

EL MODELO INCREMENTAL PARA EL DESARROLLO DE UN SISTEMA INFORMÁTICO DE GESTIÓN DE INFORMACIÓN

THE ITERATIVE AND INCREMENTAL MODEL FOR THE DEVELOPMENT OF A COMPUTER INFORMATION MANAGEMENT SYSTEM

Autora: Dayai A. Ortiz Negrín dortizn@estudiantes.udg.co.cu

Carrera: Ingeniería en Ciencias Informáticas. Universidad de Granma. Cuba.

Tutora: Ing. Yaira Caridad Bagarotti Acebo ybagarottia@udg.co.cu

Centro Universitario Municipal de Media Luna de la Universidad de Granma.

Resumen:

En el presente artículo se propone la utilización del modelo iterativo e incremental (MII) como metodología de desarrollo para un SISTEMA INFORMÁTICO DE GESTIÓN DE INFORMACIÓN relacionada con la tecnología educativa de la Dirección Municipal de Educación de Media Luna. Se cometió una revisión bibliográfica que permitió comparar dos modelos de desarrollo de software; uno tradicional, llamado modelo en cascada el cual fue comparado con el Modelo Incremental basado en iteraciones. Entre los resultados enfatizan las ventajas al manejar el Modelo Incremental que anticiparon su uso. Se concluye, que el empleo del modelo seleccionado permitirá un aprendizaje en cada iteración, ofreciendo un producto de acuerdo con las necesidades del cliente y garantizando en la investigadora una mayor experiencia para la toma de decisiones en futuros proyectos con características similares.

Palabras clave: Sistema de gestión, modelo iterativo incremental; Tecnología educativa

Introducción.

La tecnología educativa, es el resultado de las aplicaciones de diferentes concepciones y teorías educativas para la resolución de un amplio espectro de problemas y situaciones referidos a la enseñanza y al aprendizaje. Surge como disciplina en Estados Unidos de América en la década de los cincuenta del siglo pasado y ha transitado por diferentes enfoques o tendencias como enseñanza audiovisual, enseñanza programada, tecnología instruccional y diseño curricular, entre otros. Utiliza los medios y recursos de la enseñanza como componentes activos en todo proceso dirigido al desarrollo de aprendizajes. (María Vidal Ledo, Carlos Raúl del Pozo Cruz, 2008).

La educación del siglo XXI está llamada a avanzar en la dirección (y la velocidad) adecuada para enfrentar los diversos desafíos y oportunidades que ofrece la sociedad del conocimiento. Por ello, se puede postular que debe existir una estrecha relación entre

aprendizaje, generación de conocimiento, innovación continua y uso de las nuevas tecnologías [Garcés Suárez Et Al, 2016]

Por la importancia de esta tecnología en la educación es que se debe de tener en cuenta la disponibilidad técnica de la misma, pues es así que en cada centro educacional se debe de llevar el control de bajas, roturas, es decir, saber con los equipos que se cuenta para de una manera u otra influenciar para bien en el desarrollo del conocimiento además de supervisar el cuidado y conservación que se le debe de dar a cada uno de estos equipos.

En la provincia Granma se han llevado a cabo: con el objetivo de llevar la informatización a cada centro estudiantil cubano, obtener contenidos de impacto, actualizados y que potencien el alcance de una preparación tecnológica-educativa acorde con los nuevos tiempos, varios procesos como el internet en los centros de estudios. Con el devenir del tiempo además se introdujeron el uso de tabletas, mayores números de computadoras al igual que de televisores ya que se busca que el estudiantado posea las habilidades técnicas necesarias y no ingrese con carencias respecto a este tipo de tecnologías.

La Dirección Municipal de Educación de Media Luna, es un centro de trabajo que abarca varias instituciones educativas donde se utiliza esta tecnología, tales como: Laboratorios de Computación, salas de video conferencias, aulas, etc. A pesar de su avance informático, actualmente, los encargados de manipular el proceso para reportar la disponibilidad técnica de la tecnología educativa, cuentan con una base de datos realizada en Microsoft Excel, pero no cumple con los requisitos necesarios y su uso resulta engorroso ya sea para realizar una búsqueda, modificar algún dato o generar reportes; afectando así la calidad y rapidez del proceso.

Todo lo anteriormente planteado, conlleva a que se produzcan una serie de irregularidades entre las que se incluyen:

- Lentitud excesiva en búsqueda y recuperación de datos para la generación de reportes.
- Insuficiente control de los medios en proceso de bajas técnicas, lo que genera desactualización en la contabilidad.
- Demoras en el proceso de actualización de la disponibilidad técnica, de esta forma se imposibilita que los involucrados conozcan en tiempo real, el estado de los recursos informáticos con que cuentan las instituciones.

Por lo anteriormente planteado se dificulta la gestión de reportes de la disponibilidad técnica de tecnología educativa en la DME de Media Luna por lo que se identificó el siguiente **problema científico**: Insuficiencias en la gestión de los reportes automáticos sobre el estado real de la tecnología educativa, por parte de la Dirección Municipal de Educación en Media Luna.

En la presente investigación se propone como objetivo general: Identificar el modelo de proceso adecuado para lograr un desarrollo eficiente de un sistema informático que contribuya a perfeccionar el reporte automático del estado físico de la tecnología educativa en la Dirección Municipal de Educación de Media Luna.

Metodología

El desarrollo de una aplicación de software se lleva a cabo por medio de un proceso denominado Ciclo de Desarrollo de Software (SDLC, por sus siglas en inglés Software Development Life Cycle), el cual comprende un conjunto de actividades fundamentales que son comunes a todos los procesos de software [Sommerville & Galipienso, 2011], tal y como se explica en la siguiente tabla:

Tabla 1. Representación del denominado Ciclo de Desarrollo de Software. [Sommerville & Galipienso, 2011] y [Alshamrani & A. Bahattab, 2015]

Fases	Actividades
Análisis	Se define la especificación de los requerimientos funcionales y no funcionales esperados del producto de software, así como sus restricciones de operación. Estos requerimientos son recolectados, analizados y especificados en una labor por el equipo de desarrollo, generando una serie de artefactos (documentos, prototipos, arquitecturas beta) que documentan la información recopilada y acordada
Diseño	Tomando como base la especificación de requerimientos generada en la fase anterior, se establece una arquitectura de software, que implica la identificación y descripción de las abstracciones fundamentales del sistema (componentes e interfaces de conexión) y a nivel detallado, se especifica la estructura del producto por medio de diagramas lógicos, esquemas conceptuales de los datos, considerando las tecnologías apropiadas para el desarrollo del producto especificado.
Implementación	En esta etapa, los requerimientos estipulados en la primera fase se convierten en código siguiendo los lineamientos de diseño definidos en la etapa anterior y se acompaña de pruebas de unidades para verificar que cada unidad cumpla con su especificación
Integración y prueba del sistema	Las unidades del producto se integran y prueban como un sistema completo para asegurarse de que se cumplan los requerimientos de software. Además, se detectan inconsistencias de funcionamiento que se puedan producir de la fase anterior de codificación para su corrección y depuración antes de la liberación del producto
Mantenimiento	Una vez que el software ha sido liberado, el producto puede requerir modificaciones en el funcionamiento de los requerimientos cubiertos, mejoras o extensión de nuevas funcionalidades, corrección de errores y entrar en una etapa de refinamiento.

Según Solano-Fernández y Porrás-Alfaro (2020), este ciclo de desarrollo puede ser organizado por medio de modelos que establecen una propuesta y ordenamiento sobre la forma de llevar a cabo cada una de las etapas y actividades.

Iam Sommerville refiere que un modelo de proceso es una representación abstracta de un proceso de software. Estos modelos de procesos muy generales, se presentan desde

una perspectiva arquitectónica (se ve el marco de trabajo del proceso, pero no los detalles de actividades específicas); no son descripciones definitivas de los procesos del software. Más bien son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo del software. (Sommerville, 2005)

Por su parte Micky Jerzy plantea que un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software. Algunos ejemplos de estos tipos de modelos que se pueden producir son: (Micky Jerzy, 2016)

Como marco de referencia, el modelo tradicional o más básico se conoce como modelo en cascada (WM-Waterfall Model) y organiza las fases del SDLC en un esquema secuencial, donde cada una de las fases no puede ser realizada hasta que la previa haya sido concluida en su totalidad [Alshamrani & Bahattab, 2015].

Por su parte, el modelo iterativo e incremental (IID-Iterative and Incremental Model) combina elementos del modelo en cascada en una propuesta de iteraciones. Este modelo construye una implementación parcial de la aplicación; cada secuencia lineal o iteración produce un incremento funcional en el desarrollo del producto final [Alshamrani & Bahattab, 2015]. Este esquema permite la identificación y establecimiento de acciones para la mitigación de los mismos en cada iteración, así como favorece la valoración y aprovechamiento de las lecciones aprendidas durante el desarrollo de versiones que se van liberando del sistema. El aprendizaje proviene tanto del desarrollo como del uso del sistema, siempre que sea posible [Larman & V. R. Basili, 2003].

El modelo iterativo-incremental (MII) se ha constituido en una base para la generación de nuevas metodologías y prácticas en el desarrollo de software, y ha sido base para la evolución de nuevas propuestas de desarrollo en una era en la que el factor de cambio en las necesidades de información es cada vez más frecuente y juega un papel preponderante y constante, por lo que se requiere un alto grado de flexibilidad y capacidad de reacción ante las expectativas y necesidades que surgen de forma natural. (Figura 1)



Figura 2. Modelo de desarrollo iterativo e incremental.

Como parte de la revisión bibliográfica realizada en la presente investigación, se identificaron diversos tipos de modelos [Aguileta & O. S. Gómez, 2019], [Pressman, 2010], haciendo énfasis en los siguientes: modelo iterativo-incremental (MII) y modelo en cascada.

En el modelo incremental:

- El proyecto puede comenzar con un conjunto de requerimientos bien definidos, sin tener la totalidad de requerimientos
- Al ser repetitivo el proceso, se pueden crear nuevas versiones del producto en cada iteración.
- Permite un mejor manejo de los riesgos del proyecto.
- Surgen nuevos riesgos en el plan de gestión del proyecto por exceso de cambios.

En el modelo en cascada:

- Los riesgos son altos e inciertos.
- Deben estar listos todos los requerimientos de antemano.
- El progreso de cada etapa es difícil de medir mientras se está en el proceso de desarrollo.
- Permite la clasificación y priorización de tareas.

Teniendo en cuenta las particularidades del modelo iterativo incremental y las características de la investigación, en el presente trabajo el modelo de procesos que se utilizará, es el modelo incremental que es una unión de las mejores funcionalidades del modelo de cascada y del modelo de prototipos. A medida que se presenta un prototipo se produce un incremento, que es una iteración del proceso anterior, pero aplicando las experiencias aprendidas. Se tuvo en cuenta, además, los siguientes elementos.

Fases del modelo incremental:

El modelo de gestión incremental no es un modelo necesariamente rígido, es decir, que puede adaptarse a las características de cualquier tipo de proyecto, existen al menos 7 fases que debemos tener en cuenta a la hora de implementarlo [Pérez, 2016]:

1. **Requerimientos:** son los objetivos centrales y específicos que persigue el proyecto.
2. **Definición de las tareas y las iteraciones:** teniendo en cuenta lo que se busca, el siguiente paso es hacer una lista de tareas y agruparlas en las iteraciones que tendrá el proyecto. Esta agrupación no puede ser aleatoria. Cada una debe perseguir objetivos específicos que la definan como tal.
3. **Diseño de los incrementos:** establecidas las iteraciones, es preciso definir cuál será la evolución del producto en cada una de ellas. Cada iteración debe superar a la que le ha precedido. Esto es lo que se denomina incremento.
4. **Desarrollo del incremento:** posteriormente se realizan las tareas previstas y se desarrollan los incrementos establecidos en la etapa anterior.
5. **Validación de incrementos:** al término de cada iteración, los responsables de la gestión del proyecto deben dar por buenos los incrementos que cada una de ellas ha arrojado. Si no son los esperados o si ha habido algún retroceso, es necesario volver la vista atrás y buscar las causas de ello.
6. **Integración de incrementos:** una vez son validados, los incrementos dan forma a lo que se denomina línea incremental o evolución del proyecto en su conjunto. Cada incremento ha contribuido al resultado final.
7. **Entrega del producto:** cuando el producto en su conjunto ha sido validado y se confirma su correspondencia con los objetivos iniciales, se procede a su entrega final.

Conclusiones

Existen numerosos modelos de desarrollo de software basados en los requisitos y la dimensión de los proyectos que se desean desarrollar. Seleccionar el modelo a utilizar, permite asegurar la calidad de los procesos y los productos, una gestión más eficiente de los riesgos y un alto nivel de satisfacción por parte del cliente.

Referencias bibliográficas.

Alshamrani A. & Bahattab A, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model", International Journal of Computer Science Issues (IJCSI), vol. 12, (1), pp. 106, 2015. Available: <https://search.proquest.com/docview/1660801422>

Aguileta A. A & Gómez O. S., "Estudio de calidad y eficiencia de un enfoque de desarrollo software secuencial con programadores solos y en pareja", Ingeniare. Revista chilena de ingeniería, vol. 27, (2), pp. 304-318, 2019. Available: https://scielo.conicyt.cl/scielo.php?pid=S0718-33052019000200304&script=sci_arttext

Garcés Suárez, Alcívar Fajardo "LAS TECNOLOGÍAS DE LA INFORMACIÓN EN EL CAMBIO DE LA EDUCACION SUPERIOR EN EL SIGLO XXI: REFLEXIONES PARA LA PRÁCTICA" Universidad y Sociedad vol.8 no.4 Cienfuegos sep.-dic. 2016 Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000400023

Larman C. & V. R. Basili, "Iterative and incremental developments. a brief history", Mc, vol. 36, (6), pp. 47-56, 2003. Available: <https://ieeexplore.ieee.org/document/1204375> DOI: 10.1109/MC.2003.1204375.

Pérez A. "Características y fases del modelo incremental" OBS Business School. Available: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental> 2016

Pressman R. S., "Software Engineering: A Practitioner's Approach", 7th ed., New York: Editorial McGraw-Hill, 2010.

Solano-Fernández E.y Porrás-Alfaro D. "El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón_RA" Tecnología en Marcha. Edición especial. Escuela de Arquitectura y Urbanismo, diciembre 2020. Pág 165-177. <https://doi.org/10.18845/tm.v33i8.5518>

Sommerville I. & M. Alfonso-Galipienso, "Ingeniería de Software", 9th ed., México, Ciudad de México: Pearson Educación, S.A., 2011.

Vidal Ledo, M.; del Pozo Cruz R. "Tecnología educativa, medios y recursos de enseñanza-aprendizaje" Educ Med Super v.22 n.4 Ciudad de la Habana oct.-dic. 2008 Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21412008000400010