

El programari lliure dins de les TIC

The Green Software

Víctor Buján Fernández ¹
Universitat Oberta de Catalunya
Rambla del Poblenou, 156, 08018 Barcelona
vbujan@uoc.edu

Resum

Mitjançant aquest treball de recerca es vol analitzar com l'ús de tecnologies green com la virtualització i/o computació en el núvol basades en programari lliure fan que es minimitzi l'impacte mediambiental, millorant la sostenibilitat i maximitzant la viabilitat econòmica de les empreses.

A més a més, es vol analitzar quins són els possibles paràmetres d'avaluació per certificar si un projecte de programari lliure té present la sostenibilitat en les seves dimensions diverses.

Finalment, es vol certificar si el desenvolupament de solucions green de programari lliure té cabuda dins del ventall existent de metodologies de desenvolupament de software o es tenen que formular d'altres.

Abstract

Through this research work, we want to analyze how the use of green technologies such as virtualization and / or cloud computing based on free software make minimizing the environmental impact, improving sustainability and maximizing the economic viability of companies.

In addition, we want to analyze what are the possible assessment parameters to certify if a free software project has sustainability in its diverse dimensions.

Finally, it is wanted to certify if the development of green solutions of free software has room within the existing fan of methodologies of software development or have to formulate others.

Paraules clau

Green IT, virtualització, medi ambient, diòxid de carboni, computació en núvol, programari lliure, codi obert, sostenibilitat, metodologia de desenvolupament.

Key Words

Green IT, virtualization, environment, carbon dioxide, cloud computing, free software, open source, sustainability, development methodology.

1. Enginyer de Telecomunicacions, especialitat en telemàtica per la Universitat Oberta de Catalunya. Actualment cursant el Master de Programari Lliure en la mateixa universitat.

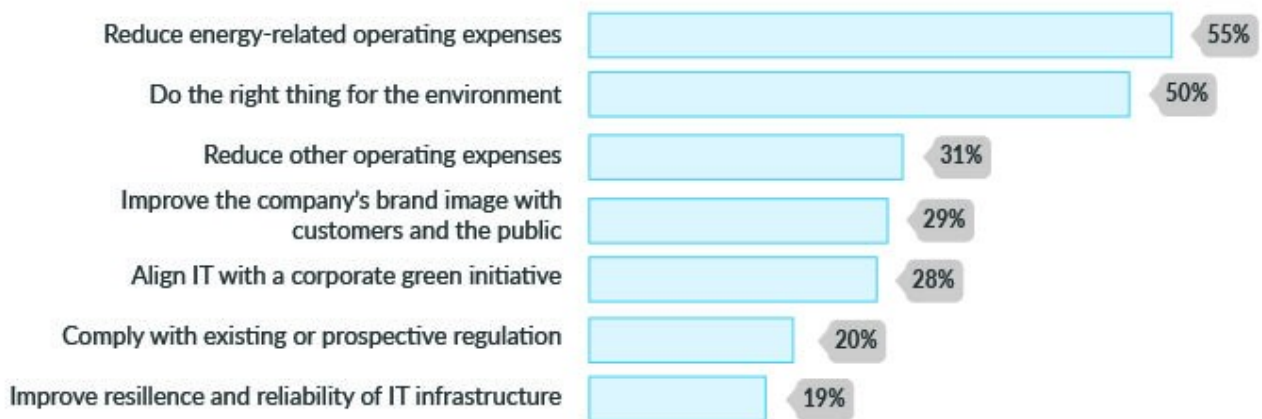
1. Introducció

L'indústria del software és una de les més importants del món i ha tingut una forta evolució [1] des de els seus orígens. Les empreses de software estan orientades a la venda de productes, altres a la venda de serveis, com moltes empreses de programari lliure, i unes altres híbrides, que combinen les dues anteriors. Al ser aquesta una indústria important, la creixent preocupació mundial pel medi ambient ha posat de manifest les emissions i les ineficiències energètiques dels sistemes TIC [2]. Per aquest motiu, les empreses del sector han incrementat la seva sensibilitat i, cada vegada, s'observen més polítiques empresarials encaminades a reduir l'impacte de les tecnologies en l'entorn mediambiental [3] [4]. Si la situació no canvia, el sector tecnològic realitza, actualment, el 2% de les emissions globals de diòxid de carboni, una dada que s'ha de reduir considerablement en un futur pròxim. [5]

Afortunadament, les TIC ofereixen a les empreses moltes eines i innovacions per contribuir a una producció molt més respectuosa amb el medi ambient que en anys anteriors.

Ja des de la dècada passada, les companyies iniciaven aquest procés cap a l'empresa 'verda', com a mostra aquest estudi de l'any 2007 de la Universitat de Missouri [6] sobre les motivacions de les corporacions per decantar-se per les Green IT. Les dos principals motivacions, en aquells temps, era la reducció de consum energètic i el respecte al medi ambient :

What are your organization's top three motivations for pursuing greener IT operations?



Estudi Universitat de Missouri [6]

Actualment, apart de les motivacions esmentades anteriorment, existeixen d'altres com reduir la petjada de carboni, combatre l'obsolescència programada del parc computacional o reduir el número de màquines en les corporacions. Aquí és on entra en joc com a protagonista el programari lliure. Aquest tipus de software permet utilitzar els recursos computacionals d'una empresa de forma ecològica i sostenible, reduint l'impacte mediambiental i maximitzant la viabilitat econòmica de la corporació. Per tant, permet que una empresa es converteixi en una Green IT utilitzant la virtualització, la computació en el núvol o la computació grid entre d'altres.

2. La Virtualització

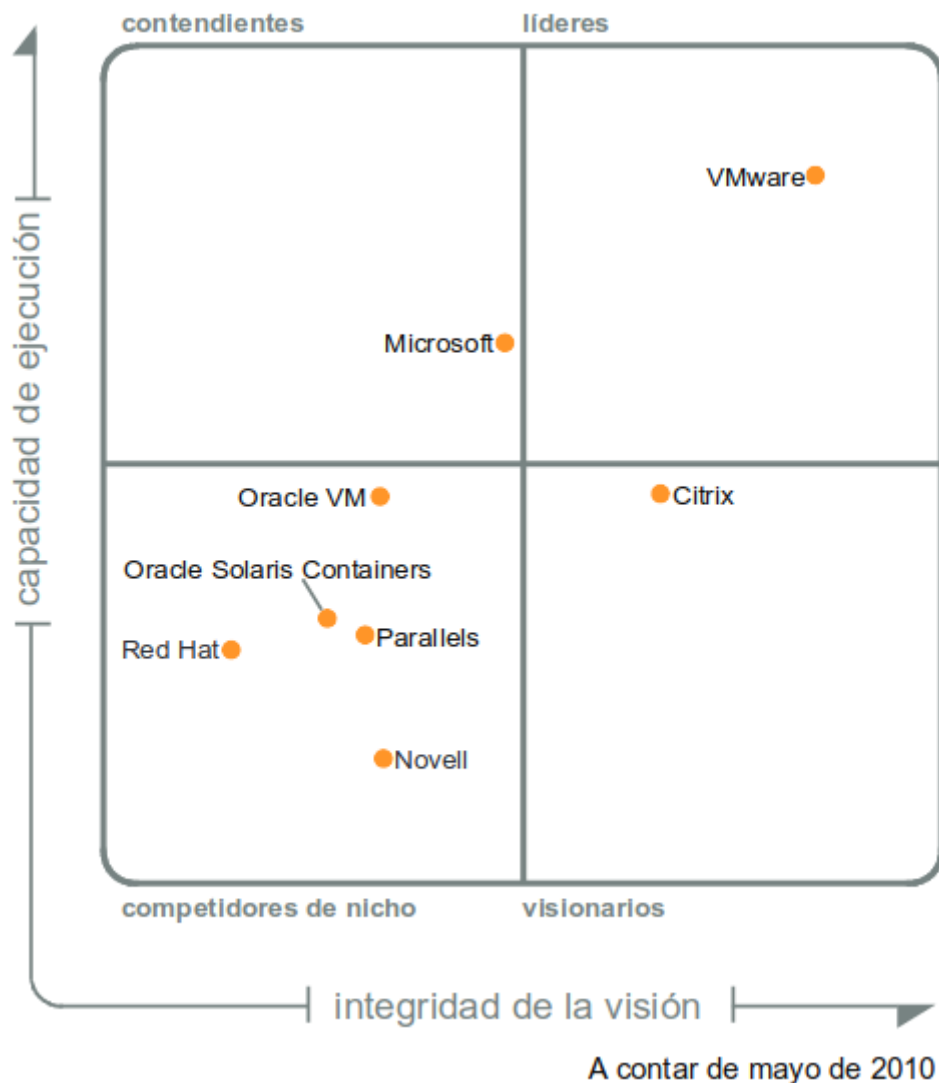
Es defineix la virtualització com la possibilitat de crear múltiples entorns simulats o recursos dedicats des d'un sol sistema de maquinari físic. El programari anomenat *hipervisor* es connecta directament amb el maquinari i permet dividir un sistema en entorns separats, diferents i assegurats, els quals es denominen *màquines virtuals* (VM). Aquestes VM depenen de la capacitat de l'hipervisor de separar els recursos de la màquina del maquinari i distribuir-los adequadament.

La màquina física original equipada amb l'hipervisor es denomina *host*, i les VM que utilitzen aquests recursos es diuen *guests*. Aquests tracten als recursos informàtics, com el processador, la memòria i l'emmagatzematge, com un hangar de recursos que poden traslladar-se amb facilitat. Els operadors poden controlar les instàncies virtuals de la CPU, la memòria, l'emmagatzematge i altres recursos perquè els guests rebin els recursos que necessiten quan els necessiten. [7]

Els beneficis d'utilitzar la virtualització són múltiples :

- *Reducció de costos de hardware*: s'aconsegueix una reducció de servidors que pot suposar un gran estalvi a una corporació.
- *Reducció del consum elèctric* : es disminueix el diòxid de carboni emès a l'atmosfera en reduir el nombre de màquines.
- *Millora de TCO i ROI*: en reduir el nombre de màquines es disminueix el cost total de la propietat (total cost of ownership, TCO) i per tant s'aconsegueix una tornada de la inversió (return of investment, ROI) millor. Si s'afegeix que es genera un estalvi en la factura elèctrica, el ROI serà encara més alt.
- *Reducció dels costos d'espai*: en tenir menys màquines es necessita menys espai físic per poder desplegar el centre de processament de dades (CPD).
- *Administració global centralitzada i simplificada*: són menys màquines físiques les que cal administrar, a més de que el programari de virtualització ajuda a la gestió remota de les màquines virtuals.
- *Ràpida incorporació de nous recursos per als servidors virtualitzats*: es tracta d'una tecnologia escalable on és fàcil l'incorporació de noves màquines virtuals en una màquina amfitriona.
- *Millora en els processos de clonació i còpia de seguretat del sistema*: major facilitat per a la creació d'entorns de prova que permeten utilitzar noves aplicacions sense afectar a la producció en un entorn controlat agilitant el procés.
- *Aïllament*: una fallada general de sistema d'una màquina virtual no afecta a la resta de màquines virtuals

La virtualització va ser i continua sent una de les tecnologies estratègiques per fomentar les Green IT. L'any 2010, la famosa consultora Gartner va publicar gràfic mostrant la situació de les empreses més importants: [8] :



Quadrant de Gartner sobre virtualització (2010) [8]

Podem veure com l'empresa VMWare, en aquells temps, liderava el quadrant en solitari, i que com a competidors, es trobaven empreses del sector del programari lliure com Novell, creador de SuSE Linux, Red Hat o Oracle VM. Ens aquells temps, el software lliure ja era una opció a tenir en compte per poder virtualitzar els servidors d'una corporació.

Era evident que VMware tingués cada vegada més competència a causa de la creixent adopció de serveis d'infraestructura de núvol, la pròpia saturació del mercat, i el creixement de Microsoft en aquest camp gràcies a Hiper-V i Azure [9] que només usaven Windows, el nombre de servidors que utilitzaven aquest sistema operatiu, i el seu menor cost d'implantació fan que el software lliure li costés incrementar quota de mercat.

L'últim quadrant sobre virtualització publicat per Gartner és d'agost del 2016 [10].

Continua amb Microsoft i VMWare al capdavant però trobem un Red Hat i la seva plataforma OpenStack en el quadrant de visionaris :

Magic Quadrant

Figure 1. Magic Quadrant for x86 Server Virtualization Infrastructure



Quadrant de Gartner sobre virtualització (2016) [10]

En aquell any, era significatiu l'augment d'OpenStack² de Red Hat, que ocupa el tercer lloc en el quadrant, i que estava prop de convertir-se en la plataforma insígnia per executar càrregues de treball en servidors Linux virtualitzats.

La virtualització en codi obert té moltes opcions a tenir en compte, com són les següents :

- **KVM**³
- **XEN**⁴
- **QEMU**⁵

2. [OpenStack](#)
3. [KVM](#)

4. [XEN](#)
5. [QEmu](#)

3. Cloud Computing [11]

Cloud computing, molt sovint denominat simplement “cloud” o “núvol”, consisteix en el subministrament de recursos informàtics sota demanda, des d'aplicacions fins a centres de dades, a través d'Internet i basat en un model de pagament per ús.

El fet d'utilitzar Cloud s'obté:

- Recursos flexibles: Es pot augmentar o disminuir els recursos de forma ràpida i senzilla per cobrir la demanda
- Servei mesurat perquè només es pagui pel que es utilitza.
- Autoservei: tots els recursos de TIC que es necessita amb accés d'autoservei

El Cloud Computing es pot dividir en :

- **Programari com a servei (SaaS)**

Les aplicacions basades en cloud, s'executen en sistemes distants, que pertanyen i són administrats per uns altres i que estan connectats als sistemes d'usuari a través d'Internet.

- **Plataforma com a servei (PaaS)**

Proporciona un entorn basat en cloud amb tots els requisits necessaris per donar suport a tot el cicle de vida de creació i engegada d'aplicacions basades en web, sense el cost i la complexitat de comprar i gestionar el maquinari, programari, aprovisionament i allotjament necessari.

- **Infraestructura com a servei (IaaS)**

Proporciona a les empreses recursos informàtics, incloent servidors, xarxes, emmagatzematge i espai en centre de dades amb pagament en funció de l'ús.

Un dels projectes estrella del software lliure que combina la virtualització i el cloud computing és **Openstack** [12].

El projecte OpenStack es defineix així mateix com una plataforma de cloud computing feta amb programari lliure per desplegar núvols públics i privades, desenvolupada amb la idea de ser senzilla d'implementar, massivament escalable i amb moltes prestacions. OpenStack proporciona una solució d'Infraestructura com a servei (IaaS) a través d'un conjunt de serveis interrelacionats.

Hi han altres projectes de cloud computing importants sota programari lliure com es OpenNebula ⁶

6. [OpenNebula](#)

4. Metodologia de desenvolupament del software green [14] [15] [16]

Fins fa poc, la major part del treball green realitzat dins de la indústria de TIC era relacionats amb l'àrea del maquinari, centrant-se principalment en la millora de l'eficiència energètica del hardware. El maquinari és, per descomptat, fonamental, però el maquinari i el programari junts formen un conjunt complet; un no té sentit sense l'altre. Per tant, sembla evident que cal continuar investigant per incloure el programari, encara que la tendència ha anat canviant en els últims anys, i nous projectes relacionats amb l'àrea del programari han sorgit en benefici de la sostenibilitat.

Les corporacions tenen diverses àrees en què cal aplicar la sostenibilitat en el programari: sistemes i productes de software, aplicacions web, centres de dades, i un llarg etcètera. Qui té major pes en tot això són els centres de dades, que consumeixen energia de forma redundat i ininterrompuda, d'una forma molt més significativa que qualsevol altra secció d'una corporació. El principal objectiu per crear i/o emprar programari sostenible és per millorant el consum d'energia. El hardware ha anat evolucionant per ser més eficient enèrgicament però el software no ha seguit el mateix camí. El cicle de vida del desenvolupament de programari i les metodologies de desenvolupament, poques vegades, consideren l'eficiència energètica com un objectiu clau que s'ha d'aconseguir. Això es degut a que mai ha sigut un requisit indispensable en el desenvolupament de programari, i per tant hi ha una gran millora a realitzar.

El software té un paper clau, tant com a part del problema com part de la solució. El programari sostenible és eficient en l'energia, minimitza l'impacte ambiental de la processos que recolza, i té un impacte positiu en la sostenibilitat social i/o econòmica de la corporació. Aquests impactes poden tenir lloc de forma directa (energia) o indirecta (mitigat pel servei que es realitza).

Es consideren que els impactes directes estan relacionats amb els recursos i consum d'energia durant la producció i ús del programari. Els impactes indirectes són efectes derivats de l'ús del producte de programari, juntament amb altres processos i efectes sistèmics a llarg termini.

Una part de la sostenibilitat del programari és la sostenibilitat de l'enginyeria del programari. Normalment, la sostenibilitat s'ha de tenir en compte des de les primeres etapes de desenvolupament de software. Això no sempre és factible, ja que no és fàcil de canviar la forma de desenvolupar de les empreses encarregades de fer-ho. A més, hi ha poca orientació, actualment, sobre com l'enginyeria de programari pot contribuir a millorar la sostenibilitat dels sistemes en desenvolupament.

Per tant, un programari verd i sostenible és un software que el consum que requereix és directe i/o indirecte dels recursos naturals, que sorgeixen del desplegament i utilització del mateix, que es monitoritza, s'analitza de manera contínua, s'avalua i s'optimitza ja en el procés de desenvolupament. Els processos de desenvolupament i producció avaluen i minimitzen cíclicament el consum directe i indirecte de recursos naturals i d'energia.

Es presenten cinc dimensions de sostenibilitat que són importants per poder realitzar un anàlisi de sistemes de programari:

- *Sostenibilitat individual*: es refereix al manteniment del bé privat del capital humà individual. Salut, educació, habilitats, coneixement, lideratge i l'accés als serveis constitueix un capital humà.
- *Sostenibilitat social*: això significa mantenir el capital social i preservar la solidaritat de les comunitats socials. El capital social són inversions i serveis són bàsics per a la societat.

- **Sostenibilitat econòmica:** es tracta de mantenir els actius. Els actius inclouen, no només capital, sinó també valor afegit. Això requereix una definició d'ingressos com la "quantitat" que es pot consumir durant un període i encara estar al final del període, ja que es dedica a consumir un valor afegit (interès), en comptes de capital.
- **Sostenibilitat ambiental:** pretén millorar el benestar humà protegint els recursos naturals com l'aigua, la terra, l'aire, els minerals i l'ecosistema. Inclou les fonts de matèries primeres utilitzades per a les necessitats humanes.
- **Sostenibilitat tècnica:** des d'un punt de vista de l'enginyeria de sistemes (programari) hi ha una altra dimensió que cal tenir en compte, la sostenibilitat tècnica que té l'objectiu central de l'ús prolongat dels sistemes i la seva adequada evolució amb canvi de les condicions de l'entorn i requisits respectius.

El software sostenible es pot dividir en “sostenibilitat gràcies al programari” i “sostenibilitat inclosa en el programari”. La principal diferència és l'objectiu que es vol aconseguir. Si la finalitat és tenir un programari més respectuós amb el medi ambient (*green by software*) o si és millor produir programari que ajudi al medi ambient (*green in software*). La següent taula mostra aquest concepte :

Green Software	
Green BY software	Green IN software
Dematerialització	Processos
Gestió Grid	Productes
Cloud Computing	Cicle de vida
Virtualització	Gestió

L'enginyeria de programari green és la suma del *green by software* i del *green in software*. La definició de l'enginyeria verda es pot realitzar com el conjunt de pràctiques que apliquen els principis d'enginyeria al programari tenint en compte aspectes mediambientals. El desenvolupament, l'operació i el manteniment del programari es realitzen de manera sostenible.

El *green by software*, en general, cobreix programari desenvolupat que té com a finalitat, la preservació del medi ambient, així com un programari que ajuda a gestionar aplicacions intensives en energia, com són els centres de processaments de dades en les empreses.

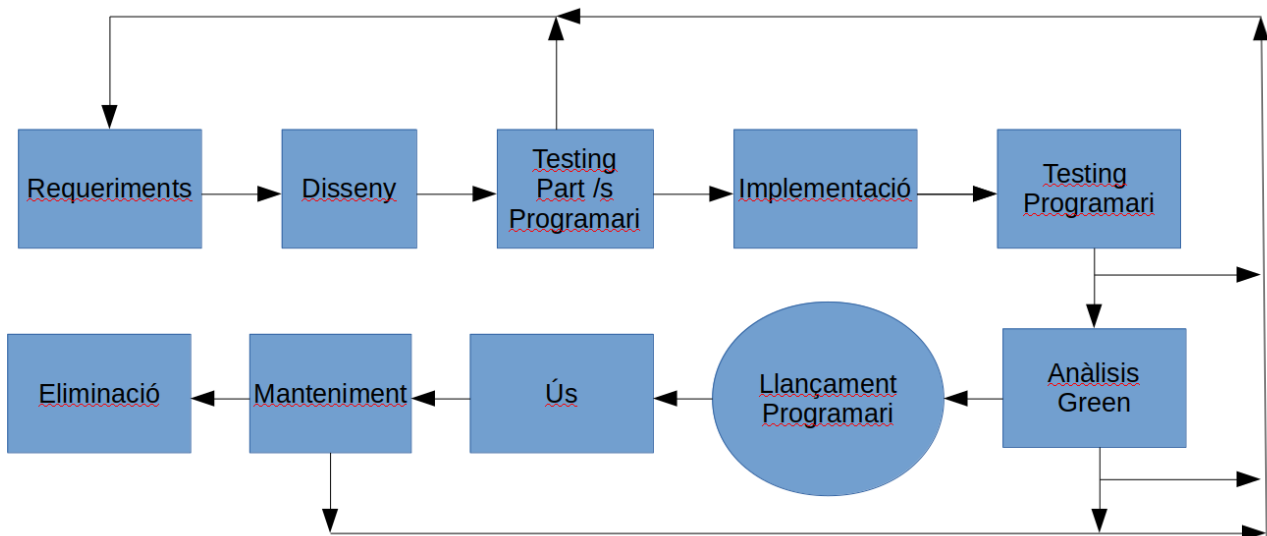
D'altra banda, el *software in green* està relacionat amb la manera de desenvolupar aquest software, i per tant, que proporcioni un producte i/o servei més sostenible. Aquí és on ens aturem i observem que les metodologies pròpies i actuals de l'enginyeria del programari, encara, no prioritzen paràmetres sostenibles en el programari a desenvolupar. Per tant, detectat el problema, la solució és la formulació d'una metodologia de desenvolupament de software perquè aquest aspecte sigui contemplat.

5. Disseny d'una metodologia com aportació de recerca. [16] [17] [18] [19] [20] [21] [22] [23]

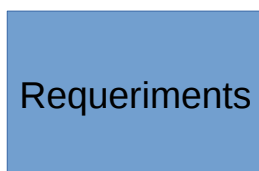
Per intentar donar solució, s'escolleix en dissenyar un model de desenvolupament de software lliure que cobreixi tots els aspectes que poden afectar al medi ambient. Un dels models més populars que tenen la mateixa finalitat és el model «*greensoft*» proposat per Naumann.

El model dissenyat dóna suport per crear, mantenir i utilitzar el programari de manera verda i s'organitza en quatre parts. Les parts inclouen el cicle de vida del producte que es produeix, mètriques que s'han d'afegir com a criteris d'avaluació, models de procediments i eines recomanades per a desenvolupadors, compradors i usuaris. El model intenta abordar la sostenibilitat i obliga a integrar-se en el difícil procés d'enginyeria de programari d'una manera senzilla.

El model dissenyat pren l'essència dels mètodes de desenvolupament de programari seqüencials, iteratius i àgils per formar un cicle de vida sostenible. El procés d'enginyeria de programari popular i genèric està format per set etapes principals: requeriments, disseny, implementació, comprovació, ús, manteniment i eliminació. La metodologia proposada està formada de nou etapes :



A continuació es comenten cadascuna de les etapes :



Els requeriments és la primera etapa i són les descripcions dels serveis que ha de proporcionar el sistema i qualsevol limitació. El treball del enginyer és saber la necessitat que el client vol cobrir i ,per tant, aquest identifica l'esbós dels serveis que desitja per al sistema a realitzar.

Aquesta etapa tenim una fase de factibilitat, que és determinar si el projecte és viable. Seguidament una fase on es realitzarà un esbós dels serveis que es volen proveir. La tercera fase és una anàlisi de riscos inspirada en el model espiral iteratiu, però en termes d'energia. Basant-se en l'anàlisi d'aquesta fase es pot realitzar una alteració dels requisits. La fase final és de prova de requeriments que s'ha d'afegir a un procés de requisits ambientalment sostenible.

Disseny

En l'etapa de disseny, es crea l'arquitectura general del sistema en funció dels requisits que figuren en aquesta etapa. Hi ha un nombre d'activitats de disseny que configuren el nivell de sostenibilitat del component de programari, com el disseny arquitectònic, l'especificació abstracta, el disseny de l'estructura de dades i el disseny d'algorismes.

A continuació es detallen tres pautes de decisió de disseny que promouen el medi ambient sostenible en aquesta etapa :

1. Els programadors han d'escriure algorismes eficients mitjançant el disseny compacte de codi i estructures de dades basades en l'aplicació, el llenguatge de programació i l'arquitectura del hardware.
2. Els subsistemes han d'adherir-se a les funcions presentades en la fase de requisits i s'han de dissenyar per minimitzar-se amb una funcionalitat eficient, per tant, produeixen un algoritme compacte i una menor quantitat de línies de programa en la fase d'implementació.
3. L'efecte dels entorns de reutilització i desenvolupament d'aplicacions perjudica en termes d'eficiència energètica a causa de les capes de programari addicionals que no són necessàries i que s'afegeixen al codi. Aquestes requereixen un treball extra del processador.

Testing Part /s Programari

En el procés de proves es pot descobrir que el programari no compleix els seus requeriments o el funcionament sigui incorrecte. El procés de testing pugui ser perjudicial per al medi ambient és perquè és iteratiu. Això significa que si el client no aprova el programari, hauria d'indicar-lo als desenvolupadors, i aquests han de tornar a fer canvis en la fase de requeriments i en la fase de desenvolupament. Per tant, més hores de treball i més energia en general. Per a una fase més ecològica i més eficient, els conceptes s'han de desenvolupar abans, com en la fase de requeriments per assegurar que el software s'ajusta a les necessitats de l'usuari.

A continuació es presenten dues mètriques d'enginyeria de programari o atributs de qualitat que poden ser específicament útils per mesurar la sostenibilitat ambiental del sistema de programari en l'etapa de prova. Si milloren aquestes mètriques, pot conduir a una fase de proves més sostenible per al medi ambient. Per a cada mètrica, proporcionem la seva definició primer i, en segon lloc, el benefici ambiental que es pot aconseguir.

1. *Tolerància a fallades*: permet que un sistema continuï operant a un nivell reduït en comptes de fallar completament.
Minimitza els residus ambientals permetent que el sistema funcioni fins i tot amb defectes en comptes de deixar de ser operatiu i requerir reparació.
2. *Gestió de fallades*: implica identificar, respondre i reparar fallades de programari.
Moltes vegades s'associa una fallada de software amb una fallada de hardware. Una bona gestió de fallades pot reduir els residus de maquinari.

Disposar de una fase de prova abans de començar el desenvolupament permet garantir que el programari funcioni tal com es pretenen els requisits, A més a més es confirma un procés de verificació de programari verd i sostenible, disminuint la quantitat d'iteracions que condueixen a tornar a les etapes anteriors.

Anàlisi Green

L'etapa d'anàlisi verda és una etapa que s'afegeix al procés d'enginyeria del programari per promoure l'eficiència energètica. Es tracta d'un escenari que presenta noves idees sobre sostenibilitat mediambiental que no s'han considerat abans. Aquesta etapa determina el nivell «green» de cada part del sistema que s'està desenvolupant. Funciona com una etapa de proves, però per a l'eficiència energètica. Les mètriques que es poden utilitzar, per exemple, és l'ús dels recursos TIC com la CPU i/o mètriques de qualitat, com ara el rendiment dels indicadors de rendiment verd acceptats pels projectes de la UE [17].

Les mètriques són criteris d'avaluació que mesuren la quantitat d'energia desaprovechada utilitzada de manera eficient i, com a resultat, indiquen com és de verd un programari. Es poden crear eines per fer un seguiment, identificar i limitar la quantitat d'energia que utilitzen les aplicacions. Aquestes eines poden ser necessàries per executar l'aplicació, o utilitzar-se durant tot el procés de desenvolupament, i aquests conceptes es poden implementar en l'aplicació de programari que s'està construint.

L'ús de mètriques que mesuren i descriuen les despeses totals del cicle de vida del procés proporcionen als desenvolupadors una visió clara del verdor de cada etapa. S'anomenen *indicadors de cost del cicle de vida*. Aquestes mètriques tenen en compte paràmetres potencials com ara l'experiència del desenvolupador, la complexitat principal de l'operació de servei, el nivell d'abstracció, la reutilització i l'integració, l'estabilitat requerida i la proximitat de l'aplicació al nucli comercial. Si tenim en compte el mètode de cost de desenvolupament, els paràmetres que es consideren són els esforços realitzats pels desenvolupadors durant el cicle de vida de l'aplicació.

Ús

Etapa que es basa en com hauria d'utilitzar l'usuari el programari de forma eficient. Perquè això succeeixi, la responsabilitat no només cau sobre aquest sinó també sobre els propis desenvolupadors. A continuació, es proporcionen dues directrius verdes en dos grups d'interès: requisits del programari i desenvolupament de programari.

Requisit del programari : s'ha d'escriure funcions que promoguin conductes sostenibles pels usuaris.

Desenvolupament de programari: Els desenvolupadors poden crear programari que participi en funcions de gestió d'energia a l'ordinador per permetre que els recursos s'utilitzin de forma eficient.

Manteniment

El manteniment del programari o l'evolució del mateix, consisteix a canviar un software després d'haver estat lliurat a causa de noves modificacions o millores. Aquesta fase és la més costosa de tot el procés d'enginyeria del programari. El cost aquí és proporcional al malbaratament energètic. Els canvis realitzats a l'aplicació poden ser senzills, com l'optimització de codi o canvis més difícils de corregir com són modificacions dels requeriments o afegir-ne d'altres.

S'ha de dur el procés de canvi energètic de forma eficient. És important que durant l'etapa de desenvolupament s'escrigui principalment un programa per a una bona estructuració i comprensibilitat. Si un programa s'entén fàcilment, serà molt més ràpid i més fàcil aplicar-hi canvis i això reflectirà l'eficiència energètica.

Eliminació

Aquesta etapa suposa la substitució de programari ja que no s'utilitza o s'ha quedat obsolet.

A continuació es detalla una pauta d'eficiència energètica :

1. Reciclatge de programari en termes de reutilització del codi per a futurs projectes, mantenint baixos els costos de desenvolupament intern.

6. Diferències respecte altres models.

Hi ha molts models de desenvolupament de software i, per tant, es troben models amb diferents variacions de les etapes i moltes empreses adopten les seves pròpies però totes tendeixen a tenir patrons similars.

Els processos d'enginyeria del programari solen reflectir un enfocament iteratiu del cicle de vida o un enfocament seqüencial. El model proposat l'enfoc és iteratiu en lloc del seqüencial per raons green, a més a més, d'incorporar alguns dels principis que es troben en metodologies àgils per ajudar a incloure la sostenibilitat ambiental.

Els models seqüencials lineals tenen desavantatges que afecten el procés green del programari. En un cicle de vida seqüencial, cadascuna de les fases s'ha de completar abans que la següent pugui començar, sense deixar espai per a la possibilitat de canvis en els requisits que poden generar canvis en el disseny i la implementació. Si es produeixin canvis, les modificacions són costoses i deriven a un increment d'ús d'energia. Els models de cicle de vida seqüencials funcionen bé només quan els requisits són ben entesos i poc probable que canviïn. A favor, els models seqüencials també poden promoure pràctiques d'eficiència energètica. com desenvolupar proves abans de la seva implementació, amb una major probabilitat d'èxit. Això pot evitar tornar a corregir errors i canviar els requisits quan es revelen problemes després de la implementació. Trobar errors i fer-ho de forma anticipada sempre és una bona pràctica energètica.

El model proposat no implementa l'idea en models iteratius on petites evolucions del sistema es lliuren anticipadament als usuaris sinó que s'implementa el concepte de desenvolupament iteratiu desenvolupant evolucions molt petites i no proporcionar-ho als usuaris finals de manera que si cal alguna modificació, es pot fer. El proposat incorpora un segon concepte de desenvolupament iteratiu que es beneficia del desenvolupament primerenc incremental que permet aclarir fàcilment els requisits per a increments posteriors, oferint-los una major possibilitat de ser immutables i correctes, donant lloc a l'eficiència energètica. D'aquestes dues maneres s'incorpora el principi d'agilitat relacionat amb acollir canvis en els requisits. El model també té una essència de cicles de vida seqüencials perquè incorpora proves primerenques abans de desenvolupar codi.

Respecte al model competidor directe, el model greensoft de Naumann, falta les fases de requisit i prova. Aquestes dues etapes tenen un paper important en la producció d'un producte de programari verd i sostenible i poden estimular els efectes sobre el medi ambient.

Conclusions

L'impacte de les TIC en el desenvolupament sostenible ha de ser clar. El programari, i en especial el software lliure, s'ha de desenvolupar amb la finalitat d'optimitzar l'ús i el consum d'energia en les corporacions tecnològiques. Per tant, no n'hi ha prou amb centrar-se en construir aplicacions/ productes de programari verds, s'ha de mirar el programari des del punt de vista que pot ajudar a controlar i utilitzar recursos de manera eficient.

Les tecnologies de virtualització i cloud computing, tan importants per convertir una corporació en Green IT, tenen un creixement positiu i nombrosos proveïdors de programari lliure, disposen d'una oferta cada vegada més sòlida, estan evolucionant l'oferta dels seus serveis per consolidar les solucions amb més acceptació al mercat.

A mesura que sorgeixen noves tecnologies i es produeixen avenços en el tema de l'enginyeria del programari, un nou enfocament en la metodologia de desenvolupament ha d'assegurar que la sostenibilitat ha vingut per quedar-se. És cert que grans esforços s'han fet en els últims anys, però és treball de tots els que formen part de les TIC, el mantenir la responsabilitat de triar i exigir un programari que sigui més respectuós amb el medi ambient.

Si s'aconsegueix, les companyies hauran d'adoptar processos de programari més ecològics, i el software lliure té una bona oportunitat per augmentar la seva quota de mercat. El camí per recórrer encara es llarg, ja que a mesura que el programari verd es desenvolupa i s'enforteix, com a contrapartida, també sorgiran molts desafiaments que s'haurà de fer front per a benefici de tothom.

Referències bibliogràfiques

- [1] [The Business of software - Michael A. Cusumano](#)
- [2] [Green IT for Sustainable Business Practice - Mark O'Neill](#)
- [3] [Harnessing Green IT: Principles and Practices – Sam Murugesan](#)
- [4] [Green IT in Practice: How One Company is Approaching the Greening - Gary Hird](#)
- [5] [Why Green IT is Better IT](#)
- [6] [Green IT. Universidad de Missouri](#)
- [7] [Understanding Full Virtualization, Paravirtualization, and Hardware Assist](#)
- [8] [Cuadrante Mágico para infraestructura de virtualización de servidores x86](#)
- [9] [VMware and Microsoft are the top virtualization leaders, according to Gartner](#)
- [10] [Microsoft a leader in Gartner x86 Server Virtualization Infrastructure 2016 Magic Quadrant](#)
- [11] [Cloud Computing Tecnología Y Negocio - Beltran Pardo Marta](#)
- [12] [OpenStack y virtualización, ¿en qué se diferencian?](#)
- [13] [Safety, security, now sustainability: The nonfunctional requirement for the 21st century](#)
- [14] [Is software green? Application development environments and energy efficiency in open source applications – Eugenio Capra, Chiara Francalanci, Sandra A. Slaughter](#)
- [15] [Software libre : técnicamente viable, económicamente sostenible y comercialmente justo – Jordi Mas i Hernández](#)
- [16] [The Green Soft Model : A reference model for green and suitable software – Stefan Naumann, Markus Dick, Eva Kern, Timo Johann](#)
- [17] [Green Performance Indicators for Energy Aware IT Systems - Sara Mahmoud, Imtiaz Ahmad](#)
- [18] [Discovering Sustainability Requirements: An Experience Report - Mahaux, Heymans, Saval](#)
- [19] [Green software development model: An approach towards sustainable software development](#)
- [20] [Supporting Sustainability Aspects in Software Engineering - Penzenstadler](#)
- [21] [Empirical studies of Agile Software Development: A Systematic Review – Dyba & Dingsoyr](#)
- [22] [Incorporating Green IT concepts in undergraduate software requirements engineering course](#)
- [23] [Measuring the sustainability performance of software project - F.Albertao, J.Xiao, C.Tian, Y](#)