ainsi, l'objectif est d'outiller la méthodologie:

- Pour gagner de la productivité
- Pour faire, à terme, l'analyse de TOUS les logiciels qui « sortent » de l'INRIA



# Le cahier des charges idéal

Identifier les auteurs et les propriétaires de droits d'auteur  
(organismes employeur)

- Définition de part de propriété dans le cadre de projet collaboratif et / ou bilatéral

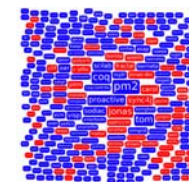
Aide à l'analyse de compatibilité de licence



# Premières expériences en 2009 –2010

## Avec les outils ANTELINK et FOSSOLOGY

- ANTELINK pour les Auteurs
- FOSSOLOGY pour les Licences



## Autres outils indispensables

- Base des dépôts APP
  - Trace les dépôts APP des logiciels
- Outil de Gestion
  - Trace les contrats de recherche, transfert, ... autour des logiciels



# Les bonnes pratiques



# Bonnes pratiques sur la prise en compte de l'IPR

L'objectif est d'insuffler dès le démarrage d'un logiciel ces bonnes pratiques afin d'avoir la situation juridique la plus saine possible au moment où une opération de transfert technologique engageant le patrimoine de l'INRIA est envisagée.

Les bonnes pratiques concernent tous les acteurs

- développement: chercheur/ingénieur
- gestion de projet; responsable d'équipe (ou de projets)
- suivi de contrat; juriste et CPPI (chargé de valorisation)

Les bonnes pratiques pour la prise en compte de l'IPR dans le développement sont issues d'un Groupe de Travail interne





# Les bonnes pratiques de développement

3 phases:

## 1) Lors de la phase initiale de développement

- 1.1) Codage
- 1.2) Utilisation de la Forge
- 1.3) Entête
- 1.4) Sur la propriété du logiciel ?
- 1.5) Dépôt APP (Agence de Protection des Programmes)
- 1.6) Envisager une licence :

## 2) Lors de la diffusion ou transfert du logiciel

- 2.1) Décision de diffusion
- 2.2) Situation juridique du logiciel
- 2.3) Choix de la licence
- 2.4) Packaging et redistribution
- 2.5) Autres droits opposables

## 3) Lors des évolutions du logiciel

- 3.1) Codage
- 3.2) Changement de licence
- 3.3) Entêtes
- 3.4) Gouvernance
- 3.5) Dépôt APP
- 4) Que faire en cas de doute ?



# Conclusion



# Sensibilisation des acteurs

Il devient nécessaire de prendre en compte l'aspect juridique tout au long de la création d'un logiciel libre pour éviter tout risque pouvant devenir paralysant et rendant le logiciel inexploitable.

L'INRIA s'engage à participer à des actions de sensibilisation, à diffuser les méthodologies internes, à étudier les outils répondant aux défis de logiciel libre de demain

- Notamment, via la création de l'IRILL (Initiative pour la Recherche et l'Innovation sur le Logiciel Libre)



# Mise en place du label « **logiciel qualifié pour le transfert@INRIA** »

Une fois les problèmes identifiés et éventuellement résolus, le label « Logiciel Qualifié pour le Transfert@INRIA » sera attribué au logiciel, ceci dans le cadre d'un schéma de transfert précis.

Ce label n'engage pas la responsabilité de l'INRIA et ne constitue pas une garantie, mais doit être considéré comme un effort de moyens et une preuve de bonne volonté et de professionnalisme de l'INRIA vis-à-vis de son (ses) partenaire(s), en vue d'une réduction du risque juridique.



# Pour en savoir plus

## Open Source

- « Stratégie de l'INRIA sur le logiciel libre » du 14 juin 2009, <http://www.inria.fr/valorisation/Strategie-inria-logiciel-libre.pdf>
- Guide d'approche et d'analyse des licences de logiciels libres V2.0, de Magali Fitzgibbon, Sylvain Steer - [http://www.inria.fr/valorisation/logiciels/INRIA\\_guide\\_analyse\\_licences\\_libres\\_vf.pdf](http://www.inria.fr/valorisation/logiciels/INRIA_guide_analyse_licences_libres_vf.pdf)
- Recueil de fiches explicatives de licences libres, V2.0 », de Magali Fitzgibbon, Sylvain Steer - [http://www.inria.fr/valorisation/logiciels/INRIA\\_recueil\\_fiches\\_licences\\_libres\\_vf.pdf](http://www.inria.fr/valorisation/logiciels/INRIA_recueil_fiches_licences_libres_vf.pdf)

## IPRA

- Qualipso, Report on the proposed IPR tracking methodology, du 16/12/2009 de Luc Gâteau, Magali Fitzgibbon, Guillaume Rousseau, Stéphane Dalmas - <http://www.inria.fr/valorisation/methodologie%20IPR%20A1.D2.1.4.pdf>



# Avez-vous des questions ?

Merci de votre attention !

