QUICK REFERENCE CARD

APPROCHE EROOM :

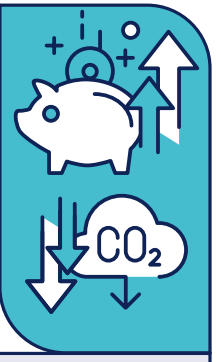
réduire massivement l'empreinte environnementale du numérique tout en continuant à innover, grâce à l'optimisation du S.I.

Canalisé avec ♥ par Tristan Nitot, Frédéric Lenci, Amandine Capelle et les contributions des groupes EROOM d'OCTO et de Boavitza.

THERE IS A BETTER WAY

EROOM : innovez plus, consommez moins !

"Faire plus avec la même chose et pour moins cher" est devenu plus que jamais le mot d'ordre. La loi de Moore nous a longtemps offert des machines toujours plus puissantes, mais l'urgence métier a souvent favorisé la rapidité de développement au détriment de la qualité du code. Aujourd'hui, avec des budgets contraints et la nécessité de réduire l'empreinte environnementale du numérique, il faut optimiser l'existant. La démarche EROOM apporte une solution.



EROOM (inverse de la loi de MOORE) part du constat que la puissance accrue des machines a poussé au développement de logiciels toujours plus gourmands, perpétuant un cycle de renouvellement matériel. Or, une partie du code actuel est vraiment sub-optimale et gaspille des ressources. L'optimiser permettrait d'accélérer les systèmes et de limiter le remplacement du matériel, dont la fabrication représente l'essentiel de l'impact environnemental du numérique.

Conditions nécessaires :

- **Des experts capables de refactorer le code et maîtrisant les technologies utilisées ;**
- **Un accès au code source, sans lequel l'optimisation est difficile.**

Repérer les bouchons.

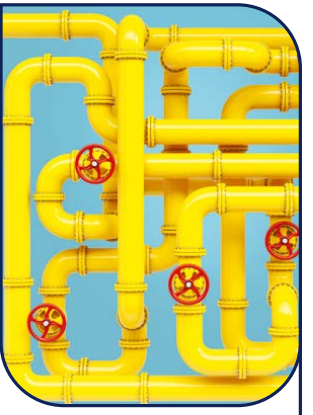
Une petite partie du S.I. est très sub-optimale. Trouver ces quelques pour cent, les corriger, les optimiser, pour faire cesser le gaspillage, permet parfois d'aller 2, 10, 100 ou 1000 fois plus vite qu'avant. Si en moyenne, en corrigeant ces points durs, on arrive à faire tourner le logiciel 2 fois plus vite tous les deux ans, on revient à l'époque de la loi de Moore, et on dispose de deux fois plus de puissance tous les 2 ans, mais sans changer le matériel. Or, c'est la fabrication du matériel qui forme l'essentiel de l'empreinte environnementale du numérique. Dans une démarche EROOM, on fait durer le matériel tout en libérant des ressources informatiques. On fait faire du travail de meilleure qualité aux développeurs tout en réduisant l'impact environnemental du numérique.

Le logiciel d'un système d'information est écrit par des humains (pour l'instant ?) et donc il est de qualité inégale. Par ailleurs, il existe dans un contexte qui évolue (nouvelles technologies qui arrivent, d'autres qui deviennent obsolètes, alors que les besoins métiers et la volumétrie des données évoluent sans cesse).

Qu'est-ce qu'un bouchon ?

Un bouchon (métaphorique), c'est un petit morceau d'une application qui est très sub-optimal. Il consomme 10, 100 ou 1000 fois plus de ressources informatiques que nécessaire. Il est possible de l'optimiser et donc de réduire le gaspillage qu'il engendre, donc de faire fonctionner l'application beaucoup plus rapidement et ainsi libérer de la ressource informatique qui pourra être allouée à d'autres usages. Avec cette ressource libérée, la DSI peut continuer d'innover tout en conservant un périmètre matériel constant.

Faire le plan des canalisations.



Chaque système d'information, chaque application, porte la marque de l'organisation qui l'a conçue, et le patrimoine informatique est le reflet de son histoire. La plupart du temps, de nombreux plombiers se sont succédé en y mettant chacun leurs outils, leurs méthodes et leur tuyauterie spécifique. Les architectures témoignent des choix (make, buy, use), des pratiques de développement internes ou externes qui ont construit ou intégré ces applications. À chacun son réseau de canalisations.

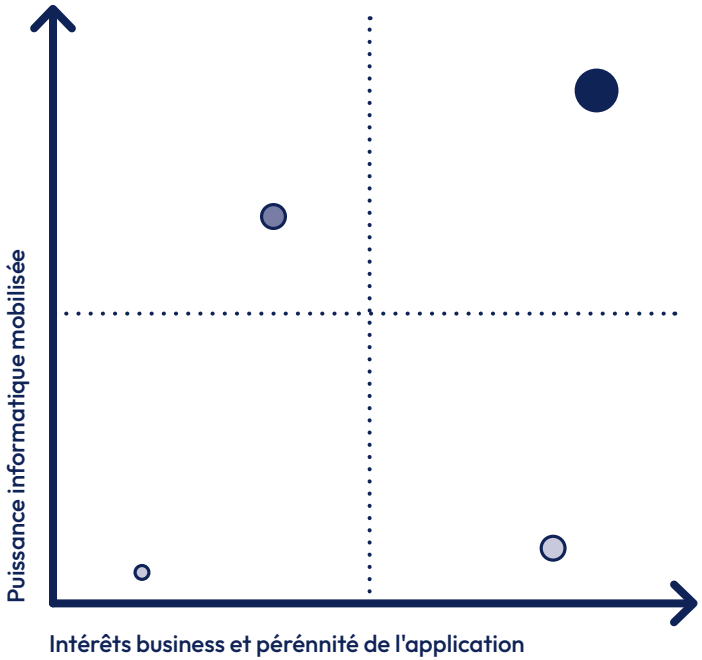
La première étape d'une démarche EROOM consiste à pouvoir se repérer dans cette complexité : comprendre les différents réseaux de tuyaux, les différentes méthodes d'assemblage et où se situent les fuites ou les bouchons, qui nuisent à la fluidité du système d'information, et sont totalement sous-optimisés.

Cet état des lieux est un pré-requis nécessaire pour savoir où et comment procéder l'optimisation logicielle d'EROOM. Et il dépend d'abord de la maturité de l'organisation vis-à-vis de sa connaissance de la réalité de son patrimoine applicatif. Selon nous, le monde se divise en deux catégories : ceux qui ont un système observable et monitoré, et les autres. Intéressons-nous d'abord aux autres.

PLASMA.

Portfolio-Led Application Sustainability Management

Sur quelles applications concentrer nos efforts ?



Taille de l'application

100 10 1

Capacité d'action

Forte Moyenne Faible

Comprendre le cadran PLASMA

Avec la méthodologie PLASMA, nous pouvons cartographier sur un cadran les différentes applications du SI, selon deux axes principaux :

Intérêt business et pérennité de l'application, en abscisse.

Là, nous apprécions les applications selon leur position dans leur cycle de vie, l'usage supposé ou réel par les métiers, leur criticité opérationnelle, leurs dépendances avec d'autres applications métiers…

Puissance informatique mobilisée, en ordonnée.

Ici nous établissons un score en fonction de paramètres tels que : facture fournisseur Cloud, nombre de machines utilisées, puissance requise sur les postes utilisateurs, volumétrie de données, données d'usage…

PLASMA met aussi en relief la taille estimée de l'application (pour faire simple, le nombre de lignes de code) qui est en soi un indicateur et un code couleur qui permet de caractériser visuellement la mesure dans laquelle l'optimisation de l'application est à la main des équipes techniques - c'est la **capacité d'action**. Cette caractéristique est importante notamment dans le cas des organisations qui ont plus massivement recours à du buy qu'à du make, et permet de tenir compte du fait que les leviers d'action sont plus faibles sur des solutions de marché que sur des développements à façon.

Au terme de cette analyse, la vision mise à jour de la matrice permet de matérialiser la priorisation préconisée pour déterminer les applications qui sont les candidates naturelles à l'optimisation.

Mesure et observabilité.

“On ne peut changer que ce que l’on mesure”, dit-on souvent. L’approche EROOM n’échappe pas à la règle. Pourtant, les organisations n’ont pas toutes le même niveau de maturité dans le domaine.

Là où elle est disponible, la donnée est utile pour positionner de façon plus précise chaque application dans le cadran PLASMA, en complément d’interviews des équipes.

Dans un second temps, celui de l’optimisation des applications dans leurs cinq dimensions, un minimum de données est nécessaire pour mesurer les progrès au fil de l’itération. Pour cela, une mise en place minimale d’un système de mesure léger est requise à base d’outils open-source et gratuits, par exemple Prometheus associé à Grafana.

EROOM, delivery et IA.

Comment intégrer EROOM dans un delivery ?

L'approche EROOM est fondamentalement une approche a posteriori, qui fonctionne sur un existant qu'on va revisiter et optimiser, pour libérer de la ressource. Si on voulait, a contrario, avoir une approche comparable, mais dès le début du projet, il faudrait se rapprocher de l'éco-conception, qui vise — avant l'écriture de l'application — à voir comment faire pour qu'elle soit la plus sobre possible.

En bref, les deux méthodes ont des objectifs comparables, l'éco-conception en amont du déploiement, EROOM en aval.

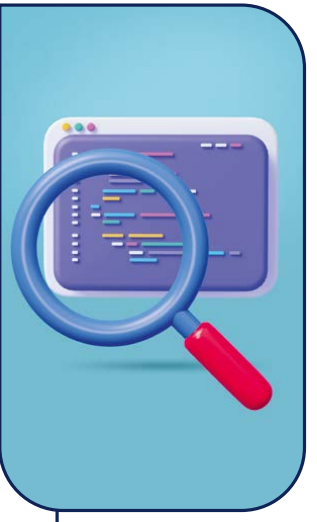
Et l'IA dans tout ça ?

L'IA peut tout chambouler au niveau du numérique. En restant optimiste, il est possible d'envisager deux grands usages de l'IA dans une approche EROOM. La première consisterait à utiliser l'IA pour repérer quelles parties du portefeuille applicatif peuvent être les plus importantes à optimiser, voire à supprimer.

Le second usage de l'IA pourrait consister à proposer des solutions d'optimisation, en modifiant le code, en changeant l'algorithme, en repensant l'architecture et en facilitant la réécriture du code.


les 5 dimensions
de l’optimisation.

La qualité du code.



Le code est écrit par des humains, et donc de qualité variable. Une mauvaise décision ou un développeur inexpérimenté peuvent mener à des performances très dégradées.

Il convient de repérer quelles parties du code provoquent des ralentissements avant de les corriger pour réduire la consommation de ressources.

OUTILS

Le profilage de code est un grand classique qui permet de déterminer quelles parties du code consomment le plus de ressources (CPU, mémoire, voire GPU). Les outils dépendent du langage utilisé. Par exemple, pour Python, il existe l'outil Scalene.

Le choix de l'algorithme.

Parfois, même si le code est de haute qualité, c'est la façon d'aborder le problème d'un point de vue logique qui peut être en cause. C'est particulièrement sensible quand la volumétrie de données augmente. Certains algorithmes ne dépendent pas du nombre de traitements, d'autres au contraire augmentent proportionnellement ou de façon exponentielle. Un algorithme pour de petits volumes peut ne plus convenir quand la volumétrie augmente.

OUTILS

Si l'application contient une partie algorithmique, alors il faut la réévaluer. Dans de nombreux cas, les bibliothèques logicielles gèrent l'aspect algorithmique. Dans ce cas, il convient de revisiter les choix de bibliothèques pour en choisir de plus adaptées au volume concerné.

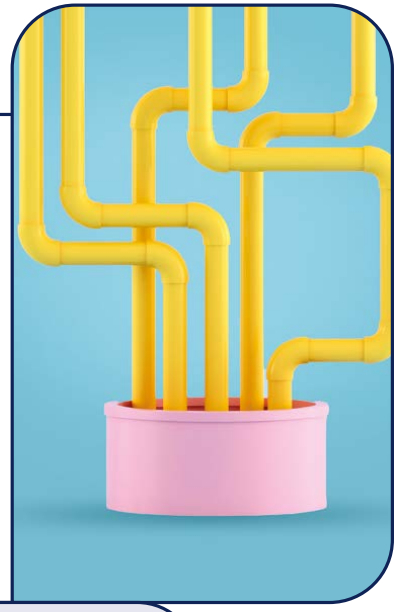
Le stockage des données.

Les bases de données constituent un domaine à part entière en matière de performance. Le stockage se reposant sur mémoire de masse, dont l'accès est très lent par rapport aux performances du CPU et de la RAM, une mauvaise configuration de la base de données peut avoir un impact très important sur la performance du système d'information.

OUTILS

Il existe de nombreuses options. Par exemple, la commande SQL EXPLAIN permet de comprendre comment se déroule une commande. Elle pourra servir à l'analyse des requêtes les plus gourmandes, décider d'éventuelles créations d'index. Il existe par ailleurs de nombreux outils plus perfectionnés permettant par exemple de bien dimensionner le pool de connexions à la base.

Architecture et réseau.

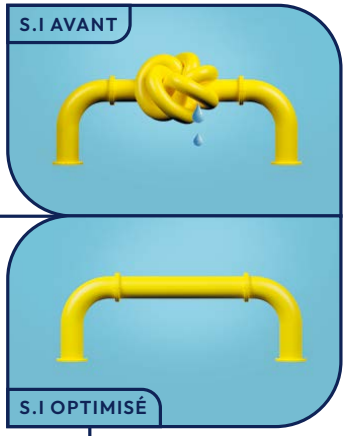


On peut avoir des machines largement dimensionnées, des applications optimisées… et avoir de mauvaises performances ! Cela peut provenir du réseau, ou bien la façon dont le système est agencé, il suffit d'un goulet d'étranglement pour tout ralentir.

OUTILS

L'approche, dans ce cas, doit être systémique, en testant plusieurs hypothèses, après s'être assuré d'avoir les bons outils d'observabilité.

La sobriété fonctionnelle.

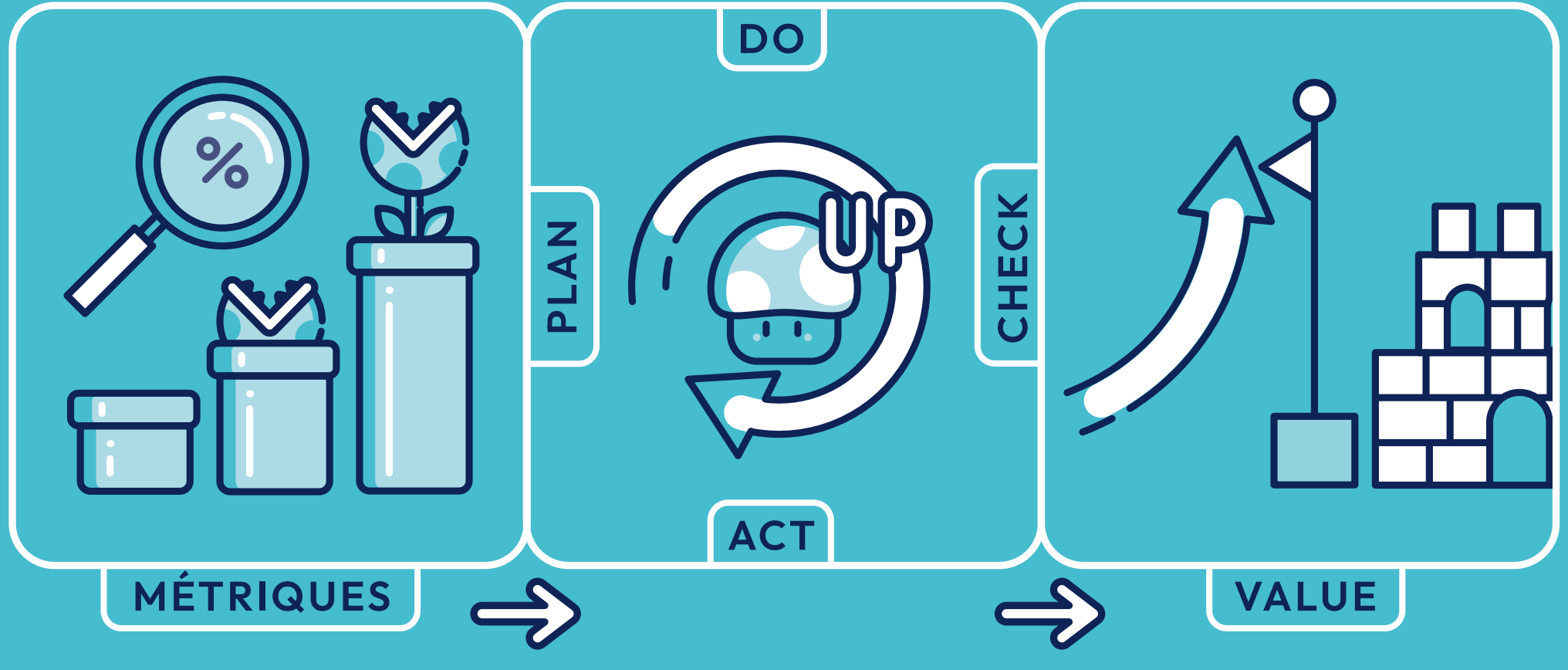


Il s'agit peut-être du point le plus important en termes d'optimisation. Certaines fonctionnalités peuvent être très gourmandes tout en étant peu utilisées. Pire, elles peuvent complexifier le système d'information sans apporter de bénéfice métier, parfois parce que le même service sera fourni par une autre application, plus récente.

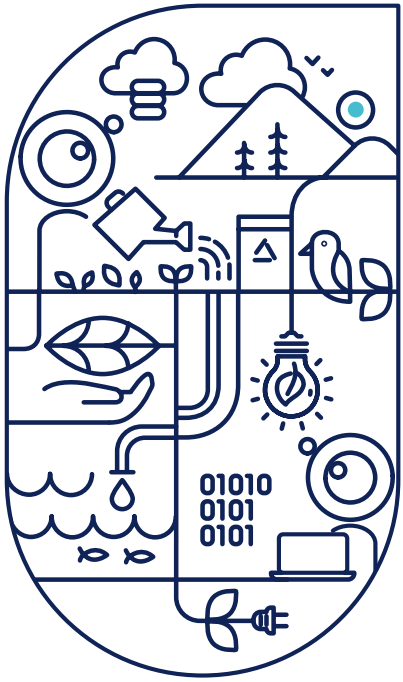
OUTILS

Identifier les fonctionnalités les plus gourmandes et moins utiles et décider de leur suppression. On pourra aussi envisager de ne pas fonctionner en temps réel, ce qui serait acceptable dans beaucoup de cas et permettrait de faire un S.I. beaucoup plus performant à ressources égales, si on accepte une dégradation de fonctionnalité. Ou bien on peut aller plus loin et supprimer des fonctionnalités, à condition, dans tous les cas, d'impliquer les équipes métier.

QUICK REFERENCE CARD



contact@octo.com



Dans un **MONDE** complexe aux **RESSOURCES** finies, nous recherchons ensemble de meilleures façons d'**AGIR**. Nous **ŒUVRONS** à concevoir et à réaliser les produits numériques **ESSENTIELS** au **PROGRÈS** de nos clients et à l'émergence d'écosystèmes **VERTUEUX**.

Cette refcard est imprimée sur du papier certifié PEFC.

Avec le soutien ♥ des power-ups
Caroline Bretagne et Charlotte Petitbon.

OCTO TECHNOLOGY
Part of Accenture