

## Décryptage : le CIR dans le domaine de l'informatique

En **informatique**, la frontière entre **R&D et innovation** est parfois floue.

Pour qualifier un projet de R&D, l'administration fiscale s'appuie sur les **5 critères** définis par le manuel de Frascati.

Ainsi, un projet de R&D doit comporter un élément :

de nouveauté

de créativité

d'incertitude

Mais aussi être :

systematique

transférable et/ou reproductible

Par ailleurs, le guide du CIR 2025, publié par le Ministère en charge de la Recherche, définit **4 grandes catégories** qui permettent de caractériser les travaux de R&D en informatique dans le cadre du CIR.

Nous détaillons ces catégories ci-après et les illustrons à travers des exemples fictifs afin d'en clarifier les contours.

En complément, une infographie en fin de document revient sur les principales idées reçues concernant la R&D informatique et le CIR.

Catégories	Création ou amélioration d'une méthode, d'un algorithme ou d'un protocole	Formalisation ou adaptation d'une méthodologie originale	Application innovante de concepts scientifiques récents ou émergents	Mise en place d'expérimentations ou de prototypes
<p><b>Définition (selon guide du CIR 2025)</b></p>	<p>Il s'agit de produire « un élément technique nouveau (...) avec une documentation détaillant les verrous identifiés, les principes de fonctionnement, les étapes de mise au point, les solutions développées et les résultats obtenus ».</p>	<p>Cela concerne « la formalisation d'un processus global de travail (...) ou l'adaptation substantielle d'une méthodologie existante à un cas d'étude spécifique », avec un cadre reproductible et évaluable.</p>	<p>Mise en œuvre « de concepts récents à la pointe de l'état de l'art (IA, calcul quantique, cryptographie avancée...) dans un contexte nouveau ou contraint », incluant une validation rigoureuse.</p>	<p>Développement de prototypes utilisés pour « tester et mesurer des avancées scientifiques ou techniques », avec protocoles expérimentaux définis et documentation des résultats.</p>
<p><b>Exemples*</b></p>	<p><b>Algorithme de navigation temps réel pour robots autonomes.</b></p>	<p><b>Méthodologie de tests reproductibles pour comportements non déterministes.</b></p>	<p><b>IA émergente pour la détection d'obstacles complexes.</b></p>	<p><b>Prototype de coordination multi-robots pour charge lourde.</b></p>
<p><b>Problématique</b></p>	<p>Lorsque plusieurs robots circulent en même temps, les algorithmes connus provoquent des congestions ou des recalculs trop lents, empêchant une circulation fluide.</p>	<p>Les robots n'adoptent pas toujours le même comportement selon l'ordre d'arrivée des signaux (messages réseau, capteurs). Les outils de test classiques ne permettent pas de rejouer les scénarios identiques.</p>	<p>Les modèles de vision classiques échouent à détecter des objets : cartons déformés, palettes abîmées, matériaux semi-transparents.</p>	<p>Certaines charges doivent être transportées à plusieurs robots, mais on ne sait pas si une synchronisation fiable est possible à vitesse réelle.</p>
<p><b>Verrous</b></p>	<p>Aucune méthode existante n'assure un recalcul quasi immédiat tout en anticipant les mouvements d'autres robots.</p>	<p>Absence de méthode reproductible pour valider le comportement d'un système multi-robots dans des conditions instables.</p>	<p>Les nouvelles architectures (ex : modèles de diffusion, vision Transformers) n'ont jamais été appliquées à la détection d'objets industriels dans un entrepôt réel. Leur pertinence est inconnue.</p>	<p>Incertitude sur la synchronisation des moteurs, la communication temps réel inter robots, la stabilité de la trajectoire partagée.</p>
<p><b>Travaux de R&amp;D réalisés</b></p>	<ul style="list-style-type: none"> <li>Conception d'un nouvel algorithme hybride combinant graphes probabilistes et prédiction de trajectoires.</li> <li>Mise au point progressive : tuning, simulations de trafic dense, comparaison objective aux méthodes existantes.</li> <li>Documentation des expérimentations, résultats intermédiaires et limites (critères : systématicité + reproductibilité).</li> </ul>	<ul style="list-style-type: none"> <li>Création d'un cadre méthodologique complet : <ul style="list-style-type: none"> <li>○ génération de scénarios contrôlés (latence, pertes, collisions de messages),</li> <li>○ capture des événements dans un "journal causal",</li> <li>○ protocole pour rejouer exactement la séquence d'événements.</li> </ul> </li> <li>Définition de critères d'évaluation (écart de trajectoire, cohérence de décision...).</li> </ul>	<ul style="list-style-type: none"> <li>Adaptation d'un modèle émergent à un flux vidéo embarqué.</li> <li>Création d'un protocole d'entraînement et d'évaluation inédit (conditions de lumière, objets partiellement cachés...).</li> <li>Comparaison rigoureuse avec les approches traditionnelles.</li> </ul>	<ul style="list-style-type: none"> <li>Développement d'un prototype expérimental qui simule différents modes de coopération.</li> <li>Tests systématiques (vitesse, latence, perte de messages, déséquilibres mécaniques).</li> <li>Analyse des résultats pour confirmer l'hypothèse : "Deux robots peuvent maintenir une trajectoire commune stable".</li> </ul>
<p><b>Nouvelles connaissances</b></p>	<p>Un protocole de gestion dynamique des trajectoires non documenté auparavant et applicable à d'autres contextes robotisés.</p>	<p>Une méthode généralisable pour tester des systèmes distribués non déterministes.</p>	<p>Validation (ou non) de l'intérêt d'un modèle d'IA d'avant-garde dans un contexte où il n'avait jamais été testé, ce qui correspond à cette catégorie.</p>	<p>Caractérisation précise des limites et conditions nécessaires à la coordination multi-robots → un résultat scientifique réutilisable.</p>

\*Exemples fictifs qui permettent d'illustrer le type de travaux en R&D informatique.

**Nouveauté vs. R&D :**

Nouveau ne suffit pas : il faut résoudre un **vrai défi scientifique/technique**.

01

**Complexité vs. R&D :**

Complexe  $\neq$  éligible : il faut une approche **originale** et réutilisable.

02

**Informatique comme outil :**

Résoudre un problème métier avec de l'informatique  $\neq$  R&D ; il faut isoler la vraie R&D informatique.

08

**Acquisition de compétences vs. Création de connaissances**

Se former n'est pas de la R&D, il faut **créer de la connaissance nouvelle**.

03

**Cycles de développement :**

Les cycles Agile/corrections ne sont pas de la R&D.

07

**8 idées reçues sur la R&D en informatique****Exploration et intégration de technologies émergentes :**

Éligible seulement s'il y a un problème technique nouveau à résoudre.

04

**Méthodologie de développement conçue ou adaptée :**

Éligible uniquement si une vraie démarche de R&D est démontrée.

06

**Prototypage :**

Un prototype n'est éligible que s'il **valide un concept de R&D**.

05