

Plan

Remerciement	2
Introduction	3
Principes et Définitions	4
La notion de la Durabilité	4
La notion du Green IT	5
La notion de l'Obsolescence Programmée	6
La notion de la famille de logiciels	8
L'importance du couplage entre Software et Hardware	12
L'empreinte environnementale de matériels informatiques	12
Démonstration de l'importance du couplage entre Software et Hardware /// reste à faire /// ..	13
Cycle de vie d'un matériel informatique /// reste à faire ///	14
Cycle de vie d'un logiciel	15
Les clés de la durabilité de logiciels (Matrice)	21
Bibliographie	22

Remerciement

Amal ALSAYED

Introduction

Amal ALSAYED

Les technologies s'accélèrent et progressent considérablement à notre époque pour répondre aux besoins sans cesse croissants de l'homme.

Comme nous le savons dorénavant, cette quête du progrès présente des effets néfastes pour l'environnement et donc pour l'humanité. L'un des impacts les plus menaçant est écologique.

Une grande majorité de scientifiques s'accordent sur la gravité du changement climatique [9]^[1]. Augmentation de la température moyenne à la surface du globe, appauvrissement de la fertilité des sols, ou encore fonte des glaces aux pôles sont autant d'effets qu'il faut limiter tant que possible et au plus tôt.

Il est clairement établi que les technologies de l'information contribuent à cet impact environnemental et au changement climatique [10]^[2]. Il s'agit donc de gérer au mieux le cycle de vie des produits technologiques.

La stratégie ou la politique que les entreprises suivent pour réduire ces effets s'appelle le **Green IT**. Dans ce domaine, il est démontré que la fabrication des produits représente classiquement un facteur 100 au regard de leur utilisation. On s'intéressera dans ce mémoire à la question d'une utilisation durable des matériels grâce à une saine gestion de leurs logiciels.

Si l'enjeu environnemental repose sur le fait de rendre durable un matériel, alors la question, posée de manière vulgarisée, peut être formulée ainsi : **Le logiciel et son cycle de vie font-ils vieillir le matériel plus ou moins rapidement ?**

L'objectif de cette thèse est donc de construire une matrice de durabilité logicielle qui expose et note les critères de la durabilité logicielle. Idéalement, cette matrice nous permettra de déterminer si un logiciel est durable ou non.

Au préalable, nous nous pencherons sur les principes et définitions afin de clarifier certains concepts puis nous illustrerons le poids relatif des émissions de Gaz à Effet de Serre du logiciel par rapport à la construction d'un équipement (un matériel). Nous pourrons ensuite décrire l'importance du couplage entre le logiciel et le matériel informatique afin de pouvoir répondre à la question que pose ce mémoire. Enfin, nous évaluerons les critères de la matrice de durabilité des logiciels pour la famille logicielle des SmartPhone.

N.B. : il existe plusieurs familles de logiciels (cf. chapitre Principes et Définitions/La notion de la famille de logiciels). Ce travail de mémoire se concentre sur la famille logicielle des SmartPhones mais pourra être étendu aux autres familles logicielles si les travaux sont repris pour poursuivre ce premier travail de recherche.

Principes et Définitions

La notion de la Durabilité

L'un des sujets les plus courants de nos jours est de savoir comment préserver l'environnement ? et comment réduire la pollution, la pollution par le gaz à effet de serre ? et l'augmentation du pourcentage de dioxyde de carbone dans l'air ?

Dans ce contexte, le concept de **Durabilité** est apparu parmi les nombreux efforts déployés. Ces dernières années, on a constaté que cette soit appliquée dans plusieurs domaines tels que l'immobilier et l'électroménager.

Bien sûr, parmi ces domaines, il y a celui des technologies de l'information, qui à son tour essaie de réduire son impact négatif sur l'environnement.

Ainsi, afin de mieux comprendre la **notion de durabilité**, nous commençons à distinguer les différents concepts de **Green IT** ou de **Gestion durable, Technologies de l'Information Green** et **Systèmes d'Information Green**.

La **Gestion durable** : est un processus de longue durée visant à optimiser les performances économiques, environnementales et sociales en tirant parti des ressources naturelles. Ce processus permet de poursuivre les activités de l'entreprise sans faire appel aux nouvelles générations. [1]

Avant d'illustrer les concepts de **Technologie de l'Information** et de **Système d'Information**, il est nécessaire de faire une distinction claire entre eux. La **Technologie de l'Information** fait référence au matériel informatique, aux logiciels et aux équipements périphériques d'information. Le **Système d'Information** est un concept large qui couvre les composantes technologiques et les activités humaines liées au processus de gestion et d'utilisation de la technologie au sein de l'organisation. [1]

Le concept de SI couvre différents aspects : [1]

- Elle comprend les TI (tels que les serveurs physiques, les ordinateurs de bureau et les dispositifs de réseau) ainsi que les services partagés. (tels que les bases de données ou le stockage).
- Les applications commerciales (telles que les systèmes ERP).
- Les ressources humaines en TI (telles que les compétences et les connaissances).
- Capacités de gestion liées aux SI pour les processus organisationnels et la transformation des activités.

Les **Technologies de l'Information Green** sont des mesures et des initiatives qui réduisent l'impact négatif de la fabrication sur l'environnement, l'exploitation et l'élimination des équipements et des infrastructures informatiques. [1]

De ce concept, le Green IT pratique : [1]

- Prise en compte de critères environnementaux lors de l'achat de matériel et de services informatiques.
- Exploitation informatique économe en énergie dans les centres de données et les

environnements de bureau.

- Pratiques respectueuses de l'environnement en ce qui concerne l'élimination des équipements informatiques.

Les **Systèmes d'Information Green** sont les pratiques qui déterminent l'investissement, le déploiement, l'utilisation et la gestion des systèmes d'information (SI) afin de minimiser les impacts environnementaux négatifs des SI, des opérations commerciales et des produits et services basés sur les SI. [1]

A partir de ce concept, les caractéristiques de l'IS green : [1]

- Réingénierie des processus d'entreprise et de production.
- Mise en œuvre de systèmes de gestion de l'environnement (SGE) basés sur les SI.
- Innovations en matière de technologies environnementales dans les produits et services destinés aux utilisateurs finaux.
- Surveillance de la demande en ressources et des émissions des produits et services (analyses du cycle de vie).

La notion du Green IT

Une bonne pratique pour résoudre un problème ou trouver une solution consiste à commencer à comprendre le problème ou les causes qui le composent. Dans notre cas, cela signifie mieux connaître les impacts environnementaux des technologies de l'information et comment rendre nos infrastructures, produits, services, opérations, applications et pratiques informatiques respectueux de l'environnement.

L'impact le plus important sur l'environnement est l'électricité. En effet, le charbon ou le pétrole qui contribue à la production d'électricité libère également du dioxyde de carbone, des polluants et du soufre dans l'atmosphère. La réduction de la consommation d'énergie électrique est essentielle pour diminuer les émissions de dioxyde de carbone et leur impact sur notre environnement et le réchauffement climatique. [2]

Comme indiqué ci-dessus, nous nous intéressons aux technologies de l'information. En effet, la technologie de l'information affectent notre environnement de plusieurs manières différentes. Chaque étape de la vie d'un ordinateur, de sa production à son élimination, en passant par son utilisation, pose des problèmes environnementaux. Chaque PC utilisé génère environ une tonne de dioxyde de carbone chaque année. [2]

Le Green IT est l'étude et la pratique de la conception, de la fabrication, de l'utilisation et de l'élimination des ordinateurs, des serveurs et des sous-systèmes associés d'une manière efficace et efficiente avec un impact minimal ou nul sur l'environnement. Le Green IT s'efforce également d'atteindre la viabilité économique et d'améliorer les performances et l'utilisation des systèmes, tout en respectant nos responsabilités sociales et éthiques. [2]

Elle nous conduit donc à appliquer une approche globale pour réduire les impacts environnementaux. Nous devons adopter une approche holistique qui aborde les problèmes de quatre manières complémentaires : [2]

- Green utilisation : réduire la consommation d'énergie des ordinateurs et autres systèmes d'information et les utiliser de manière écologique.
- Le Green élimination : remettre à neuf et réutiliser les vieux ordinateurs et recycler correctement les ordinateurs et autres équipements électroniques non désirés.
- Green conception : concevoir des composants, des ordinateurs, des serveurs et des équipements de refroidissement efficaces sur le plan énergétique et respectueux de l'environnement.
- Green fabrication : fabriquer des composants électroniques, des ordinateurs et d'autres sous-systèmes associés avec un impact minimal ou nul sur l'environnement.

Un objectif Green IT essentiel dans l'utilisation des systèmes informatiques et des centres de données opérationnels est de réduire leur consommation d'énergie, minimisant ainsi les émissions de gaz à effet de serre. Il existe quelques pratiques : [\[2\]](#)

- Réduire la consommation d'énergie des PC.
- Activation des fonctions de gestion de l'énergie.
- Éteindre le système lorsqu'il n'est pas utilisé.
- Utiliser des économiseurs d'écran.
- Utilisation d'ordinateurs clients légers.
- Conservation de l'énergie.
- Conception écologique.
- Virtualisation.

Il est important de noter que les Green IT couvrent un certain nombre de domaines et d'activités prioritaires : [\[2\]](#)

- Conception pour un environnement durable.
- L'informatique économe en énergie.
- Gestion de l'énergie.
- Disposition et emplacement des centres de données.
- Virtualisation des serveurs.
- Élimination et recyclage responsables.
- Conformité à la réglementation.
- Outils et méthodologie d'évaluation des mesures écologiques.
- Atténuation des risques liés à l'environnement.
- Utilisation de sources d'énergie renouvelables et éco-labellisation des produits informatiques.

La notion de l'Obsolescence Programmée

L'article "Art. L. 213-4-1.-I." définit l'obsolescence programmée comme :

l'Obsolescence Programmée: se définit par l'ensemble des techniques par lesquelles un metteur sur le marché vise à réduire délibérément la durée de vie d'un produit pour en augmenter le taux

de remplacement. [3]

Pour bien expliquer la notion d'obsolescence programmée, voici la différenciation de ses différents types : [7]

- **L'obsolescence indirecte** : est représentée s'il y a un défaut du produit et qu'il n'est pas possible de le réparer. Par exemple, la perte d'un chargeur de téléphone.
- **L'obsolescence d'incompatibilité** : cela signifie que s'il y a une mise à jour d'un système d'exploitation ou d'une application, il y aura un logiciel qui ne fonctionnera plus avec cette mise à jour.
- **L'obsolescence esthétique** : se fait remarquer par les nouveaux produits régulièrement mis sur le marché avec une esthétique nouvelle, voire améliorée, qui rend obsolètes les versions précédentes de ces produits.
- **L'obsolescence de fonctionnement** : c'est-à-dire la cessation de l'exploitation d'un produit à partir d'une certaine date, ou sa mauvaise qualité qui le rend impropre à la consommation. Par exemple, certaines imprimantes étaient équipées d'une puce de comptage qui bloquait l'impression au-delà d'un certain nombre de feuilles.
- **L'obsolescence de service après-vente** : est trouvée s'il est nécessaire de réparer ou d'entretenir un produit, cela coûtera plus cher que d'acheter un nouveau produit. Ou s'il est nécessaire de changer une pièce d'équipement qui est détachée du produit, il n'y aura pas souvent la possibilité de la vendre séparément, ce qui conduit à l'obligation d'acheter un tout nouvel appareil.

Il existe un accord sur l'existence d'une obsolescence conjoncturelle des produits. Il se peut que ce soit le cas : [8]

- **Raisons techniques** : par exemple, les sauts technologiques ou l'apparition d'incompatibilités entre les équipements.
- **Raisons économiques** : par exemple, les questions de coûts de stockage des pièces de rechange.
- **Raisons réglementaires** : par exemple, l'évolution des technologies ou des règles de sécurité (par exemple, l'interdiction ou la restriction de certaines substances dangereuses).
- **Les choix des consommateurs** à la lumière d'une évolution naturelle de la gamme des produits disponibles sur le marché, en fonction de la maturité de la technologie en question, des innovations et de la concurrence. L'apparition de fonctionnalités améliorées stimule le lancement de nouveaux produits, avec un déclin des produits les moins efficaces (par exemple, performance énergétique, apparition de la 3D pour les téléviseurs).

Le rapport a approuvé les définitions suivantes de l'obsolescence : [8]

- **L'obsolescence fonctionnelle** : correspond au fait qu'un produit ne réponde plus aux nouveaux usages attendus, pour des raisons techniques (exemple incompatibilité avec de nouveaux équipements), réglementaires et/ou économiques.
- **L'obsolescence d'évolution** : correspond au fait qu'un produit ne réponde plus aux envies des utilisateurs qui souhaitent acquérir un nouveau modèle du fait d'une évolution de fonctionnalité ou de design.

La notion de la famille de logiciels

Ce chapitre illustre les différents types de logiciels liés aux types de matériel utilisés (ordinateur, SmartPhone, serveur).

Avant de commencer à présenter les types de logiciels, notons la différence entre **Logiciel**, **Program**, **Application** (en **Client Léger** et/ou en **Client Lourd**).

Le **Logiciel** : applique une ou plusieurs opérations pour transformer les données d'un état A à un état B. Un logiciel traite donc les données par une ou plusieurs séquences d'opérations.

Le **Programme** : est une séquence d'opérations. A partir de cette définition, on remarque qu'un **logiciel** est la compilation d'un ensemble de **programmes**.

L' **Application** : est la mise à disposition de fonctionnalités pour un usage précis et identifié. Elle s'instancie soit :

- En **Client Léger** (Thin Client en anglais) : est une interface utilisateur qui sert principalement à afficher des informations et à répondre aux interactions de l'utilisateur. Peu ou pas de logique applicative réside sur le client (à part la vérification de base des entrées), la plupart du travail est effectué sur un serveur central. [4]
- En **Client Lourd** (Fat Client en anglais) : est une application client qui intègre une logique commerciale. Le traitement est donc décentralisé - le serveur n'est responsable que de parties rudimentaires de la logique et sert principalement à stocker les données. [4]

Ces types de composants interagissent avec différents classes de Matériels.

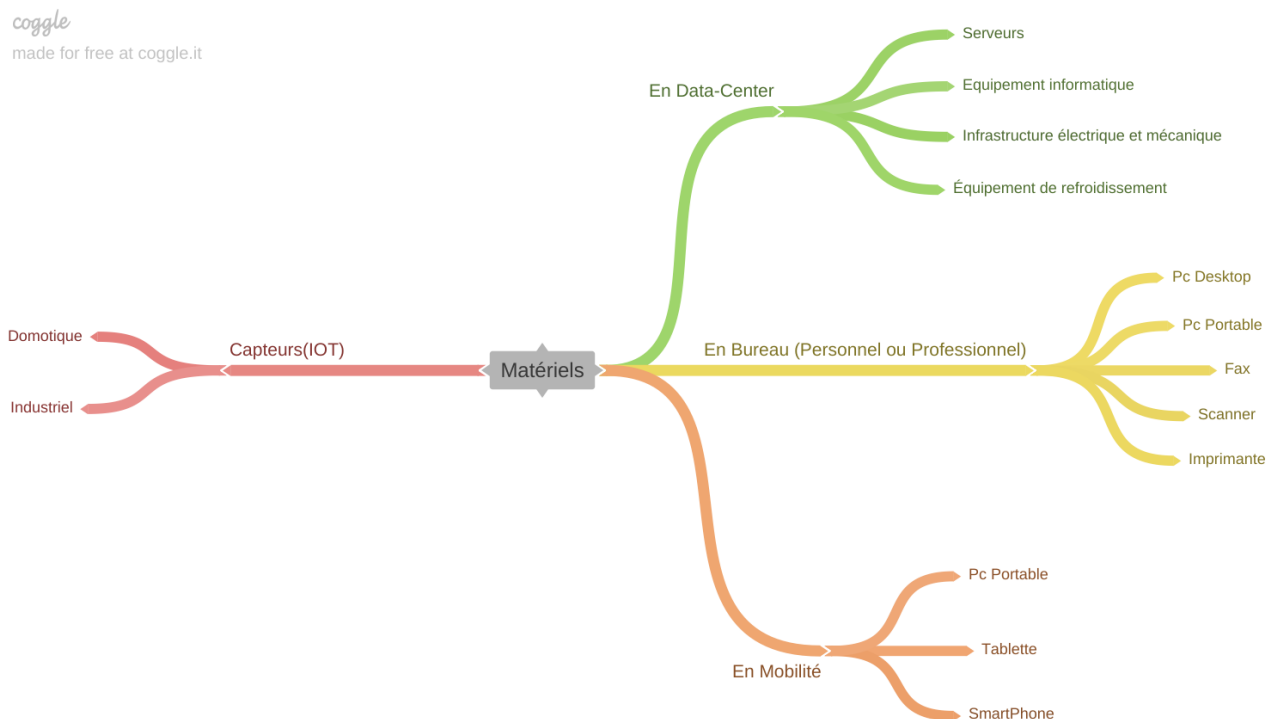


Figure 1: Les Matériels

Sur ces matériels, nous installons un Système d'Exploitation qui offre à l'utilisateur la possibilité d'interagir avec un lot d'applications pour ses besoins. Les Systèmes d'Exploitation les plus connus :

Windows, Linux et Mac-OS.

Sur ces systèmes d'exploitation, des logiciels peuvent être installés. Ces logiciels sont variés en fonction de leur utilisation, tels que : **Logiciels d'Application**, **Logiciels d'Exploitation** et **Logiciels de programmation**. Ces logiciels peuvent être soit **Payants**, soit **Gratuits**, soit **Open-Source**.

La famille de logiciels sur un SmartPhone

Système d'exploitation mobile est conçu pour fonctionner sur un smartphone, une tablette ou tout autre dispositif mobile. Aujourd'hui, les trois systèmes les plus importants sont Android, iOS et Windows Phone. [5]

- Le système d'exploitation **Android** : [5]

Le système d'exploitation d'Android fonctionne sur plusieurs appareils tels que les smartphones, tablettes, montres, téléviseurs et voitures.

L'architecture d'Android est composée de plusieurs couches :

- **Linux Kernel** : le kernel est basé sur Linux. Il permet de faire le lien entre la partie hardware et le logiciel. Il contient les différents drivers de la partie matérielle.
- **Libraries** : la couche suivante contient plusieurs librairies telles que SQLite, SSL, OpenGL...
- **Android Runtime** : la couche Android Runtime contient des librairies qui permettent aux développeurs d'utiliser un sous-ensemble des libraires Java ainsi qu'une machine virtuelle.
- **Applications Framework** : les applications Android interagissent avec la couche application Framework. Cette couche permet d'accéder au travers d'API aux fonctionnalités du dispositif.
- **Application** : toutes les applications installées se trouvent dans cette couche.

Les applications écrites en Java sont exécutées dans une machine virtuelle. Cependant, au lieu d'utiliser une JVM, Android utilise sa propre machine virtuelle (Dalvik Machine, ART).

- Le système d'exploitation **iOS** : [5]

iOS est le système d'exploitation mobile développé par Apple et fonctionnant uniquement sur du matériel Apple.

L'architecture d'iOS est composée de 4 couches différentes :

- **Core OS** : cette couche est responsable du système d'exploitation. Elle est en charge de la gestion de la mémoire ainsi que l'accès au matériel du dispositif.
- **Core Services** : elle contient les API qui permettent aux applications d'accéder à différents services tels que le réseau, les contacts, la base de données.
- **Media** : la couche média permet d'implémenter des fonctionnalités audio, vidéo et graphiques.
- **Cocoa Touch** : cette couche définit les bases de l'application. Elle contient de nombreuses fonctionnalités comme la gestion des vues, la reconnaissance des gestes, le multitâche, les notifications.

- Le système d'exploitation **Windows** : [\[5\]](#)

L'architecture d'une application Windows est composée de 4 couches :

- **Core** : la couche la plus basse dans la hiérarchie, c'est le kernel qui permet de gérer le matériel du dispositif.
- **System Services** : au-dessus, la couche contient les API qui permettent d'accéder aux fonctionnalités des appareils.
- **Model Controller** : la couche suivante contient le code qui sera exécuté.
- **View** : la dernière couche contient les vues avec lesquelles les utilisateurs vont interagir.

Il existe plusieurs types de développement pour mettre au point une application mobile. Ces développements comprennent : [\[5\]](#)

- Le **Développement Native** : est spécialement développée pour un système d'exploitation. Ainsi, Android, iOS et Windows Phone, nous avons besoin de trois versions de logiciels. Sachant que chaque système d'exploitation possède ses propres outils, langages et spécificités. Ces applications nécessitent d'installer sur l'appareil.
- Le **Développement Web** : sont conçues pour fonctionner dans un navigateur web et sont généralement adaptées pour qu'elles puissent s'exécuter dans un navigateur mobile. Elles n'ont besoin que d'un navigateur mobile pour les utiliser.
- Le **Développement Multiplateforme** : Leur objectif est de développer une application qui fonctionne sur tous les systèmes d'exploitation.

La famille de logiciels sur un serveur

Le serveur est un dispositif qui met des ressources, des données, des services ou des logiciels à la disposition des clients et qui est connecté au réseau. Il existe plusieurs types de serveurs en fonction de leurs services. En outre, le serveur dispose d'un système d'exploitation, tel que Windows, Linux/Unix et Mac-OS.

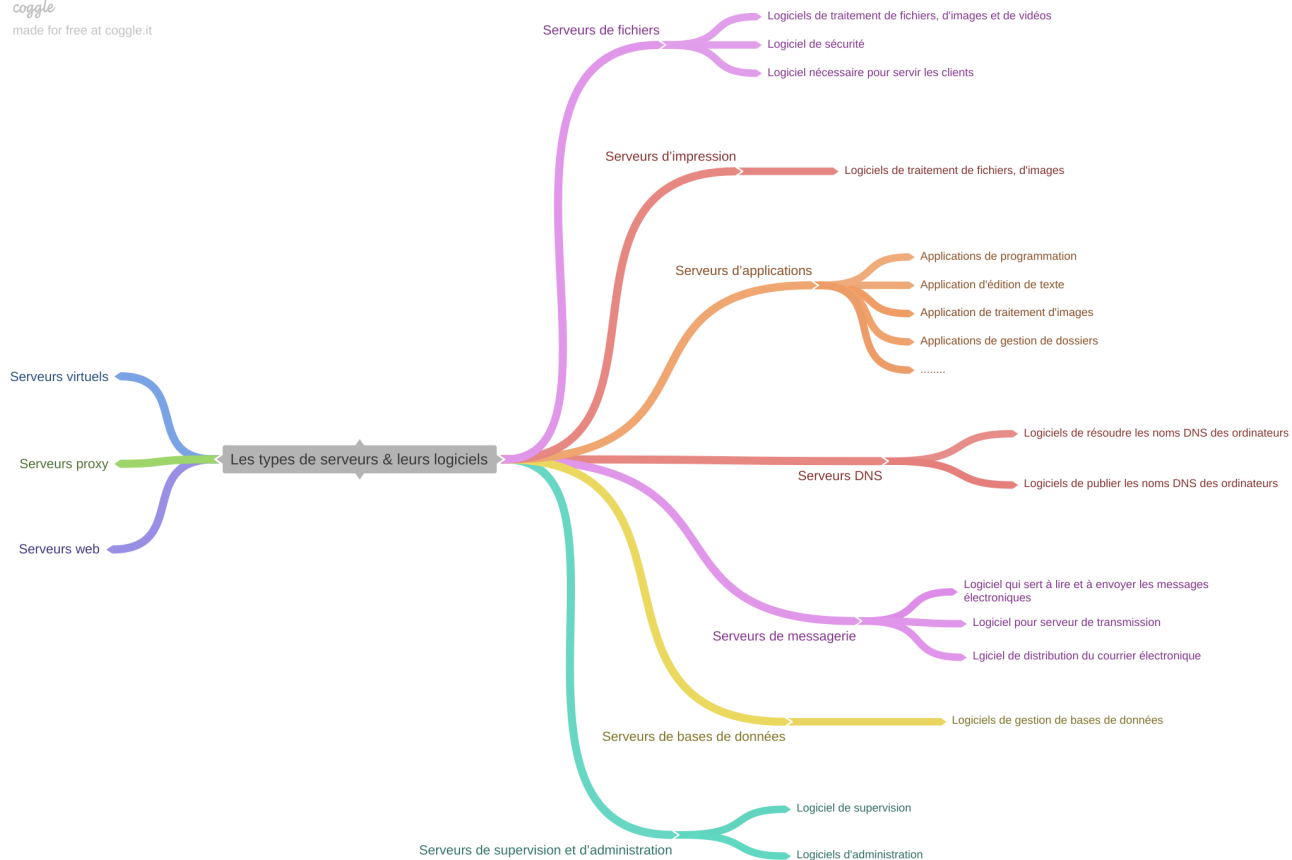


Figure 8: Différents types de serveurs et leurs logiciels

L'importance du couplage entre Software et Hardware

L'objectif essentiel de la durabilité des logiciels qu'elle s'agit de travailler sur la durabilité des matériels. Sans l'existence de la durabilité des matériels, il y n'a pas de sens de la durabilité des logiciels. Car l'empreinte écologique dépend de la durabilité du matériel !

L'empreinte environnementale de matériels informatiques

Il existe de nombreux types de matériel informatique, tels que : ordinateurs, smartphones, imprimantes, consoles de jeux vidéo, téléviseurs connectés à un boîtier, objets connectés, etc. Chaque type a une empreinte écologique.

Bien sûr, il existe plusieurs études qui clarifient l'empreinte écologique. L'une des ces études est **Empreinte environnementale du numérique mondial** [6] que l'équipe de GreenIT.fr a publié, qui se concentre sur la quantification de l'empreinte environnementale numérique globale et son évolution entre 2010 et 2025.

Cette étude s'applique à tous les équipements électroniques qui manipulent des données binaires. Elle est basée sur une méthodologie d'analyse du cycle de vie (ACV), et s'appuie sur trois modèles de quantification des impacts environnementaux (utilisateurs, réseaux, centres informatiques) agrégés par un méta-modèle.

Les quatre indicateurs qui ont été pris en compte dans cette étude : [6]

- Épuisement des ressources abiotiques (ADP) : cet indicateur, qui évalue l'impact de la technologie numérique sur l'épuisement des stocks de minéraux.
- Réchauffement global (GES) : cet indicateur reflète le changement climatique.
- Bilan énergétique (EP) : L'eau douce est considérée comme la deuxième ressource physiologique de base la plus importante pour l'homme et des millions d'autres formes de vie. Par conséquent, l'eau bleue est plus accessible aux êtres humains que l'eau verte, qui n'est accessible qu'aux plantes. Cela signifie que si la technologie numérique augmente la consommation d'eau bleue, cette dernière sera moins disponible pour d'autres usages à un moment donné.
- Tension sur l'eau douce (Eau) : L'énergie primaire est l'énergie nécessaire pour produire l'énergie finale. Ainsi, en numérique, selon l'étape du cycle de vie d'un équipement, il y aura différentes énergies primaires pour produire différentes énergies finales.

Le mot **Numérique** n'est pas abstrait. Il est donc traduit par équipement et câbles. Pour cela, l'équipe de GreenIT.fr a divisé le monde numérique en trois tiers : les **utilisateurs**, les **centres informatiques** et les **réseaux** qui relient les utilisateurs entre eux et aux centres informatiques.

Pour illustrer l'empreinte écologique du numérique mondial, l'équipe de GreenIT.fr a étudié l'ensemble des impacts environnementaux pour chaque tiers du numérique. Nous présentons quelques résultats de l'étude : [6]

"L'empreinte environnementale du numérique mondial est de l'ordre de :

- * 6 800 TWh d'énergie primaire (EP).
- * 1 400 millions de tonnes de gaz à effet de serre (GES).
- * 7,8 millions de m³ d'eau douce (Eau).
- * 22 millions de tonnes d'antimoine (ADP)."

"L'étude indique que les émissions de gaz à effet de serre sont directement liées à la combustion de l'énergie primaire fossile que nous produisons à tous les stades du cycle de vie des équipements numériques. Cela signifie que la fabrication des équipements utilisateurs prend en charge la station d'émission principale, suivie de l'alimentation électrique, puis des équipements de réseau et des centres informatiques."

"C'est la phase d'utilisation qui émet le plus de gaz à effet de serre. Cela signifie qu'à volume constant d'émissions de GES, plus la phase d'utilisation augmente, plus la durée de vie des équipements s'allonge."

"Les équipements des utilisateurs sont la principale source d'impacts du numérique mondial. Leur fabrication concentre systématiquement le plus d'impacts avec 30% du bilan énergétique global, 39 % des émissions de GES, 74 % de la consommation d'eau et 76 % de la contribution à l'épuisement des ressources abiotiques. Si on y ajoute les impacts associés à la production de l'électricité qu'ils consomment, les équipements utilisateurs (hors box DSL / fibre) totalisent de 59 % à 84 % des impacts !"

"Tous les matériels vont augmenter leurs impacts multiplier de 2 à 3 fois à 2025, mais les objets connectés vont avoir le plus d'impact multiplier 5 fois à 2025, La croissance exponentielle du nombre d'objets connectés (de 1 milliard en 2010 à 48 milliards en 2025)."

Ces résultats nous encouragent donc à réduire autant que possible ces impacts environnementaux. Car après tout ce qui précède, nous constatons le grand effet de la technologie numérique sur l'environnement, qui augmente chaque année avec la croissance des équipements numériques et de leurs utilisations. Cette croissance est liée au soutien de l'économie, il n'est donc plus acceptable d'accélérer volontairement notre empreinte numérique uniquement pour le bien de l'économie, ce qui est la principale raison de la croissance effrénée des impacts numériques.

En outre, cette étude illustre certaines hypothèses et recommandations visant à réduire les impacts environnementaux (pour plus d'information [\[6\]](#)).

Démonstration de l'importance du couplage entre Software et Hardware /// reste à faire ///

L'objectif essentiel de la durabilité des logiciels qu'elle s'agit de travailler sur la durabilité des matériels. Sans l'existence de la durabilité des matériels, il y a pas de sens de la durabilité des

logiciels. Car l'empreinte écologique dépend de la durabilité du matériel !

Comme l'explique Anne-Cécile Orgerie dans l'article "L'informatique émet plus de gaz à effet de serre que l'aviation [12]", que TIC et tous les équipements informatiques représentent entre 2 à 10 % des émissions de dioxyde de carbone selon les études, avec un consensus autour de 4-5 %. En comparaison, l'aviation représente environ 2 % des émissions de dioxyde de carbone.

Des études montrent que les deux phases de fabrication et de recyclage sont responsables de la plus forte proportion d'émissions de dioxyde de carbone. D'où l'importance du couplage entre le matériel et les logiciels pour développer des logiciels durables afin de rendre le matériel plus durable avec la plus longue période d'utilisation possible.

La durabilité des logiciels ne suffit pas à elle seule à rendre le matériel durable. En même temps, il y a aussi des conditions préalables à la durabilité du matériel. Celles-ci sont présentées ci-dessous.

Conditions préalables pour des matériels durables

Une bonne utilisation de l'équipement permettra de le faire fonctionner le plus longtemps possible. Voici les conditions de maintenir de l'équipement :

- Ne mettez pas l'équipement à la chaleur ou à l'humidité.
- Évitez de charger les ordinateurs portables et les smartphones pendant leur utilisation.
- Évitez de télécharger des applications et des logiciels qui ne sont pas nécessaires, car ils prennent de l'espace mémoire et s'ils fonctionnent toujours en arrière-plan, cela entraîne une consommation d'énergie plus élevée et donc un cycle de vie des batteries plus court. *

Décrire les mauvaises conditions de maintenir les machines les bonnes smartphone tap pc serveur

le logiciel peut aussi endommager le matériel /// reste à faire ///

sur chauffe temp monte trop

Cycle de vie d'un matériel informatique /// reste à faire ///

Comme il le souligne, la définition de la **Technologie de l'Information** dans le chapitre **Principes et définitions** : est l'ensemble du matériel informatique, des logiciels. Et selon l'objectif de ce travail, il est nécessaire de connaître le cycle de vie du matériel et des logiciels.

Dans ce paragraphe, nous parlerons du cycle de vie du matériel informatique, puis du cycle de vie du deuxième ingrédient de la technologie de l'information, le logiciel.

Les impacts du cycle de vie d'un matériel :

Pour illustrer l'empreinte des gaz à effet de serre (GES) d'un ordinateur, nous allons résumer l'article "Quelle est l'empreinte carbone d'un ordinateur" publié sur le site GreenIT.fr [13]. L'article présente les résultats du travail de Fujitsu avec l'Institut de recherche Bifa à Ausbourg pour présenter l'empreinte en gaz à effet de serre (GES) de son ordinateur de bureau Espresso E9900. L'étude a été menée conformément aux normes internationales ISO 14040 et ISO 14044, qui

définissent la méthodologie pour réaliser une analyse du cycle de vie (ACV). Son empreinte totale de GES sans l'écran est de 339 kg d'équivalent CO₂, ce qui comprend les trois phases de fabrication, de transport et d'assemblage des composants. Pour la phase d'utilisation, sa consommation annuelle est de 113,6 kWh. Avec l'ajout de l'écran, qui émet 676 kg d'équivalent CO₂. Cela nous donne un ordre de grandeur de l'empreinte globale d'un poste de travail récent : 1015 kg d'équivalent CO₂ pour un poste de travail complet !

Cycle de vie d'un logiciel

Pour illustrer la notion de logiciel, l'Organisation internationale de normalisation (ISO), a défini en 1981 le logiciel (en anglais software) comme une création intellectuelle rassemblant des programmes, des procédures, des règles et de la documentation utilisés pour faire fonctionner un système informatique. Pour renforcer le caractère intellectuel du logiciel, l'ISO précise en plus que le logiciel existe indépendamment des supports utilisés pour le transporter, ce qui sous-entend qu'il ne faut pas confondre le logiciel avec son support. [11]

Nous définissons le cycle de vie des logiciels : est la période de temps s'étalant du début à la fin du processus du logiciel. Il commence donc avec la proposition ou la décision de développer un logiciel et se termine avec sa mise hors service. [11]

Un autre terme pour décrire le cycle de vie des logiciels est le **Processus du logiciel** (en anglais **Software process**). Ce processus est composé de plusieurs sous-processus qui échangent et communiquent entre eux. Pour chaque processus, il y a des conditions à déclencher, des caractéristiques uniques, des activités déployées, certaines ressources utilisées et les produits qui en résultent et ses critères de fin. Les activités qui font partie d'un processus sont à leur tour divisées en sous-activités et en tâches plus détaillées. [11]

Modélisation du cycle de vie du logiciel (software life cycle model)

Gestion de projet (project management)

Initiation du projet (project initiation)

Pilotage et suivi du projet (project monitoring and control)

Gestion de qualité du logiciel (software quality management)

Pré-développement (pre-development)

Besoins de l'utilisateur ou du système (user/system requirements)

Conception du système (system design)

Développement (development)

Besoins en logiciels (software requirements)

Conception architecturale (architectural design)

Conception détaillée (detailed design)

Codage (code)

Intégration (integration)

Réception du logiciel (software acceptance)

Intégration du système (system integration)

Test de terrain (field test)

Post-développement (post-development)

Installation (installation and checkout)

Exploitation (operation)

Maintenance et support du logiciel (software maintenance and support)

Retrait (retirement)

Processus globaux (integral processes)

Vérification et validation (verification and validation)

Gestion de la configuration du logiciel (software configuration management)

Développement de la documentation (documentation development)

Formation (training)

Figure 12: Architecture du processus du logiciel (Software process architecture, ISO/JTC1/SC7/WG3) [11]

En partant de la définition du cycle de vie des logiciels, nous découvrons que le cycle de vie des logiciels se compose de quatre phases [11] qui se succèdent les unes aux autres :

- **Avant-projet/ Etude préalable** : c'est la phase préparatoire qui a des tâches telles que répondre à des questions comme la clarification de l'objectif du logiciel et de son importance. Il s'agit notamment de clarifier les aspects techniques et de gestion, les phases de ce logiciel et les besoins des utilisateurs. Pour chacune de ces phases, il existe plusieurs petites tâches détaillées. Les résultats de cette phase sont documentés dans un document appelé **Cahier de charge/Spécification du projet** (en anglais projet specification).
- **Développement** : nous pouvons aussi dire le **cycle de développement du logiciel** qui commence avec la décision de développer un logiciel et se termine avec la livraison du logiciel et son installation. Nous en parlons en détail par la suite.
- **Exploitation et Maintenance** : après l'achèvement de la phase précédente et avant d'entrer

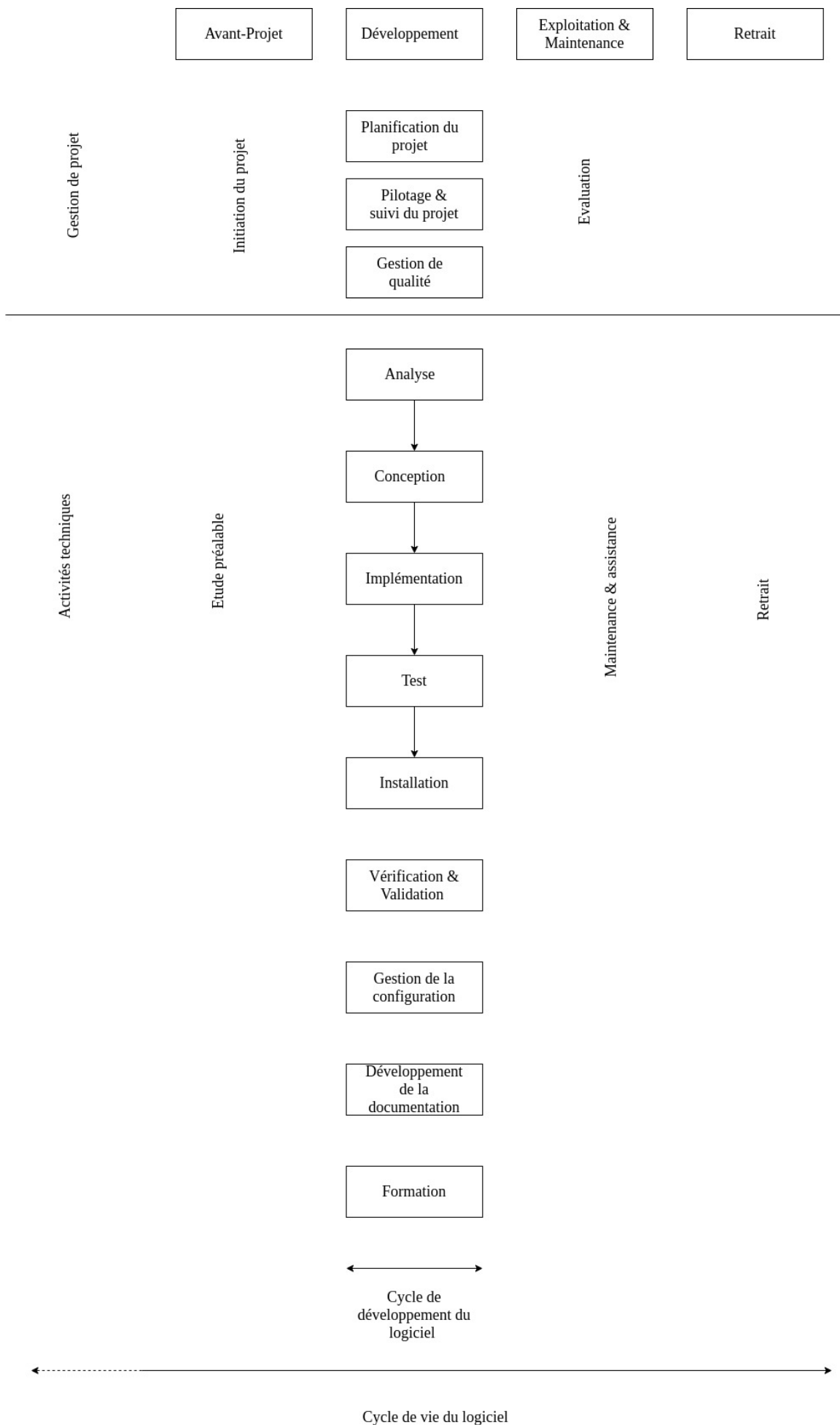
dans la phase d'exploitation, le logiciel passe au stade de **installation**. Cette étape est prise en charge soit par l'équipe de développement pour les logiciels en développement interne, soit par l'utilisateur qui effectue l'installation dans le cas de logiciels largement distribués, et est soutenue par l'assistance du distributeur (service après-vente). Après cette étape vient la phase d'exploitation et de maintenance, qui comprend l'exploitation du logiciel dans son environnement d'exploitation, sa surveillance et sa modification si nécessaire. N.B. : la maintenance du logiciel consisterait à corriger les erreurs, à ajouter de nouvelles fonctionnalités, à l'adopter avec un nouvel environnement, etc. On distingue donc plusieurs types de maintenance : la maintenance corrective, la maintenance perfective et la maintenance adaptative. Souvent, la maintenance d'un logiciel nécessite de revenir à la phase de développement pour le réappliquer. Après chaque maintenance d'un logiciel, il faut bien sûr le distribuer. En outre, il faut fournir une assistance technique et un soutien en matière de conseil.

- **Retrait** : la dernière phase que nous pouvons également appeler **Mettre le logiciel hors de service**. Cette phase comprend : avertir les utilisateurs, effectuer une exploitation en parallèle du logiciel à retirer, arrêter le support du logiciel.

Pour revenir au cycle de développement du logiciel [11]. Ce dernier se compose de plusieurs phases qui sont les suivantes :

- **La gestion du projet.**
- **La vérification et validation.**
- **Le développement de la documentation.**
- **La gestion de la configuration.**
- **La formation.**

De tout ce qui précède, nous avons résumé le cycle de vie des logiciels et le cycle de développement des logiciels, ce qui correspond au but de cette thèse (pour aller plus en détail [11]).



Que savent les programmeurs de la consommation d'énergie des logiciels ?

Les chercheurs ont mené une enquête présentée dans l'article "What Do Programmers Know about Software Energy Consumption? [14]", qui a révélé que les programmeurs avaient une connaissance limitée de l'efficacité énergétique, n'étaient pas au courant des meilleures pratiques pour réduire la consommation d'énergie des logiciels et n'étaient pas sûrs de la manière dont les logiciels consomment l'énergie. Cette question d'une exigence non fonctionnelle devient très importante avec la popularité croissante de l'informatique mobile et l'émergence de déploiements de cloud à grande échelle, et en sachant que si la consommation d'énergie au niveau individuel était négligeable, elle ne le serait pas à l'échelle mondiale car l'énergie consommée par tous les appareils mobiles et les centres de données se multiplie.

L'enquête menée par ces chercheurs était anonyme en ligne et comportait 13 questions en quatre étapes. Les résultats mettent en évidence le besoin de formation sur la consommation d'énergie et l'efficacité énergétique des logiciels, montrant que le sujet de l'efficacité énergétique est rarement abordé par les programmeurs, et qu'il y a peu de demande de la part des utilisateurs, qui s'intéressent à la fois à la vitesse de développement et aux performances raisonnables des logiciels, ce qui signifie que le manque d'attention à la consommation d'énergie des logiciels est une priorité. Cela signifie également que les programmeurs ne s'intéresseront probablement pas à l'efficacité énergétique des logiciels si les clients ne le demandent pas. Il est donc nécessaire d'éduquer le public de manière appropriée afin que les clients et les programmeurs soient conscients de la consommation énergétique des logiciels. Les résultats de l'enquête montrent également que les programmeurs ne savent pas comment mesurer avec précision la consommation énergétique de leurs logiciels, puisque les méthodes de mesure (par compteur électrique, batterie, alimentation électrique, mesure des ressources, outils logiciels et temps CPU), mesurent la consommation énergétique globale du matériel, et non la consommation énergétique des logiciels. [14]

Les chercheurs indiquent que pour analyser correctement les raisons sous-jacentes de la consommation d'énergie des logiciels, les programmeurs doivent comprendre les interactions entre les composants de haut niveau et de bas niveau. En outre, et sur la base des résultats de l'enquête montrant que les opinions des personnes interrogées varient largement sur la meilleure façon de réduire la consommation d'énergie des logiciels, les chercheurs indiquent que les cours universitaires n'enseignent généralement pas le lien entre de meilleurs algorithmes et la consommation d'énergie. [14]

Selon le rapport publié par les chercheurs, leurs conclusions sont très similaires à celles de Gustavo Pinto et de ses collègues, qui ont extrait des données de StackOverflow pour découvrir sept raisons de la consommation d'énergie inutile des logiciels : [14]

- Utilisation inutile des ressources.
- Comportement défectueux du GPS.
- Activité de fond.
- Synchronisation excessive.
- Les fonds d'écran.
- Publicité.

- Utilisation élevée du GPU.

Selon le rapport des chercheurs, Pinto et ses collègues ont identifié huit stratégies pour réduire la consommation d'énergie en modifiant le logiciel : [\[14\]](#)

- Minimiser les E / S.
- Traitement par lots (batch processing en anglais).
- Interrogé fonctionnement.
- Coordination matérielle.
- Programmation simultanée.
- Initialisation paresseuse.
- Tourner au ralenti (race to idle, en anglais).
- Structure de données efficace.

De la part des chercheurs, ces stratégies devraient faire partie de la formation des programmeurs. De plus, selon eux, des outils de développement peuvent être créés pour identifier les consommations d'énergie inutiles et faire des recommandations pour réduire la consommation d'énergie, et les enseignants peuvent développer des diapositives, des vidéos, des projets et des devoirs dans le cadre du cours de premier cycle sur l'efficacité et la durabilité énergétiques. [\[14\]](#)

Les clés de la durabilité de logiciels (Matrice)

Les critères : /// reste à faire ///

- Mise à jour selectif : batch/ sécurité/fonctionnalité
- CPU
- Mémoire
- Batterie
- Utiliser des ressources
- Bilan ?
- Temps-Bugs ?
- Open source
- Travailler en arrière plan
- Consommation d'énergie
- Low-tech / High tech
- APK/Space/
- Préparation de logiciels
- Consommer des ressources
- Lancer automatiquement au démarrage par défaut
- Mode nuit / jour
- Logiciels audio / travailler sans allumer l'écran
- Éviter les publicités
- Des applications installées et ne pas les utiliser
- La gourmandise des logiciels en ressources. → L'obsolescence programmée
- Lazy loading
- Congruence ?

Bibliographie

Amal ALSAYED

- [1] Fabian Loeser. Technische Universitaet Berlin. Green IT and Green IS: Definition of Constructs and Overview of Current Practices. Cité 67 fois researchgate.net [L'article](#)
- [2] S Murugesan. 2008 - ieeexplore.ieee.org. Harnessing Green IT: Principles and Practices. Cité 1318 fois. [L'article](#)
- [3] LOI n° 2015-992 du 17 août 2015 relative à la transition énergétique pour la croissance verte (1). Article 99. legifrance.gouv.fr. [L'article](#)
- [4] Viktor C. Pavlu. 2003 – 2004. Thin-Client/Fat-Client Comparison. UNIVERSITY OF DERBY Derbyshire Business School. [L'article](#)
- [5] Luca FALVO. 2016. Haute École de Gestion de Genève (HEG-GE). Développements Natif, Xamarin, Hybride ou Web : Critères de choix et arbre de décision pour un développement mobile. [L'article](#)
- [6] Frédéric Bordage. GreenIT.fr. 2019. Empreinte environnementale du numérique mondial. [L'article](#)
- [7] Christophe BOUILLON, Michel HAVARD. Rapport d'information déposé en application de l'article 145 du Règlement par la mission d'information sur la gestion durable des matières premières minérales. assemblee-nationale.fr [L'article](#)
- [8] Shailendra Mudgal, Benoît Tinetti, Thibault Faninger, Sarah Lockwood, Gina Anderson. Étude sur la durée de vie des équipements électriques et électroniques. Juillet 2012. Agence de la transition écologique (ADEME). ademe.fr. [L'article](#)
- [9] Mohan MUNASINGHE, Osvaldo CANZIANI, Ogunlade DAVIDSON, Bert METZ, Martin PARRY, Mike Harrison. Integrating sustainable development and climate change in the IPCC fourth assessment report. 2003. [L'article](#)
- [10] L'informatique émet plus de gaz à effet de serre que l'aviation. 21 octobre 2015 [L'article](#)
- [11] Didier Buchs, 20 May 2014, Cycle de vie du logiciel. [L'article](#)
- [12] L'informatique émet plus de gaz à effet de serre que l'aviation, 21 octobre 2015, [L'article](#)
- [13] Quelle est l'empreinte carbone d'un ordinateur ?. site GreenIT.fr [L'article](#)
- [14] Candy Pang and Abram Hindle, University of Alberta, Bram Adams, Polytechnique Montréal, Ahmed E. Hassan, Queen's University. What Do Programmers Know about Software Energy Consumption? IEEE Software, 2015 - ieeexplore.ieee.org. Cité 85 fois. [L'article](#)
 - « Programme / Logiciel ou Application », synonymes ou non ?. Apsynet.fr. [L'article](#)
 - Différents types de logiciel d'application. Microordinateur.blogspot.com [L'article](#)
 - La norme CEI 62402 utilise le mot « obsolète » ???
 - IT Explained: Serveurs : <https://www.fr.paessler.com/it-explained/server>

[1] Integrating sustainable development and climate change in the IPCC fourth assessment report. <https://archive.ipcc.ch/pdf/supporting-material/ipcc-4th-assessment-2003-03.pdf>

[2] L'informatique émet plus de gaz à effet de serre que l'aviation. <https://www.pourlascience.fr/sd/informatique/linformatique-emet-plus-de-gaz-a-effet-de-serre-que-laviation-8741.php>