

## **CAPÍTULO 5. EL SOFTWARE LIBRE COMO PIVOTE TECNOLÓGICO PARA LA EVOLUCIÓN NACIONAL HACIA LA SOCIEDAD DE LA INFORMACIÓN Y EL CONOCIMIENTO**

El desarrollo tecnológico va ligado siempre a una necesidad, solo con el Software Libre podemos encontrar una posible solución tecnológica a un problema de una localidad, ya que se hace obligatorio poseer una masa de conocimiento y herramientas de análisis con las cuales se pueda diseñar una solución viable, capaz de cubrir adecuadamente la necesidad. La alternativa ideada será válida cuando el grupo de personas que integran dicha localidad y que tenían la necesidad se encuentren satisfechos con la tecnología desarrollada.

De esta forma es posible decir que se ha presentado un desarrollo tecnológico cuando la tecnología y los desarrolladores de la misma pertenecen a la misma localidad (un pueblo, una región o un país). Sin embargo, el uso de la tecnología no implica un desarrollo de la misma, lo cual nos lleva a afirmar que en este caso existe un estancamiento tecnológico ya que no se tiene una capacidad autosuficiente para adecuar la tecnología a las necesidades particulares de la localidad. Se hace evidente que el desarrollo tecnológico es un proceso continuo que involucra a todas las personas de una localidad, por lo que el conocimiento libre tiene un papel protagónico, comenzando por las escuelas y universidades, las cuales tienen la obligación de convertirse en un generador de conocimiento útil.

Como ya se ha observado, la realidad en México muestra que nos encontramos con una falta de producción de tecnología de software. Un factor importante han sido los proyectos puestos en marcha por el gobierno que no han considerado las áreas de investigación y desarrollo local. En este sentido, cualquier iniciativa que involucre desarrollo tecnológico nos llevará sin lugar a dudas a considerar la obtención de la tecnología fuera del país. Este aspecto nos subordina a una posición de dependencia tecnológica, aceptando las condiciones que el proveedor de la tecnología desee otorgar.

En América Latina se están definiendo posturas ante las necesidades informáticas en la región, países como Cuba, Venezuela, Uruguay entre otros, han escuchado propuestas de los proveedores de tecnología y han elegido utilizar Software Libre al menos en uno de sus

proyectos nacionales debido a que detrás de él ven una filosofía de desarrollo, un modelo de implementación, un sistema de distribución, un conjunto de licencias que involucran los derechos de autor, entre otras características. Pero tal vez, la característica más apreciada del Software Libre radica en la ausencia de una empresa o conjunto de empresas que lo promuevan; detrás del Software Libre se encuentra una comunidad que puede integrar a los países en la SIC.

El software es un elemento de gran importancia para el desarrollo tecnológico de una región, dentro del campo de las TIC el software es el motor de la tecnología aplicada, no existe equipo de hardware que pueda funcionar sin algún tipo de software.

Dentro de una comunidad, el software tiene una característica muy particular que hace diferencia dentro los elementos TIC: La propiedad del software no se adquiere<sup>81</sup>, es decir, se obtiene un permiso o licencia que habilita al usuario ciertas libertades o derechos para su uso, copia, modificación y distribución.

Generalmente, si hablamos de un modelo de Código Cerrado, obtendremos únicamente una licencia de uso la cual permite al usuario únicamente usar el programa para el fin específico que el autor señale o para no usarlo, ya que no existe obligatoriedad de uso.

Las licencias antes mencionadas influyen directamente en el elemento para el cual se han desarrollado las TIC: la información. Al contrario del software, la información si es propiedad del usuario y en muchos casos es generada por uno o varios usuarios del software.

En el caso de la información almacenada en archivos de un formato “no libre”, el usuario tendrá acceso a su información siempre y cuando cumpla con las condiciones de licenciamiento del software que ha impuesto el proveedor. Así es como aparece una de las grandes barreras para el desarrollo tecnológico, al compartir la información generada por un usuario, el proveedor obliga a que el receptor de dicha información acepte también las mismas condiciones de licenciamiento. Esto limita la generación y divulgación del conocimiento existente en una comunidad debido a que esta latente el riesgo de la imposibilidad de acceso, filtración y manipulación de datos.

---

81 PILAS, Rodolfo., PÉREZ, Manuel. Software Libre para el desarrollo tecnológico. Montevideo: Asociación para el Progreso de las Comunidades (APC), 2001.

Todo lo anteriormente señalado se agrava terriblemente cuando el usuario que obliga a los demás es el propio Estado. En el caso mexicano, al usar Software de Código Cerrado, el Estado propicia que los ciudadanos acepten licencias y paguen regalías a proveedores, generalmente extranjeros, para cumplir con sus deberes como miembros de la sociedad civil. Adicionalmente también se condiciona una dependencia al proveedor cada vez que se requiera un cambio, corrección, mejora u optimización del software.

En este sentido, lo que en un principio era decisión del usuario ahora es decisión de un proveedor. Por el contrario, el Software Libre hace al usuario independiente del proveedor y ahora la vinculación se da mediante la calidad del producto y los servicios prestados. El proveedor brindará su mejor trabajo intelectual y el usuario decidirá cuando pagar por un servicio adicional. Además la información es liberada de la exigencia comercial y es posible impulsar la compartición y generación de conocimiento.

La consecuencia inmediata de la adopción del Software Libre es un intercambio tecnológico completo ya que el usuario dispone de toda la tecnología informática involucrada en sus sistemas, facilitando así el desarrollo tecnológico.

Estos argumentos se vuelven cruciales cuando hablamos de definir políticas que involucran al conjunto de nuestra sociedad y su desarrollo tecnológico. Adoptando Software Libre estaremos entonces entrando en el terreno de la información comunitaria, propiedad de los ciudadanos<sup>82</sup>.

También desde los puntos de vista social y estratégico es imperativo el uso del Software Libre, pues no solo es la única manera de garantizar la democratización del acceso a la información y los sistemas del Estado, sino también la competitividad de la industria local de software, potencial fuente de trabajo de altísimo valor agregado y además fuente y divulgación de conocimiento científico.

Al final nos encontramos de nuevo con el costo del Software Libre el cual permite

---

82 FINQUELIEVICH, Susana. Derechos ciudadanos en la era de Internet: una propuesta tentativa. Sala de Prensa [en línea]. 20 noviembre 2000. [Fecha de consulta: enero 2010]. Sección Artículos. Disponible en: <http://www.saladeprensa.org/art171.htm>

generar nuevas soluciones lógicas sin tener que gastar dinero en la herramientas de desarrollo o en la distribución del conocimiento.

### 5.1. Promoción y aplicación del Software Libre

Como ya se ha mencionado anteriormente, el Software Libre en México no ha sido tomado en cuenta, sin embargo, podemos mencionar ejemplos internacionales que pueden ser aplicables al caso mexicano.

Lo más destacable es que Gobierno del Distrito Federal incluye dentro de sus políticas y lineamientos en materia de informática cierta preferencia al uso del Software Libre. La Delegación Tlalpan en el 2005 creó Gobierno GDF/Linux, basada en Fedora. Además, estudiantes de la UNAM desarrollan el sistema operativo GNU/Linux mediante Jarro Negro el cual puede instalarse o usarse en forma de LiveCD.

En este sentido los países que integran la Unión Europea han tenido avances notables. Las principales conclusiones del informe Study into the use of Open Source Software in the Public Sector. Use of Open Source in Europe<sup>83</sup> indican que el uso del Software Libre en el sector público está concentrado principalmente en los servidores del los cuales la mayor parte sustituyen a anteriores soluciones propietarias de Unix.

No más del 8% de los servidores de las administraciones públicas incluían en 2001 un componente dominante de Software Libre. No obstante, desde entonces el crecimiento ha sido progresivo y rápido,

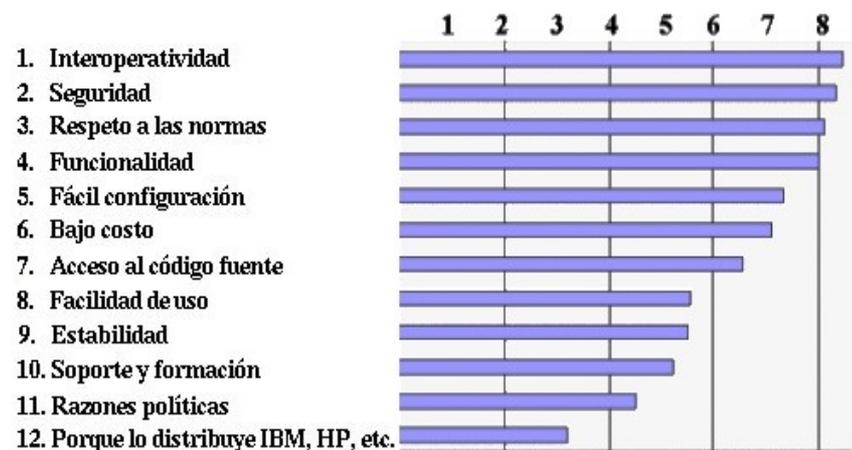


Ilustración 1: Razones para usar SL en la administración pública

especialmente a partir de la presión ejercida en 2003 por parte de algunos países como Francia

<sup>83</sup> COMISIÓN Europea., DG Enterprise. Study into the use of Open Source Software in the Public Sector:

The Open Source Market Structure. 3ª ed. Europa: Comisión Europea, 2001.

y desde que el nivel de satisfacción en la utilización de este tipo de soluciones fue en aumento a partir de la implantación de soluciones basadas en Software Libre para servidores (ver Ilustración 1).

Según las encuestas realizadas por el estudio anteriormente citado:

- El 63% de los encuestados afirmaron utilizar de alguna manera herramientas basadas en Software Libre en su organización.
- El 26.9% de los servidores contaban con sistemas basados en Software de Código Abierto y en la previsión era que en 2003 este número aumentaría hasta en 32.5%

IDABC es un programa de administración electrónica establecido para el periodo 2005-2009. Sustituye al programa IDA (Interchange of Data between Administrations – intercambio de datos entre administraciones), con un campo de acción más amplio, IDABC entra en el marco de las iniciativas eEurope 2005 e i2010. La interoperatividad y las normas abiertas son campos de acción prioritarios para este programa. Además tienen como objetivos:

- Permitir el intercambio entre las administraciones públicas y entre éstas y las instituciones comunitarias.
- Facilitar el suministro de servicios paneuropeos a las empresas y a los ciudadanos, teniendo en cuenta sus necesidades.
- Lograr la interoperatividad entre los distintos ámbitos de acción, en particular, basándose en un marco de interoperatividad europeo.
- Promover la difusión de buenas prácticas y fomentar la elaboración de soluciones telemáticas innovadoras en las administraciones públicas<sup>84</sup>.

---

84 Para difundir las buenas prácticas en soluciones libres dentro de las administraciones públicas europeas, IDABC ha fundado el Open Source Observatory & Repository (Europe): <http://www.osor.eu> donde publica periódicamente información del uso de FLOSS en las distintas administraciones públicas de los estados miembros de la Unión Europea.

La Fundación para el Software Libre Europa tiene como algunos de sus objetivos:

- Dar soporte a todos los aspectos del Software Libre en Europa, especialmente al proyecto GNU.
- Crear conciencia sobre la temática FLOSS.
- Asegurar el Software Libre política y legalmente.

Por estas razones, la FSFE se fundó en 2001 como la organización “hermana” de la Free Software Foundation de Boston, MA, USA. Ambas son financiera , legal y personalmente independientes entre sí, como partes de la red internacional del Software Libre.

El hasta ahora Open Source Observatory que ha servido a la comunidad europea de Open Source forma ahora parte de un ente de servicios de mayor envergadura que el IDABC esta ofreciendo a su comunidad. El nuevo Open Source Observatory and Repository for European Public Administrations ha sido designado como la plataforma para el intercambio de información y experiencias en torno al mundo FLOSS para el sector público europeo.

Con el apoyo internacional que el Software Libre tiene en la región, países como Alemania ya ha hecho avances importantes en la aplicación del Software Libre. Florian Schießl, líder del proyecto LiMux, la distribución de Linux que se está utilizando en la masiva migración a Software Libre en la administración de Munich, publicó un recuento de lo ocurrido durante el año 2009 en donde relata que finalmente se ha logrado la migración completa al formato de archivos ODF<sup>85</sup>.

El formato de archivos ODF ahora es el estándar de documentos internos de Munich, mientras que PDF se está utilizando para distribuir contenido no editable. La estandarización sobre ODF también ha provocado que OpenOffice.org se haya convertido en la suite de oficina oficial para la administración de Munich. Schießl dice que no se trata simplemente de reemplazar una suite de oficina por otra, ya que se requirió un gran esfuerzo para eliminar

---

85 SCHIESSL Florian. Communications manager free software & open stardards [en línea]. Actualizada: 30 diciembre 2009. [Fecha de consulta: diciembre 2009]. Disponible en: <http://www.floschi.info/2009/12/limux-review-2009/>.

todas las dependencias que existían con la suite de Microsoft. No menciona dependencias que involucren usuarios como uno esperaría, sino que dependencias de procesos y software.

Por ejemplo, se tuvieron que migrar y consolidar cerca de 20.000 plantillas que fueron convertidas a nuevas plantillas, macros y aplicaciones web. Para gestionar esta gran cantidad de plantillas desarrollaron una extensión para OpenOffice.org llamada WollMux con la que integraron sus plantillas con servicios internos de la organización, agregando funcionalidad que impacta positivamente en la productividad de sus usuarios.

Algo a destacar es que este proyecto de migración también ha considerado compartir la experiencia con la comunidad, de tal forma que otras administraciones puedan replicar el proyecto en su propia realidad. Así lo han estado realizando en varias conferencias, ayudando directamente a otras administraciones, y publicando sus proyectos como WollMux.

Por su parte España, es el país que ha avanzado más rápidamente en los últimos años debido a que sus administraciones locales han elegido el Software Libre como herramienta para la evolución a la SIC. Uno de los grandes motores de desarrollo ha sido en las escuelas de educación superior.

El 83% de las universidades españolas cuentan con algún grupo de investigación sobre el Software Libre y buscan la transferencia tecnológica de estos estudios a la industria, según el informe sobre el Software de Código Abierto en las universidades y centros de investigación españoles<sup>86</sup>.

El informe fue presentado en la III Asamblea de Morfeo en el Auditorio de Telefónica I+D y refleja que este tipo de software es un componente esencial de la investigación en España.

El estudio ha sido elaborado por el Centro de Referencia de Aplicación de las TIC basadas en fuentes abiertas (CENATIC)<sup>87</sup> y señala que en las universidades españolas, basan

---

86 El 83% de las universidades investigan sobre el software libre. ADN.es [en línea]. Madrid, 10 marzo 2010. [Fecha de consulta: 10 marzo 2010]. Sección Economía. Disponible en: <http://www.adn.es/economia/20100310/NWS-1982-universidades-investigan-software-libre.html>

87 CENATIC es el proyecto estratégico del Gobierno de España cuya misión es fomentar y difundir las TIC

en el Software Libre el 60% de los servidores y el 42% de sus bases de datos.

También, el 67% de su correo institucional; el 86% de sus herramientas de gestión de contenidos, el 67% de sus sistemas antisпам, el 40% de sus campus virtuales y el 44% de sus soluciones de administración electrónica.

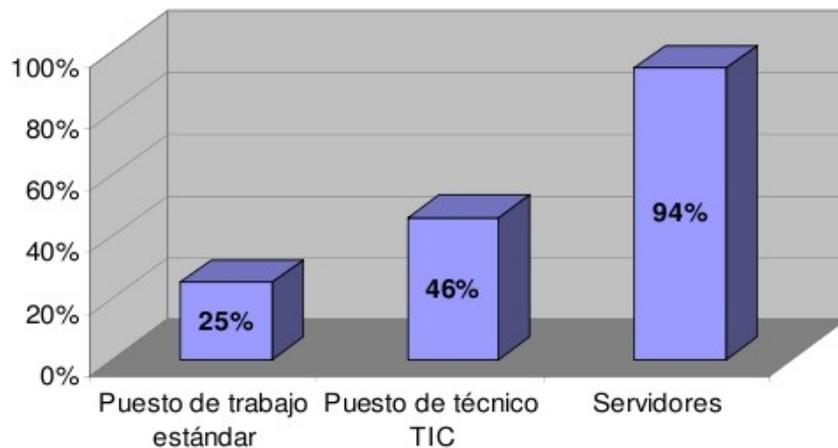


Ilustración 2: Grado de uso del SL para Internet y comunicaciones en Gipuzkoa, España

Además, en el 57% de las universidades españolas existen asociaciones de usuarios de apoyo a las tecnologías basadas en Software Libre.

El informe señala que los principales beneficios de este tipo de software son la flexibilidad, el ahorro en licencias, la independencia de proveedores, la posibilidad de reutilización, la existencia de una comunidad alrededor de los proyectos y las enormes posibilidades de innovación y desarrollo.

La investigación de los centros de enseñanza se ve reflejada en la industria porque las PyME en España empiezan a optar por utilizar Software Libre, esto debido a las limitaciones económicas que enfrentan al momento de pagar licencias de software. Además cuentan con herramientas que les ayudan a escoger la mejor opción a la hora de decidirse por utilizar un nuevo software, por ejemplo, OpenPYME, proyecto promovido por la Oficina de Software Libre de la Universidad de La Laguna, bajo el marco de colaboración científico-tecnológico entre la Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias y la Universidad de La Laguna, tiene por finalidad fomentar y mejorar

---

de fuentes abiertas en todos los ámbitos de la sociedad. Actualmente forman parte del Patronato de CENATIC, además del Ministerio de Industria, Turismo y Comercio, a través de red.es, la Junta de Extremadura, la Junta de Andalucía, el Principado de Asturias, el Gobierno de Aragón, el Gobierno de Cantabria, la Generalitat de Catalunya, el Govern de les Illes Balears y el Gobierno Vasco, así como las empresas Atos Origin, Bull, Telefónica y Gpex.

el nivel de utilización del Software Libre, tanto en el ámbito académico como a la sociedad en general y ha puesto en línea una selecta colección de los mejores programas de negocios diseñados para MS-Windows con el objetivo de acercar las herramientas TIC a las PYME para mejorar su gestión y competitividad incorporando, para ello, tecnología de Código Abierto. Además, se pretende una activación de la demanda de asesoramiento especializado de empresas TIC.

El catálogo de aplicaciones de Código Abierto que ha recopilado el sitio OpenPYME de la Universidad de La Laguna engloba una gran cantidad de programas divididos en las diferentes áreas que más interesan a una empresa para desarrollar su actividad. Estas categorías son inteligencia de negocio, gestión de contenidos, relación con el cliente, distribuciones empresariales, gestión documental, comercio electrónico, recursos empresariales, gestión financiera, servicio e infraestructuras, software colaborativo, Internet, ofimática, gestión de proyectos, Punto de venta y Utilidades.

De esta forma, OpenPYME facilita la descarga de programas de compresión de archivos (7Zip), gestión de comercio electrónico avanzado (Apache OfBiz), de documentos (Knowledgetree), videoconferencia o llamadas IP (Ekiga) o generación de estadísticas (AWStats).

Con el aumento de la demanda de Software Libre por parte de las empresas, se generaron nuevos modelos de negocios para un mercado en crecimiento como son<sup>88</sup>:

- Suscripción: mediante esta metodología, las empresas brindan soporte especializado y actualización del software post suscripción. El cliente accederá a estas prestaciones antes que cualquier otro usuario que no esté suscrito a la empresa.
- Desarrollo modular para aplicaciones libres: se ofrece un servicio de adaptación de Software de Código Abierto sea de ERP, CRM u OpenOffice.org. Por lo general, cuando al software no cuenta con una utilidad necesaria para el cliente se brinda un

---

88 COLETTI, Daniel. Nuevos modelos de negocios con Software Libre. El Argentino [en línea]. Buenos Aires, 15 marzo 2010.[Fecha de consulta: 15 marzo 2010]. Sección Economía. Disponible en: <http://www.elargentino.com/nota-86535-Nuevos-modelos-de-negocios-con-software-libre.html>.

desarrollo personalizado, que después a la vez puede ofrecerse como un Software Libre ya armado, que se seguirá adaptando para otras necesidades.

- **Software como un servicio:** se trata de la posibilidad de utilizar cualquier aplicación que esté en Internet o cualquier otra red accediendo a ella mediante un modelo de software como servicio. Es decir, acceder “virtualmente” a las aplicaciones necesarias. Así, no existe necesidad de crear una infraestructura interna, ya que el servicio y las aplicaciones estarán en la red bajo demanda y podrá ampliarse o reducirse según las necesidades de cada empresa. La utilización de Software Libre por las empresas proveedoras genera muchos beneficios económicos y le da mayor flexibilidad al negocio.
- **Recursos humanos especializados:** ante la creciente necesidad de la utilización de Software Libre por parte de las empresas, son cada vez más necesarios recursos especializados, que puedan hacer una evaluación según su profesionalismo y conocimiento. Puede darse como técnico asignado-soporte pro activo, adelantándose a los problemas a través de uno o varios técnicos periódicamente en la empresa del cliente. También se ofrecen recursos a través de un técnico telefónico o remoto, que brindan asistencia en horario de oficinas. Y por último, el modelo 24x7x365, donde el cliente puede acceder a la ayuda técnica en todo momento.
- **Personalización de instalaciones masivas en redes:** las personalizaciones del software para empresas que tienen muchos equipos (servidores de sucursal o estaciones de trabajo) empieza a ser un servicio en sí mismo ya que la flexibilidad del Software Libre permite que se ajusten las aplicaciones y sus configuraciones exactamente a la necesidad del cliente, luego se utiliza alguno de los múltiples sistemas para hacer replicaciones automáticas, de forma tal que puede hacer muchas instalaciones de manera simultánea.
- **Armado de hardware:** se trata de armar dispositivos combinando las aplicaciones del software basadas en plataformas Linux o cualquier otro Software Libre para lograr un paquete listo para funcionar. Así, el experto es quien hace todos los pasos de instalación e integración para entregar soluciones completas que simplifican la

implementación, la gestión y el mantenimiento.

En este sentido, las pequeñas empresas de software mexicanas no se ha mantenido al margen. Desde hace un tiempo, el Software Libre es ofrecido como un servicio relacionado con las instalaciones, brindando soporte, mantenimiento, capacitaciones, migraciones y consultoría, sin embargo, debido a la falta de promoción y el hecho de que estos modelos son para uso específico muchas veces el cliente no sabe que en realidad está utilizando Software Libre.

## **5.2. Sensibilización a las ventajas del Software Libre**

Hasta este momento solamente se ha examinado que es el Software Libre y las razones por las cuales es importante. También logramos ver la confrontación de la comunidad de Software Libre contra ciertos grupos de poder en México y el progreso (escaso hasta el momento) respecto al interés y uso del Software Libre.

A simple vista, podría parecer innecesario implementar estrategias que permitan discernir las bondades del Software Libre sin embargo, el hecho solamente de proponer la reflexión sugiere de entrada que estamos en una situación de vulnerabilidad cuando se aborda el tema de ganar espacio social o legitimidad cuando se propone una metodología sobre el desarrollo y adquisición de software que suponemos es distinta, por lo menos, a la forma dominante con la cual se ha abordado el tema cotidianamente. Por eso uno de los retos principales del Software Libre es la sensibilización.

La sensibilización del usuario supone para éste comprender entre otras cosas:

- Qué es el Software Libre.
- Ventajas del Software Libre.
- Para quién es realmente la libertad que se otorga.
- Uso del Software Libre.

Desgraciadamente, a diferencia de otras opciones comerciales el Software Libre no cuenta con un departamento de mercadotecnia que se encargue de la sensibilización. Ante la ausencia de tal departamento se han formado gran número de comunidades que organizan eventos cuyo objetivo principal es la sensibilización de los asistentes.

Uno de los eventos más destacados en México es el Congreso Nacional de Software Libre (CONSOL), creado en la Facultad de Ingeniería de la UNAM en 2002. Este congreso tenía en sus inicios como principio acercar a destacadas personalidades del Software Libre a la comunidad formada en su gran parte por estudiantes de nivel superior y licenciatura, así como profesionistas y cualquier interesado y simpatizante del Software Libre. Esto incluye a desarrolladores, administradores, promotores y usuarios en general que estén relacionados con él, principalmente para aprender unos de otros y de convivir, fortaleciendo la comunidad.

Esto ha cambiado para mejorar, el CONSOL hoy día es una oportunidad idónea para conocer a las personas que contribuyen, trabajan y viven con el software libre y también se imparten cursos y conferencias para varios niveles, desde principiantes hasta empresas convirtiéndose en un evento de sensibilización sobre Software Libre, incluyendo las siguientes temáticas:

- Desarrollo de Software
- Conferencias
- Diseño Gráfico
- Talleres
- Software en General
- Pláticas Técnicas
- Seguridad en Cómputo e Informática
- Pláticas Informales
- Redes
- Manuales
- Filosofía
- Presentación de Proyectos
- Negocios
- Exposición de Soluciones
- Casos de Éxito
- Mesas Redondas

Por su parte el Festival Latinoamericano de Instalación de Software Libre (FLISoL) es el evento de difusión de Software Libre más grande en Latinoamérica. Se realiza desde el año 2005 y su principal objetivo es promover el uso del Software Libre, dando a conocer al público en general su filosofía, alcances, avances y desarrollo.

Para tal fin, las diversas comunidades locales de Software Libre (en cada país, en cada ciudad/localidad), organizan simultáneamente eventos en los que se instala, de manera gratuita y totalmente legal, Software Libre en las computadoras que llevan los asistentes. Además, en forma paralela, se ofrecen charlas, ponencias y talleres, sobre temáticas locales, nacionales y latinoamericanas en torno al Software Libre, en toda su gama de expresiones: artística, académica, empresarial y social.

Debido a estos esfuerzos la sociedad civil, incluyendo el sector privado esta aceptando cada vez mejor el Software Libre sin embargo, como ya se mencionó anteriormente en los niveles gubernamentales aun no hay una influencia importante. Un ejemplo adecuado de cómo se puede sensibilizar a los funcionarios de gobierno – e incluso extender la idea a otros sectores –, es la estrategia llamada “Un courrier de sensibilisation des acteurs locaux par jour”<sup>89</sup> en la que cada día durante el año 2010, se escribirá a un agente de planificación para esbozar las cuestiones de Software Libre. Para esto se elige cada día un funcionario, agente económico, activista, asociación sindical, periodista, profesor entre otros actores importantes en la vida pública. Se trata de 365 personas en los Alpes del Sur para recibir información sobre el Software Libre y sus desafíos. Esta estrategia es formulada en el sitio web Apitux.org, un sitio que se dedica a la sensibilización al Software Libre, a los estándares abiertos y a la interoperatividad. Fundado en 1997 por Jean-Christophe Becquet Vice-Presidente y animador del grupo de trabajo sensibilización en la APRIL, la principal asociación francesa dedicada a promover y proteger el Software Libre en aquel país.

---

89 BECQUET, Jean-Christophe. Un courrier de sensibilisation des acteurs locaux par jour dans les Alpes du sud en 2010. APITUX [en línea], 4 enero 2010, [Fecha de consulta: 4 enero 2010]. Sección Ressources. Disponible en: <http://www.apitux.org/index.php?2010/01/04/215-un-courrier-de-sensibilisation-des-acteurs-locaux-par-jour-dans-les-alpes-du-sud-en-2010>.

### 5.3. Enseñanza del Software Libre

Todo proceso de enseñanza es continuo y el presentado cuando se aprende Software Libre no es la excepción. Como ya se mencionó anteriormente cuando se señalaron las desventajas de este modelo la curva de aprendizaje es mayor en comparación con la presentada con el Software de Código Cerrado.

En este sentido es necesario puntualizar que al aprender el uso de cualquier software, la curva de aprendizaje será directamente proporcional al conocimiento y funcionalidades que este software pueda contener. En el caso del Software Libre la curva de aprendizaje es extendida debido a las cualidades que se presentan, el conocimiento técnico y algunas veces científico contenido en su código fuente y la capacidad de ampliación del mismo, son algunos de los motivos por los cuales se da este fenómeno.

Después de que un usuario de Software de Código Cerrado es sensibilizado a las ventajas de Software Libre, lo primero que deberá aprender es a elegir un sistema operativo. En este primer paso el usuario debe distinguir que los sistemas operativos de Software Libre son diseñados para un perfil específico de usuario y están centradas para realizar algunas tareas con mayor eficiencia y eficacia.

Si se da el caso de que el usuario es mal orientado en la elección de su nuevo sistema operativo, es muy posible que surja el desaliento y claudique en la migración.

Una vez elegido el sistema operativo adecuado, el usuario debe aprender los programas equivalentes que le permitirán hacer sus actividades cotidianas en la computadora. En este sentido lo más importante es permitirle al usuario descubrir el concepto de innovación. El funcionamiento del Software Libre es, por regla general, mucho más ergonómico que los anteriores modelos de software lo cual obliga a un usuario que pretende migrar acostumbrarse a nuevas funcionalidades. En forma de metáfora Neal Stephenson compara así los modelos de software:

*Imagínense un cruce de carreteras donde hay cuatro puntos de venta de coches. Uno de ellos (Microsoft) es mucho, mucho mayor que los demás. Comenzó hace años vendiendo bicicletas de tres velocidades (MS-DOS); no eran perfectas, pero*

*funcionaban, y cuando se rompían se arreglaban fácilmente. Enfrente estaba la tienda de bicicletas rival (Apple), que un día empezó a vender vehículos motorizados — coches caros, pero de estilo atractivo, con los mecanismos herméticamente sellados, de tal modo que su funcionamiento era algo misterioso. (...) Al otro lado de la carretera hay dos competidores que llegaron más recientemente. Uno de ellos, (Be, Inc.) vende Batmóviles plenamente operativos (los BeOS). Son más bonitos y elegantes incluso que los eurosedanes, mejor diseñados, más avanzados tecnológicamente, y al menos tan fiables como cualquier otra cosa en el mercado - y sin embargo son más baratos que los demás. Con una excepción, claro: Linux, que está enfrente mismo, y que no es un negocio en absoluto. Es un conjunto de tiendas de campaña, yurtas, tipis, y cúpulas geodésicas levantadas en un prado y organizadas por consenso. La gente que vive allí fabrica tanques. (...) Estos tanques se producen ahí mismo a un ritmo aterrador, y hay un número enorme de ellos alineados junto a la carretera con las llaves puestas. Cualquiera que quiera puede simplemente montarse en uno y marcharse con él gratis.<sup>90</sup>*

Para ejemplificar lo anteriormente dicho podemos repasar y comparar paso a paso el proceso de configuración e instalación de software en dos sistemas operativos representativos: MS-Windows y GNU/Linux. Los sistemas de instalación actuales en GNU/Linux y MS-Windows tienen un enfoque totalmente distinto. En MS-Windows normalmente la aplicación es autocontenida y posee todo lo que necesita para funcionar, en cambio en GNU/Linux y en general en las aplicaciones de código abierto se evitan complicaciones y una aplicación está dividida en muchas partes pequeñas que se desarrollan en forma independiente, compartiendo muchos componentes entre ellas.

Un instalador para una aplicación en MS-Windows puede ser bastante grande, ya que además de incluir la aplicación a ser instalada, debe incluir todas las bibliotecas que necesita esa aplicación para funcionar, lo que en algunos casos hace inevitable el uso de CD's para contener sólo una aplicación. En el caso de GNU/Linux, sólo se descarga e instala en el sistema lo que no se había instalado previamente, por ejemplo, si una aplicación ya instaló

---

90 STEPHENSON, Neal. In the beginning was the command line. E.E.U.U.: Altediciones, 1999.

una biblioteca de criptografía, la siguiente aplicación no necesita volverla a instalar.

En MS-Windows, lo más cercano al enfoque de Linux podría ser la relación entre los juegos y DirectX, en donde Microsoft desarrolla DirectX mientras que terceros desarrollan los juegos: Primero se instala DirectX y luego los juegos pueden usar el DirectX que ya se encuentra instalado en el sistema. Lamentablemente o afortunadamente depende del punto de vista de cada uno, no es mucho más lo que se comparte entre aplicaciones de MS-Windows.

Desde el punto de vista positivo en las instalaciones de MS-Windows, se asegura de que el instalador tiene todo lo que se necesita para que la aplicación funcione. Desde el punto de vista negativo, cada aplicación usa su propia versión de cada componente, existen múltiples copias y no hay una forma unificada en que se pueda actualizar el sistema y las aplicaciones al mismo tiempo, por ejemplo para corregir errores o problemas de seguridad, además el sistema requiere mayor espacio de almacenamiento. Por ejemplo si una aplicación contiene la funcionalidad para decodificar mp3 y contiene una falla, cuando aparece la corrección se debe descargar e instalar la aplicación completa nuevamente.

En el caso de GNU/Linux también se tiene un punto de vista positivo y otro negativo: Por el lado positivo, una aplicación es sólo un componente del sistema y puede ser instalada y actualizada en forma sencilla e independiente. El instalador sólo tiene lo necesario y si la aplicación necesita otro componente para funcionar, se usa el que ya está instalado o bien se descarga automáticamente de Internet. A veces una actualización puede significar inmediatamente la mejora de varias aplicaciones, ya que los componentes se comparten. Desde el punto de vista negativo, instalar estas aplicaciones sin acceso a Internet puede resultar en la falta de los componentes necesarios, sin embargo siempre existen alternativas debido a que los programas se pueden compartir libremente.

Para simplificar la instalación de estas aplicaciones, en la mayoría de las distribuciones GNU/Linux se utiliza un sistema de paquetes, en donde cada componente está constituido por uno o más paquetes. Al instalar un paquete, el sistema determina qué más necesita, lo descarga de Internet, lo instala y configura en forma completamente automática. En MS-Windows, que no cuenta con un sistema de paquetes, instalar aplicaciones es más difícil, porque el usuario debe instalar cada componente en forma manual y configurarlos para que

conversen entre ellos.

Por ejemplo, si en GNU/Linux queremos tener un servidor web con soporte de PHP y base de datos, con una corta línea de comandos o muy pocos movimientos del ratón, el sistema instala todo lo necesario en forma automática y queda listo en pocos minutos o segundos dependiendo de la calidad de la conexión a Internet. En MS-Windows, instalar lo mismo puede llevar horas, ya que el usuario se debe encargar de todo, desde buscar la aplicación en Internet, hasta configurarlas para que se integren. Si a esto agregamos que no hay una coordinación de versiones, es muy probable que en algunos casos no encuentre versiones que sean compatibles entre ellas.

Además de la instalación y configuración del software también pueden cambiar otros aspectos de uso diario. Por eso es importante que el usuario comience usando software que pueda ser utilizado de forma similar al que usaba antes de migrar. Poco a poco el usuario podrá ver el uso de software que contenga innovaciones para hacer tareas específicas y aprender a usar nuevas herramientas particulares del sistema libre que ha instalado.

Tal es el caso del entorno gráfico. Si bien existen entornos gráficos que utilizan la metáfora del escritorio como lo hacen los entornos MS-Windows o MacOS también se ha innovado y se han creado entornos basados en la idea de mosaicos.

Los escritorios comunes utilizan la interfaz WIMP<sup>91</sup> que es acrónimo de “ventana, icono, menú, apuntador” - haciendo referencia a las palabras en inglés -. Este paradigma ha sido dominante desde la década de los ochenta cuando Microsoft y Apple adaptaron sus aplicaciones a este modelo. En este paradigma las ventanas son áreas de visualización con una barra de título y controles para cambiar el tamaño de la ventana y su posición, que se acomodan en forma de pila y solo es visible una ventana a la vez.

Este paradigma ofrece realmente pocas ventajas, la principal es un funcionamiento bastante sencillo. Las opciones son visibles en los menús, se crea dependencia con el dispositivo apuntador, el cual facilita el acceso a los menús. Las capacidades pueden ser

---

91 GEORGE Taylor, Ashley. WIMP Interfaces [en línea]. Actualizada: 1997. [Fecha de consulta: abril 2010].

Disponible en: [http://www.cc.gatech.edu/classes/cs6751\\_97\\_winter/Topics/dialog-wimp/](http://www.cc.gatech.edu/classes/cs6751_97_winter/Topics/dialog-wimp/).

exploradas sin comprometerse con ninguna acción. La mayoría de las acciones se pueden invertir con atajos visibles a simple vista lo que fomenta la exploración. La mayoría de las aplicaciones tienen formas similares, por lo que las transferencias de conocimiento entre ellas se facilita.

Sin embargo, paralelamente a estos entornos se han desarrollado entornos gráficos basados en el paradigma de mosaicos, el cual pone especial detalle en los defectos que presenta WIMP. En este paradigma la ventana esta concebida como un espacio de la pantalla que permite observar una aplicación. Las ventanas se gestionan dinámicamente y son agrupadas por etiquetas. Cada ventana puede ser marcada con una o varias etiquetas. La selección de ciertas etiquetas muestra todas las ventanas con estas etiquetas, de esta manera las ventanas no necesitan estar en una pila, la pantalla solo mostrará las ventanas deseadas y el tamaño de la ventana sera en principio seleccionado por el entorno y el usuario podrá cambiarlo con cualquier dispositivo de entrada, retirando así la dependencia a un apuntador. Las ventajas son inmediatas, las ventanas siempre están en una posición adecuada para trabajar con varias aplicaciones de forma paralela, la organización de las mismas es más intuitiva y el escritorio nunca se verá saturado.

Como se puede observar el proceso de aprendizaje puede complicarse solo porque es difícil luchar contra lo que el usuario ya conoce, por malo que sea. Entonces, la innovación se pierde. Sin embargo es parte de los procesos de sensibilización y enseñanza que los usuarios tengan inquietud o necesidad de aprender nuevas cosas y sobre todo en que las nuevas generaciones que tienen que aprender todo desde el inicio siempre tengan la iniciativa de mejorar lo ya aprendido.<sup>92</sup>

## **5.4. Uso del Software Libre**

### ***5.4.1. Uso del Software Libre en la universidad, la empresa y el gobierno***

La universidad, es una institución que mantiene estrechas relaciones con otras instituciones, entre ellas la propia sociedad, ya que la universidad es la institución encargada

---

92 SERRADILLA, Francisco. El gran problema de la innovación. Libro de notas [en línea]. 25 febrero 2010.

[Fecha de consulta: 25 febrero 2010]. Sección Computación. Disponible en: <http://www.librodenotas.com/computacion/17809/el-gran-problema-de-la-innovacion?>

de educar a los ciudadanos, con la finalidad de que éstos últimos los puedan aplicar su conocimiento para el mejoramiento de la vida social.

El filósofo español Ortega y Gasset afirmó que la universidad, es la “institución normal de un país, [la cual] depende más del aire público (...) que el aire pedagógico (...) producido dentro de sus muros.”<sup>93</sup>

Y esto se debe a que la universidad tiene el reconocimiento y un prestigio social debido a su actividad principal contemporánea, la de mantener siempre constante los objetivos de preservación y difusión tanto de ciencia-investigación, docencia y cultura.

El mismo Ortega y Gasset mencionó que la universidad “(...) debe hacerse cargo [del] proyecto (...) técnico (la preparación de los profesionales), moral (la transmisión de la cultura y la civilización), científico (la formación de los científicos) y social (la preparación de cuadros de dirigentes conscientes de los problemas que debieran afrontar)”.

La relación entre la universidad con la ciencia y la tecnología tiene sentido en la medida que el personal académico, administrativo, laboral y los estudiantes estén en la capacidad de articular el trabajo académico con una mentalidad reflexiva, crítica y metódica adecuadas a las condiciones básicas impuestas por la dinámica de la sociedad.

En este sentido, la educación universitaria debe orientarse hacia la construcción autónoma del individuo que garantice su desarrollo libre dentro de la sociedad. Por ello la formación de dicho individuo pretende hacerlo participe activo de los múltiples procesos por medio de la amplia gama de canales comunicativos e informativos que ofrecen las TIC. Por ello, el cambio tecnológico dentro de la universidad se plantea como un fenómeno de importancia tal, ya que es en este espacio y tiempo cuando la formación profesional pretende capacitar al sujeto en habilidades tales como planificar, analizar, diseñar, implementar y administrar información, por medio de las TIC dentro de todos los ámbitos sociales en las que se encuentre el sujeto.

---

93 ORTEGA y Gasset, José. Obras Completas: Misión de la universidad. Madrid: Alianza Editorial, 1987. 4

Condición que nos lleva a la necesidad de usar el Software Libre, debido a que es una herramienta que puede usarse para formar al estudiante como un ser integral y enseñarle la importancia de la libertad, guiarlo en el sentido de saber hacer uso de esa libertad. Hacerle entender al estudiante que debe tener la posibilidad de estudiar hasta donde su curiosidad lo lleve, que debe poder profundizar en el conocimiento, y que además existe una responsabilidad inherente a ello, un Software Libre es tan bueno como la responsabilidad con que encararan sus realizadores su trabajo en torno a él. Algunos usos específicos en el sector educativo son:

- Educación a distancia.
- Laboratorios de computo.
- Comunidades docentes virtuales.
- Comunidades estudiantiles virtuales y portales comunicativos.
- Administración de la institución.
- Servicios de búsqueda en bibliotecas y bibliotecas digitales.
- Proyectos: académicos, científicos, culturales, etc.

Los costos para las universidades e instituciones educativas en general son menores, no solo por el ahorro en el pago de licencias, sino también porque, el Software Libre es más confiable y seguro que otros. El poder acceder al código fuente es un estímulo permanente para la apropiación de las TIC y la innovación. Lo cual pasa también por entender que el software se puede adaptar a las necesidades locales y que cualquier persona, con los conocimientos necesarios, puede participar de la construcción, adaptación y uso de programas de computadora. Esta es una forma real y tangible de acercar las TIC a la gente, acercando las posibilidades, por ejemplo por medio de laboratorios de computo.

La posibilidad distribuir el software, muestra al alumno y al docente que hay muchas cosas por hacer, que la obra nunca está acabada y que puede colaborar en su construcción, que

su aporte es bienvenido, que puede cooperar con otros, que puede difundir el conocimiento, todo ello participando en la construcción, modificación y difusión del software.

En cuanto a la investigación, tarea fundamental de la universidad, el uso de Software Libre es básico dado que el software de código cerrado nos impide conocer como funciona un programa. Si en un trabajo de investigación utilizamos software de código cerrado en el procesamiento de los datos, no es posible refutar los resultados obtenidos debido a que no podemos revisar todos los pasos que llevaron a la obtención de un determinado resultado. Es imposible, dado que el software de código cerrado actúa como una caja negra y no podemos acceder a los algoritmos utilizados para el procesamiento de los datos porque no hay acceso al código fuente.

Además la universidad se encarga de la socialización de los ciudadanos, su educación, de la misma forma se encarga de la formación de profesionales destinados a abastecer los sectores productivos, con miras de que los egresados se puedan incorporar a un mercado laboral.

En este sentido el Software Libre es también una opción favorable. Organizaciones y personas que han comprendido las ventajas y los beneficios que este tipo de software les ofrece hoy en día están migrando. Así como gente y empresas que entienden los nuevos modelos de negocio y procesos de trabajo. No importa su área de negocio o industria, el tamaño empresarial y zona geográfica todos buscan mejorar su sistema o plataforma ya sea en el ambiente técnico o económico, aumentar la productividad y la agilidad para responder a cambios en el mercado, reducir dependencias, mantener una solución efectiva en costos y ofrecer mejores servicios o productos.

Para ellos se han desarrollado y se siguen desarrollando dentro del modelo de Software Libre mejores sistemas operativos con la finalidad de resolver problemas y no necesariamente para adherir nuevas funcionalidades para uso de venta. La calidad de las aplicaciones esta a la par de las aplicaciones de código cerrado y evitan la dependencia en proveedores, logrando así una armonía al elegir entre los mejores productos y servicios reduciendo costos en base a un mejor costo total de uso y un mejor retorno en inversión.

De manera general, las empresas suelen usar software básico: sistemas operativos, antivirus, herramientas de ofimática; particularmente usan software para agilizar sus procesos o controlarlos remota u automáticamente. Esto implica costos por licenciamiento, instalación (de hardware y software) capacitación y asistencia técnica.

Dada la evidencia respecto de los ahorros de incorporar Software Libre en las computadoras, es razonable recomendar a las empresas evaluar seriamente esta alternativa.

Idealmente, una empresa podría optar por el uso exclusivo de Software Libre, sin embargo la decisión de reemplazar el software de código cerrado no implica necesariamente cambios radicales e inmediatos. En lugar de ello, puede ser más apropiado hacer cambios graduales de manera que se mantengan productos como el sistema operativo y se reemplacen elementos como los programas ofimáticos.

Podemos observar que este modelo de software otorga derechos a sus usuarios, algunas de estas ventajas pueden ser más apreciadas por usuarios particulares, otras por empresas sin embargo, las instituciones públicas y otros organismos pueden también beneficiarse con la confiabilidad y estabilidad, al ser público, esta sometido a la inspección de una multitud de personas, que pueden buscar problemas, solucionarlos, y compartir la solución con los demás. Debido a esto, y a la mencionada “Ley de Linus” (dada la suficiente cantidad de ojos, cualquier error del software es evidente), los programas libres gozan de un excelente nivel de confiabilidad y estabilidad, requerido para las aplicaciones críticas del estado.

Algunos de los usos del Software Libre en la administración pública son:

- Para crear ciudades virtuales. Que brindan servicios variados a la comunidad, desde el pago de impuestos, acceso información política y creación de comunidades virtuales.
- Para manejar información y servicios sensibles o que requieren un alto grado de seguridad. Mucha de la información que el Estado maneja puede ser peligrosa en manos incorrectas. Es por esto que es crítico que el Estado pueda fiscalizar que su software no tenga puertas de entrada traseras, voluntarias o accidentales, y que pueda cerrarlas en caso de encontrarlas; tal inspección sólo es posible con el software libre.

- Desarrollo de las TIC. Con la implementación de estándares abiertos para la administración pública se facilita el impulso, uso y gestión de las TIC a todos niveles.

Bajo este panorama las TIC se convierten en herramientas e instrumentos indispensables para vivir dentro del mundo. El Software Libre facilita de forma notable que a partir de ellas se den nuevos procesos y ambientes donde se establece la socialización entre las personas de diversas regiones del mundo; donde se establecen espacios para la colaboración en diversos proyectos; donde se hacen perfectibles los procesos de educación. Es por ello, que las universidades ocupan un rol estratégico en el impulso al uso del Software Libre y las empresas junto con la administración pública deben favorecer y beneficiarse de la formación de nuevos profesionistas, las actividades de investigación y extensión del conocimiento.

#### **5.4.2. Migración al Software Libre<sup>94</sup>**

La migración al Software Libre<sup>95</sup> es lo más importante, para el usuario común implica el aprendizaje de nuevas opciones en el mundo informático y mejor aprovechamiento de las TIC que se encuentran a su alcance. En las organizaciones, la migración es una decisión estratégica, para aumentar la competitividad y la productividad. Todo tipo de empresas y organizaciones en todas las áreas están migrando lo que es posible a su tiempo, para obtener los beneficios que esta migración les brinda. Una Migración al Software Libre es exitosa si se desarrolla un buen “Plan de Migración”.

Sin importar el caso es necesario tener en cuenta ciertas consideraciones antes de

---

94 Este texto contiene los puntos más relevantes de la publicación Materiales de Directrices IDA (Intercambio de Datos entre Administradores) de migración a software de fuente abierta. El texto completo esta disponible como: COMUNIDAD Europea. The IDA Open Source Migration Guidelines. 18ª Ed. Comunidad Europea, 2003.

95 La migración es un proceso, es el conjunto de actuaciones que cuya finalidad es sustituir una TIC por otra de diferente origen pero capaz de hacer la misma tarea. La migración al Software Libre tiene como objetivo la sustitución de las infraestructuras basadas en Software de Código Cerrado por otras con funciones equivalentes basadas en Software Libre. Fuentes: EIRAWORKS. Eiraworks [en línea]. [Fecha de consulta: febrero 2009]. Disponible en: <http://www.eiraworks.com/>. INVESTIC. Investic [en línea]. [Fecha de consulta: febrero 2009]. Disponible en: <http://www.investic.net/>.

comenzar una migración. Si bien muchos consejos son aplicables a cualquier ámbito, no vamos a profundizar en los aspectos técnicos, los cuales varían muy rápidamente y dependen en muchos casos de la infraestructura existente, sino que vamos a hacer énfasis en lo metodológico y las opciones disponibles para encarar la migración. En todo caso, siempre será conveniente contar con ayuda que en buena medida puede surgir de los integrantes de la misma comunidad de Software Libre.

#### *5.4.2.1. Aspectos importantes relacionados con la migración*

- Lo ideal sería partir de cero utilizando Software Libre, pero hablando de migración ese no es el caso. Estamos partiendo de una situación en la cual utilizamos software de código cerrado, y deseamos pasar a Software Libre. Esto conlleva una serie de problemas inherentes a cualquier migración (inclusive de una versión de software de código cerrado a otra) y otros inherentes a la migración de software de código cerrado a Software Libre, que presenta algunas características especiales.
- Es necesario pensar que una parte del diseño de los sistemas privativos apunta justamente a evitar la migración, lograr la “fidelidad forzosa” del cliente, por lo cual siempre hay que sortear una serie de inconvenientes para migrar con éxito.
- Para una migración exitosa, el usuario debe estar convencido de empezar este proceso, en el caso del sector privado es importante contar con defensores del cambio dentro del equipo de trabajo y el apoyo de los usuarios del sistema. Para lograr esto, siempre es conveniente, antes de migrar, informarse sobre el nuevo software que se utilizará y sus ventajas, en el caso del sector privado realizar pláticas explicativas informando sobre el por qué del cambio y de los beneficios esperados. Involucrar positivamente a los usuarios del sistema en el cambio siempre es un diferencial importante a la hora de realizar la migración.
- Recolectar información sobre el equipo de hardware en el que se realizará la migración para verificar y prever la compatibilidad del mismo. En el sector privado es necesario realizar un claro diagnóstico de la situación de partida o escenario, el cual incluye la arquitectura del sistema, configuración de la red, hardware y software

utilizado. Además de las características de los usuarios del sistema, conocimientos de informática, necesidad de capacitación, etc.

- Definir con la misma precisión la situación que se espera encontrar al terminar la migración.
- En el caso del sector privado, realizar un análisis de costos de la migración y proyectar los ahorros factibles, con el correr del tiempo, en pago de licencias.
- Planear el avance en la migración, así el usuario evitará sentirse frustrado en sus actividades. En el sector privado se debe escribir un plan de migración, teniendo en cuenta la forma de migración por la que se vaya a optar.
- Uno de los elementos más delicados es el procesamiento de los datos existentes, los cuales en muchos casos se encuentran guardados en un formato con un estándar desconocido, y deben ser convertidos a un formato abierto, lo que en algunos casos es muy sencillo. Otra opción, en caso de que dichos archivos no se puedan convertir, es mantener un equipo dentro de la red funcionando con el software específico para manejar esos archivos. En todo caso deberemos, de ahí en adelante, usar siempre formatos abiertos y estándar en el manejo de la información, ya que ésta es la única forma de asegurar un acceso permanente a dicha información, además de su conservación en mediano y largo plazo.
- Es importante asesorarse y buscar portales y comunidades de ayuda, con la idea de resolver de manera rápida las dudas que puedan surgir en el proceso. Para el sector privado es fundamental, de acuerdo al tamaño de la organización, contar con un equipo técnico que pueda brindar asesoramiento a los usuarios luego de la migración, evacuar consultas y solucionar problemas que puedan surgir. También es importante contar con cursos de capacitación previos a la migración y posteriormente a la misma.

#### 5.4.2.2. *Migración paso a paso*

La forma más simple de iniciar una migración es comenzar a usar Software Libre en el

sistema operativo privativo que estemos utilizando. Dado que el Software Libre puede ser visto por un usuario novato básicamente como una forma de licenciamiento, existe Software Libre para todos los sistemas operativos existentes. Por lo tanto, podemos comenzar sustituyendo aplicaciones de software de código cerrado que usamos por sus contrapartidas en el ámbito del Software Libre. Un buen ejemplo de lo anterior está en el área de ofimática, en la que el paquete de Software Libre Open Office ofrece las mismas funciones que sus equivalentes de código cerrado, al tiempo que sustituye los formatos cerrados (.doc, .xls, .ppt) por el formato abierto y estándar OpenDocument.

Una de las principales ventajas de esta forma de migración es que se pueden ir pasando de los formatos cerrados (por ejemplo los archivos .doc) a formatos abiertos (como por ejemplo .odt) en forma progresiva, y sin cambiar el entorno operativo. Por otra parte, posibilita el intercambio de archivos con usuarios de Software Libre bajo cualquier sistema operativo.

Una vez que el usuario se ha familiarizado con las aplicaciones libres de uso cotidiano, se puede comenzar la migración del sistema operativo. Eso se puede hacer en forma gradual máquina a máquina en el caso de un particular o por secciones si se trata de una organización. Una de las ventajas de utilizar archivos con formato abierto y estándar es, justamente, liberarnos de la dependencia a un determinado software sobre algún sistema operativo. El utilizar dentro de la red equipos con sistemas operativos diversos tiene algunas implicaciones desde el punto de vista técnico, pero es totalmente factible. Inclusive una de las migraciones más fáciles, desde el punto de vista de los usuarios, es la del servidor, dado que su funcionamiento es transparente para los usuarios.

La meta de este cambio gradual es terminar en un sistema basado en un cien por ciento en Software Libre. Es importante señalar que migrar de una sola vez, sustituyendo de entrada el sistema operativo y todas las aplicaciones por Software Libre es la opción más rápida, pero la más riesgosa, se necesita de una asesoría completa y/o de personal capacitado muy convencido del cambio y una mayor planificación.

#### 5.4.2.3. *Consideraciones especiales sobre los recursos humanos en las organizaciones*

1. Es importante que todo el personal de la institución esté informado del tema y sienta que la migración va a generar beneficios en un cierto plazo.
2. Es importante brindar opciones de capacitación para el personal que lo considere necesario, es una forma de vencer el “miedo a lo desconocido” uno de los problemas más frecuentes en el momento de encarar una migración.
3. Es importante que el personal sienta que la migración le abre nuevas puertas en el ámbito de su capacitación personal, dado que el Software Libre es una opción de futuro y una forma de apropiarse realmente de las nuevas tecnologías.
4. Es importante lograr que el personal actual del área de informática esté involucrado y comprometido en el proceso ya que son las personas clave en la migración, ellos deben ser especialmente considerados y los primeros en ser capacitados en el ámbito del Software Libre, en caso de ser necesario dada su formación en el área.

Desde este momento, puede tomar algunas medidas para facilitar una migración futura y disminuir su nivel de dependencia con relación al software de código cerrado:

- Insistir en el uso de Formatos Abiertos y Estándar es la única forma de garantizar que la organización pueda acceder hoy y en el futuro a los datos y la información que maneja. Además, es una base muy importante para facilitar una futura migración y el intercambio de información con los que ya están utilizando Software Libre.
- Desarrollar sistemas basados en tres niveles, donde el código sea independiente de la interfaz y de los métodos de acceso a los datos.
- Insistir en que las nuevas aplicaciones generadas sean portables, o sea, usar lenguajes portables como ANSI C, JAVA, Perl, Python, etc. Evitar lenguajes de arquitecturas específicas.
- Evitar la construcción de aplicaciones que requieran la presencia de otras aplicaciones

privativas.

- Insistir en que cualquier desarrollo web de la organización pueda ser visualizado en los navegadores más comúnmente usados en la web, principalmente los licenciados como libres, (por ejemplo Firefox).
- Insistir en que los desarrollos web cumplan con los estándares de la W3C. Desarrollar los sitios web de forma tal que sean fácilmente portables a un servidor que corra sobre Software Libre. Un ejemplo: Un sitio web en un servidor que corra sobre MS-Windows no distingue la diferencia entre mayúsculas y minúsculas en el nombre de los archivos, pero en un servidor Apache bajo GNU/Linux esa diferencia se manifiesta, por lo tanto es conveniente organizar el sitio web de tal manera que se tenga en cuenta que un archivo llamado autores.html no es igual a otro llamado Autores.html: los dos serán vistos como archivos diferentes, por lo que al realizar un enlace a dicho archivo se debe mantener el criterio de llamarlo con mayúscula o minúscula.

#### **5.4.3. Selección del Software**

Una vez que se ha decidido hacer la migración al Software Libre se debe hacer una selección de los nuevos programas a utilizar. Para hacer esto lo primero que se debe tomar en cuenta es el hardware disponible bajo los siguientes criterios:

- Tipo de hardware: Marca y modelo. Con esto se puede identificar fácilmente si el fabricante de este hardware da soporte a la plataforma libre a la que pensamos migrar.
- Hardware requerido y necesario: Esto incluye los dispositivos conectados, por ejemplo escáner, impresora, teclados con funcionamiento especial, dispositivos de almacenamiento como memorias USB, etc.

Adicionalmente en las organizaciones es importante tener en cuenta:

- Estandarización del hardware: Si todas las máquinas son iguales, esto facilita la búsqueda de soporte para controladores y elección de la nueva plataforma.

- Componentes requeridos: Los usuarios no siempre utilizan todo el hardware disponible en un equipo, es importante distinguir las necesidades de los usuarios para poder construir un sistema homogéneo en la organización y que cumpla con todas las necesidades de la misma.

Después de tener bien claro todos estos puntos se debe proceder a la elección e instalación del nuevo software con el fin de probar su funcionamiento (pueden existir varias opciones, con estas pruebas se pretende elegir el mejor adaptado a las necesidades del usuario). En esta elección se deben contemplar todos los entornos posibles en la cual una aplicación podría ser usada en el día a día, verificando lo siguiente:

- Compatibilidad con formatos usados actualmente.
- Comprobar que todas las características mencionadas para dicho software se puedan ejecutar.
- Que la aplicación cumpla con las exigencias requeridas.
- Comprobar la estabilidad del software.
- Soporte del software. Actualizaciones y documentación.

En la Guía de migración a Software Libre de Venezuela<sup>96</sup> el Gobierno Venezolano instruye a sus dependencias (y aplicable a las organizaciones) para la creación de un laboratorio de Software Libre, un entorno adecuado para efectuar pruebas de compatibilidad. El mismo deberá tener las condiciones adecuadas para albergar temporalmente los dispositivos que deberán ser homologados. El objetivo principal de estos laboratorios es llevar a cero los riesgos de incorporación, migración o actualización de tecnologías libres, para maximizar la continuidad operativa de la organización.

De estas pruebas obtendremos la siguiente clasificación:

---

96 CENTRO Nacional de Tecnologías de la Información. Guía para el plan de migración a Software Libre en la Administración Pública Nacional (APN) de la República Bolivariana de Venezuela. Venezuela: Ministerio del Poder Popular para las Telecomunicaciones y la Informática, 2008.

- Software con un equivalente libre.
- Software sin un equivalente libre.

El software que se encuentre en la segunda categoría no puede ser reemplazado por aplicaciones nativas sin embargo, es posible implementar soluciones basadas en virtualización, emulación o bien con terminales que se conecten a un servidor que ejecute la aplicación.

### **5.5. Programación y producción del Software Libre**

Quizás la ventaja más importante del Software Libre es la disponibilidad del código fuente del programa. Sin embargo es poco útil si el usuario no aprende a hacer uso del mismo. Hacer cambios básicos en el programa, reportar errores y proponer soluciones son actividades importantes del uso activo de este tipo de software.

Es importante recordar por ese motivo que la libertad que ofrece el Software Libre es una ventaja para todos, y la Administración Pública tiene la obligación de probar y estandarizar el software que utiliza.

Este proceso permitirá verificar y revelar la calidad de la programación, implementación de mejores prácticas entre otras cosas; mediante la revisión exhaustiva del código fuente. Existen multitud de conceptos y palabras clave asociadas a las tareas de prueba. Clasificarlas es difícil, porque no son mutuamente disjuntas, sino muy entrelazadas. La calidad del código es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar medidas o pruebas que permitan comprobar que se cumplen con los estándares de programación.

#### **5.5.1. *Requerimientos tecnológicos y mecanismos***

El desarrollo del Software Libre por medio de una comunidad es descrito brevemente en el ensayo La Catedral y el Bazar de Eric Raymond. En este ensayo se señala un desarrollo muy activo, donde los usuarios son la parte elemental del progreso del software. Con una sola jerarquía que implica ser el coordinador del proyecto, el usuario del programa se entusiasma

al tener versiones nuevas en poco tiempo. Las versiones incluyen correcciones y mejoras, muchas veces propuestas por los propios usuarios, cuya función principal es reportar errores al probar el programa.

Hasta este momento parece que el único nivel de calidad existente es el que el usuario propone sin embargo, esto es totalmente falso. Cada proyecto de Software Libre tiene un “Plan de lanzamientos”, que contiene el camino que seguirá el proyecto, la labor real del coordinador es un trabajo de Ingeniería de Software.

Según la definición del IEEE el software “es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo. [...] Un producto de software es un producto diseñado para un usuario”<sup>97</sup>. En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, se considera que “la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software”, es decir, “permite elaborar consistentemente productos correctos, utilizables y costo-efectivos”<sup>98</sup>.

El proceso de ingeniería de software se define como “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad”<sup>99</sup>. El proceso de desarrollo de software “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del

---

97 Lewis G. "What is Software Engineering?" DataPro (4015). Feb 1994.

98 Cota A. "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994.

99 Jacobson, I. Object-Oriented Software Engineering; A Use Case Driven Approach. ACM Press. Addison-Wesley Publishing. Co. U.S.A. 1998.

software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.<sup>100</sup>

El Software Libre usa un enfoque llamado Programación Extrema. La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la Ingeniería del Software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquéllas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia que hay alrededor tanto como sea posible<sup>101</sup>.

Una de las características de la Programación Extrema es que muchas de sus claves son ampliamente conocidas dentro de la Ingeniería de Software desde hace tiempo, incluso desde sus comienzos. Los autores han seleccionado las que han considerado como las mejores y han profundizado en sus relaciones y en cómo se refuerzan unas a otras. El resultado ha sido una metodología única y compacta.

La metodología de la Programación Extrema es sencilla, son 5 etapas con reglas fáciles de cumplir<sup>102</sup>:

- Planeación.
  - El cliente escribe una “Historia de usuario”. En ella se especifican los tiempos en los que él requiere el software.
  - El programador escribe un plan de lanzamientos.

---

100Zavala-Ruiz, J. (2008) "Organizational Analysis of Small Software Organizations: Framework and Case Study" en H. Oktaba y M. Piattini (eds) *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*.

101CONGRESO HispaLinux (5º. 2001: Madrid, España). Programación eXtrema, software libre y aplicabilidad. Madrid, España: ROBLES, Gregorio. FERRER Jorge, 2001.

102WELLS, Don. Extreme Programming: A gentle Introduction [en línea]. Actualizada: 28 septiembre 2009. [Fecha de consulta: 7 mayo 2009]. Disponible en: <http://www.extremeprogramming.org>

- Se hacen frecuentemente versiones pequeñas o revisiones.
- El proyecto se divide en iteraciones planeadas.
- El plan de iteración (señalando los cambios realizados) inicia cada iteración.
- Gestión.
  - Se da al equipo un espacio de trabajo adecuado.
  - Se establece un ritmo de trabajo adecuado.
  - Cada día se inicia con una reunión corta, solo se conversan temas de alta prioridad.
  - La velocidad del proyecto se mide constantemente (los avances logrados).
  - Rotación de actividades de los miembros del equipo. Así todos los miembros conocen todos los elementos del software y se evitan cuellos de botella.
  - Si algo falla, se replantea rápidamente el proyecto.
- Diseño.
  - Simple.
  - Escoger un paradigma de programación.
  - Incluya a todo su equipo al hacer el diseño de software.
  - Al seleccionar las soluciones al problema, use una metodología adecuada, así reduce riesgos.
  - No agregué ninguna funcionalidad que no sea esencial.
  - Reescriba código cuando y donde sea necesario y posible. Esto implica plantear la

solución nuevamente.

- Programación:
  - El código siempre debe estar disponible para el cliente.
  - Escribir código respetando estándares.
  - Primero escriba las pruebas a realizar junto con un código prueba, luego el código final.
  - Utilice un gestor de versiones para controlar la integración de código.
  - Integre código a menudo y en tiempos específicos.
  - Deje que todo su equipo pueda hacer cambios en cualquier parte del código.
  
- Pruebas:
  - Todo el código debe tener sus propias pruebas.
  - Todo el código debe pasar exitosamente sus pruebas antes de ser agregado a una versión mayor.
  - Si encuentra un error debe crear nuevas pruebas para asegurarse de que está solucionado.
  - Las pruebas de aceptación (cuando el usuario prueba el software) deben suceder a menudo y debe publicarse su resultado.

También es importante señalar que en los proyectos de Software Libre es común la utilización de un paradigma de programación, destacando:

- Programación Orientada a Objetos: Como paradigma es una forma de pensar, una filosofía, de la cual surge una cultura nueva que incorpora técnicas y metodologías

diferentes. Es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose por sí mismo, y comunicándose con los demás por medio de mensajes.

- Programación K.I.S.S.: Por sus siglas en Inglés significa “Dejarlo Simple y Conciso”. Como paradigma, busca que todos los elementos del universo computacional tengan la misma importancia, es decir, es homogéneo. Este paradigma busca concentrar completamente la resolución de un problema en un solo elemento.

Como se puede observar, estos dos paradigmas son opuestos y dependiendo del proyecto y del lenguaje de programación seleccionado los programadores eligen que paradigma de programación seguir.

### **5.5.2. Aplicación de la normalización técnica**

La normalización en la Ingeniería de Software es una pieza clave para crear un producto útil y de calidad. El Software Libre a primera vista puede parecer indiferente a estas cuestiones sin embargo, la normalización es un pilar al momento de elaborarlo.

El conjunto de herramientas de compilación del Sistema Operativo GNU (GCC por sus siglas en inglés) es un ejemplo de esto. En la documentación oficial del proyecto GCC<sup>103</sup> se incluye una sección especial donde se indican las normas que el programador debe seguir para que pueda compilar un programa adecuadamente con GCC, algunos ejemplos son:

**Lenguaje C:** Se soportan tres versiones estándar de C, el ANSI-C original (X3.159-1989) ratificado en 1989 y publicado en 1990. La versión ISO/IEC de la misma norma ANSI (ISO/IEC 9899:1990). Es importante señalar que no hay diferencia entre estas dos normas pero las secciones de la norma ANSI fueron ordenadas y convertidas en cláusulas en la norma ISO. Existe una revisión de la norma ISO/IEC llamada ISO/IEC 9899:1999 la cual también esta disponible.

**Lenguaje C++:** Se encuentra disponible para este lenguaje la norma ISO/IEC original 103FSF. GCC [descargable, en línea]. Versión 4.4.4. Actualizada: 1 julio de 2010. [Fecha de consulta: 19 junio 2009]. Disponible en: <http://gcc.gnu.org>

publicada en 1998 (ISO/IEC 14882:1998) y su modificación publicada en 2003 (ISO/IEC 14882:2003). Se está trabajando en una nueva norma ISO/IEC para este lenguaje, el borrador ha sido publicado y también se ha implementado en GCC.

Lenguaje Fortran: El compilador GNU Fortran implementa la norma ISO/IEC 1539:1997 (Fortran 95). También es posible trabajar con Fortran 90 y Fortran 77 gracias a las mejoras de la norma ISO/IEC TR-15581. En el futuro se implementará la revisión de la norma ISO/IEC para Fortran 2003 ISO/IEC 1539-1:2004, los avances están disponibles en la versión de prueba de GCC. Adicionalmente también es posible trabajar con la especificación OpenMP versión 3.0<sup>104</sup>.

Existen numerosas normas aplicadas en el Software Libre y también es común que el programador siga la documentación estándar sobre Internet conocida como RFC (Request For Comments). En esta serie de documentos se detalla prácticamente todo lo relacionado con la tecnología de la que se sirve Internet: protocolos, recomendaciones, comunicaciones entre otros.

La colección completa de RFC está formada por más de 3000 documentos que especifican estándares, recomendaciones, informativos y obsoletos. El encargado de publicarlos es el Editor de RFC y, aunque cualquiera puede proponer un RFC, se realiza para una detallada revisión antes de ser publicado, el IETF<sup>105</sup> es una de las principales fuentes.

En México, las normas existentes son enfocadas al seguimiento de modelos en procesos y gestiones, tal es el caso de la norma NMX-I-086/01-NYCE-2006. Esta Norma Mexicana constituye una guía para la gestión de la seguridad de TI. Presenta los modelos y conceptos de gestión básicos, esenciales para una introducción a la gestión de la seguridad de TI. Estos conceptos y modelos se discuten y desarrollan en mayor profundidad en las restantes partes de la norma (dos normas adicionales: NMX-I-086/02-NYCE-2006, NMX-I-086/03-NYCE-2006) para proporcionar una orientación más detallada. El global de todas estas partes ayuda a

---

104FRIEDMAN, Richard. OpenMP Specifications [en línea]. Actualizada: 13 enero 2009. [Fecha de consulta: 19 junio 2009]. Disponible en: <http://openmp.org/wp/openmp-specifications/>

105Internet Engineering Task Force (IETF) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, como transporte, encaminamiento, seguridad. Fue creada en E.E. U.U. en 1986.

identificar y gestionar todos los aspectos de la seguridad de TI.

También existen normas como la NMX-I-153/01-NYCE-2008 la cual establece las características que han de cumplir los contenidos disponibles mediante tecnología Web en Internet, Intranet y otro tipo de redes informáticas, para que puedan ser utilizados por la mayor parte de las personas, incluyendo personas discapacitadas y personas de edad avanzada, de forma autónoma o mediante las ayudas técnicas pertinentes.

Un caso destacado en México es la norma NMX-I-059 aprobada en 2005. La historia de esta norma<sup>106</sup> comienza en 2002 cuando la Secretaría de Economía empezó a organizar mesas de trabajo para definir las estrategias del programa para el desarrollo de la industria de software, hoy conocido bajo el nombre de PROSOFT. Alcanzar niveles internacionales en capacidad de procesos fue una de las estrategias del programa. Como presidente, en ese entonces, de la Asociación Mexicana para la Calidad en Ingeniería de Software (AMCIS) Hanna Oktaba empezó a coordinar los trabajos de un grupo interesado en este tema.

Se investigaron las características básicas de la industria de software mexicana con respecto a su tamaño y las capacidades de sus procesos y quedó confirmado con los estudios posteriores, que la gran mayoría de las empresas son PyME.

Las empresas con esas características necesitaban algo fácil de entender, práctico y barato. En este mismo momento la SE pidió que fuera norma mexicana. Se revisaron los modelos de procesos disponibles incluyendo ISO9000:2000, CMM-SW, ISO12207, ISO15504 y la versión inicial de CMMI sin embargo, ninguno cumplía tales requerimientos. Por lo tanto se propuso un proyecto para desarrollar un modelo de procesos y un método de evaluación adecuado a la industria mexicana.

El modelo de procesos debía cubrir por lo menos las prácticas de CMM-SW nivel 3 e ISO9000:2000 y cumplir con los lineamientos de ISO15504, con respecto al método de evaluación. Al principio de 2004 ya se contaba con los elementos básicos, el modelo de procesos (MoProSoft) y el método de evaluación (EvalProSoft), para empezar los trámites de

---

106OKTABA, Hanna. Historia de una norma. Software Guru, (3): columna de Tejiendo nuestra red, Mayo/Junio 2005.

normalización en la subcomisión de Software del NYCE<sup>107</sup> (Normalización y Certificación en Electrónica). La norma fue probada, aprobada y dividida en cuatro partes:

- NMX-I-059/01-NYCE-2005: Tecnología de la información – Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software - Parte 01: Definición de conceptos y productos. Esta Norma Mexicana tiene por objeto definir los conceptos y describir los productos para las demás partes de la NMX-I-059-NYCE. Es conveniente que los usuarios de esta norma se familiaricen con la terminología y estructura de la serie de normas que constituyen las NMX-I-059/02-NYCE, NMX-I-059/03-NYCE y la NMX-I-059/04-NYCE.
- NMX-I-059/02-NYCE-2005: Tecnología de la información – Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software - Parte 02: Requisitos de procesos (MoProSoft). Esta Norma Mexicana tiene por objeto definir el modelo de procesos para la industria de software. MoProSoft está dirigido a las organizaciones dedicadas al desarrollo y mantenimiento de software. Es aplicable tanto para las organizaciones que tiene procesos establecidos, así como para las que no cuenten con ellos.
- NMX-I-059/03-NYCE-2005: Tecnología de la información – Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software - Parte 03: Guía de implantación de procesos. Esta Norma Mexicana tiene por objeto proporcionar a las organizaciones de desarrollo y mantenimiento de software un ejemplo de la implantación del modelo de procesos MoProSoft basado en las mejores prácticas de ingeniería de software. Este ejemplo puede servir de apoyo para la definición de procesos en las organizaciones sin procesos establecidos o para la actualización de procesos en las que cuenten con ellos.
- NMX-I-059/04-NYCE-2005: Tecnología de la información – Software - Modelos de procesos y evaluación para desarrollo y mantenimiento de software - Parte 04: Directrices para la evaluación de procesos (EvalProSoft). Esta Norma Mexicana tiene

---

<sup>107</sup>NYCE, Portal de Normalización [en línea]. [Fecha de consulta: 19 junio 2009]. Disponible en: <http://www.normalizacion-nyce.org.mx/>.

por objeto definir las directrices para la evaluación de procesos para la industria de software. Esta Norma Mexicana es aplicable a los organismos de certificación y a las organizaciones dedicadas al desarrollo y mantenimiento de software, que han utilizado la NMX-I-059/02-NYCE para la implantación de sus procesos.

MoProSoft es una norma ideal para cualquier PyME dedicada al desarrollo de software y tiene como ventajas principales<sup>108</sup>:

- Mejorar la calidad del software producido por la empresa que adopta el modelo.
- Elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.
- Integrar todos los procesos de la organización y mantiene la alineación con los objetivos estratégicos.
- Iniciar el camino a la adopción de los modelos ISO9000 o CMMI.
- Sirve para implantar un programa de mejora continua.
- Permite reconocer a las organizaciones mexicanas por su nivel de madurez de procesos.
- Facilita la selección de proveedores.
- Permite obtener acceso a las prácticas de ingeniería de software de clase mundial.

---

108OKTABA, Hanna. MOPROSOFT [en línea]. [Fecha de consulta: 19 junio 2009]. Disponible en:  
<http://uxmcc1.iimas.unam.mx/~ggarzon/moprosoft.html>

