AUTOMATIZACIÓN DE LA PRODUCCIÓN DE CONTENIDO PARA E-LEARNING USANDO OPENOFFICE Y OTRAS HERRAMIENTAS OPENSOURCE: CASO DE ESTUDIO

APLICACION DE TECNOLOGIAS OPENSOURCE PARA OPTIMIZAR PROCESOS DE PRODUCCIÓN DE E-LEARNING EN ORGANIZACIONES PEQUEÑAS

Autor: Cesar Pachón
cesarpachon@gmail.com
www.cesarpachon.com
2011
Bogotá, Colombia

CONTENIDO

Resumen

Se presenta un caso de estudio en el que el desarrollo de una herramienta inhouse utilizando openoffice y otras tecnologías libres que permite optimizar la cadena de producción de contenido e-learning. Se enfatiza en la descripción precisa de los componentes de la solución así como la justificación de las decisiones de diseño y el impacto al integrar la herramienta dentro de la cadena de producción de la organización pequeña.

Introducción

La producción de contenidos para e-learning es un proceso de múltiples etapas en donde cada una de ellas posee sus propios objetivos, requerimientos de entrada y productos que son entregados al siguiente eslabón de la cadena de producción.

Implementar esta cadena de manera exitosa implica tener en cuenta la importancia de cada uno de sus eslabones, sus características, el perfil del personal responsable y las tareas a desarrollar. Este conocimiento permite optimizar el proceso seleccionando herramientas menos costosas, cambiando procedimientos y automatizando algunas de las tareas, allí donde la especificación y estandarización de las mismas lo haga posible.

Este documento se enfoca en la etapa de **implementación**, en la cuál ya está el contenido diseñado por los autores expertos, y debe ser transformado en el producto final ya sea para ser desplegado en un LMS ó para su envío a imprenta.

Se presenta la suite ofimatica libre "OpenOffice" [openoffice], y se describen todos los pasos llevados a cabo para integrarla con un desarrollo propio con el fin de automatizar al máximo la etapa de implementación de contenido, sin generar impacto negativo en la entrega del contenido por parte del autor, y como esta herramienta afectó positivamente la cadena de producción.

El documento hace énfasis en describir todos los componentes de la solución y su integración dentro del flujo de trabajo, manteniendo al mínimo aspectos demasiado técnicos como detalles de programación. Sin embargo, se incluye en las referencias otro documento del mismo autor en donde se enfatiza en estos detalles, para aquellos lectores con perfil de desarrollador que quieran tomarlo como base para sus propias soluciones.

Generalidades

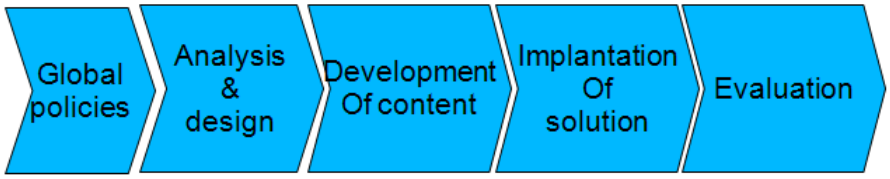

Figura 1: Cadena de valor del e-learning (tomado de [PACHON01])

La figura 1 presenta las etapas de implementación de un proyecto de e-learning. La primera, "políticas globales" pretende explicar que la implementación de todo proyecto dentro de una organización está limitada, parametrizada o definida por factores ambientales y organizacionales que desde el contexto del proyecto no son modificables.

En organizaciones pequeñas, estas políticas (ya sean explícitas ó tácitas) suelen incluir limitaciones presupuestales y de recursos humanos por lo que maximizar el grado de optimización de los procesos de producción suele ser un componente crítico para la viabilidad del proyecto.

La etapa de "análisis y diseño" incluye la especificación de requerimientos y determinación de objetivos de aprendizaje, así como el esbozo de la estructura de contenidos y otras directrices tanto de tipo pedagógico como técnico. El rol principal en esta etapa es el de "diseñador instruccional".

Después aparece la etapa de desarrollo de contenido. En esta etapa ocurre la creación de contenido original por parte de autores expertos, la creación de artes y otros recursos multimedia y la unificación del contenido en el producto final, usualmente módulos SCORM listos para ser desplegados en un LMS (más adelante se realiza una ampliación de estos términos).

Es en esta etapa en donde se centra el tema de este documento. Como puede deducirse de la descripción anterior, la cantidad de roles involucrados y su nivel de especialización es mucho más amplia, como lo ilustra la figura 2: En ella podemos apreciar que por una parte está el autor experto, quien no necesariamente debe saber detalles de diseño instruccional ni e-learning en general; lo mismo aplica para el artista, mientras que el desarrollador, además de saber de lenguajes de programación también debe conocer un mínimo de los protocolos de empaquetamiento y estándares de comunicación propios de los paquetes SCORM.

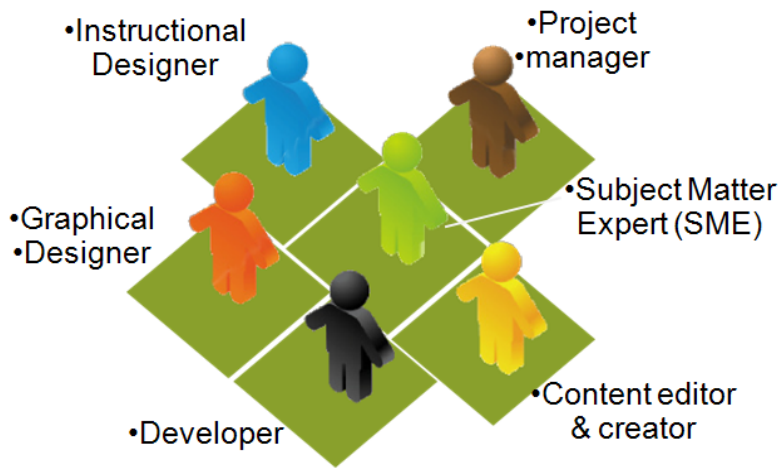

Figura 2: Roles involucrados en la etapa de desarrollo. tomado de [PACHON01]

Es importante aclarar que el esquema representa ROLES, no individuos. Es muy usual que varios roles sean asumidos por la misma persona, o que un rol esté repartido entre varios miembros del equipo[1].

Tendencias

Como se mencionó en la sección anterior, el diseño de una solución está supeditado tanto a nivel organizacional, pedagógico y tecnológico a restricciones y directrices. En el caso de la etapa de desarrollo, existen determinadas tendencias y prácticas comunes en esta industria, como son el uso de plataformas de gestión de aprendizaje ó LMS, y empaquetamiento de contenido usando el estándar SCORM.

LMS

Con la popularización de las aplicaciones web surgió la posibilidad de crear aplicaciones centralizadas, basadas en web, que facilitaran tareas de gestión y soporte relacionadas con los procesos de aprendizaje. En escenarios de aprendizaje altamente estructurados, como instituciones educativas y cursos al interior de organizaciones, estas plataformas permiten la gestión de estudiantes y grupos de estudiantes restringiendo el acceso a contenidos de acuerdo al calendario académico, registrando calificaciones y monitoreando la actividad de los estudiantes con el fin de facilitar los procesos de auditoría y control de calidad.

Hoy en día los LMS ofrecen muchas otras funciones, como soporte para evaluaciones en línea,

1 Esto es particularmente cierto en el caso de la producción de contenidos basados en Flash: usualmente la persona a cargo es una mezcla de artista y programador, con conocimientos básicos de Action Script, el lenguaje de programación empleado en esta popular tecnología de Adobe.

espacios de discusión (foros, chats, wikis) y en general, muchas de las herramientas propias de la web 2.0.

Existe una gran oferta de plataformas LMS en el mercado, que van desde aquellas basadas en software libre, como Moodle, Dokeos y Claroline, hasta soluciones costosas como BlackBoard.

Sea cual sea la plataforma elegida, es importante entender que un LMS no es una solución de elearning empaquetada ylista para usar. Es un elemento más dentro del ecosistema completo de un proyecto de e-learning, y por ello su elección debe basarse en aspectos integrales que tomen en cuenta toda la cadena de valor.

Publicación de contenidos

Antes de la aparición de los LMS, la principal manera de distribuir contenido era utilizando CD-ROMs que permitían una experiencia multimedia rica al poder llevar en el CD todos los videos, audios y animaciones asociadas al material, sin depender de las limitaciones de conectividad y ancho de banda de la época. Sin embargo era claro que estas herramientas desconectadas no permitían el monitoreo y control de las actividades del estudiante, ni tampoco la creación de actividades grupales, además de poseer sus propios problemas de distribución, en especial en proyectos de educación a distancia.

Por esta razón, la primera aplicación obvia de los LMS fue la de medio de publicación y distribución del contenido. Todos los LMS permiten la creación directa de contenido a la manera en que se edita hoy en día un blog o una wiki, y en muchas organizaciones se comete el error de instruir a los autores expertos para que directamente "suban" contenidos al LMS, con la idea de reducir costos de artistas, desarrolladores, etc.

Esto es un gran error ya que como se mencionó al presentar la cadena de valor, un autor no necesariamente conoce la mejor manera de implementar el contenido.

Otra enorme desventaja de subir el contenido directamente al LMS es que este contenido quedará atado a dicha plataforma. No será un proceso simple hacer que el contenido creado dentro de moodle se pueda publicar directamente en blackBoard, por ejemplo. Si el lector piensa que este problema sólo afecta a aquellas organizaciones cuyo modelo de negocio es la venta de contenido para terceros, esto no es así. Una organización que produce su propio contenido podría verse en la necesidad de migrar de LMS en algún momento en el futuro, viéndose en el tremendo problema de liberar sus contenidos de la vieja plataforma, que ahora actúa como una "prisión de contenidos".

SCORM

La necesidad de darle a los problemas de publicación en diversos LMS hizo que se establecieran estándares para que el contenido fuera transportable de un LMS a otro. Los estándares más aceptados actualmente son la familia de estándares SCORM, de la ADL [ADL]. La especificación más reciente del estándar SCORM datan del año 2004, y si bien hoy en día reciben muchas críticas por no estar actualizados para funcionar con las tecnologías web más recientes, lo cierto es que siguen siendo la mejor manera de crear contenido portable.

Hablar de que nuestro contenido es "compatible con SCORM" puede ser tan ambiguo como decir que un auto es "compatible con las carreteras".. como la mayoría de estándares abiertos, SCORM define un conjunto de directrices en donde las diferentes implementaciones tienen un buen margen de flexibilidad para operar. Las formas más comunes de implementación incluyen:

1. ignorar la especificación de secuenciación de SCORM. El estandar permite definir dentro de un paquete diferentes "SCO" (shareable content objects) junto con una serie de reglas para navegar entre estos. Aunque es un concepto interesante, en la práctica se tiende a generar paquetes independientes para cada SCO, en vez de generar un paquete monolítico. Esto tiene una ventaja importante que es el mejor aprovechamiento de recursos de web 2.0 que no se pueden (o no se deben..) integrar al SCO. Al menos si se quiere ser puristas con el estándar![2]
2. Utilizar el modelo de manifiesto en su forma más básica (una organización, un SCO listando todos los archivos del paquete).
3. Hacer llamadas básicas a la API de runtime del LMS, al menos, enviar una calificación final: SCORM especifica un sistema de comunicación entre el SCO y el LMS a través de javascript. La lista de llamadas posibles es larga, sin embargo en la práctica su uso real es limitado.
4. Usar el estado del SCO para almacenar información propia en forma codificada. La especificación original de SCORM permite guardar una cadena de texto de hasta 255 caracteres cuyo contenido es de libre interpretación. Muchas herramientas de authoring (ver sección siguiente) hacen uso de esta variable para mantener el estado de navegación del SCO de manera automática, por ejemplo.

Authoring

Consientes de las necesidades de las pequeñas organizaciones de reducir los costos de recurso humano las casas desarrolladoras de software pronto se volcaron a la tarea de crear herramientas que redujeran la brecha entre autor y publicación en el LMS.

Algunas de estas herramientas se enfocan en tomar presentaciones de power point y transformarlas en paquetes SCORM (tendencia a la que se le conoce como "rapid e-learning"). Otras incluyen entornos de diseño más elaborados, donde se pueden insertar animaciones, hacer algo de programación, e insertar actividades interactivas a partir de plantillas. No deja de ser irónico que esta "profesionalización" de una herramienta de authoring va en contra de la naturaleza misma de la herramienta: se supone que la herramienta de authoring debe ser tan simple que no requiera artistas ni desarrolladores, pero a medida que aumenta su funcionalidad, pasa lo inevitable: se contrata un desarrollador para que aprenda a usar la herramienta, y el autor le pasa el contenido a este..

Presentación del caso

El caso de estudio se trata de una institución educativa pequeña con un proyecto de educación virtual de uso interno (no para proveer contenido a terceros). El equipo de desarrollo original incluía un diseñador instruccional, un editor de contenido, un ingeniero multimedia y un diseñador gráfico. Los autores se contrataban por outsourcing, de manera que estaban aislados de el resto del proceso de producción.

2 Un ejemplo de esto es cuando se tiene un blog o foro como actividad de aprendizaje, y se coloca un hipervínculo dentro del contenido SCO. SCORM especifica que un paquete debe ser autocontenido, por lo que el link al recurso externo rompe este paradigma.

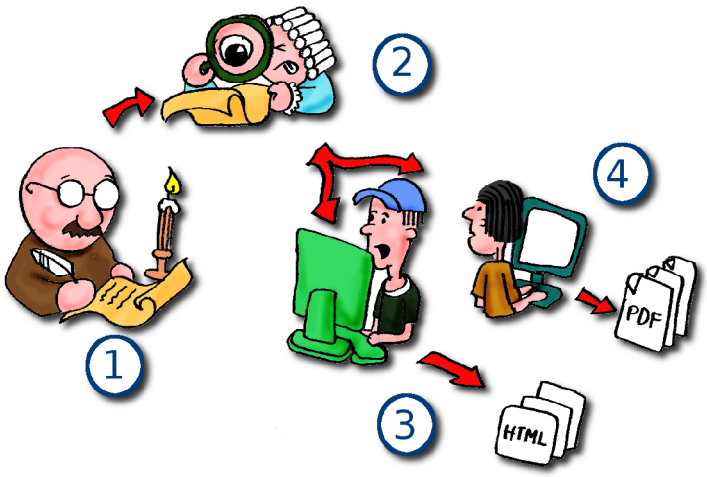

Figura 3: Proceso original de producción de contenido

Los entregables de este equipo de trabajo debían ser contenido en línea, publicable en un LMS, y contenido impreso, listo para enviar a imprenta.

La forma de trabajo era la siguiente (ver figura 3):

- el autor y el diseñador instruccional tenían una reunión previa, en donde se especificaban los requerimientos del contenido a producir (1).
- El autor envía avances que son revisados por el editor de contenido, en cuanto a forma y estilo. También debe realizar una revisión de la originalidad del trabajo, con el fin de evitar problemas legales por plagio (2).
- Se realiza una entrega formal de la revisión final.
- Una vez aprobado, este contenido se entrega al equipo de desarrollo.
- El ingeniero multimedia parte de una plantilla SCORM prediseñada para los cursos de la institución, realiza cambios mínimos como título, portada, etc.
- El ingeniero multimedia empieza a transcribir a HTML y algo de flash el contenido del autor. Algunas actividades se implementan en flash (3).
- El diseñador gráfico elabora algunas de las ilustraciones.
- El diseñador gráfico utiliza un programa avanzado para crear la versión impresa del contenido. Repite el proceso de transcripción que está haciendo el ingeniero multimedia, pero en otra herramienta, con el fin de producir un PDF (4).

Descripción del problema

Los problemas de esta forma de trabajo son:

1. Duplicidad de esfuerzos: el ingeniero y el desarrollador realizan el mismo trabajo:

transcriben el mismo contenido, cada uno en su propia herramienta. Qué pasa si hay que hacer un cambio a futuro en el contenido?

2. Desperdicio de recursos: un ingeniero multimedia y un diseñador gráfico son recursos valiosos. El ingeniero debería concentrarse en crear simulaciones, animaciones y otros recursos interactivos. El diseñador, en las ilustraciones y plantillas. Tener a ambos realizando una tarea tediosa y repetitiva (transcribir contenido) es un claro desperdicio de recursos.
3. Riesgos legales y de seguridad: cada profesional trae mentalizado desde la universidad que "sólo es profesional" usar determinada herramienta comercial de software para hacer su trabajo. Coincidencialmente se trata de las herramientas cuyas licencias son las más costosas del mercado! La organización no cuenta con los recursos para ese tipo de licencias, asi que aprovechando la informalidad de una organización pequeña, instalan por su cuenta versiones sin licenciar de dichas herramientas. Esto no sólo pone en riesgos legales a la organización, sino que también la expone a riesgos de seguridad, ya que la manera más fácil de transmitir virus, programas troyanos y otras amenazas es ocultándolos dentro de instaladores de programas conocidos que los hackers liberan en la red y que los usuarios descargan e instalan voluntariamente, pensando que se están ahorrando un buen dinero.
4. Tiempos de entrega: los tiempos de entrega se estiran innecesariamente. El proceso de transcripción toma la mayor parte del tiempo del equipo de trabajo.

Planteamiento de la solución

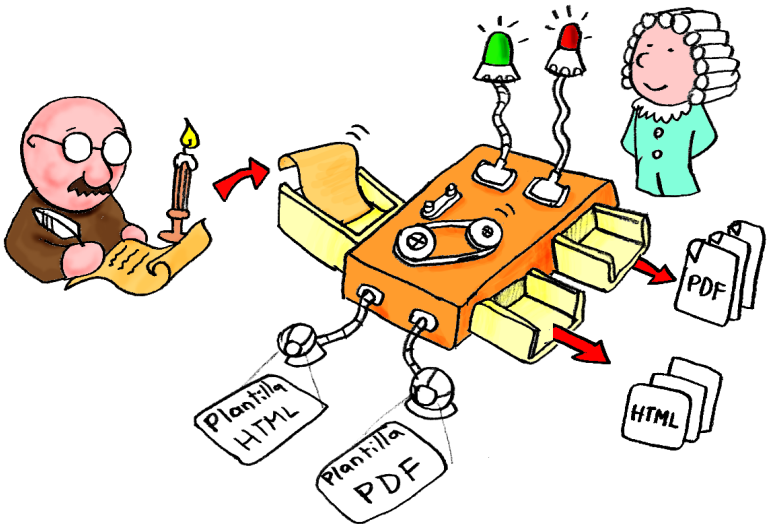

Figura 4: Planteamiento de la solución

Se plantea desarrollar un sistema que automatize al máximo el desarrollo de contenido, con el fin de agilizar el proceso de producción y liberar recurso humano para tareas más especializadas. En particular se quiere eliminar el proceso de transcripción tanto para el contenido en línea como para el contenido impreso. Idealmente, también ayudar al editor de contenido en sus revisiones

de forma y originalidad.

Componentes de la solución

El componente central de la solución es un programa desarrollado en JAVA, conectado a la suite ofimática "openoffice". De esta manera, el programa es capaz de "leer" documentos de texto, ya sea en el formato de openoffice (.odt) ó en el formato de Microsoft Word (.doc, .docx), analizar su estructura (ver figura 5) y extraer contenidos.

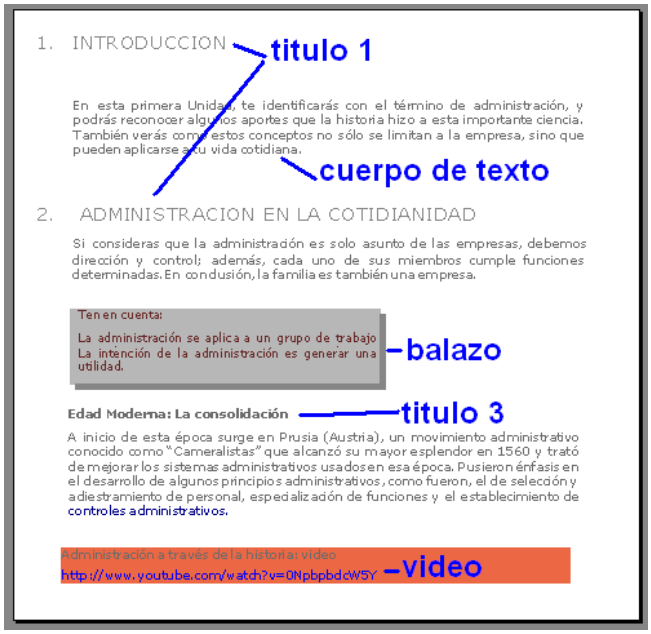

Figura 5: Aplicación de estilos en OpenOffice

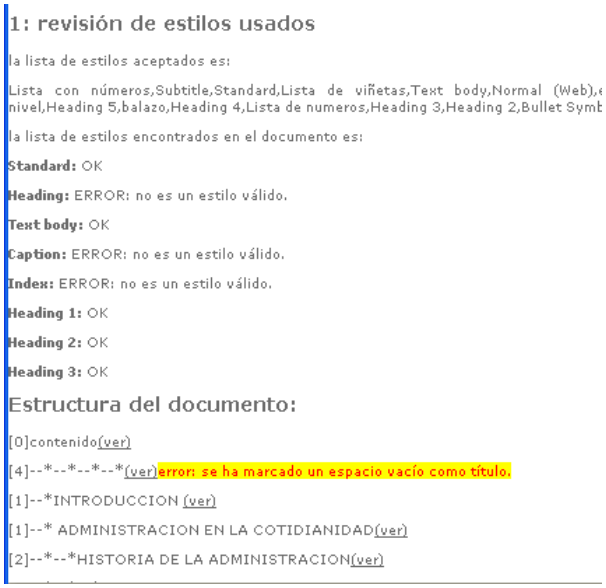

Figura 6: Resultado de la revisión automática de estilos

El programa puede reconocer el estilo de cada párrafo (título 1, título 2, comentario, etc) y de esta manera, generar automáticamente tablas de contenido y paginar el texto. El texto producido se pasa a HTML para integrarlo al paquete SCORM.

El programa también es capaz de:

- extraer las imágenes y diagramas que se encuentren dentro del documento, guardarlas como archivos PNG para ser visualizados en internet, y referenciarlos correctamente dentro del contenido HTML.
- Detectar las tablas con estructura básica y reproducir dicha estructura en HTML.
- Detectar estilos personalizados especiales, por ejemplo: "balazo". Un párrafo marcado como balazo producirá en el HTML y el PDF para imprenta, un cuadro secundario, en la margen del texto.
- Insertar videos de youtube dentro del contenido HTML: si el programa encuentra un párrafo marcado con el estilo personalizado "video", asumirá que el contenido es un hipervínculo a un video de youtube, y generará todo el código necesario para incrustar el video dentro del contenido del paquete SCORM.
- Producir correctamente el manifiesto para el paquete SCORM, empaquetar todo el contenido en un archivo ZIP listo para ser subido al LMS.

revisión anti-plagios

los siguientes párrafos necesitan verificar su originalidad o que hayan sido citados adecuadamente:

---

se encontraron coincidencias para este parrafo:

La partida doble es la base de la contabilidad actual y es parte fundamental de la Ecuación Patrimonial.

porcentaje= 75.0

texto en web:

La contabilidad es un campo o disciplina perteneciente a la ciencia de la ... La partida doble es la esencia de l Contabilidad actual y parte integral de la Ecuación ...

ruta original:

http://www.temas-estudio.com/ecologia-y-contabilidad/

---

se encontraron coincidencias para este parrafo:

No existe deudor sin acreedor ni acreedor sin deudor (Los recursos no surgen por si solos, deben proceder de algún lado)

porcentaje= 80.0

texto en web:

"No existe deudor sin acreedor, ni acreedor sin deudor" Claro esta que estos principios no son tan obvios en ... (Lo recursos no surgen por si solos, debe proceder de algíºn lado) ...

ruta original:

Figura 7: Resultados de revisión antiplagios

Adicionalmente, el programa es capaz de tomar cada uno de los párrafos del documento de texto y realizar una búsqueda automática con el motor de búsquedas de Yahoo!. Luego, genera un "índice de originalidad", un porcentaje que indica qué tanto del contenido del documento podría no ser original, así como un reporte de los fragmentos con más alto grado de sospecha de plagio, para su revisión manual por parte del inspector de contenido (ver figura 7).

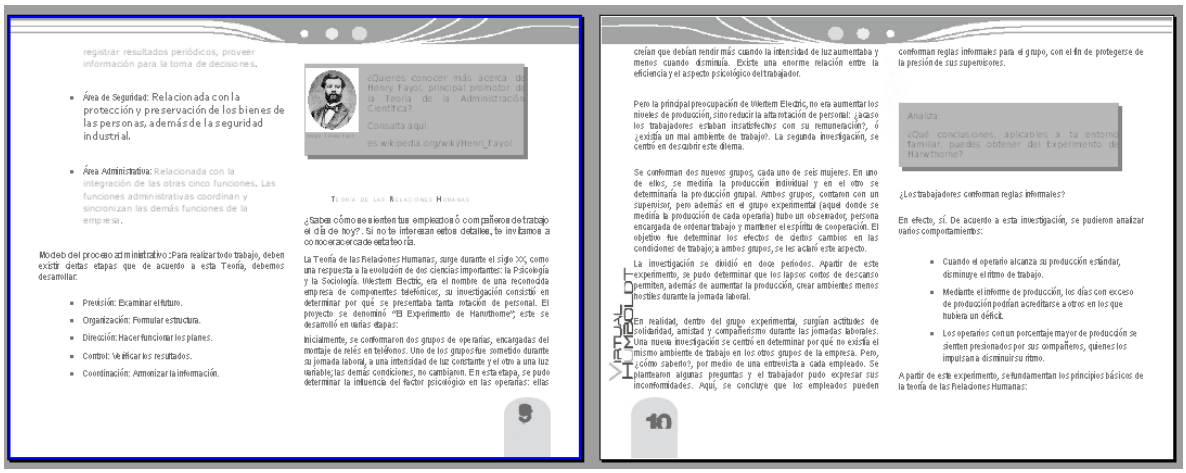

Figura 8: Documento maestro en openoffice

Para producir los PDF, se pensó originalmente en desarrollar un módulo del programa para

realizar esa labor, sin embargo, un poco de investigación sobre las funcionalidades avanzadas de openoffice, en particular el "merging" o fusionado de archivos para crear meta-documentos, permitió implementar un mecanismo para producir PDF usando sólamente funcionalidad de openoffice y sin recurrir a escribir ni una sola línea de código en JAVA. Lo mejor es que este sistema de documentos maestros permitía que al llegar un cambio en el documento del autor el documento maestro se actualizara de manera automática.

Implementación y resultados

Toda nueva herramienta o nueva forma de hacer las cosas genera un impacto en la organización, y esta herramienta no es la excepción. Sin embargo es notable que la resistencia a la nueva solución haya sido mínima. El proceso de implementación involucró:

1. Capacitar a los autores para que aplicaran estilos en vez de dar formato directo a los textos que entregaban.
2. Concientizar al desarrollador y diseñador gráfico de que la herramienta les liberaría tiempo para tareas más creativas.

Los resultados obtenidos al implementar la herramienta fueron:

1. Mejora en la estandarización de la forma de entrega de contenidos de los autores. Estos ya se concentran en aplicar estilos en vez de dar diseño visual a sus textos.
2. Reducción de cerca del 60% de los tiempos de revisión del editor de contenidos. Esto gracias a dos cosas: la validación automática de forma (títulos, estructura, profundidad del árbol de contenido) y a la validación anti-plagio.
3. Reducción de más del 80% en los tiempos de transcripción del desarrollador. Actualmente el trabajo se reduce a una revisión del producto de la ejecución del programa para realizar ajustes menores.
4. Reducción del 100% en el tiempo de transcripción del diseñador gráfico para producir PDF. Se liberó este importante recurso quedando disponible para diseñar plantillas, ilustraciones y otras tareas de alta creatividad.
5. Reducción cercana al 100% en los tiempos de actualización de contenidos. Antes una corrección requería un reproceso manual tanto del SCORM como del PDF. Actualmente la actualización de ambos productos es totalmente automática luego de haber sido producida por primera vez.
6. Eliminación de todo el software no licenciado así como de la necesidad de adquirir software costoso.
7. Estandarización del proceso de producción, integración de gestión de calidad en una sola herramienta.

Justificación de la solución

En este punto vale la pena preguntarse: por qué OpenOffice, por qué Java? La respuesta simplemente tiene que ver con aprovechar los recursos limitados con los que cuenta una organización pequeña, y el más importante de estos es el conocimiento que está en las cabezas de los miembros de la organización. En este caso, el autor poseía amplia experiencia en el uso de openoffice como usuario final y como desarrollador, también mucha experiencia en el uso de

Java como lenguaje de programación y predilección por el software libre. En una organización que contase con licencias de Microsoft Office es totalmente posible realizar este mismo desarrollo usando Microsoft Office y VBA (Visual Basic for Applications).

Conclusiones

1. Las organizaciones pequeñas tienen que ser altamente creativas a la hora de optimizar sus procesos productivos. Esta necesidad no aplica sólamente a incrementar utilidades sino que en muchos casos es crítica para la supervivencia de la misma organización.
2. El recurso más importante de una organización, y en especial una organización pequeña, es el know-how de sus integrantes. La gente es un recurso demasiado valioso para desperdiciar en tareas que tienen un alto potencial de automatización.
3. El software libre es una solución real para las organizaciones. Si bien es cierto que en ciertos nichos específicos las alternativas comerciales suelen contar con funcionalidades más avanzadas, la relación costo/beneficio que resulta de comparar los costos de licenciamiento con la subutilización a la que se someten estas herramientas resulta a favor del software libre. Si en vez de comparar costos de licenciamiento se reemplaza por la instalación de software pirata, las implicaciones de seguridad siguen inclinando la balanza a favor de las alternativas abiertas.

Referencias

[PACHON01] e-learning industry overview, Pachón Cesar, http://www.cesarpachon.com/index.php?id=presentations.php#p03

[openoffice] www.openoffice.org: suite ofimatica libre (opensource).

[moodle] www.moodle.org: LMS opensource basado en PHP.

[ADL] http://www.adlnet.gov/ organización encargada de la especificación SCORM y estándares relacionados.