



*Tout ou partie des compétences ou des capacités peut faire l'objet de négociation avec un partenaire en vue de leur utilisation ou intégration dans ses services ou produits.*

Le métier de mezzonomy est de concevoir, réaliser, déployer et maintenir des solutions logicielles dans des écosystèmes *règlementés* – ou *exigeants* – et dédié à la résolution **coopérative** de problèmes **prospectifs**, concernant des produits, des dispositifs ou des systèmes qui n'existent pas encore physiquement.

Dans ce contexte, nous développons une méthodologie fondée sur la notion de réseaux d'apprentissage. Basée sur un modèle mathématique et disposant de ressorts graphiques éprouvés, cette méthode est à l'origine adaptée à la prise de note lors de l'expression du besoin par le client, besoins très volatiles dans ce contexte.

**Compétence** : consigner finement l'expression du besoin par l'usage de réseaux d'apprentissage.

La notion clé, l'apprentissage, est la valeur confiée à la machine : cette notion permet de ne plus faire de différence de nature entre la programmation par l'utilisateur (formules, macros, ...) et la programmation par un tiers spécialisé (SI, SSII, ...). Cette notion permet de mettre tous les acteurs autour de la table sans mettre des distinctions entre interlocuteurs métier et informatiques.

**Compétence** : établir des situations de coopérations sereines entre des acteurs d'un projet.

La méthodologie différencie trois niveaux de service dans les apprentissages :

- **satisfaction** : les fonctions satisfont les besoins
- **proposition** : les applications proposent des fonctions
- **articulation** : les notions articulent des applications

**Compétence** : satisfaire le besoin en nous concentrant sur l'articulation et déléguant la proposition au client

Pour accroître sa notoriété et disposer d'une capacité différenciante, mezzonomy développe une plateforme ouverte au calcul scientifique, technique et industriel appelé Mirza. Cette plateforme propose les capacités suivantes :

- **Traitement des matrices, dessins de courbes et de volumes** : le noyau des plateformes de calcul libres existantes qu'elles soient grand public comme Gnumeric ou plus spécialisées comme Scilab, nous proposons un mécanisme tactile de programmation sans variable capable de trouver un plus large public que des programmes impératifs.

**Capacité** : apprentissage tactile sans variable

- **Gestion des systèmes d'unités et des langues** : nécessaire pour envisager la coopération dans des écosystèmes continentaux comme AIRBUS, cette capacité pose la différence entre le reflet (la vue) et l'apprentissage (le modèle) sur une base intuitive.
- **Sécurisation des savoir-faire** : la principale capacité différenciante de Mirza est que l'apprentissage n'est pas dupliqué seuls les reflets sont transmis. Par exemple, un utilisateur définit un formulaire permettant de dimensionner une poutre, il peut publier ce formulaire en étant sûr que tous les calculs ne seront réalisés que sur sa machine.

**Capacité** : supprimer la duplication pour restaurer la propriété

- **Gestion des espaces de conceptions** : comme tous les reflets d'un apprentissage sont connus et mis à jour par la machine hôte de l'apprentissage, il est possible de travailler sur l'ensemble des reflets. Ce type de gestion, appelé gestion des espaces de conceptions, peut se concevoir comme gestion de la non-régression, mais aussi par la définition de domaine de linéarité permettant de limiter l'usage de modèles lourds.

**Capacité** : manipuler l'ensemble des usages d'une fonction

- **Administration des réseaux d'apprentissages** : dans le cadre d'une application professionnelle il est nécessaire de pouvoir inspecter, répartir, archiver et certifier les réseaux d'apprentissages implémentés en Mirza.

**Capacité** : utiliser un réseau d'apprentissage intégré