

EDI. ESTRATEGIA DIDÁCTICA PARA LA ELABORACIÓN DE MATERIALES EDUCATIVOS COMPUTARIZADOS

EDI. A TEACHING STRATEGY FOR THE DEVELOPMENT OF EDUCATIONAL MATERIALS COMPUTERIZED

Victor Esteller
vestelle@uc.edu.ve

Elsy Medina
emedina@uc.edu.ve

Universidad de Carabobo,
Valencia, Edo. Carabobo, Venezuela

Recibido: 15/12/2014
Aprobado: 12/02/2015

Resumen

La presente investigación expone la estrategia didáctica (EDI), definiendo una arquitectura como lineamiento para la creación de materiales educativos computarizados de calidad, con altos niveles de reutilización, representando un recurso facilitador capaz de mejorar los procesos de enseñanza y de aprendizaje. Para esto, se hizo una investigación acción; recabando información exhaustiva de procesos de desarrollo, ingeniería de *software*, metodologías ágiles y otras disciplinas en el ámbito tecnológico educativo. Se realizó, por un lado, el análisis riguroso, tanto técnico como didáctico, de las posibilidades funcionales de las TIC y, por otro lado, su integración en un planteamiento didáctico general.

Palabras clave: estrategia didáctica, materiales educativos computarizados, ingeniería de *software*.

Abstract

This research presents the teaching strategy EDI, defining architecture as a guideline for creating high quality computerized educational materials, with high levels of reuse, representing

a facilitator resource capable of improving teaching and learning processes. It was an action research gathering comprehensive information from processes of development, software engineering, agile methodologies and other disciplines in the educational technology field. Firstly, it was performed a technical and didactic rigorous analysis related to ICT functional possibilities, and secondly, their integration into a comprehensive didactic approach.

Keywords: Teaching strategy. Computerized educational materials. Software engineering.

1. Introducción

La estrategia propuesta, (EDI) está enfocada a la construcción de materiales educativos computarizados con pequeños grupos de desarrollo, de no más de 15 personas involucradas en el proyecto, las cuales deben compartir un ambiente de trabajo común, con mínima separación física.

Cabe mencionar que en caso de contar con una mayor cantidad de recursos se deberán dividir en subgrupos del tamaño antes indicado, trabajando en forma paralela. Es así que se presenta EDI como una estrategia para ser utilizada en proyectos que tienen pequeños grupos de personas en el equipo de desarrollo.

En particular, EDI es creada por el investigador para ser utilizada en la Facultad de Educación de la Universidad de Carabobo como parte de un proyecto en el contexto de la Línea de Investigación Software Educativo del Departamento de Computación de la Facultad de Ingeniería de la misma Universidad, para especificar la arquitectura y elaboración de materiales educativos computarizados y ser aplicada en pequeños grupos de desarrollo, sin necesidad de invertir en costosas herramientas o de capacitar en forma extensa a las personas involucradas. Por tratarse de una estrategia ágil (Sommerville 2011), se intenta minimizar la burocracia que está presente en la utilización de procesos complejos, que son utilizados en proyectos con gran cantidad de recursos.

EDI no establece técnicas, sólo algunos casos particulares, que son mencionadas en conjunto con las actividades correspondientes. EDI es una estrategia con orientación a las

personas y a los productos; no señala ningún tipo de herramienta para llevar a cabo las actividades. Sin embargo, en algunos casos se mencionarán ciertas herramientas como ejemplificación.

2. Filosofía de EDI

La estrategia didáctica EDI tiene como pilares fundamentales en su accionar, tres aspectos clave, como son las técnicas de moderación (Del Pozo, 2013), el trabajo colaborativo Lucero (2013) y la gestión del conocimiento (Fernández, 2012). Es de resaltar que estos fundamentos deben ejecutarse de forma iterativa e incremental.

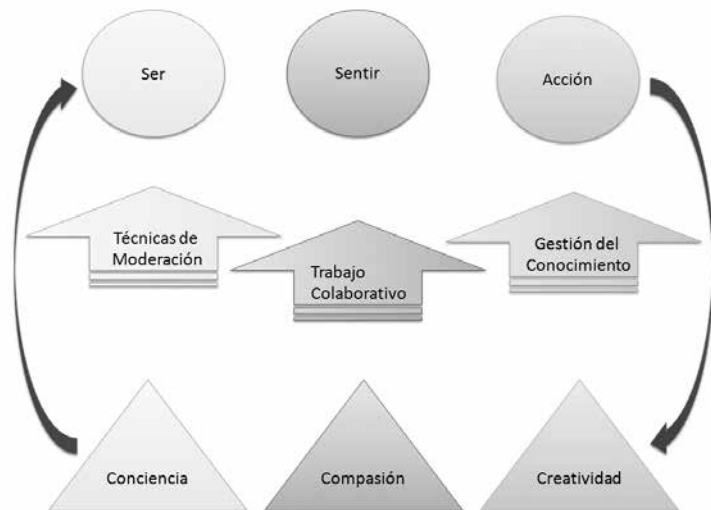
A su vez, deben representar el espíritu del equipo de desarrollo. Es responsabilidad del líder de proyecto garantizar el cumplimiento de la filosofía, para ello pudiera emplear técnicas de motivación y el uso de metáforas. En la Figura 1 se puede observar la filosofía de EDI.

3. Elementos de la filosofía de EDI

3.1. Técnicas de moderación

Las técnicas de moderación son responsabilidad del líder de proyecto y del coordinador, para tal fin se debe emplear un lenguaje amigable, respetuoso, paciente, crítico, abierto siempre al dialogo y al intercambio de ideas, creando de esta manera un grupo de trabajo consciente de sus responsabilidades y orgulloso de ser parte del proyecto, (Del Pozo, 2013). Este proceso innovador, hace necesario el diseño de estructuras pedagógicas específicas, que propicien la interacción y la interactividad y el trabajo colaborativo entre los actores. Acortar la distancia entre asesor y estudiante, y dar respuesta inmediata a los requerimientos de este último. Para ello es inminente la utilización de las tecnologías comunicacionales como herramientas didácticas, bajo la premisa de que su eficacia estará supeditada a la posibilidad de que a ellas accedan masivamente los usuarios.

Figura 1: Filosofía de EDI. Fuente Autor (2014).



En este orden de ideas, se propone como línea de acción la humanización de la actividad académica en el entorno de desarrollo de los materiales educativos computarizados y dentro de los mismos; igualmente se deben emplear las TIC para maximizar la interactividad del proceso instruccional y las teorías del aprendizaje (humanismo, constructivismo y aprendizaje colaborativo). Aun cuando el docente y estudiantes sean miembros de una comunidad en aras de la tecnificación o virtualización, es necesario conservar ese tradicional ingrediente de la cultura que llamamos formación humanista.

3.2. Trabajo colaborativo

El segundo fundamento de EDI se corresponde con el trabajo colaborativo (Lucero, 2013), pues si bien, se dividen las responsabilidades en roles y se descompone el desarrollo del material educativo en fases, los diferentes roles deben trabajar de forma simultánea y en conjunto, donde exista una retroalimentación del grupo de trabajo, para que así el equipo aprenda de los éxitos y fracasos en la ejecución de alguna actividad. Para tal fin, el equipo debe estar compaginado como un engranaje, donde se valore y se tenga compasión del desempeño de los

demás compañeros, para que así, cada uno de los miembros del equipo se sienta como parte de una familia, y se logre que cada uno de lo mejor de sí.

3.3. Gestión del conocimiento

La gestión del conocimiento representa el tercer pilar fundamental de EDI, éste tiene presencia en la acción de los diferentes grupos de trabajo (Fernández, 2012), a la hora de generar los diferentes artefactos como resultado de la ejecución de su desempeño (Ej. La satisfacción de los requerimientos y el cumplimiento de los objetivos del proyecto, el diseño de los diferentes modelos UML, (Debrauwer y Van der Heyde, 2009) y la generación de código, entre otros). Este pilar es responsabilidad directa del administrador del conocimiento, el cual debe registrar los diferentes sucesos que puedan ocurrir en los diferentes grupos de trabajo, y de forma muy creativa administrar experiencias, resultados y productos de *software*, en repositorios controlados por bases de datos.

4. Técnicas aplicadas en la estrategia EDI

Las técnicas propuestas por EDI son técnicas que han sido adoptadas por distintas metodologías en las distintas fases del desarrollo de un sistema.

Dadas las características de los materiales educativos computarizados, que concuerdan con los proyectos ágiles descritos por (Sommerville, 2011), se mostrarán las técnicas que más se ajustan a la estrategia EDI y que colaboran al desarrollo de las tareas que son encaradas en cada iteración. Las técnicas propuestas serán mencionadas en relación a las fases en que se divide EDI, ver Figura 2.

5. Fases de la estrategia EDI

Como cualquier estrategia para el desarrollo de software, EDI se compone de un conjunto de fases que son realizadas a lo largo del tiempo, las mismas conforman un período de tiempo encuadrado entre puntos significativos para el proyecto. Específicamente estas fases responden al ¿Qué? y al ¿Cuándo? desarrollar y avanzar en cuanto a la elaboración de un material educativo computarizado.

5.1. Fase de concepción

Se observa en la Figura 2 la primera fase denominada concepción que consiste en la asignación de roles, definición del planteamiento del problema, la justificación del proyecto, aproximación de prototipos (basado en antecedentes) que tendrá la aplicación, el alcance de la misma, la identificación de los involucrados en el desarrollo del proyecto, y llevar a cabo la planificación general del proyecto.

En esta fase es crítico llevar a cabo las primeras iteraciones que se realizan con el cliente de forma de adquirir conocimiento del dominio y analizar si los costos y duración del proyecto estarán justificados o bien conviene comprar algún *software* o replantear la ejecución. Esta fase finaliza con la Redacción de Objetivos y la Especificación de Requerimientos del Ciclo de Vida en el que se evalúa lo realizado contra las expectativas del cliente y del equipo de desarrollo, (Kendall, 2007).

Figura 2: Fases de la Estrategia EDI. Fuente Autor (2014).



5.2. Fase de elaboración

La siguiente fase denominada elaboración, se refiere a la exploración de los requerimientos más críticos (funcionales y no funcionales), (plantilla de especificación de requerimientos), que involucra el proyecto, así como las decisiones técnicas más importantes que quedarán plasmadas en el documento de arquitectura (estructura o estructuras del sistema, que incluye elementos de *software*, las propiedades externamente visibles de esos elementos y la relación entre ellos, (Sommerville, 2011)).

El objetivo principal consiste en asegurar la factibilidad técnica respecto a la realización del proyecto. Es a partir de la próxima fase cuando se comienza con la construcción a gran escala del *software*, en donde se comprometen la totalidad de los recursos necesarios para que el desarrollo se complete en las iteraciones planificadas. Asimismo, se podrán incorporar recursos en forma limitada tratando de no obstaculizar el normal transcurso del proyecto debido a la capacitación que estos últimos requerirán.

La fase de elaboración finaliza con la arquitectura del material educativo computarizado en el que se evalúa lo realizado contra las expectativas del cliente y del equipo de desarrollo. Una vez que se pasan estas etapas de producción se tienen dos fases concatenadas que se realizan en forma repetitiva por cada entrega de la aplicación. Estas fases son construcción y transición.

5.3. Fase de construcción

Durante la fase de construcción se terminan de especificar los casos de uso correspondientes a la iteración (utilizar la plantilla de especificación de casos de uso ver anexo 4), se diseñan los mismos bajo la arquitectura presentada, y se codifican todos los componentes definidos por los casos de uso, ejecutándose las pruebas correspondientes y la integración.

5.4. Fase de transición

Cuando se quiera pasar un cierto conjunto de componentes al entorno productivo se tendrá una fase de transición en la cual se llevarán a cabo las actividades de despliegue necesarias.

La fase de transición finaliza con la muestra (*Release*) del Producto en el que se evalúa lo realizado contra las expectativas del cliente y del equipo de desarrollo.

Cabe mencionar que dentro de las iteraciones de construcción se llevan a cabo actividades relacionadas con requerimientos, diseño, pruebas de usabilidad y satisfacción, verificación de ejercicios prácticos, gestión del conocimiento, redes sociales, entre otras, ya que se trata de un proceso iterativo e incremental ya que se van llevando a cabo tareas en paralelo y la aplicación va evolucionando hasta cumplir los casos de uso definidos.

6. Roles definidos en EDI

Una de las razones principales de la adopción de una estrategia para el desarrollo de materiales educativos computarizados consiste en la definición de las tareas que serán llevadas a cabo por los individuos (profesores y estudiantes), (Responde al ¿Quién?) que participan en el proyecto. Los roles definen ese conjunto ligado de actividades realizadas y artefactos mantenibles que son llevados a cabo por personas o por grupos de personas. No sólo se refieren a personas internas al desarrollo del *software*, sino que también involucran a los usuarios u otras personas que se vean afectadas por el proyecto.

Los roles recién mencionados no son roles exclusivos asociados a una única persona, aunque en algunos casos pueden serlo (Patrocinante). Algunos roles podrán ser abarcados por más de una persona (Desarrollador). Asimismo, una persona podrá cubrir más de un rol (Arquitecto/Escritor Técnico)

Las asignaciones serán llevadas a cabo por el Líder, el cual en base a las aptitudes de los recursos que dispone repartirá las tareas a ser realizadas en cada iteración. Como punto de partida en la definición de roles dentro de EDI (ver Figura 3), se enumera brevemente cada uno, con una descripción concisa de las actividades que abarcan.

Figura 3: Flujos de comunicación durante el proyecto entre los roles propuestos en EDI. Fuente Autor (2014).



6.1. Patrocinante (Cliente)

Actividades: tiene a su cargo el soporte gerencial del proyecto; es el encargado de proveer, comunicar y mantener actualizada la Visión del proyecto; provee el presupuesto para la viabilidad económica del desarrollo; es responsable por la consecución del proyecto del lado del cliente.

Importancia del rol: es esencial para el éxito del mismo, ya que un *software* que no tiene aceptación dentro de la organización que lo financia jamás llegará a ser construido en tiempo, forma y con consentimiento de los usuarios, no siendo utilizado eventualmente si se concreta el proyecto.

6.2. Líder de Proyecto (Docente Responsable)

Actividades: tiene a su cargo la planificación del proyecto, a lo largo de todo el ciclo de vida, incluida la planificación en detalle de cada iteración; asigna recursos y delega responsabilidades en los mismos; fomenta la cohesión del grupo y lleva a cabo actividades destinadas a eliminar fricciones; organiza las reuniones a ser realizadas; monitorea el progreso del proyecto y establece estrategias para mitigar los riesgos que se puedan presentar.

Importancia del rol: el Líder de Proyecto representa la cara visible del equipo de desarrollo, es el nexo existente entre la gerencia (sujeto a quien está dirigido el *software*) y el equipo de desarrollo.

6.3. Experto en el Dominio (Puede ser el cliente o algún especialista externo)

Actividades: tiene a su cargo brindar su conocimiento del negocio contribuyendo al modelado del sistema que llevan a cabo los Analistas durante la disciplina de requerimientos-análisis; participará junto con los *Testers* en la definición del contenido de las pruebas funcionales a ser realizadas; será el responsable de la aprobación de las pruebas de aceptación por cada paquete entregado.

Importancia del rol: el Experto en el Dominio permite al Equipo de Desarrollo aprender sobre el negocio para el cual está siendo construida la aplicación; son encargados de resolver cualquier cuestión relacionada con la funcionalidad de la aplicación junto con los Analistas.

Destrezas: el Experto en el Dominio deberá conocer en detalle el negocio para prestar respuesta a cualquier duda que pueda surgir del mismo. En general será un miembro de la empresa Cliente.

6.4. Coordinador (Estudiante Responsable)

Actividades: tiene a su cargo la supervisión del proceso, y cualquier actividad orientada al mejoramiento del mismo. Durante las primeras etapas de utilización de EDI supervisará la implementación del proceso. *Importancia del rol:* en las metodologías ágiles este rol permite reforzar la adherencia al proceso en aquellos momentos en que el tiempo apremia y se suele caer en el modelo Codificar y Probar.

Destrezas: el coordinador deberá conocer en detalle el negocio para prestar respuesta a cualquier duda que pueda surgir del mismo.

6.5. Analista Didáctico (Estudiante o Grupo de estudiantes encargados)

Actividades: tiene a su cargo el relevar, mediante el cual se obtienen los requerimientos de la aplicación a ser construidos en cada iteración; realiza la especificación de los requerimientos; redacta los objetivos de aprendizaje y aspectos didácticos a ser considerados en el material educativo computarizado; prepara el documento de Visión para la elaboración de la arquitectura.

Importancia del rol: el aprendizaje del dominio de la aplicación y de los requerimientos que deberá tener la misma son claves para el éxito del proyecto y la aceptación del mismo por parte del usuario.

Destrezas: el Analista deberá tener amplio conocimiento de técnicas de especificación de requerimientos.

6.6. Arquitecto (Estudiante o Grupo de estudiantes encargados)

Actividades: tiene a su cargo la definición de la arquitectura que guiará el desarrollo, y de la continua refinación de la misma en cada iteración; deberá construir cualquier prototipo necesario para probar aspectos riesgosos desde el punto de vista técnico en el proyecto; definirá los lineamientos generales del diseño (Diagramas UML, (Debrauwer y Van der Heyde, 2009)) y la implementación (Lenguaje de programación adecuado).

Importancia del rol: la arquitectura es imprescindible en los proyectos de *software* actuales en donde cada vez existe mayor complejidad; el arquitecto puede ser considerado como el Experto en la parte técnica del desarrollo y debe mantener a todo el equipo en conocimiento de los lineamientos fundamentales de la construcción.

Destrezas: el Arquitecto deberá tener una buena formación técnica, contar con experiencia en las herramientas y técnicas utilizadas; aptitudes comunicacionales deseadas para que la arquitectura sea comunicada a todos los miembros del equipo; también deberá ser perseverante en conseguir los hitos técnicos planteados mediante entregables para asegurar el progreso de la construcción.

6.7. Programador o desarrollador (Estudiante o Grupo de estudiantes encargados)

Actividades: tiene a su cargo la codificación de los componentes a desarrollar en la iteración; debe crear y ejecutar las pruebas unitarias realizadas sobre el código desarrollado; es responsable de las clases que ha desarrollado, siendo su responsabilidad el documentarlas, actualizarlas ante cambios y mantenerlas bajo el control de configuración de las mismas.

Importancia del rol: el Programador es la persona que tiene acceso a los materiales educativos computarizados ya elaborados, así como a los diferentes artefactos relacionados y lleva a cabo la implementación de los casos de uso en el lenguaje de programación elegido, el Programador definirá las clases y métodos que realicen los correspondientes casos de uso.

Destrezas: el programador deberá tener amplio conocimiento de las herramientas de desarrollo, del lenguaje de programación, de los aspectos técnicos involucrados.

6.8. Tester (Estudiante o Grupo de estudiantes encargados)

Actividades: tiene a su cargo la generación de pruebas funcionales a partir de los requerimientos extraídos por los Analistas.

Importancia del rol: la importancia del *Tester* radica en la necesidad de construir un *software* de calidad, ISO 25010 (2011), que cumpla con los requerimientos del usuario; mediante la utilización de un proceso y el armado de un grupo cohesivo de desarrollo, se tienen prácticas para garantizar la calidad en el producto desde el punto de vista técnico. Sin embargo, para asegurarnos de que la aplicación satisface las necesidades del usuario debemos realizar todo tipo de pruebas de carácter funcional. Aquí radica el rol del *Tester*, quien crea, ejecuta, analiza y mantiene el conjunto de pruebas automatizadas y manuales que son utilizados.

Destrezas: el Tester debe tener amplio conocimiento de técnicas de *testing*, debe conocer a fondo la aplicación que probará. Asimismo, debe tener conocimientos de programación para trabajar con las pruebas automatizadas.

6.9. Administrador del Conocimiento. (Grupo de estudiantes encargados)

Actividades: tiene a su cargo la captura, refinamiento, empaquetamiento, y transferencia del conocimiento, ya sea tácito o explícito, en el grupo de desarrollo.

Importancia del rol: el Administrador del Conocimiento es uno de los pilares sobre los que se establece EDI. Su importancia consiste en la capacidad del equipo de desarrollo de aprender de la experiencia que éste y que otros equipos dentro de la organización generan a diario durante el transcurso de los proyectos. Para tal fin debe registrar los éxitos y fracasos en una base de datos.

Destrezas: el Administrador del Conocimiento debe poseer aptitudes en comunicación para capturar el conocimiento de aquellas personas que lo generan.

7. Conclusiones

Se diseñó una estrategia didáctica (EDI), para la elaboración de materiales educativos computarizados, considerando aspectos de calidad, enfocados en las Tecnologías de Información y Comunicación, aplicadas al escenario educativo, con el objetivo de que sea fácil de usar y de aprender, intuitiva, memorizable y pueda emplearse sobre diferentes situaciones de forma ágil y rápida, al separar su construcción en componentes, lo que permite trabajar por separado y en paralelo cada una de las fases. De esta manera el diseño, está centrado en el usuario, atendiendo a las necesidades expresadas directamente por los mismos y a estándares internacionales de calidad, lo cual prácticamente garantiza que los materiales educativos computarizados como producto final tengan un nivel de satisfacción alto entre los alumnos y profesores de la Universidad de Carabobo y usuarios en general.

Las estrategias para el desarrollo de *software* ágiles permiten a los pequeños grupos de desarrollo concentrarse en la tarea de construir *software* fomentando prácticas de fácil adopción y un entorno ordenado que ayude a que las personas trabajen mejor y permita que los proyectos finalicen exitosamente. Las mismas están basadas en los cuatro principios del *Manifiesto Ágil* señalados anteriormente.

EDI, la estrategia propuesta en esta investigación, avanza en el conocimiento teórico de estos procesos analizando principios, prácticas y patrones que contribuyan posteriormente a la implementación y adaptación del proceso según la realidad de cada organización.

8. Referencias bibliográficas

- Debrauwer, L.y Van der Heyde, F. (2009). UML 2, Iniciación, ejemplos y ejercicios corregidos. Segunda Edición. Ediciones ENI. España.
- Esteller, V. (2009). Técnicas y herramientas para el modelado de *software*. Universidad de Carabobo.
- Esteller, V. y Medina, E. (2009). Evaluación de cuatro modelos instruccionales para la aplicación de una estrategia didáctica en el contexto de la tecnología. *Eduweb, Revista de Tecnología de Información y Comunicación en Educación*. Universidad de Carabobo. Volumen 3, N° 1.
- Esteller, V. y Medina, E. (2012). Procesos de desarrollo de *Software* y materiales educativos Computarizados. *Eduweb, Revista de Tecnología de Información y Comunicación en Educación*, Volumen 6, No. 1. Venezuela.
- Fernandez, M. (2012). Red de investigadores educativos: gestión del conocimiento y formación en prácticas democráticas. *Ciencia y Sociedad*, Vol. 2, pp. 36.
- ISO/IEC 25010. (2011). Software engineering – Software product quality requirements and evaluation (SQuaRE)- Quality model. International Organization for Standardization (ISO).
- Kendall, K. y Kendall J. (2007). *Systems Analysis and Design*. 7° edition. Prentice Hall
- Lucero, M. (2013). Entre el trabajo colaborativo y el aprendizaje colaborativo. *Revista Iberoamericana de Educación*, pp. 1-21. Key: citeulike:12314766.
- Sommerville, I. (2011). *Ingeniería de Software*. 9na edición. Prentice Hall.