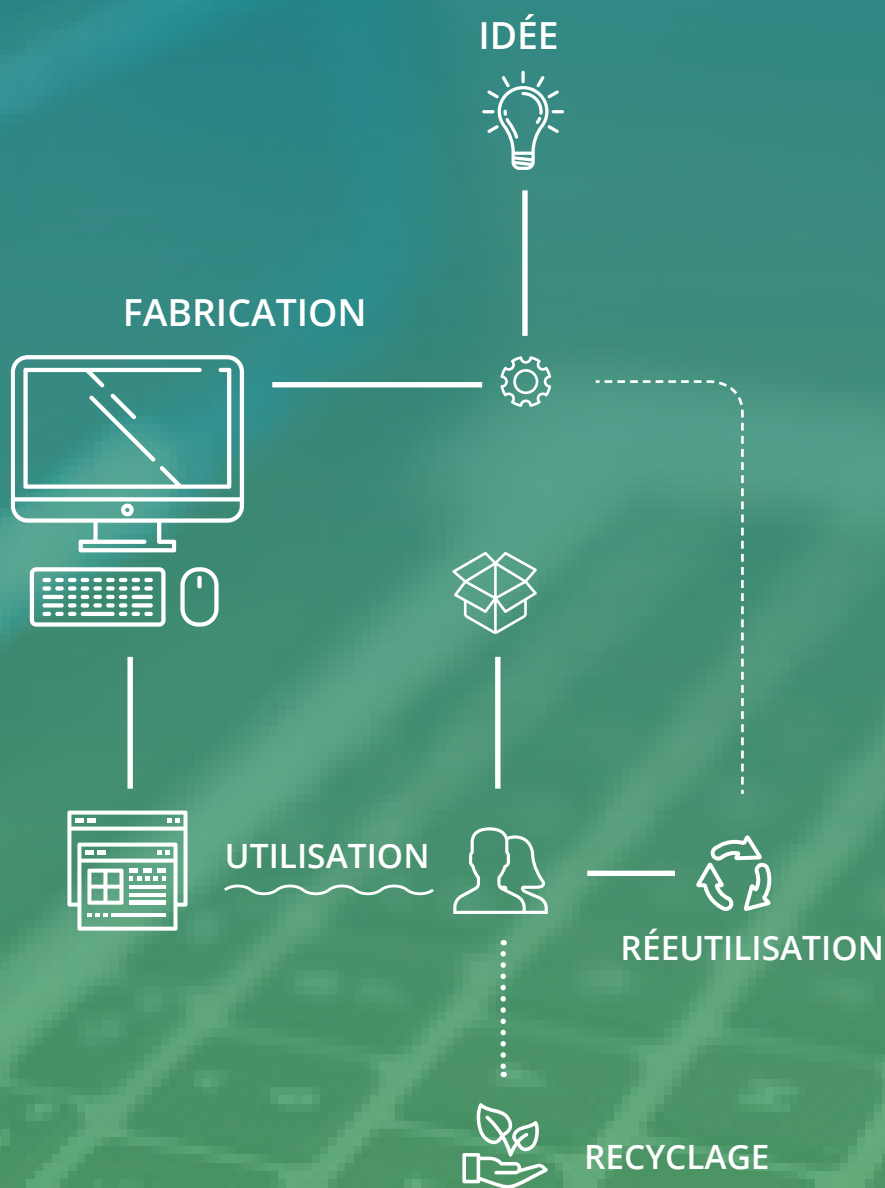


# GUIDE MÉTHODOLOGIQUE



## ACV des logiciels



GREENSPECTOR





# MERCI POUR VOTRE TÉLÉCHARGEMENT !

Ne manquez pas nos autres ressources sur  
l'écoconception des logiciels :

[OUI, JE VEUX EN SAVOIR PLUS !](#)

+33 (0) 9 51 44 55 79  
[contact@greenspector.com](mailto:contact@greenspector.com)



GREENSPECTOR

# I Pourquoi réaliser une ACV des logiciels ?

L'écoconception, qui consiste à tenir compte des impacts environnementaux et sanitaires lors de la conception ou l'amélioration d'un produit (bien ou service), s'impose progressivement dans tous les secteurs économiques comme une démarche créatrice de valeur. Ceci parce que les entreprises sont de plus en plus sensibles à la responsabilité qu'elles ont vis-à-vis de notre planète et des générations futures, mais surtout parce qu'elles prennent conscience des multiples bénéfices qu'elles peuvent tirer de la mise en œuvre d'une telle démarche.

Il est cependant un domaine où l'écoconception n'en est qu'à ses balbutiements : **il s'agit du monde du logiciel, dans lequel la plupart des méthodes et bonnes pratiques en la matière sont encore à inventer.** Pourtant, comme dans tous les autres secteurs économiques, **les avantages que peuvent en retirer les différents acteurs du monde du logiciel sont nombreux :**



## Réduction des coûts

En veillant à réduire les ressources ou matières premières nécessaires à la fabrication d'un produit, l'écoconception permet du même coup de réduire les coûts de fabrication. Cela est bien entendu aussi valable pour un logiciel : dans la phase de production d'un logiciel, réduire le nombre de postes de travail, le nombre d'impressions, la quantité d'énergie ou le nombre de déplacements nécessaires sont autant de moyens de réduire les pollutions engendrées par cette activité mais également de **réduire les coûts de fabrication du logiciel.**



**L'écoconception des logiciels s'impose progressivement comme une démarche créatrice de valeur**



## Anticipation des réglementations environnementales

De plus en plus de normes sont imposées aux entreprises pour rendre les produits, et l'économie en général, plus vertueux sur le plan environnemental. On pense par exemple aux directives qui visent les Equipements Electriques et Electroniques (EEE) RoHS et WEEE, REACH ou ErP visant à rendre les produits moins polluants. Mais également aux tentatives actuelles ou à venir des pouvoirs publics d'intégrer à notre économie les coûts de la dégradation de l'environnement qui ne sont pas aujourd'hui assumés par les entreprises (externalités négatives) : droits d'émission de CO<sub>2</sub>, taxe carbone etc. **Face à l'arrivée de ces nouvelles réglementations, nul doute que les entreprises ayant déjà mûri la problématique de l'écoconception en tireront un avantage concurrentiel.**



## Différenciation du produit

Éco-concevoir, c'est aussi créer un produit de meilleure qualité, plus robuste, plus durable et plus économe pour l'utilisateur, puisque ces qualités sont intimement liées à la réduction de l'impact du produit sur l'environnement et/ou par l'allongement de sa durée de vie active. L'utilisateur y trouve alors son compte. La

consommation d'énergie est par exemple un souci bien réel pour le responsable d'un data center, qui verra ainsi un logiciel moins consommateur d'électricité avec un œil plus favorable, surtout à l'ère où les « opérateurs Cloud » fleurissent et ont intérêt à optimiser l'utilisation de leurs ressources. De même, l'autonomie des plates-formes mobiles est un enjeu essentiel pour les fabricants et utilisateurs de smartphones et tablettes, et le logiciel y est pour quelque chose.



## **Facteur d'innovation**

Le ministère de l'écologie, du développement durable et de l'environnement affirme sur son site web<sup>1</sup> :

*« L'écoconception est un aiguillon pour l'innovation, aussi bien en ce qui concerne la fonction du produit que les différentes étapes de son cycle de vie. Un regard nouveau pour optimiser les consommations (matière et énergie) et pour réduire les pollutions débouche parfois sur des idées entièrement nouvelles sur les composants du produit, son fonctionnement ou les technologies auxquelles il fait appel. ».*

Ce constat est valable aussi bien pour les logiciels que pour tout autre type de produit.



## **Image de l'entreprise**

A une époque où les consommateurs sont de plus en plus attentifs à la responsabilité sociale et environnementale des entreprises, s'engager activement à appliquer les principes d'écoconception bénéficie sans aucun doute à l'image et au prestige de l'entreprise, avec les avantages en termes de retombées économiques.

C'est fort de ces constats qu'un groupe d'experts du « Green IT » a fondé le Green Code Lab dont le but est de promouvoir l'écoconception des logiciels et de proposer des outils et méthodes pour aider à sa mise en œuvre.

Dans le cadre de la collaboration d'Orange avec GREENSPECTOR, lauréat d'un appel à projet ADEME sur l'écoconception des logiciels, les 2 organisations ont apporté leurs expertises respectives pour poursuivre les travaux, objet de ce document.

La méthodologie que nous proposons ici pour réaliser une Analyse de Cycle de Vie (ACV) de logiciel s'inscrit pleinement dans un objectif de définition d'une méthodologie à diffuser largement pour initier de futures démarches d'évaluation des impacts de logiciels.



***L'ACV est un outil incontournable de l'écoconception. Cette méthodologie permet d'évaluer de façon complète les impacts environnementaux de biens manufacturés, de services et procédés.***

L'ACV est en effet un outil central et incontournable de l'écoconception. Il s'agit d'une méthodologie standardisée (ISO14040 et ISO14044 notamment) qui permet d'évaluer les impacts environnementaux de biens manufacturés, services et procédés, ceci de façon globale et complète. Etudier les pollutions générées à toutes les étapes du cycle de vie d'un produit (conception, fabrication, utilisation, fin de vie), permet de n'en oublier aucune et de mettre en évidence la phase du cycle de vie la plus polluante (là où il faudrait porter l'effort prioritaire). Cet effort sera fonction des décisions de l'entreprise et de ses choix & contraintes stratégiques.

<sup>1</sup> <http://www.developpement-durable.gouv.fr/L-eco-conception-c-est-quoi.html>

Cette vision de toutes les phases permet de s'assurer aussi qu'une solution réduisant l'impact sur l'environnement à une étape ne va pas en générer un plus important à une autre étape du cycle de vie (transferts d'impact et/ou de pollution).

**L'objet de ce document est donc de proposer une méthodologie pour réaliser l'ACV d'un logiciel.**

Définir une méthodologie commune à cette catégorie de produits se justifie par le fait que les logiciels, souvent considérés à tort comme immatériels, présentent des spécificités par rapport aux produits dits « matériels ». Cette immatérialité soulève des questions quant à la meilleure façon de réaliser une telle analyse sur un logiciel.

Nous nous attacherons donc à décrire ces spécificités et à proposer ce qui nous semble être la meilleure approche au regard des objectifs que nous nous serons fixés. Puis nous détaillerons un peu plus concrètement comment mettre en œuvre cette approche dans les différentes phases normalisées de l'ACV.

Notons que l'aspect social, qui est l'un des 3 piliers du développement durable et auquel le Green Code Lab prête également une attention particulière, n'est pas traité directement dans ce document (autrement qu'en termes d'impacts sanitaires indirects). Néanmoins, des méthodologies d'ACV sociales<sup>2</sup> existent et dans une large mesure, ce qui est dit ici est tout à fait applicable et transposable à une analyse des impacts sociaux et sociétaux.

---

2 Outil GSF de NTT NTT-Orange collaboration et Rapsodie Gross Social Feel-good Index—Social Impact Assessment for ICT Services : <https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr200703043.pdf>

## II Quels sont les objectifs d'une ACV des logiciels ?



*Comparer les impacts environnementaux de plusieurs produits ou solutions logiciels afin de choisir celui ayant le moindre impact environnemental.*

Comme nous le verrons plus loin, la première étape d'une ACV est la définition des objectifs et du champ de l'étude.

C'est une étape essentielle car elle va déterminer de nombreux choix dans la façon de réaliser les étapes suivantes de l'étude, mais également le résultat même de l'étude. C'est pourquoi les ACVs sont dites « goal dépendent ».

Pour un logiciel, on peut identifier plusieurs objectifs:



***Etudier les impacts environnementaux***

Etudier les impacts environnementaux d'un logiciel donné (déjà réalisé): consommation de ressources non renouvelables, d'énergie et émissions de polluants (chimiques ou particules) dans l'eau, l'air et les sols.



***Identifier les phases impactantes du cycle de vie***

Déterminer les phases les plus impactantes de son cycle de vie fabrication/développement, utilisation, transport, fin de vie. Ce type d'étude pourra se restreindre à des catégories de logiciels (logiciels de messagerie, traitements de texte, CMS, page web...)



***Détecter les opportunités d'amélioration des impacts***

Identifier les opportunités d'amélioration et de réduction des impacts sur l'environnement pour de futurs produits. Cet objectif intéressera particulièrement les éditeurs et autres créateurs de logiciels soucieux d'améliorer la qualité environnementale de leurs produits.



***Comparer les impacts environnementaux***

Comparer les impacts environnementaux de plusieurs produits ou solutions logiciels afin de choisir celui ayant le moindre impact environnemental. Les utilisateurs (DSI, particuliers etc.) ou les développeurs / intégrateurs confrontés à des choix technologiques pourront ainsi utiliser cet outil. Dans le cadre d'une ACV comparée (évolution d'un logiciel ou nouveau produit), seules seront calculées les phases qui diffèrent entre les deux produits / services à comparer. La comparaison entre deux ACV est toujours risquée. Pour être crédible celle-ci devra se faire avec le même logiciel, à la même date, avec les mêmes règles de cut-off et de préférence avec le même praticien.

Comme tout ACV de produit et de service, pour être publié, l'ACV doit faire l'objet d'une revue critique indépendante.



## III Les spécificités des produits logiciels

L'objectif de cette partie est de présenter les caractéristiques propres aux logiciels.

Pour chacun des problèmes que soulèvent ces spécificités, nous indiquerons l'approche que nous recommandons pour l'évaluation des impacts environnementaux.

### 3.1 Logiciel : bien matériel ou immatériel ?

Le logiciel est un bien particulier :

- Il ne génère pas de déchet physique direct
- Il n'est pas directement branché à une source d'alimentation et n'est donc pas perçu comme « consommant ».
- Il est néanmoins impactant sur l'environnement à travers une consommation de ressources et d'énergie, par le matériel, nécessaire à son développement et à son utilisation.

L'ACV a pour but d'évaluer les impacts environnementaux de biens manufacturés, services et procédés. Mais où se situent les logiciels parmi ces trois types de produits ?

A première vue, les logiciels s'apparentent fortement à des biens matériels tels que ceux créés par l'industrie traditionnelle puisqu'ils sont matérialisés par un ensemble de données informatiques (code source et / ou exécutable) que l'on peut échanger, posséder et utiliser pour satisfaire un besoin précis.

Mais il faut bien distinguer le support de stockage et les interfaces physiques d'interaction du logiciel lui-même : le logiciel n'est en réalité rien de plus qu'un état du support de stockage (constitué par une suite bien définie et unique de 0 et de 1), un état du réseau transportant ces données, un ensemble d'états de l'écran

affichant la représentation graphique du logiciel, etc. Devrions-nous donc considérer le logiciel plutôt comme un bien « immatériel » ?

Pour répondre à ces questions, il convient de bien faire la distinction entre le logiciel lui-même et le service qu'il rend. Ainsi, on peut effectivement considérer un logiciel comme un bien immatériel qui rend un ou plusieurs services particuliers (ses fonctionnalités ou contenus). En tant que bien immatériel, ses impacts environnementaux résulteront de l'utilisation de ressources (humaines, physiques/matériels...) nécessaires à la mise en œuvre des différentes phases de son cycle de vie : fabrication/développement, fonctionnement, distribution, fin de vie.

### 3.2 Faut-il isoler un logiciel de son environnement de fonctionnement ?

Il est évident qu'un logiciel ne fonctionne jamais seul, mais toujours dans un écosystème avec d'autres logiciels dont il dépend, à commencer par l'OS (système d'exploitation), ou avec lesquels il est en communication ou en interaction. Ainsi, mesurer les impacts générés par le seul logiciel étudié en phase d'utilisation est compliqué.

L'impact du logiciel est indissociable du matériel & de l'OS avec lequel il fonctionne : il n'est pas possible avec une ACV d'identifier directement les impacts environnementaux liés à l'OS ou au hard. Ces impacts peuvent malgré tout être obtenus au travers d'ACV comparatives, c'est-à-dire en comparant par exemple les ACV de deux configurations bien spécifiques : par exemple Logiciel A sur

équipement X avec OS1 versus Logiciel A sur équipement X avec OS2 ; c'est ainsi qu'il sera possible d'identifier le delta d'impacts entre les OS1 et OS2. D'autres analyses de sensibilité permettraient d'évaluer les deltas d'impact liés aux équipements support.

Un équipement informatique ne fonctionne pas uniquement pour le logiciel étudié. Généralement d'autres applications/logiciels fonctionnent en parallèle sur le même équipement et donc consomment des ressources. Aussi la totalité de la puissance consommée par l'équipement ne peut pas être attribué au logiciel considéré.

La stratégie prise dans le cadre du projet de recherche Web Energy Archive ([www.webenergyarchive.com](http://www.webenergyarchive.com)) pour attribuer au logiciel l'énergie qu'il consomme, est de retrancher la consommation d'énergie due à l'OS et aux services tels que l'antivirus (on parle de consommation en mode idle), à la consommation totale de l'équipement.

### 3.3 Logiciel : quel périmètre considérer ?

L'une des principales difficultés à laquelle on se heurte lorsque l'on réfléchit à l'évaluation des impacts environnementaux d'un logiciel est que ce dernier évolue régulièrement au travers des différentes versions (correctrices ou fonctionnelles) et peut avoir une architecture modulaire, voire fonctionner simultanément sur différents équipements.

#### 3.3.1 Un logiciel évolue sans cesse

Un logiciel se décline la plupart du temps en une multitude de versions et sous-versions qui comprennent des ensembles de fonctionnalités différents.

On peut être tenté de dire que cela ne pose pas de problème majeur car les versions sont en général très espacées dans le temps, mais c'est rarement le cas.

Suite à la sortie par un éditeur d'une « release » officielle de version bien identifiée, le logiciel peut très rapidement faire l'objet de patches correctifs ou de modules complémentaires qui peuvent devenir très nombreux et fréquents.

Il faut donc différencier les évolutions mineures, des évolutions majeures d'un logiciel :

- Les évolutions **majeures** apportent de nouvelles fonctionnalités, voire restructurent complètement une application.
- Les évolutions **mineures** apportent principalement des corrections de bugs ou des ajouts de fonctionnalités secondaires. Puisqu'il est difficile de parler de version « finie » d'un logiciel, nous proposons de limiter l'étude à la dernière version stable la plus largement diffusée et utilisée. Quoiqu'il en soit, la version étudiée devra figurer clairement dans les hypothèses de l'étude.

L'impact des versions correctrices et/ou fonctionnelles, qu'elles soient mineures ou majeures ne pourront être prises en compte qu'au travers d'une analyse de sensibilité.

C'est-à-dire que l'on modélisera l'impact d'une correction de bug et une évolution fonctionnelle par l'ajout de ressources supplémentaires (RH, consommation de papier, matériel...) lors de la phase de fabrication/développement ou au travers d'une nouvelle phase spécifique (maintenance).



### 3.3.2 Un logiciel est souvent modulaire

Un logiciel peut être lui-même découpé en plusieurs modules que l'on peut choisir d'installer ou non, ou bien il peut offrir la possibilité d'être étendu par des plugins ou add-ons (comme la plupart des navigateurs web par exemple).

La notion de modularité d'un logiciel ne pourra pas être modélisée comme telle au travers d'une ACV, car il sera difficile voire impossible d'identifier les ressources spécifiques nécessaires au développement de tels ou tels module.

Il faudra donc considérer la configuration la plus standard possible, puis des analyses de sensibilité pourront être faites pour évaluer les impacts liés à des ressources nécessaires pour le développement de modules particuliers (RH, matériels...).

### 3.4 Quel est le cycle de vie d'un logiciel ?

La grande majorité des cycles de vie des produits étudiés par des ACV peuvent être considérés comme étant composés des 6 phases suivantes:

- Etude de la conception du produit
- L'extraction des matières premières
- Le processus de fabrication et de conditionnement
- Le processus logistique et d'acheminement
- L'usage du produit
- La fin de vie (démontage, transport, tri, recyclage, déchets)

Cependant, si ce cycle de vie est pertinent pour un produit matériel usuel, il l'est moins pour un logiciel :

-En tant que « bien immatériel », un logiciel ne nécessite pas directement d'extraction de matières premières.

La phase de fabrication n'est pas envisageable comme un processus de fabrication répété N fois pour produire N exemplaires du produit : il faut plutôt la considérer comme une phase unique permettant d'aboutir à une version du logiciel théoriquement reproductible et réutilisable à l'infini.

Transport amont et distribution.

Si le logiciel est conçu à partir de différentes modules développés sur différents sites, il faudra (dans la mesure du possible) prendre en compte l'envoi depuis les différents sites vers le site d'agrégation des différents modules. Dans une première approche ces impacts ne seront pas pris en compte car pour être négligés il faut qu'ils aient moins de 5% des impacts totaux.

Si la distribution vers les utilisateurs finaux est réalisée via un téléchargement sur internet, l'impact environnemental de ce téléchargement devra être pris en compte. Si la distribution se fait par CD rom la fabrication et le transport devront être pris en compte.

L'installation du logiciel peut être intégrée à la phase d'usage.

La maintenance pourra être modélisée par un surcout de la phase de fabrication. La fin de vie d'un logiciel semble à première vue inexistante, ou du moins sans impact (nous verrons plus loin qu'il n'en est rien en réalité) On devra en réalité intégrer la désinstallation des programmes et la destruction /récupération des données associées.

Comme proposé<sup>3</sup>, on peut, pour un logiciel, simplifier ce cycle de vie en ne retenant que 4 phases: la fabrication, la distribution vers l'utilisateur final, l'utilisation, et la fin de vie/réutilisation/recyclage.



*Pour un logiciel, on peut simplifier son cycle de vie en 4 étapes.*



**La fabrication** (processus de conception / développement), considérée donc comme une phase unique permettant de produire le logiciel. Cette phase intègre tout le processus de conception du logiciel :

- analyse des besoins,
  - conception,
  - programmation,
  - test,
  - stabilisation,
  - déploiement.
- Les ressources associées aux maintenances correctives (correction de bug) et les enrichissements fonctionnels, sont à inclure dans cette phase.
- Un logiciel est souvent et en partie constitué de composants tels que des frameworks, bibliothèques, librairies. Dans ce cas, on pourra considérer que la fabrication de ces composants a un impact négligeable au regard du nombre de copies.



La **distribution/copie** vers l'utilisateur final : différents scénarios sont possibles, ici seuls 3 sont rapidement présentés (d'autres sont possibles).

- Téléchargement : le logiciel ainsi que la documentation sont envoyés par voie électronique. Il faudra ici prendre en

compte le périmètre de l'émetteur du programme (serveurs de téléchargement) mais également le récepteur (les ordinateurs des utilisateurs finaux) ainsi que l'infrastructure qui a permis de véhiculer les fichiers électroniques (réseau, routeurs, ...) en prenant une quote-part de la construction du matériel et de l'énergie nécessaire au téléchargement du logiciel en fonction de la ressource utilisée.

- Le logiciel ainsi que la documentation sont packagés et envoyés par courrier postal, il faudra prendre en compte l'emballage (CDrom, DVD, clé USB, documentation) et le transport postal associé.

- L'utilisateur peut récupérer la licence et la notice d'utilisation dans un commerce ou par courrier postal et télécharger le logiciel. Il faut donc prendre en compte l'étape de packaging (fabrication & transport) ainsi que le téléchargement du logiciel. Dans ce cas particulier, les impacts liés au déplacement de l'utilisateur peuvent être forts et peuvent écraser les autres impacts. Des ACV précédentes, menées par Orange sur des terminaux, mobiles, modem, CDrom, ont montré que les déplacements client pouvaient être très variés et très impactants, s'ils ont lieu en voiture notamment, (plusieurs kg de CO2 eq)



**L'utilisation** du logiciel par l'utilisateur final démarre par la phase d'installation sur son poste (mise en service) suite au téléchargement (distribution) par exemple et couvre toute la phase d'usage du logiciel sur le matériel adéquat de l'utilisateur. Le périmètre intègre :

- Le matériel nécessaire ou prérequis pour utiliser le logiciel. Dans ce cas, on prend en compte la quote-part de :
  - la fabrication du matériel (l'équipement de l'utilisateur, accès réseau, accès serveur),

- l'utilisation de l'énergie en phase d'usage du matériel (l'équipement de l'utilisateur avec éventuellement l'accès réseau et serveur) et qui peut intégrer par défaut la consommation des logiciels pré-requis ;

- Les logiciels prérequis intégrant leurs consommations de ressources (OS, machines virtuelles, ...) ; on peut isoler la consommation de ressources de ces logiciels pré-requis en établissant une valeur étalon ou appelé « Idle » qui correspond à la consommation de ressources avant tout traitement du logiciel étudié ; cette valeur peut-être décomposée en autant de valeur si on souhaite isoler l'OS, du navigateur par exemple ;

- Le logiciel étudié intégrant sa consommation d'énergie ;

- Les données nécessaires pour utiliser le logiciel ou créées par le logiciel et qui sont stockées sur les différentes ressources de matériel de l'application ; la consommation d'énergie associée à ces données est intégrée par défaut dans l'équipement.



### La fin de vie / réutilisation / recyclage :

On fait l'hypothèse qu'en fin de vie un logiciel est effacé ou désinstallé côté usager / côté éditeur du logiciel. Il y a plusieurs points à prendre en compte pour cette étape : la fin de vie du matériel support et la fin de vie des données générées par le logiciel.

- Fin de vie du matériel support : ce point fait l'objet du § 3.5

- Fin de vie des données : on pourra considérer la bonne désinstallation du logiciel selon une procédure qui supprime

tous les fichiers de configuration sur le poste client. On doit prendre également en considération dans cette phase les données générées par le logiciel et qui ont été créées volontairement par l'utilisateur. Plusieurs cas se présentent :

- l'utilisateur ne souhaite pas récupérer les données qu'il a produites ;

- l'utilisateur souhaite récupérer ces données pour les utiliser avec un autre outil comparable et un processus de conversion est prévu ; ce processus a été pris en compte dans l'outil dans le cadre de sa phase de fabrication ;

- l'outil ne permet pas le processus de récupération et de conversion des données pour une nouvelle utilisation ; on devra alors estimer l'impact de la conversion pour l'utilisateur dans cette phase de fin de vie.

### 3.5 Fin de vie : obsolescence et déchets ?

La phase de fin de vie d'un logiciel est également difficile à appréhender



*Il n'existe pas d'obsolescence intrinsèque à un logiciel [...] théoriquement utilisable à l'infini...*

Elle fait donc ici l'objet d'un chapitre dédié. Les deux raisons mises en avant sont les suivantes :

Il n'existe pas d'obsolescence intrinsèque à un logiciel. Un logiciel est théoriquement utilisable à l'infini, tant qu'il existe des matériels pour le faire fonctionner. Le logiciel ignore l'usure et ne risque pas de tomber en panne parce qu'il est trop vieux. On ne peut donc pas déterminer une durée de vie d'un logiciel liée au fait que ses composants vont se dégrader au cours du temps. Les seules raisons pour

un logiciel de devenir obsolète sont des facteurs extérieurs au produit lui-même :

- décision de l'utilisateur de le supprimer,
- politique de maintenance d'une version,
- obsolescence des matériels supportant le logiciel,
- obsolescence des autres logiciels en interaction avec le logiciel (système d'exploitation, bases de données...),
- disparition du besoin etc.

### **Un logiciel ne génère pas en apparence de déchets matériels en fin de vie.**

C'est-à-dire que lorsque l'on décide de ne plus l'utiliser, ou que l'on ne peut plus l'utiliser, il est simplement supprimé du poste sur lequel il est installé sans que cela génère de déchets matériels. Au pire, il reste des fichiers qui occupent inutilement de l'espace disque et dans ce cas il existe bien souvent des moyens pour le récupérer.

Mais en réalité si l'on y regarde de plus près on trouve :

- des déchets matériels (déchets liés au processus de conception / développement, CD + boîte + Manuel papier si le logiciel était emballé... )
- et surtout le matériel (ordinateur, tablette, équipement réseau ...) utilisé pour faire fonctionner le logiciel.

Ces déchets sont pris en compte lors de l'étude des autres phases du cycle de vie.

### **- Remplacement ou mise à jour logiciel nécessitant de nouveaux équipements :**

Si le logiciel subit une mise à jour majeure ou s'il est remplacé par un autre logiciel nécessitant d'autres ressources matérielles (machines plus puissantes, technologies différentes). On peut alors considérer les anciens matériels comme des déchets. C'est le phénomène d'obsolescence des matériels entraîné par le renouvellement

des logiciels. A noter que pour prendre en compte cet aspect obsolescence matérielle, on pourra s'intéresser au SLI (Software Longevity Index), indicateur synthétique et prédictif mis au point par le site GreenIT.fr pour évaluer la durabilité d'un logiciel<sup>4</sup>.

### **- Effet de bord de la désinstallation sur d'autres logiciels :**

Il faudra également s'intéresser aux autres logiciels que la désinstallation en fin de vie rend obsolètes, donc être attentif par exemple aux dépendances nécessaires avec d'autres logiciels.

Si un logiciel contribue à rendre un équipement obsolète, cela signifie que les mises à jour du logiciel pour l'enrichissement fonctionnel du service, consomment de plus en plus de ressources...et in fine l'équipement support n'est plus compatible. Dans le cas présent c'est donc le logiciel et le service qui sont responsables du déchet. Un logiciel mature (sans évolution fonctionnelle) n'a pas de raison d'être à l'origine de déchets.

### **- Mauvaise désinstallation :**

Une mauvaise désinstallation peut créer une obsolescence. En effet, clés de registre laissées, fichier temporaire ; si le logiciel ne laisse pas le système comme il était avant, on crée une empreinte ressource résiduelle. Si un éditeur décide de ne plus maintenir un logiciel majeur d'un équipement type PC, le terminal deviendra obsolète et générera des déchets. C'est au logiciel que l'on attribuera tous les effets de la fin de vie du produit.

<sup>4</sup> SLI : un indicateur pour évaluer la durabilité des logiciels, Frédéric Bordage, <http://www.greenit.fr/article/logiciels/sli-un-indicateur-pour-evaluer-la-durabilite-des-logiciels-4237>

## IV Quelles différences entre une ACV de logiciel et une ACV de service ?

---

Un service est rendu/fourni par l'utilisation d'un logiciel s'appuyant sur les ressources matérielles d'un terminal (ordinateur, mobile, tablette) nécessitant éventuellement des ressources réseaux voire des ressources d'équipements informatiques distants (plate-forme de services). Un logiciel est un élément constitutif d'un service.

L'évaluation des impacts environnementaux d'un logiciel sera donc différente de celle d'un service.

L'Analyse de Cycle de Vie d'un logiciel concernera uniquement les ressources nécessaires au développement du logiciel ainsi que la quote part des ressources matérielles nécessaires à son fonctionnement.

L'Analyse de cycle de vie d'un service doit prendre en compte l'ensemble des ressources nécessaires pour rendre le service considéré (terminaux, logiciels, réseaux, plates-formes de services (serveurs)...) et s'appuiera sur les ACV de ces éléments.

Ainsi pour effectuer l'ACV d'un site web, il est nécessaire d'effectuer l'ACV de toute la chaîne supportant le service : les serveurs, le réseau et le terminal client tant pour la partie matérielle que logicielle (serveurs, client, réseau...). On prendra donc en compte l'impact du logiciel étudié (ici une ou plusieurs pages web) sur les différents composants matériels qui composent le système et qui permettent à l'utilisateur final d'accéder à la ou aux pages web.

Pour certaines ACV de service, l'impact du logiciel sur certains matériels sera négligeable et ne sera donc pas pris en compte. C'est le cas par exemple pour la partie réseau du site web (à confirmer par l'ACV logiciel des éléments réseaux).

Les résultats des ACV des différents systèmes (équipements) serviront donc de données pour l'ACV du service. Il faudra s'assurer que les unités fonctionnelles des éléments constitutifs du service sont cohérentes (temps d'usage, données transportées, allocations, etc.).

Il faut aussi que les ACV aient été faites avec le même ensemble d'indicateurs, avec la même version de logiciel, avec la même précision, avec les mêmes règles de cut-off ou « Coupure » et de préférence avec le même praticien.



## V Unités fonctionnelles d'un logiciel

**D**éfinir l'unité fonctionnelle du produit étudié est un préalable nécessaire à l'ACV car la comparaison des performances environnementales de deux produits n'a de sens qu'à service rendu identique.

**L'unité fonctionnelle représente ainsi une quantification de la fonction d'un produit.**

C'est à partir de cette unité qu'il sera possible de comparer des scénarios de produits a priori différents. Comme toute unité, elle se doit d'être précise, mesurable et additive. D'une manière générale, l'unité fonctionnelle devrait contenir une composante fonctionnelle, un critère de performance, et une durée.

Voici quelques **exemples d'unités fonctionnelles** pour d'autres types de produits ou procédés industriels<sup>5</sup> . :

- **Pour une peinture** : couvrir 1m<sup>2</sup> de mur avec une opacité mesurée à l'opacimètre d'au moins 0,98 pendant 20 ans.

- Pour un **téléphone mobile** : un an de fonctionnement sur le réseau 3G.

Pour un logiciel, l'exercice est différent car : les services rendus sont très divers et souvent très complexes. Un logiciel possède en général une multitude de fonctionnalités.

- Pour un **ensemble de fonctionnalités** données, le nombre de cas d'utilisation possibles est lui aussi très élevé.

Un logiciel peut être destiné à plusieurs usages et à plusieurs types d'utilisateurs : un grand nombre n'utilisant que quelques fonctions « de base », et quelques utilisateurs des fonctions « évoluées ». Par exemple, Word peut être utilisé comme « Notepad », ou bien comme modèle pour des macros, outil de présentation, outil pour imprimer...

Cependant, lorsque l'on cherchera à définir l'unité fonctionnelle on s'intéressera aux services rendus par le logiciel, comme par exemple :

- Ecrire n pages de document dans un **traitement de texte**

- Ecrire et envoyer un **e-mail** de n lignes à x destinataires pour un logiciel de messagerie

- **Lecteur de vidéo** : lire n minutes de vidéo d'une qualité donnée (résolution, niveau de compression)

- **Serveur Web** : traiter n requêtes HTTP

- **Base de données** : gérer N Mo de données ou répondre à n requêtes

Dans le cadre d'un service comme par exemple l'accès à un serveur Web depuis un terminal, il faudra identifier une unité fonctionnelle (« consulter la page d'accueil pendant 40s sur un PC portable ») ainsi que l'architecture support :

- 1 serveur principal

- 2 serveurs CDN

- plusieurs éléments réseaux traversés en moyenne

- 1 box ou un switch entreprise

- 1 terminal client : PC portable



## VI Frontière du système et collecte des données

Ce paragraphe a pour but d'expliquer comment constituer le périmètre de l'étude, de lister les données nécessaires et comment les obtenir.

Il existe deux types d'analyses du cycle de vie, dites « d'attribution » et « de conséquence ». L'analyse du cycle de vie « d'attribution<sup>6</sup> » décrit les flux physiques nécessaires et émis pour un produit ou un procédé, tandis que l'analyse du cycle de vie « de conséquence » décrit comment les flux environnementaux pertinents changeront en réponse aux décisions prises. Ce dernier type a un effet sur la définition des frontières.

Nous utiliserons de préférence l'analyse du cycle de vie « d'attribution » dont l'objectif est d'évaluer les impacts environnementaux émis par un processus/service.

### 6.1 Définition du périmètre d'une étude ACV

Lorsqu'on procède à une analyse de cycle de vie, après avoir défini l'objectif de l'étude et l'unité fonctionnelle il faut fixer le périmètre de l'étude c'est-à-dire lister tous les processus élémentaires qui entrent en compte dans l'étude du produit « logiciel ». Il est très utile de décrire le système à l'aide d'un diagramme des flux indiquant les processus élémentaires (intrants et extrants) et leurs relations. Les intrants et les extrants énergétiques doivent être traités comme n'importe quel intrant ou extrant d'une ACV.

La définition des frontières est itérative il sera souvent nécessaire de revenir sur les frontières pour inclure ou exclure des processus, soit parce que des données

précises ne sont pas disponibles, soit parce qu'un processus doit être inclus car présentant un impact qu'il faut qualifier plus précisément.

La suppression d'étapes du cycle de vie, de processus, ou d'intrants ou d'extrants est permise uniquement si elle ne change pas significativement les conclusions générales de l'étude. Ces décisions doivent être clairement mentionnées et les raisons et les implications de leur omission doivent être expliquées.

En matière d'étude d'analyse du cycle de vie, on utilise plusieurs critères de coupure ou cut-off pour décider des intrants à inclure dans l'analyse, tels que la masse, l'énergie et la portée environnementale. Des critères de coupure semblables peuvent également permettre d'identifier quels extrants il convient de suivre dans l'environnement, par exemple en incluant les processus finaux de traitement des déchets.

Les critères de coupure sont également établis pour permettre la réalisation des analyses de cycle de vie dans des délais raisonnables. En effet, il est contreproductif de consacrer énormément de temps pour rechercher des données pour des éléments dont les impacts environnementaux seront négligeables.

### 6.2 Collecte des données

La collecte des données concerne les mesures de consommation d'énergie (développement, tests, stockage, hébergement, usage...), l'identification du mix électrique utilisé en fonction du pays, des quantités de consommables (papier,

<sup>6</sup> Finnveden, G. et al. 2009. "Recent Developments in life cycle assessment". *Journal of Environmental Management* vol 91 p. 1-21

cartouches d'imprimante...), des distances (déplacements, livraison...), l'identification des matériaux et des composants électroniques des terminaux utilisés, des mesures de caractéristiques physiques réelles des éléments (masse, surface, volume...), les consommations réseaux, etc. Les données primaires doivent être recueillies auprès des développeurs. Des données secondaires peuvent être utilisées là où les mesures directes ne sont pas possibles.

La base de données Energy Star, disponible en ligne, est une source de données pertinentes à utiliser pour les ordinateurs de bureau, ordinateurs portables et serveurs. Cependant, si le périphérique et la configuration ne sont pas les mêmes, l'incertitude pourra être importante.

Pour les données secondaires issues de logiciels d'ACV, la zone géographique, la date de création ainsi que la source de chacune des données utilisées pour l'étude doivent être fournies.

*Il est possible aussi d'utiliser des résultats d'ACV pour les machines (serveurs, ordinateurs...) si elles ont été faites avec le même logiciel d'ACV (indicateurs d'impact) et la même version de base de données (sinon une mise à jour est nécessaire).*



## 6.2.1 Phase de fabrication

### Bibliothèques de logiciels :

Lorsque les bibliothèques de logiciels ont été développées par des tiers, il est très difficile d'obtenir des données primaires relatives au développement du logiciel. Seuls les impacts de la partie développée en interne pourront être calculés.

Des techniques d'estimation seront utilisées pour la taille ou la complexité du logiciel; cela peut être une estimation de l'effort en h.j pour le développement, ou la taille du code (nombre de lignes de code ou Mo).

### Développement et tests :

Prendre en compte :

- les ressources humaines en h.j nécessaires au développement et maintenance légère du logiciel à travers le relevé des temps associé à la conception, développement et aux tests du logiciel avant son déploiement.

Les consommations liées au développement et tests en termes d'énergie (ordinateurs, serveurs) mais également les flux d'énergie plus globaux (énergie, éclairage, climatisation) des lieux de fabrication du logiciel. *Les ordinateurs utilisés pour développer le logiciel sont considérés comme des biens d'équipement et donc hors périmètre.*

- le nombre de déplacements et les distances (voiture, train...) nécessaires au développement de l'application, le type de voiture utilisée,  
- les consommables utilisés au cours du processus de développement et de test (par exemple, papier et autres fournitures de bureau),  
- les réunions physiques et visioconférences.

Une méthode appropriée pour l'attribution des consommations dans ces cas, est basée sur le nombre d'h.j pour le développement de chaque logiciel, et sur le calcul d'un facteur d'émission par employé à allouer au développement et au test.



## 6.2.2 Phase de distribution et d'installation

La distribution du logiciel peut se faire soit par une distribution électronique (par exemple via des téléchargements sur Internet), ou en support physique (par exemple DVD).

Ou ce peut être une combinaison des deux: distribution par un support physique vers un emplacement central d'entreprise, puis distribution par voie électronique à des utilisateurs individuels au sein de la société. Lorsqu'une combinaison est utilisée, une moyenne pondérée devra être utilisée.

### **Pour la distribution électronique on devra inclure:**

- le stockage et l'hébergement du logiciel par les serveurs (y compris les serveurs miroirs),
- l'utilisation du réseau ou des réseaux (WAN & LAN) pour le transfert et le téléchargement du logiciel,
- l'utilisation de l'ordinateur/terminal de l'utilisateur final pour le téléchargement du logiciel.

### **Pour une diffusion physique on devra inclure:**

- les matières premières et la production des médias (DVD ou CD),
- le boîtier et l'emballage,
- la documentation physique livrée avec le logiciel,
- le transport des médias (y compris le stockage le cas échéant).

**Pour l'installation du logiciel on prendra en compte :** (la phase d'installation pure devrait être incluse dans la phase d'utilisation, sauf s'il faut ajouter une session de formation...)

-la durée d'utilisation de l'ordinateur/terminal de l'utilisateur final pour le téléchargement du logiciel,

-la consommation électrique nécessaire à l'installation du logiciel, associée au mix électrique ad hoc,

-l'utilisation du réseau (volume de données, par exemple en Mo ou Go) pour le transfert et le téléchargement du logiciel,

-pour l'installation, on peut prendre en compte le temps d'installation du logiciel sur le terminal sur lequel il a été téléchargé,

- la formation initiale en h.j, la consommation d'énergie électrique de l'ordinateur/terminal du client, et un mix électrique correspondant,

- des tests d'acceptation utilisateur en h.j, la consommation d'énergie électrique de l'ordinateur/terminal du client, et un mix électrique correspondant.

*Pour les logiciels d'entreprise complexes (par exemple les systèmes ERP) il y a un niveau significatif d'activité dans cette phase alors que pour les logiciels "sous cellophane", cette étape peut n'inclure que la livraison physique (ou distribution) du logiciel.*



### 6.2.3 Phase d'utilisation

• Mesure de l'énergie consommée par le logiciel lors de son utilisation.

Pour cela il faut un scénario d'utilisation du logiciel (x heures par jour), connaître le mix électrique du pays d'utilisation et déclarer le type d'instrument de mesure et la méthodologie.

L'énergie à l'utilisation d'un logiciel peut représenter la majorité de l'énergie consommée par un matériel de TIC, qui peut être significativement affectée par la conception du logiciel.

*Lors de la réalisation d'une ACV de services TIC, la mesure de l'énergie utilisée par le matériel inclut automatiquement l'énergie utilisée par le logiciel, donc dans ce cas, il n'est pas nécessaire d'évaluer séparément l'énergie consommée par le logiciel. Dans le cas où plusieurs logiciels peuvent tourner simultanément sur un même terminal il faut donc considérer uniquement la consommation liée à l'usage du logiciel.*



### 6.2.4 Phase de fin de vie

• Pour les logiciels distribués par les médias physiques, les émissions associées à la fin de vie des médias (exemple CDrom) doivent être prises en compte.

• Le matériel utilisé pour faire fonctionner le logiciel doit être pris en compte au cas où l'arrêt du logiciel le rend obsolète.

Si pour prolonger un « service » l'installation d'une version plus puissante du logiciel (version majeure) se révèle nécessaire et que ceci nécessite d'autres ressources matérielles (machines plus puissantes, technologies différentes) on peut alors considérer les anciens matériels comme des déchets.

Il faut donc leur appliquer les règles de fin de vie des DEEE : réutilisation ou collecte, démantèlement, tri sélectif, recyclage, enfouissement des déchets ultimes et transports associés.

La collecte des informations se fera auprès des gestionnaires des équipements informatiques en fin de vie et des éco-organismes de DEEE choisis par les entreprises, ou de la déchetterie pour les particuliers.

Par manque de données ou en fonction des résultats déjà acquis sur d'autres ACV de produits ou service les données suivantes pourront être mises hors périmètre :

- les services support (R&D / Capital Goods / Sales & Marketing),
- le trajet du client en voiture allant chercher son logiciel,
- les emballages secondaires et tertiaires,
- les transports amont des composants,
- les logiciels que la désinstallation du logiciel cible de l'étude rend obsolètes si l'on ne les a pas créés ou si on n'a pas d'information,
- les clés de registre laissées, les fichiers temporaires (empreinte ressource résiduelle) en cas de mauvaise désinstallation.

Dans l'alignement du GHG Protocol, les émissions dues à la fabrication des biens d'équipement (par exemple les bâtiments, machines) peuvent être exclues (notez que dans ce cas les ordinateurs utilisés pour développer le logiciel seraient considérés comme des biens d'équipement).

## 6.2.5 Qualité des données

La qualité des données recueillies pour constituer l'inventaire doit être évaluée afin de déterminer leur pertinence et leur fiabilité.

La méthode utilisée s'appuie sur les recommandations du guide Product Environmental Footprint (PEF) (Commission Européenne, 2012).

Un indicateur de de Qualité de Données est calculé selon la formule suivante :

$$QD = \frac{(MC + AM + C + U + DR + T + GC + TC + Q_{min}.4)}{(6 + 4)}$$

avec :

MC, cohérence de la méthodologie

AM, méthode d'acquisition,

C, complétude,

U, incertitude,

DR, représentativité des données,

T, âge des données,

GC, c Corrélacion géographique,

TC, corrélation technologique,

Q<sub>min</sub>, la qualité la plus faible obtenue.

<i>Note de qualité des données</i>	<i>Niveau de Qualité des données</i>
<i>≤1.6</i>	<i>Excellente qualité</i>
<i>&gt;1.6 à ≤2.0</i>	<i>Très bonne qualité</i>
<i>&gt;2 à ≤3.0</i>	<i>Bonne Qualité</i>
<i>&gt;3 à ≤ 4.0</i>	<i>Qualité correcte</i>
<i>&gt;4</i>	<i>Qualité pauvre</i>

Seuls les éléments les plus impactants

## VII *Évaluation des impacts du cycle de vie*

---

**L**'évaluation des impacts du cycle de vie (EICV) transforme l'inventaire de flux en une série d'impacts clairement identifiables grâce aux logiciels et bases de données d'ACV.

Comme le reste de l'analyse de cycle de vie, l'évaluation des impacts est fondée sur une unité fonctionnelle.

L'évaluation des impacts du cycle de vie prend comme données d'entrée une liste de flux entrants (les matières premières, matières transformées, énergies) et sortants (les rejets, déchets, émissions, etc.) agrégés sur l'ensemble du système étudié à toutes ses étapes de vie. Ces flux sont agrégés dans des catégories d'impacts pour ensuite donner des indicateurs de catégorie.

Ultimement, il est possible d'arriver à un score environnemental unique, bien que ceci implique une pondération entre les catégories d'impact.

Il existe plusieurs méthodes pour réaliser une telle évaluation.

### **Méthodes orientées problèmes**

La chaîne de cause à effet pour les problématiques environnementales est assez complexe. On peut généralement distinguer des effets primaires, découlant directement des activités étudiées, comme l'émission de CFC, et les effets secondaires, qui sont en fait les conséquences comme la diminution de l'ozone stratosphérique, résultant en une augmentation des rayons UV touchant le sol, ce qui cause des problèmes de cataracte et de cancer.

Ces méthodes sont également connues sous le nom de méthode « mid-point ».

### **Méthodes orientées dommages**

Contrairement aux méthodes orientées problèmes, les méthodes orientées dommages vont s'attacher à regrouper les impacts en fonction des résultats, aussi loin que possible dans la chaîne de cause à effet. C'est pour cela que ces méthodes sont également qualifiées de « end-point ».

**Remarque** (commentaire issu de la normation) :

Dans de nombreux cas de produits et services, les émissions inhérentes au logiciel (portant sur toutes les étapes à l'exclusion de la phase d'utilisation) ne sont pas importantes par rapport aux émissions globales du système en cours d'évaluation, en particulier lorsque les émissions dues au développement du logiciel sont amorties sur un grand nombre de copies du logiciel. Dans ces cas, il ne sera donc pas nécessaire de procéder à une évaluation détaillée du cycle de vie du logiciel.

Mais si un logiciel est développé sur mesure avec un petit nombre d'instances, il est recommandé d'effectuer une évaluation préalable de toutes les étapes afin de déterminer si une évaluation détaillée des émissions est nécessaire.



# VIII Indicateurs

Il est important de déterminer les indicateurs d'impact (mid point categories) et les catégories de dommage (damage categories) qui vont être mesurés pour l'étude. Ce choix dépendra de l'objectif que s'est fixé l'entreprise mais également de la disponibilité des données que l'on pourra exploiter.

Ci-dessous quelques exemples d'indicateurs d'impact utilisés dans le logiciel Simapro.

- **GW** (Global Warming), production de gaz à effet de serre : cet indicateur calcule la contribution au réchauffement global de la planète par l'émission de gaz à effet de serre. Il est exprimé en g eq CO<sub>2</sub>.

- **OD** (Ozone Layer Depletion), diminution de la couche d'ozone : cet indicateur calcule la contribution à la diminution de la couche d'ozone stratosphérique par les émissions atmosphériques. Il est exprimé en g eq CFC11.

- **PO** (Photochemical Oxidation), création d'ozone photochimique : cet indicateur calcule l'ozone produit dans la couche troposphérique par l'action des radiations solaires sur les émissions de gaz oxydants. Il est exprimé en g eq C<sub>2</sub>H<sub>4</sub>.

- **AE** (Aquatic Eutrophication), eutrophisation de l'eau : cet indicateur calcule l'eutrophisation (enrichissement en éléments nutritifs) des océans et des lacs par les effluents. Il est exprimé en g eq PO<sub>4</sub>.

- **AT** (Aquatic Toxicity), toxicité de l'eau : cet indicateur calcule la toxicité de l'eau en prenant en compte les concentrations limites autorisées des effluents. Il est exprimé en dm<sup>3</sup>.

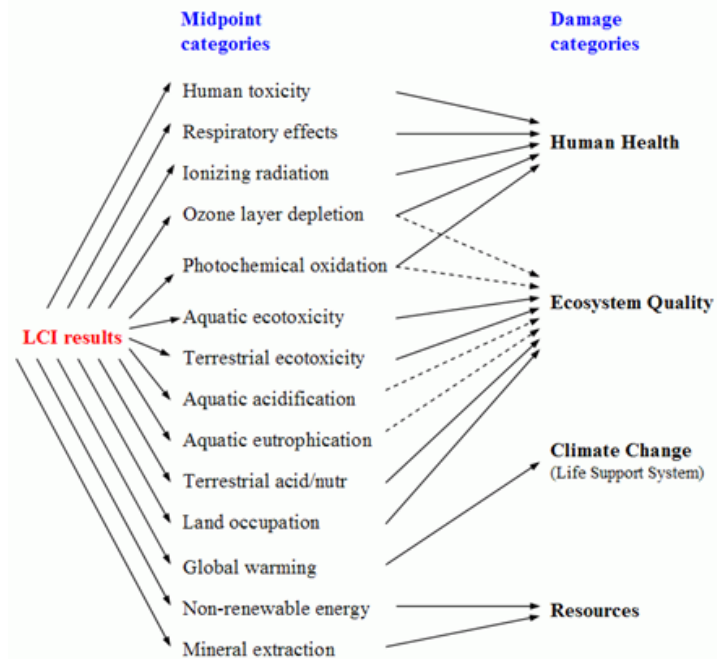


Fig. : Schéma global d'IMPACT 2002+, faisant le lien entre l'inventaire du cycle de vie (LCI) et les catégories de dommages, via les indicateurs d'impact.

Pour exemple les catégories de dommages (End Point) :

- santé humaine (DALY),
- qualité des éco systèmes (PDF.m<sup>2</sup>.an),
- changement climatique (kg CO<sub>2</sub> Eq),
- consommation de ressource (MJ énergie primaire).

Tous ces impacts peuvent être étudiés séparément puis, en définitive, être ramenés à une unité d'impact unique telle que « l'empreinte écologique » ou le « temps d'utilisation planète », comme le proposent classiquement les logiciels d'ACV.

Catégorie de dommage	Facteur de normalisation	Unités
Santé humaine	0,0071	DALY/point
Qualité des écosystèmes	13700	PDF.m <sup>2</sup> .an/point
Changements climatiques	9950	kg CO <sub>2</sub> /point
Ressources	152000	MJ/point

Table: Facteurs de normalisation pour les quatre catégories de dommage pour l'Europe de l'Ouest (Jolliet et al. 2003, Humbert et al. 2005)

## Indicateurs de flux (exemple)

- ED (Energy Depletion), consommation d'énergie : cet indicateur calcule la consommation d'énergie, aussi bien issue de combustibles (fossiles, uranium pour l'énergie nucléaire, bois, etc.) que de sources alternatives (hydroélectricité, solaire, éolienne, marée motrice, etc.).

L'indicateur prend aussi en compte l'énergie contenue dans les matériaux (qui est produite lors de leur combustion en fin de vie par exemple). Il est exprimé en MJ.

- **WD** (Water Depletion), consommation d'eau : cet indicateur calcule la consommation d'eau utilisée quelle que soit son origine ou sa qualité (potable, industrielle, etc.). Il est exprimé en dm<sup>3</sup>.

Indicateurs de design (exemple)

Les **indicateurs de Design** (conception) permettent de mettre en avant, entre autres : - le nombre de matériaux différents utilisés,

- les **indicateurs de fin de vie du produit** :

- % de réutilisation,
- % de recyclabilité,
- % de valorisation énergétique,
- % de déchets.

**Pour un logiciel, les indicateurs d'impact et les flux que nous avons identifiés sont les suivants :**

- Changement climatique « Climate change (CC for ILCD) » kg eq CO<sub>2</sub>,
- Eutrophisation eau douce « Aquatic eutrophication – Freshwater (AEF for ILCD) » kg eq P,
- Diminution des ressources naturelles non renouvelables (minérales et fossiles) « Abiotic resource depletion (ARD for ILCD) » kg eq Sb,

- Emission particules fines dans l'air « Respiratory inorganics (RI for ILCD) » kg eq PM<sub>2.5</sub>,

- Radiations ionisantes « Ionizing radiation – Human health (IRHH for ILCD) » kg eq U235.

## Les flux:

- Consommation d'énergie « Energy depletion » MJ

- Consommation d'eau « Water depletion » m<sup>3</sup>

## IX *Interprétation de l'analyse de cycle de vie*

---

Les contributions aux impacts et flux sont identifiables grâce au logiciel d'ACV :

- les impacts par phase permettent de les hiérarchiser en fonction des préoccupations de l'entreprise (CO<sub>2</sub>, eutrophisation d'eau douce...) et de ses possibilités d'action ;

- de même à l'intérieur des phases de cycle de vie il est possible de savoir si les impacts sont liés à la réalisation ou à la maintenance du logiciel, aux déplacements des équipes de développement, aux matériels utilisés pour la réalisation, au stockage ou aux impressions papier, etc.

### **Contrôle de cohérence**

L'objectif de ce contrôle est de s'assurer que les résultats obtenus sont conformes au champ de l'étude initialement formulé. Dans le cas de comparaisons entre différents scénarios, il est également conseillé de démontrer que les hypothèses choisies dans chacun des scénarios sont cohérentes les unes par rapport aux autres. Ces différences entre les scénarios peuvent venir de différences dans les sources des données, dans la précision des données, dans les représentations technologiques, etc. Les différences liées au facteur temps, au facteur géographique, à l'âge des données, et aux indicateurs doivent être également prises en compte.

Durant la phase de vérification, les données utilisées doivent être comparées aux recommandations initiales. Les écarts doivent être documentés et justifiés.

### **Analyse d'incertitude**

Elle vise à vérifier l'impact de l'incertitude des données principales sur les résultats du modèle. Ceci se fait habituellement avec des outils informatiques en utilisant par exemple une méthode de Monte-Carlo.

### **Analyse de sensibilité**

L'objectif est de valider la fiabilité des résultats finaux en déterminant l'influence sur ceux-ci de variations dans les hypothèses, les données sources et la méthodologie.

Le contrôle de sensibilité peut s'appliquer à n'importe quel élément de l'analyse : imputation, critère d'exclusion, frontière du système, catégories d'impact choisies, données de normalisation, etc.

## X *Les limites de l'ACV*

---

Bien que l'analyse de cycle de vie soit une méthode globale permettant d'évaluer les impacts sur l'environnement, certaines limitations sont à prendre en compte.

Tout d'abord, il ne s'agit que d'évaluer les impacts potentiels et non réels. De plus, les résultats sont particulièrement dépendants des hypothèses choisies au début de l'étude (périmètre de l'étude, unité fonctionnelle, etc.) mais aussi de la qualité des données (disponibilité, confidentialité, complexité, etc.).

De ce fait, la réalisation de ce genre d'étude requiert un niveau de connaissances et de compétences important dans le domaine. En ce qui concerne la conception du service ou du logiciel, l'un des facteurs limitants de l'ACV est le volume de données nécessaires à la réalisation de l'étude.

Lors du développement d'un nouveau service ou logiciel, l'analyse de cycle de vie nécessite d'avoir une connaissance suffisamment avancée du service ou logiciel et donc d'avoir arrêté un nombre important de choix techniques qui détermineront par la suite l'impact du nouveau service ou logiciel.

Il y a effectivement un risque avec les bibliothèques auxquelles on fait appel pour l'application que l'on développe car il sera quasiment impossible de faire une évaluation de leurs impacts sur la phase de fabrication et sur la phase d'usage.

Enfin, l'analyse de cycle de vie étant centrée sur l'évaluation des impacts environnementaux, il n'est pas rare que les recommandations qui peuvent émerger de l'interprétation des résultats soient en conflit avec d'autres intérêts liés au service ou logiciel tels que des considérations économiques ou sociales.

L'objectif de cette partie est de présenter les premiers résultats d'évaluation des impacts environnementaux de l'application étudiée sur laquelle ont été menées des actions de réduction de consommation d'énergie.

Cette première analyse de cycle de vie de logiciels s'appuie sur la série de recommandations ISO14 040.

Ainsi nous avons considéré les étapes suivantes du cycle de vie :

### **Fabrication**

- des équipements sur lesquels ont été développées & testés l'application,
- de l'application étudiée,

### **Le transport**

- des équipements de leurs lieux de fabrication jusque sur le site d'Orange où l'application a été développée,
- des différents modules de l'application étudiée,

### **La phase d'usage de l'application.**

- pour des raisons pratiques et de simplification nous avons installé sur un même terminal (Raspberry pi) différentes instances du logiciel SDS, ce qui permet de modéliser différents équipements sensé communiquer entre eux,

### **La fin de vie**

- des équipements qui ont servis au développement de l'application,
- de l'application.

L'ACV est par défaut multicritère mais pour des raisons de simplifications et de clarté nous nous focaliserons pour cette première étude d'impacts des logiciels uniquement sur l'impact émission CO2.

### **11.1.1 L'unité fonctionnelle**

Pour cette étude nous proposons de prendre l'unité fonctionnelle suivante :

« activer à raison de 20 fois par jour pendant 1 ans 20 petits équipements (ex : réveil, cafetière, 2 volets électriques, éclairage...) »

Chaque sollicitation a une durée en fonction de la version logicielle choisie :

- La version non optimisée a une durée de test de 78,5s,
- La version optimisée a une durée de test plus courte : 74,38s.

L'objectif de l'étude est de mesurer les impacts environnementaux (consommation d'énergie) liés au fonctionnement de l'application SDS sur l'ensemble des 20 équipements.

On installera donc sur un Raspberry pi 20 instances de l'application SDS, que l'on sollicitera 20 fois par jour sur une année (365j).

### **11.1.2 Détail des différentes phases du cycle de vie**

Tous les calculs explicités ci-après sont contenus dans une feuille de calcul Excel (.xls). L'objet des paragraphes suivants est de présenter succinctement l'approche mise en œuvre.

### 11.1.2.1 Phase de fabrication

#### Phase de fabrication/développement de l'application :

L'application a été développée sur 1 an et demi par un développeur, soit 300h.j (1 h.an Orange = 200h.j).

La consommation d'énergie de l'ensemble des PC de développement & laptops a été mesurée sur 2j ouvrables (11,75 kWh), et cette valeur a été comparée à un calcul estimatif de la consommation de ces équipements. Les résultats obtenus sont très proches. Ces valeurs ont permis d'obtenir la consommation d'énergie liée au développement de l'application : 1,7 MWh.

Cette quantité d'énergie peut être convertie en équivalent kg de CO<sub>2</sub> en prenant le facteur d'émission pour la France (0,14927 Eq CO<sub>2</sub> kg/kWh). Ainsi pour le développement de l'application 263kg eq CO<sub>2</sub> auront été émis.

A cette consommation d'énergie liée au développement de l'application il faut prendre en compte une contribution des bâtiments (éclairage, chauffage...). Le bâtiment d'Orange Labs de Rennes qui héberge les équipes date des années 1970, et le site d'Orange Labs héberge aussi le restaurant d'entreprise. Les consommations d'eau/gaz/électricité regroupent celles des bâtiments tertiaires et du restaurant d'entreprise. Comme il n'a pas été possible d'extraire des valeurs fournies celles liées aux bâtiments d'Orange Labs, il a été décidé de se référer aux consommations d'eau/gaz/électricité fournies par l'ADEME et le CEREN pour un bâtiment de bureau situé dans une zone climat normale. La surface du bâtiment d'Orange Labs est de 14 456 m<sup>2</sup> pour 602 personnes en moyenne dans le bâtiment soit 30,5m<sup>2</sup>/p. La contribution du bâtiment ramené à un développeur pour les émissions de CO<sub>2</sub> incluant tous les flux est de 2240 kg eqCO<sub>2</sub> et uniquement 1520 kg eq CO<sub>2</sub> si on ne prend en compte que le chauffage.

#### Phase de fabrication des PC de développements

Pour cette phase nous avons considéré l'environnement de travail du développeur:

Le PC de développement,  
- UC : HP Z420 Workstation,  
- écran : Samsung SyncMaster BX2240,

Le PC bureautique ,  
- laptop DELL Latitude,  
- écran Dell U2312HM,

D'un switch Netgear (pour le raccordement des deux équipements sur le réseau intranet d'Orange).

La durée de vie du PC de développement a été fixée à 5 ans et à 4 ans pour celle du laptop.

Pour des raisons de simplification, il n'a pas été pris en compte dans l'étude l'environnement de développement au sens large : gestion de configuration, sauvegarde/stockage...

Les impacts environnementaux (consommation d'énergie primaire) liés à la phase de fabrication de ces équipements ont été récupérés auprès des fabricants. Lorsqu'il n'était pas possible de récupérer l'information pour la référence voulue de matériel c'est la moyenne d'équipements similaires qui a été considérée.

La contribution de la phase de fabrication des équipements aux impacts globaux est proportionnelle à la durée de développement de l'application (300hj). Ainsi le pour le PC de développement, la durée de vie réelle d'utilisation est de 1000j (5ans à 200j/an).

La contribution de la phase de fabrication est donc de 30% des impacts totaux fournis par le fabricant.

Pour le laptop si sa durée de vie est de 4 ans, seul 37 % des impacts de la phase de fabrication seront pris en compte.



Au total la fabrication des PC +Laptop émet 394 k eq CO2.

Phase de fabrication du Raspberry Pi utilisé pour modéliser les 5 équipements

Tout comme les PC de développement nous ne prenons en compte pour le Raspberry Pi uniquement que la quote part de fabrication proportionnelle à la phase de sollicitation.

Le Raspberry Pi est utilisé 20 fois par jour pendant 19s, pour une durée de vie de 7 ans.

Sur 5 ans de fonctionnement de l'application, le ratio phase d'usage/phase de fabrication est de 3.10<sup>-8</sup>. On peut donc faire l'hypothèse que la phase de fabrication de l'équipement Raspberry Pi est négligeable.

### 11.1.2.2 Phase de Transport amont

Phase de transport amont pour l'application  
Dans le cadre de l'étude aucune phase de transport n'est à considérer pour l'application.

Phase de transport des équipements de développement.

Tout comme la phase de fabrication seul 30% des impacts du PC de développement et 37% du laptop seront pris en compte pour le transport.

Sur les sites de fabricants d'équipements sont indiqués les lieux de fabrication ainsi que les modes de transport. Le PC de développement est transporté en bateau entre l'Asie & l'Europe alors que pour le laptop le transport se fait en avion. Ces sites fournissent aussi le poids des équipements emballés (kg)

Les facteurs d'émission (bateau/camion/avion) sont ceux d'IME (eq kg CO2 par kg.km de matière transportée).

Les distances retenues sont les suivantes :

Camion Chine	2000 km
Bateau Chine - Europe	18000 km
Avion Chine - Europe	10000 km
Camion Europe - France	500 km

Le transport des PC & Laptop aura émis 33kg eq CO2

### 11.1.2.3 Phase d'usage

La consommation d'énergie liée au fonctionnement du Raspberry Pi sur une session d'usage (78,5 s) pour les 20 instances logicielles est de 0,43 mWh. La session est jouée 20 fois par jour, tous les jours de l'année. La consommation d'énergie sur un an est donc de 39 Wh, soit une émission de 5,9 g eq. CO2.

### 11.1.2.4 Fin de vie des équipements informatiques

Les équipements de développement sont supposés avoir une fin de vie de D3E standard. Les calculs donnent 7,84 kg eq CO2.

### 11.1.2.5 Récapitulatif

Le tableau ci-dessous récapitule les émissions de CO2 par phase du cycle de vie de la version du code non optimisée.

Fabrication PC + laptop	394,39	kg eq. CO <sub>2</sub>
Distribution PC + laptop	33,110	kg eq. CO <sub>2</sub>
Développement du code (300j)	263,088	kg eq. CO <sub>2</sub>
Fin de vie	7,84	kg eq. CO <sub>2</sub>
Bâtiment avec chauffage + autres flux (éclairage...)	2 240	kg eq. CO <sub>2</sub>
Bâtiment avec uniquement chauffage	1 520	kg eq. CO <sub>2</sub>
Phase d'usage du code (1 an)	4.72 10 <sup>-4</sup>	kg eq. CO <sub>2</sub>

Le tableau suivant regroupe l'ensemble des lignes liées à la phase de fabrication (hors bâtiment).

Phase de développement (hors bâtiment)	698.428	kg eq. CO <sub>2</sub>
Bâtiment avec chauffage + autres flux (éclairage...)	2 240	kg eq. CO <sub>2</sub>
Bâtiment avec uniquement chauffage	1 520	kg eq. CO <sub>2</sub>
Phase d'usage du code (1 an)	4.72 10 <sup>-4</sup>	kg eq. CO <sub>2</sub>

Suite à réception des recommandations de GREENSPECTOR pour optimiser l'application, il aura fallu 5 jours pour les implémenter soit 305 j au total pour développer cette nouvelle version du code.

Aussi les émissions de CO<sub>2</sub> liée à la phase d'usage de l'application sont de 4,5 g eq. CO<sub>2</sub> (pour une session). Toutes les valeurs d'impacts augmentent car la quote part liée au développement de l'application augmente (de 300j à 305j).

Le tableau des émissions pour la version optimisée est donc le suivant :

Phase de Fabrication	Fabrication PC + Laptop	400,96	kg eq. CO <sub>2</sub>
	Distribution PC + Laptop	33,662	kg eq. CO <sub>2</sub>
	Développement du code (300j)	267,473	kg eq. CO <sub>2</sub>
	Fin de vie	7,97	kg eq. CO <sub>2</sub>
	Phase d'usage du code (1 ans)	4.39 10 <sup>-4</sup>	Kg eq. CO <sub>2</sub>
	Bâtiment avec chauffage+ autres flux (éclairage...)	2 280	kg eq. CO <sub>2</sub>
	Bâtiment avec uniquement chauffage	1 550	kg eq. CO <sub>2</sub>

Comme précédemment le tableau ci-dessous regroupe l'ensemble des lignes liées à la phase de fabrication du code (hors bâtiment)

Phase de développement (hors bâtiment)	710,07	kg eq. CO <sub>2</sub>
Bâtiment avec chauffage+ autres flux (éclairage...)	2 280	kg eq. CO <sub>2</sub>
Bâtiment avec uniquement chauffage	1 550	kg eq. CO <sub>2</sub>
Phase d'usage du code (1 an)	4.39 10 <sup>-4</sup>	kg eq. CO <sub>2</sub>

### 11.1.3 Interprétation

Les points marquant de ces résultats d'ACV sont les suivants.

- La prédominance de l'empreinte des bâtiments dans les émissions de CO<sub>2</sub> qui sont près de dix fois plus importants que les émissions liées au développement du logiciel lui-même. Une première

recommandation serait disposer de bâtiments basse consommation voire à énergie positive.

- Par ailleurs on peut aussi noter la contribution de la fabrication des PC & laptop qui est légèrement supérieure à celle de la phase de développement.

- Enfin les émissions de CO<sub>2</sub> associées à la phase d'usage sont sur une année extrêmement faibles dans l'absolu et comparées à celle de la phase de développement du logiciel. Le ratio est encore plus important si on inclut la fabrication des équipements. Rappelons que ces émissions de la phase d'usage sont celles liées à une application qui ne fonctionne que par intermittence et sur des périodes de temps court (78s).

Ce type de ratio entre les impacts de la phase de fabrication versus ceux de la phase d'usage traduit un fonctionnement optimisé d'un point de vue consommation d'énergie. En effet dès l'usage terminé, la consommation devient nulle.

- L'application prise en compte n'est peut-être pas l'application idéale pour la mise en œuvre d'une opération d'éco conception logicielle. En effet les gains d'énergie e sont que de 7% et les valeurs absolues de gains d'émission de CO<sub>2</sub> par usage (0,472 & 0,439g CO<sub>2</sub>) sont elles aussi très faibles. Aussi calculer le nombre de versions logicielles optimisées à déployer pour compenser les surcoûts liés à l'optimisation du code (5j de développement supplémentaires) conduit à des valeurs très élevées qui ne montrent pas la pertinence de l'approche.

- Mais ces 7% de gains sont à mettre en perspective avec les 40% de gains obtenus par Orange<sup>7</sup> lors du refactoring complet de l'application Business Every Where pour laquelle tous les axes d'éco conception avaient été mis en œuvre : besoin fonctionnels et l'architecture. Dans le cas présent les 7% n'ont été obtenus que par le remplacement des séquences les plus énergivores, les résultats peuvent donc être considérés comme satisfaisant, sachant que seul un tiers des recommandations a été implémenté

- L'intérêt de cette opération est malgré tout de valider l'approche et sa potentialité pour réduire les impacts environnementaux des logiciels.

## XII Conclusion, prochaines étapes

---

Ce document a permis d'utiliser la démarche méthodologique et théorique pour évaluer les impacts environnementaux d'un logiciel.

A travers ce premier exemple d'application logicielle dans le monde des objets connectés, nous n'avons pas pu ni nous assurer de la portabilité de la méthode, ni valider les hypothèses retenues pour les prochaines explorations.

Cette méthodologie a besoin d'être éprouvée pour progresser et poser concrètement les bases de la mesure des impacts environnementaux des logiciels selon la méthode normée de l'ACV.

Malgré tout, ces premiers résultats peuvent être considérés comme encourageant au regard des travaux menés par Orange & Greenspector sur ce sujet.

Ce document pourra être complété dans le cadre de futures analyses réalisées par ORANGE et GREENSPECTOR et la communauté nationale d'écoconception des logiciels du Green Code Lab.

<sup>7</sup> « Comparaison des coûts d'implémentation entre les versions BEW V8 - V9 » document interne Orange - confidentiel.

### 13.1 Rappel des normes sur les ACV

#### **Normes ISO :**

- ISO 14040 : ACV – Principes et cadre
- ISO 14044 : ACV – Exigences et lignes directrices
- ISO 14048 : Formats d'échanges de données
- ISO 14049 : Rapports techniques sur des exemples d'analyse des inventaires selon ISO 14041

#### **Normes complémentaires pour les TIC**

- ITU-T SG5 - Q18 L Methodology: Goods networks and services, part 1
- ETSI TS 103 199 V1.1.1 Life Cycle Assessment (LCA) of ICT equipment, networks and services; General methodology and common requirements
- GeSI /Carbon Trust - ICT guidance on WRI/WBSCD product and value chain standards
- IEC TC 111 - IEC/TR 62725, "Quantification methodology of greenhouse gas emissions (CO<sub>2</sub>e) for electrical and electronic products and systems".



GREENSPECTOR



Auteurs :

Marc VAUTIER - ORANGE

Elisabeth DECHENAUX - ORANGE

Thierry LEBOUQCQ - GREENSPECTOR

Olivier PHILIPPOT - GREENSPECTOR

+33 (0) 9 51 44 55 79  
contact@greenspector.com



GREENSPECTOR