



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5

**PROCESO DE DESARROLLO PARA LA
CONSTRUCCIÓN DE APLICACIONES ENRIQUECIDAS
DE INTERNET QUE INCLUYAN ESCENARIOS
TRIDIMENSIONALES INTERACTIVOS.**

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS

Autor: Maybé Gil Cabrera.

Tutor: Ing. Gilberto Cao Tarrero.

Co-Tutor: Msc. Yaimí Trujillo Casañola

Ciudad de la Habana

abril de 2011

Declaración de autoría

Yo, Maybé Gil Cabrera, declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmo la presente declaración de autoría en Ciudad de la Habana a los _ días del mes de _ del año _.

Nombre y Apellidos del Autor.

Datos de Contacto

Ing. Gilberto Cao Tarrero

Centro de Tecnologías para la Formación (FORTES), Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 1\2, Boyeros, Ciudad de La Habana, Cuba. gcao@uci.cu.

MsC.Yaimí Trujillo Casañola

Profesor Asistente, Centro de Calidad para Soluciones Tecnológicas (CALISOFT), Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 1\2, Boyeros, Ciudad de La Habana, Cuba. yaimi@uci.cu.

Agradecimientos

A mis padres que jamás dudaron de mi y son mi vida,
A mi hermana por ser mi ídolo y ponerme metas altas,
A mis abuelos por su experiencia y cuidado,
A mi familia por exigirme el estudio constante,
A mi preciado tesoro por minimizar los problemas y por
tanto amor diario,
A mis amigos por obligarme a mantener una sonrisa,
A los que influyeron en mi vida, sembrando el sueño de
graduarme,
A todos, mil gracias y un beso...

Resumen

Durante los últimos años en Cuba se ha investigado en busca de implantar estrategias que ayuden a fortalecer la industria del software y la calidad de sus productos. La creación de la Universidad de las Ciencias Informáticas es una de ellas, en la cual se ejerce el concepto estudio-trabajo, convirtiendo la universidad en un área productiva.

Desconocimiento en los flujos de trabajo, los roles, sus responsabilidades y en general la ausencia de un proceso de desarrollo establecido para la producción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos influyen en la obtención de un producto sin la adecuada calidad.

De ahí que el presente trabajo defina como problema científico: no se cuenta con un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos, adaptado a las características y necesidades de la actividad productiva en la UCI. Con el objetivo: describir los elementos necesarios de un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

La aplicación del proceso de desarrollo de software contribuye a la industrialización, pues define el flujo de procesos, los roles y las responsabilidades en el proyecto, mejora la planificación e identifica al desarrollador con las características de las aplicaciones que produce.

Palabras Clave: *proceso de desarrollo, Aplicaciones Enriquecidas de Internet, productividad, calidad.*

Índice de contenidos

Introducción	1
1. Fundamentación Teórica	6
1.1. Aplicaciones Enriquecidas de Internet	6
1.2. Proceso de desarrollo	8
1.3. Características de los procesos de desarrollo	9
1.4. Modelo de Proceso de Desarrollo	10
1.4.1. Modelo Lineal Secuencial	11
1.4.2. Modelo de Construcción de Prototipos	12
1.4.3. Modelo de Desarrollo Rápido de Aplicaciones	13
1.4.4. Modelo evolutivo del proceso de software	13
1.4.5. Modelo de desarrollo basado en componentes	15
1.4.6. Consideraciones sobre Modelos de Proceso de Software	16
1.5. Metodologías de desarrollo de software	17
1.5.1. Metodologías Ágiles	18
1.5.2. Consideraciones sobre las metodologías de desarrollo de software	23
1.6. Calidad de Software	24
1.6.1. Modelos y estándares de calidad de software	25
1.7. Bases tecnológicas	26
1.7.1. Ambiente Integrado de Desarrollo	28
1.7.2. Herramientas de programación: Eclipse	28
1.7.3. Plugins: Adobe Flash Builder y Subclipse	29
1.7.4. Framework:Flex SDK 4	29
1.7.5. Lenguajes de programación: ActionScript 3.0	30
1.7.6. Sistemas de gestión de base de datos: MySQL y PostgreSQL	31
1.7.7. Herramientas de modelado 3D: Blender	32
1.7.8. Herramientas de modelado: Visual Paradigm y Rational Rose	33
1.7.9. Servidores de Aplicaciones:Apache	35
1.7.10. Consideraciones sobre las herramientas	35
1.8. Conclusiones parciales	36
2. Solución técnica	38
2.1. Métodos, procedimientos y técnicas utilizadas	38
2.1.1. Entrevista # 1	39
2.1.2. Entrevista # 2	40

2.1.3. Método para la evaluación del proceso de desarrollo propuesto. Consulta a expertos	41
2.2. Método para la evaluación técnica del proceso de desarrollo propuesto	42
2.3. Entorno de implantación: Universidad de las Ciencias Informáticas .	47
2.4. Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.	49
2.4.1. Estudio Preliminar	51
2.4.2. Requerimientos	52
2.4.3. Análisis y Diseño	53
2.4.4. Implementación	54
2.4.5. Pruebas	55
2.4.6. Despliegue	57
2.4.7. Soporte	58
2.5. Conclusiones Parciales	59
3. Evaluación de la propuesta	60
3.1. Introducción	60
3.2. Análisis de los resultados de la aplicación de las encuestas y las entre- vistas	61
3.3. Análisis de los resultados de evaluación técnica del proceso de desarrollo	68
3.4. Elementos de implantación del proceso de desarrollo	70
3.4.1. Prácticas Generales para CMMI nivel 2	70
3.5. Conclusiones Parciales	73
Conclusiones	74
Recomendaciones	75
Bibliografía	76

Introducción

En la actualidad, las aplicaciones web se han transformado en Aplicaciones Enriquecidas de Internet (RIA, por sus siglas en inglés). Este término fue utilizado por primera vez en el año 2005 por Dale Dougherty y O'Reilly en una conferencia sobre el renacimiento y evolución de la Web [31]. Varias empresas de punta como Macromedia, Magic Software, Sun y Microsoft aprovechan en sus aplicaciones el enriquecimiento de la experiencia visual, mejoras en la conectividad y despliegue instantáneo de la información, características básicas de las RIA.[38]

La continua demanda de las mencionadas RIA exige que su calidad no solo se mida en la usabilidad, flexibilidad y la mantenibilidad que puedan tener, sino que además, interesa una disminución de costo y tiempo durante su construcción, utilizando para esto técnicas y metodologías de desarrollo que permiten una correcta gestión de proyecto, hasta alcanzar un nivel calificado de madurez del software.[18]

Cuba está experimentando un crecimiento considerable en el desarrollo de aplicaciones web, y con esto un incremento en el interés por garantizar y automatizar su proceso de desarrollo.[14]

Teniendo en cuenta la tendencia mundial de migrar la mayoría de las aplicaciones de escritorio a aplicaciones en la web, el creciente atractivo de escenarios tridimensionales en busca del realismo visual y el nivel de la competencia internacional en

el desarrollo de las RIA [6] es de primordial interés para la industria de software cubano profesionalizar la creación de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos, no solo por los beneficios que trae desde el punto de vista de desarrollo de sistemas para el uso interno, sino también con el objetivo de introducirse en el mercado.

La Universidad de las Ciencias Informáticas (UCI) forma parte de la industria cubana de software. En dicha institución se están desarrollando algunas RIA que incluyan escenarios tridimensionales interactivos. La primera tarea en este sentido fue realizada por el equipo del proyecto Visualización Web de la Facultad 5 el cual logró productos finales como "Paseo Virtual del Salón de Productos Informáticos Cubanos", presentado en la Feria de Soluciones Informáticas 2010 (FESI), donde alcanzó el galardón de producto más popular.

El Centro de Tecnologías para la Formación (FORTES), ubicado en la Facultad 4, desarrolla la Colección Multisaber y El Navegante en la web. Sin embargo, no existe una planificación concreta y eficaz ajustada a las características del proyecto que permita la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos, la cual garantice una organización del equipo de desarrollo y el aumento de la calidad del producto. Basándose en lo anterior surge la interrogante:

¿Cuáles son los pasos que guían la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos?

El **objeto de estudio** se centra en los Procesos de Ingeniería de Software.

Para dar solución al problema antes mencionado se propone como **objetivo general**: Describir los elementos necesarios de un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Los objetivos específicos planteados son:

1. Analizar el estado del arte relacionado con las metodologías y los modelos de proceso de desarrollo asociados al desarrollo de soluciones RIA.
2. Elaborar el Libro de Proceso de Desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.
3. Validar las características de un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos guiados por las Prácticas Generales establecidas por CMMI para el nivel 2.

En esta investigación, teniendo en cuenta la relación entre el problema, el objeto de estudio y el objetivo de la investigación, se definen como **campo de acción** las Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Donde las tareas de investigación propuestas se definen como:

1. Caracterización de las Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.
2. Análisis de proceso de desarrollo en la ingeniería de software.
3. Descripción de los Modelos de Procesos de Desarrollo.
4. Caracterización y selección de las Metodologías Ágiles.
5. Elaboración del Libro de Proceso de Desarrollo para guiar la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos

6. Validación de las características del proceso de desarrollo de software para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Se utiliza como **Estrategias de Investigación**, la Investigación Descriptiva, la cual tiene como objetivo describir el fenómeno y reflejar lo esencial y más significativo del mismo; y la Investigación Experimental, que se basa en determinar las causas que producen el fenómeno en estudio, es decir, se determinan las razones por las cuales no se tiene un proceso de desarrollo ajustado a la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos y partiendo de ello, se propone una solución.

Dentro de los **Métodos Científicos de la Investigación** a emplear, se eligieron los Métodos Teóricos en particular el método histórico-lógico y analítico-sintético en los cuales se analiza la trayectoria completa del fenómeno en estudio y su condicionamiento a los diferentes períodos de la historia. Este método se usó específicamente a la hora de caracterizar los principales términos a dominar en el marco teórico.

Además, se utilizaron métodos empíricos, específicamente la entrevista (para obtener información de cómo es el flujo de trabajo para la creación de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos) y la consulta a expertos en vistas de validar la propuesta. Se determinó que los desarrolladores y personas vinculadas a la producción de software fueran entrevistadas para conocer la efectividad de Libro de Procesos de Desarrollo y así poder validar la calidad del proceso de desarrollo propuesto.

El documento presenta una estructura por capítulos como se muestra a continuación:

CAPÍTULO 1. Fundamentación Teórica: Se describen los conceptos fundamentales asociados al dominio del problema, se expone el estado del arte del tema tratado, tanto a nivel nacional como internacional, se hace un análisis crítico entre las soluciones ya existentes. Explica y justifica, además, las tendencias actuales al desarrollar estas Aplicaciones Enriquecidas de Internet y las tecnologías y herramientas en las que se apoya la propuesta presentada.

CAPÍTULO 2. Personalización del proceso de desarrollo: se describen los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación. Se presenta la propuesta de modificación del proceso de desarrollo para la construcción de las Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos haciendo referencia en algunos casos a la consulta del Libro de Proceso de Desarrollo. Se identifican los elementos fundamentales de la propuesta y se brindan las partes que componen el proceso ajustado. Se establecen las fases, los roles involucrados, las actividades y los artefactos generados.

CAPÍTULO 3. Evaluación de la propuesta: Se describen las principales características de un proceso de desarrollo, permitiendo mediante consultas a expertos comprobar la existencia de estas en el proceso de desarrollo propuesto. Analizando los resultados de las encuestas realizadas y haciéndole tratamiento estadístico, se llega a conclusiones. Se recomienda la forma de implantación del proceso para lograr la institucionalización de este, siguiendo las prácticas generales de CMMI nivel 2.

Capítulo 1

Fundamentación Teórica

Introducción

En el presente capítulo se brinda una panorámica de los aspectos relacionados con las Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos y los conceptos que se hacen necesarios para su estudio. Proceso de Desarrollo, Metodologías Ágiles y Modelo de Procesos de Desarrollo constituyen los principales conceptos a dominar durante la investigación y para el desarrollo de la propuesta de solución.

1.1. Aplicaciones Enriquecidas de Internet

En el siglo XXI la tendencia a la utilización de las aplicaciones web ha aumentado considerablemente. A causa de las facilidades que nos brinda y sus beneficios, se opta, cada vez más, por la utilización de este tipo de aplicaciones en lugar de las tradicionales.

Gracias a la intensa campaña de comunicación realizada, el concepto de Aplica-

ciones Enriquecidas de Internet ha adquirido gran popularidad al punto de que muchas empresas lo están usando como moda, sin lograr una comprensión plena de este término.

Sin embargo algunas empresas de punta como Adobe tienen claro que las RIA ofrecen una experiencia sofisticada y atractiva que mejora la satisfacción del usuario y aumenta su productividad.[1]

La IEEE Computer Society en el artículo “Aplicaciones Enriquecidas de Internet” aclara que el término RIA se refiere a una familia heterogénea de soluciones, caracterizadas por la meta común de agregar nuevas capacidades a las aplicaciones tradicionales. Combina además la arquitectura de distribución ligera con la interactividad de la interfaz de las aplicaciones de escritorio, y la combinación resultante mejora todos los elementos de la aplicación web (datos, lógica del negocio, comunicación, y presentación).[11]

Es indudable las ventajas que brinda el desarrollo de las RIA, grandes y prósperas empresas obtuvieron su sello gracias a que sus productos brindan una experiencia visual mucho mayor que las de su competencia, ejemplo de esto es Google, eBay y Amazon las cuales han centrado su negocio en las RIA.[11]

Adobe aclara que las RIA ofrecen a las empresas un modo económico y de probada eficiencia para proporcionar modernas aplicaciones con las ventajas de los negocios reales. Además, aumenta la fidelidad de clientes aprovechando el 99 % de los ordenadores conectados a Internet y a 300 millones de dispositivos. [1]

1.2. Proceso de desarrollo

El proceso de desarrollo de software según Jacob específicamente define: “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo. En general es, “el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software”[19]. En la figura 1.1 se representa los elementos que conforman un proceso de desarrollo de software.

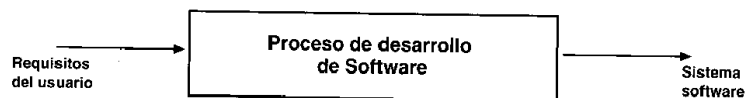


Figura 1.1: Proceso de desarrollo de software según Jacobson.

Jacob concibe el proceso de desarrollo de software como la transformación de los requisitos del usuario en un producto final, el cual se realiza desarrollando una serie de actividades ejecutadas por los desarrolladores durante las etapas o fases del proceso. El proceso debe contar con un flujo de trabajo, ordenando el conjunto de actividades y asignándolas a un determinado rol teniendo en cuenta los artefactos de entrada y salida, los cuales deben seguir con la norma de calidad establecida.

Por su parte Pressman define el proceso de desarrollo de software como: “. . . un marco de trabajo de las tareas que se requieren para construir software de alta calidad”. [33]

En este caso Pressman especifica que el proceso de desarrollo está conformado por un marco de trabajo común en el cual se definen las actividades a realizar formadas por un conjunto de tareas que pueden estar formadas, a su vez, por varias tareas. Este conjunto de tareas responden a hitos del proyecto y deben generar artefactos cumpliendo siempre con los puntos de garantía de calidad. Las actividades de protección son independientes de las demás actividades y se ejecutan

durante todo el ciclo de vida del software. En la figura 1.2 puede aclararse gráficamente lo antes descrito.

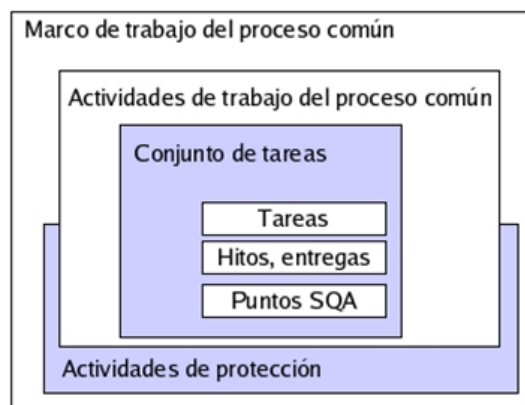


Figura 1.2: Proceso de desarrollo de software según Pressman.

Es importante aclarar que ningún proceso de desarrollo está diseñado para uso universal debido a que tienen lugar en contextos diferentes, se desarrollan distintos tipos de sistemas y se ajustan a restricciones del negocio variables. Por tanto un proceso de desarrollo debe ser adaptable y configurable para cumplir con las necesidades del proyecto.

Crear un proceso de desarrollo sólido no garantiza la calidad total del software pero sí nos guía para la obtención de este. Como plantea Davis si un proceso es débil, el producto final es el que sufre [8], dándole una marcada importancia a la correcta definición de las normas de desarrollo a seguir por los desarrolladores con el objetivo de lograr un software de calidad.

1.3. Características de los procesos de desarrollo

Los procesos de desarrollo tienen atributos o características que ayudan a mantener o mejorar la calidad del proceso y su efectividad para el desarrollo de software.[39]

Estas características son:

- **Comprensión:** El proceso se define completamente y se comprende fácilmente su definición.
- **Visibilidad:** Las actividades terminan en resultados claros, así el proceso es visible externamente.
- **Apoyo:** Las actividades del proceso se apoyan en las herramientas CASE.
- **Aceptación:** El proceso definido es aceptable y utilizable por los ingenieros responsables de producir software.
- **Fiabilidad:** El proceso está diseñado para que los errores del proceso se eviten o identifiquen antes de que se conviertan en errores de producto.
- **Robustez:** El proceso continua a pesar de los problemas inesperados.
- **Mantenibilidad:** El proceso evoluciona para reflejar los requerimientos cambiantes o las mejoras identificadas.
- **Rapidez:** Agilidad en la construcción de un sistema a partir de las especificaciones dadas.

Resulta imposible realizar una mejora de procesos que optimice todos los atributos de forma simultánea, sin embargo, un proceso de desarrollo efectivo debe cumplir cada una de las características por encima del 50 %.

1.4. Modelo de Proceso de Desarrollo

Un modelo de proceso es una descripción simplificada de un proceso de software, el cual presenta una visión de ese proceso. Pueden incluir actividades y los rol involucrados durante el ciclo de vida del software.[39]

En la bibliografía consultada se plantean varios tipos de modelos de procesos los cuales no son de cumplimiento estricto para ningún proyecto, en cada desarrollo de software se utilizan los que más se ajusten a las características propias. La mayoría de los autores referencian a Pressman, el cual define los modelos de procesos de desarrollo como una estrategia que acompaña al proceso, métodos y capas de herramientas durante las fases genéricas.[33]

En el siguiente epígrafe se nombran y caracterizan los modelos de procesos de software más importante y comúnmente usados según la investigación realizada.

1.4.1. Modelo Lineal Secuencial

Presentado inicialmente por Royce (1970), el modelo lineal secuencial también conocido como modelo en cascada por su enfoque sistemático y secuencial o ciclo de vida básico[33] se caracteriza por un refinamiento del modelo de etapas comenzando en un nivel de sistemas y progresando con el análisis, diseño, codificación, pruebas y mantenimiento. En el Anexo 1 se muestran las actividades del modelo lineal secuencial.

Sus principales características son:

- Cada fase comienza cuando se ha terminado la fase anterior.
- Para pasar de una fase a otra es necesario que se hayan cumplido todos los objetivos de la etapa previa.
- Al final de cada fase el personal técnico y los usuarios pueden revisar el progreso del proyecto.

El modelo lineal secuencial es un modelo sencillo que se basa en seguir pasos intuitivos a la hora de desarrollar el software. Además, ayuda a que los tiempos de

entrega y costos del proyecto no sobrepasen la planificación inicial y disminuye los riesgos para desarrollos bien comprendidos con tecnología conocida.[41]

Sin embargo, consta de poca flexibilidad a la hora de dividir el proyecto en etapas, lo que dificulta responder a los cambios en los requerimientos del cliente. Este modelo se tarda mucho tiempo en pasar por todo el ciclo de vida del software y las revisiones en proyectos muy complejos se hacen verdaderamente difíciles. Además, solo se obtendrá un resultado en la etapa final del proyecto, lo que trae consigo que no se pueda detectar un error hasta que el programa este funcionando, y esto puede ser desastroso.[41]

1.4.2. Modelo de Construcción de Prototipos

El modelo de construcción de prototipos consiste en crear una implementación del sistema con un diseño rápido, que ayude a los clientes, usuarios o representantes de estos a elegir los requisitos del sistema mientras experimentan con el prototipo. El equipo de desarrollo se retroalimenta con la experiencia de los usuarios y toman los requerimientos para el desarrollo del sistema real. En el Anexo 2 se muestra el paradigma de construcción de prototipos.[33]

El paradigma de construcción de prototipos puede ser efectivo, siempre y cuando tengamos en cuenta que el cliente y el desarrollador debe establecer reglas al inicio para que el prototipo que se construya sirva como mecanismo de definición de requisitos. Sin embargo, la gestión de desarrollo es lenta porque hasta que el usuario no esté de acuerdo con el prototipo el producto real no se desarrolla, imposibilitando saber el tiempo de desarrollo. En general, es muy difícil y complejo realizarlo.[37]

1.4.3. Modelo de Desarrollo Rápido de Aplicaciones

El modelo de desarrollo rápido de aplicaciones (DRA, por sus siglas en inglés) surge a partir de la propuesta del modelo lineal secuencial el cual tiene un ciclo de desarrollo extremadamente corto utilizando una construcción basada en componentes.[33]

Cuando se conocen bien los requisitos, el proceso DRA permite la construcción de un sistema completamente funcional dentro de períodos cortos de tiempo.[24]

Este modelo permite que varios equipos se desarrollen en paralelo y la integración del producto al final tenga varios ciclos de vida evolutivo o en espiral, gráficamente se puede ver en el Anexo 3.

Como todos los modelos, DRA tiene sus inconvenientes. Hay que tener en cuenta que en proyectos grandes se requiere recursos humanos suficientes para la creación del número correcto de equipos DRA; los clientes y desarrolladores deben estar comprometidos en las rápidas actividades para completar un sistema en el corto período de tiempo y sobre todo hay que tener claro que no es un modelo adecuado para proyectos con tecnología nueva o cuando el software requiere un alto nivel de interoperatividad con programas de computadoras que ya existen.[5]

1.4.4. Modelo evolutivo del proceso de software

Los modelos evolutivos tienen dentro de sus principales características que son iterativos lo que le permite gestionar con efectividad la naturaleza evolutiva del software, logrando adaptarse a los cambios continuos de requisitos y a las rigurosas fechas de entrega que llegan a ser poco realistas.[33]

Los modelos evolutivos tiene tres variantes: incrementales, en espiral y desarrollo concurrente.[26]

1. Modelo incremental

Utilizando una combinación de algunos elementos del modelo lineal secuencial y la filosofía interactiva de la construcción de prototipos, este modelo produce un "incremento" del software.[26]

El modelo incremental se centra en la entrega de un producto operacional en cada iteración. Al inicio cada entrega son versiones incompletas del producto final pero le facilita al usuario una plataforma para evaluar la funcionalidad que precisa. Además, este modelo se hace particularmente necesario cuando no se encuentra en el equipo de trabajo la cantidad de personas necesarias para cumplir con la implementación completa del producto en el tiempo establecido por el proyecto, más bien da la posibilidad de implementar las primeras versiones con menos personal.[33]

2. Modelo espiral

Clasificado dentro de los procesos de software evolutivos, el modelo espiral, propuesto inicialmente por Boehm, conjuga la naturaleza iterativa de construcción de prototipos con el modelo lineal secuencial, más específicamente con la sistematicidad y el control que se propone en el modelo lineal.[3]

El modelo espiral consiste en definir un número de actividades de marco de trabajo o regiones de tareas (generalmente existen entre tres y seis regiones de tareas), las cuales están compuestas por un conjunto de tareas adaptadas a las características del proyecto que se va a desarrollar, entonces el equipo de ingeniería de software comienza a desarrollar estas actividades desde el centro hacia afuera a favor de las agujas del reloj. Cada ciclo comienza identificando los objetivos de la región correspondiente, las alternativas y las

restricciones. Se plantea el próximo prototipo después de evaluar las alternativas respecto a los objetivos y a las restricciones, lo cual resuelve los riesgos definidos permitiendo seguir el ciclo en cascada. Al finalizar el ciclo se revisa el ciclo anterior y el plan para el siguiente.[33]

Pero como es de suponer, ningún modelo es perfecto. En situaciones de contrato puede resultar difícil convencer al cliente que el modelo evolutivo puede ser controlable y contar con la continua participación de este durante el proceso puede volverse una exigencia que el cliente no está dispuesto a cumplir. Además necesita contar con la experiencia en la identificación de riesgos para disminuir los costos.[33]

1.4.5. Modelo de desarrollo basado en componentes

Este enfoque se basa en la reutilización de componentes de software. Generalmente esto sucede cuando los desarrolladores ven alguna similitud entre los diseños o códigos requeridos y los elaborados para otro sistema.[39]

El modelo basado en componentes incluye muchas características del modelo espiral, sobre todo es por naturaleza evolutivo y exige para su desarrollo un enfoque iterativo.[30]

Con la reutilización podemos obtener varias ventajas en la reducción de la cantidad de software a desarrollar, además de los costos y riesgos, lo cual agilizaría la entrega del producto. Sin embargo, igual que todos los modelos, presenta inconvenientes. El principal inconveniente sería que la reutilización puede conllevar a que un sistema no cumpla con las necesidades reales de los usuarios definidas en los requerimientos.

1.4.6. Consideraciones sobre Modelos de Proceso de Software

Después de analizar los distintos modelos de procesos y ver las características de cada uno, se realizó una tabla comparativa basada en las ventajas y desventajas de cada uno, para ello ver Anexo 4.

Basados en los elementos significativos y teniendo en cuenta como características del proyecto que no tiene más de 30 personas en el equipo de desarrollo, que el tiempo de producción no debe sobrepasar los 4 meses, que la línea de producción en que van a ser desarrollados los productos utiliza una arquitectura cliente/servidor y los requisitos pueden ser modelados en prototipos, se seleccionaron dos modelos para combinarlos y utilizarlos en el proyecto, los cuales son modelo incremental y el modelo de desarrollo basado en componentes.

La unión de estos ayuda a utilizar lo mejor de cada uno y ponerlo en función del éxito del producto. El modelo incremental brinda una continua posibilidad de ajustar los requerimientos mientras el cliente interactúa con versiones del producto, disminuyendo los riesgos de fracaso del producto final. Además, apoyado por el modelo de desarrollo basado en componentes se logra asegurar una disminución en los costos y riesgos, influyendo en la eficiencia de los tiempos de entrega y una reducción de la cantidad de software a desarrollar.

Es importante señalar que se necesita un gran conocimiento sobre las necesidades del cliente e ir chequeándolas en cada iteración para no correr el riesgo de que el sistema no cumpla con los requisitos reales definidos por el cliente.

1.5. Metodologías de desarrollo de software

Una metodología es un sistema de prácticas, técnicas, procedimientos y normas utilizado por quienes trabajan en una disciplina.[40]

Según la norma 1074 de la IEEE toda metodología de desarrollo de software debe incluir la forma en que se va a realizar la captura de requisitos, el diseño, la implementación y prueba.[25]

En la disciplina de desarrollo de software se han agrupado las metodologías en dos grandes grupos: Metodologías Tradicionales y Metodologías Ágiles.

Como es de suponer cada una tiene sus características distintivas que la hace más conveniente de usar o no, de acuerdo con las especificaciones del proyecto que se lleva a cabo.

En el artículo Metodologías tradicionales vs Metodologías Ágiles, Roberth G. Figueroa [9] realiza una comparación para ayudarnos a identificar sus principales diferencias y poder elegir la metodología a utilizar durante el ciclo de vida del software.(Ver Anexo 5).

Basándose en esta comparación se eligen las metodologías ágiles para que guíen el proceso de desarrollo propuesto, el cual necesita:

- Aligerarse en cuanto a documentación.
- Responder a los cambios.
- Tener un equipo de desarrollo pequeño.

1.5.1. Metodologías Ágiles

Las metodologías ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.[13]

Cada metodología ágil responde al Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto de software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Entre las principales metodologías ágiles se destacan eXtreme Programming (XP), SCRUM y el Proceso Unificado Ágil (AUP por sus siglas en inglés).

1. **XP**

Metodología ágil formulada por Kent Beck, el cual la definió como un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido a la hora de desarrollar software.[2]

Basada en una serie de reglas y principios, XP le da prioridad a las tareas que dan resultados directos y reducen la burocracia tanto como sea posible.[10]

Los principios definidos por Beck son:

- **El juego de planeamiento:** Determinar con rapidez el alcance del próximo *release* combinando prioridades del negocio con estimaciones técnicas. Se va actualizando el plan.
- **Pequeños Releases:** Producto muy simple al inicio e ir liberando versiones en ciclos muy cortos.
- **Metáfora:** Guiar el proceso de desarrollo con una historia simple de cómo funciona el sistema.
- **Diseño simple:** Sistema diseñado lo más simple que se pueda. Cuando se detecta complejidad extra se elimina.
- **Testing:** Los programadores hacen pruebas unitarias continuamente para garantizar que el sistema funcione. Los clientes realizan pruebas de funcionalidad.
- **Refactoring:** Los programadores hacen reestructuración del sistema para eliminar duplicados, mejorar la comunicación, simplificar y añadir flexibilidad.
- **Programación de a pares:** Se programa con dos programadores por máquina.
- **Propiedad colectiva del código:** Cualquiera puede cambiar código, en cualquier parte del sistema, en cualquier momento.
- **Integración continua:** Integrar y hacer compilación del sistema cada vez que se termine una tarea.
- **Semanas de 40 horas:** Trabajar no más de 40 horas semanales.
- **Cliente en el lugar de desarrollo:** Incluir al cliente en el equipo de trabajo, disponible a tiempo completo para responder preguntas.
- **Estándares de codificación:** Escribir el código de acuerdo con las reglas establecidas.

Como se puede observar contribuyen a maximizar la comunicación entre las personas, permitiendo de esa forma una mayor transferencia de conocimiento entre los desarrolladores y el cliente, quien también es parte del equipo.

Sin embargo, coincidiendo con Larman, muchas veces puede llegar a ser una fantasía que el cliente esté a tiempo completo en un sitio y además hay que contar con la resistencia de muchos programadores a trabajar en dúo.[20]. Otro inconveniente es que no se menciona la arquitectura en XP y por tanto hay falta de diseño arquitectónico.

2. SCRUM

En el año 1995 Ken Schwaber, Jeff Sutherland y Ken Schwaber formalizan el proceso conocido como Scrum, el cual es un método iterativo e incremental que enfatiza prácticas y valores de administración de proyecto por encima de las demás disciplinas de desarrollo.[36]

Su proceso de desarrollo se inicia definiendo los requisitos funcionales y no funcionales con los que debe cumplir el sistema, los cuales son especificados durante reuniones de planteamiento con los stakeholders. Seguidamente se establecen las iteraciones o sprint por las que irá evolucionando el producto. Cada sprint genera un subconjunto del producto final con los requerimientos a cumplir en la próxima iteración. Se recomienda que un sprint no debe durar más de 30 días.[36]

El encargado de guiar y auditar el proceso es el Scrum Master (un equivalente al Líder del proyecto) convocando a reuniones diarias de no más de 15 minutos para obtener retroalimentación del equipo de desarrollo. Al finalizar cada sprint se realiza una revisión para evaluar los artefactos construidos y comenzar la planificación del próximo sprint.[36]

En general las ventajas de esta metodología se resumen en:

- Existe una colaboración estrecha con el cliente.
- Respuesta al cambio.
- Desarrollo incremental con entregas funcionales frecuentes.
- Simplicidad.

Sin embargo, ninguna metodología escapa de posibles desventajas que hay que valorar a la hora de elegir una específica que guíe el proceso de desarrollo de nuestro producto. En este caso hay que tener bien en cuenta que la definición de los requisitos en cada sprint debe ser inviolable y bien detallada. Además, al igual que XP no tiene arquitectura definida.

3. AUP

El AUP es un acercamiento al desarrollo de software basado en el Proceso Unificado del Rational (RUP) de IBM, teniendo en cuenta disciplinas y entregables incrementales con el tiempo.[15]

Al igual que el RUP tiene cuatro fases: Inicio, Elaboración, Construcción y Transición pero las disciplinas están más resumidas para poder cumplir con el manifiesto ágil mencionado anteriormente, quedando al final definidos como: Modelado, Implementación, Prueba, Despliegue, Administración de configuración, Administración de proyecto y Ambiente.[15]

Estas disciplinas son ejecutadas de forma iterativa, en las cuales se define las actividades que el equipo de desarrollo va a realizar para construir, validar y liberar el producto.[15]

Según Hirsch aplicando la metodología AUP se llegan a eliminar gran cantidad de artefactos que no son realmente necesarios y entorpecen la fluidez en el desarrollo, incluso hacer una modificación en la estructura de los documentos. Además, las actividades no tienen que ser tan detalladas cuando

se cuenta con un equipo de desarrollo experimentado y en caso de que alguno tenga duda con la actividad que tiene que realizar puede dirigirse a la documentación sobre esa actividad en específico.

La asignación de roles puede verse como una forma de verificar si el equipo de desarrollo tiene todas las habilidades necesarias, y no necesariamente asignarle a cada persona un rol.

Es importante que la planeación del proyecto esté bien definida, puede establecerse dos niveles como en RUP:

- En primer nivel (el más básico) se definen las fechas de inicio y fin de las fases e iteraciones, además de los objetivos de cada iteración.
- En el segundo nivel ya sería detallar el plan por cada iteración definida, es importante tener en cuenta que debe ser creado antes de comenzar con la iteración y puede ser modificado durante esta. Las iteraciones se basan en los resultados a obtener (código, documentación, entre otros.), las cuales tendrán un responsable encargado de velar por su cumplimiento.

Se mantienen las cuatro fases definidas por RUP con sus hitos, pero las iteraciones se recomienda que sean alrededor de 4 semanas con una lista detallada de sus objetivos. No hay que esperar a que se termine la iteración para hacer pruebas de integración del sistema, es más, se recomienda hacerlo diariamente e incluir al usuario en estas pruebas para que revise las funcionalidades que él desea. Así obtenemos una retroalimentación del cliente, si sus cambios son mínimos se pueden incluir en esa misma iteración sino es recomendable posponerla para la próxima ya que necesita un replaneamiento de toda la iteración.

Para el control de proyecto tiene establecido que se realicen reuniones semanales en las que los desarrolladores estiman el tiempo que les queda de

trabajo y los objetivos a cumplir. En caso de que no se cumplieran todos los objetivos y el tiempo de la iteración se agotase, es preferible hacer la entrega con menos funcionalidades que aplazar la iteración.[15]

1.5.2. Consideraciones sobre las metodologías de desarrollo de software

Las metodologías en su conjunto no son fáciles de comparar entre sí con un reducido conjunto de criterios. Hay algunas que son más específicas y están más detalladas en cuanto a todo su proceso como son AUP y XP, sin embargo Scrum es más general y deja mucho más a la imaginación de quien la vaya a aplicar.

Las metodologías analizadas tienen en común que son incrementales (pequeñas entregas en cortos períodos de tiempo), son cooperativas (los desarrolladores y los clientes trabajan muy estrechamente), directa (fácil de aprender) y adaptativas (capaz de incorporar cambios). A pesar de esto tienen una gran diferencia basada en el diseño arquitectónico, siendo AUP la única que está centrada en la arquitectura y guiada por casos de uso, lo cual coinciden con la línea de producción utilizada.

Debido a que el software que se pretende desarrollar es de ciclos cortos y con equipo de desarrollo pequeño se escoge como metodología que guía el proceso de desarrollo a AUP, debido a que su base (RUP) tiene bien establecido los roles, artefactos, actividades y flujos de trabajo. Es la metodología más detallada y completa para cumplir con los Lineamientos de Calidad y el Expediente de Proyecto establecido de manera obligatoria por la Dirección de Calidad en la UCI.

1.6. Calidad de Software

En la actualidad los expertos de las grandes empresas industrializadas reconocen que la alta calidad del producto reduce los costos y proporciona una mejora general. Sin embargo, cuando se desarrolla un proyecto de software no basta con conocer que la calidad de software es importante sino que hay que definir para la empresa el término de “calidad de software” y crear un conjunto de actividades que permitan garantizar una calidad en el producto. Después que se ejecuten estas actividades podemos utilizar las métricas para el perfeccionamiento del proceso de desarrollo y como consecuencia mejorar la calidad en el producto final.[33]

Cada quien establece su definición de calidad de software, por ejemplo, la IEEE plantea que calidad de software es “el grado con el cual el cliente o usuario percibe que el software satisface sus expectativas”(IEEE 729-83). Por otra parte, la Organización Internacional de Estándares(ISO) plantea que es “un conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer una necesidades explícitas o implícitas”. (ISO 8402:1984)

Guiados por la empresa Calisoft¹ (entidad responsable de la calidad externa de los productos de software de la Universidad de Ciencias Informáticas) la calidad del software es definida por el cumplimiento con los requisitos para los que fue desarrollado, aunque no se puede dejar de tener en cuenta que el objetivo principal cuando se busca la calidad es lograr la satisfacción del cliente; para alcanzarla debe existir calidad en :

- El análisis y diseño del producto informático.
- Abastecimientos de herramientas propias para desarrollo del sistema.
- Producción.

¹Centro de Calidad para Soluciones Tecnológicas

- Entrega del sistema terminado.
- Venta o/ servicio.
- Administración general.

1.6.1. Modelos y estándares de calidad de software

Un modelo de calidad de software es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos.

Existe un gran número de modelos y estándares que se aplican en la industria de desarrollo de software, sin embargo un estudio realizado por el Consorcio de Productividad de Software (SPC, por sus siglas en inglés) determinó que los más usados son CMMI y la Norma ISO/IEC.[35]

La Universidad de las Ciencias Informáticas está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (Capability Maturity Model Integration). El proceso de mejora está encaminado a que la universidad alcance en el 2011 una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.

CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software. Con su instauración se espera alcanzar beneficios como:

- Calendarios y presupuestos predecibles en los proyectos.
- Mejora del ciclo de vida dentro del desarrollo de software.
- Mayor productividad.

- Mayor calidad de los productos y servicios que ofrece la universidad a sus clientes y por ende la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro.

En la construcción de RIA que incluyan escenarios tridimensionales interactivos este modelo permite identificar las áreas claves que pudieran afectar la calidad del producto, mejorar los tiempos de entrega, los costos y los cronogramas. Además estas aplicaciones tienen como características que son muy cambiables, lo cual se controla logrando que el equipo de desarrollo y los procesos sean adaptables a este ambiente.

1.7. Bases tecnológicas

En la actualidad el desarrollo a gran escala de software es asistido por herramientas que soportan todos los aspectos del ciclo de vida del software como la gestión de requisitos, modelado visual, herramientas de programación y aseguramiento de la calidad.[19]

El objetivo de contar durante el proceso de desarrollo con herramientas necesarias es precisamente para automatizar tanto como sea posible el proceso. No obstante se debe tener en cuenta que deben ser fáciles de usar y manejable para que merezca la pena su aprendizaje y proporcione un incremento en la producción.[19]

Las herramientas de la ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos.[33]

En la Universidad de las Ciencias Informáticas coexisten un grupo de Direcciones de Servicios subordinadas a la Dirección General de la Infraestructura Productiva encargadas de establecer los lineamientos, disposiciones y procesos en diferentes

áreas del conocimiento que se manifiestan en la producción. Ellas han definido un conjunto de herramientas, mencionando y analizando las que han sido establecidas en el siguiente epígrafe y se seleccionaran aquellas que son necesarias para el desarrollo de la línea de producción que se propone pero que no han sido identificadas.

La Dirección de Calidad de la Universidad coordina un proyecto de mejora de proceso con el fin de fortalecer la producción, reorganizando e industrializando sus procesos bajo los paradigmas de calidad, con el objetivo de alcanzar un proceso de desarrollo de software administrado. Actualmente está trabajando por institucionalizar las buenas prácticas correspondientes al nivel 2 del Modelo de Capacidad de Madurez Integrada de CMMI.

En camino a esto han establecido las siguientes herramientas:

- Herramienta para la trazabilidad: Open Source Requirements Management Tool (OSRMT).
- Herramientas para el modelado: Visual Paradigm o StarUML.
- Herramientas para la gestión de proyecto: RedMine, Project.Net.

El Laboratorio Industrial de Pruebas de Software ubicado en la misma Universidad de las Ciencias Informáticas tiene la responsabilidad de desarrollar y coordinar las pruebas de liberación de los productos que se desarrollan en las diferentes áreas de la universidad. En el mismo se han estado aplicando diferentes herramientas entre ellas el J-Meter y Selenium. La Dirección Técnica de la Infraestructura Productiva está desarrollando el entorno para realizar las salvadas automáticas de los proyectos usando Bacula.

1.7.1. Ambiente Integrado de Desarrollo

Un ambiente integrado de desarrollo (IDE por sus siglas en inglés) es un programa informático compuesto por un conjunto de herramientas para el programador, pueden usarse uno o varios lenguajes de programación.

Se compone de un editor de texto, un compilador, un intérprete, un depurador, un sistema de control de versiones y ayuda para la construcción de interfaces gráficas de usuario.

1.7.2. Herramientas de programación: Eclipse

Eclipse es un entorno libre de desarrollo en Java, flexible para experimentar con lenguajes de programación nuevos o extensiones de estos lenguajes. Es un framework comprensivo que despliega técnicas avanzadas y modernas de implementación y diseño de software.[12]

Debido a que su diseño permite una fácil extensión por parte de terceros se considera una herramienta abierta. En general es un ambiente integrado de desarrollo que proporciona herramientas para gestionar espacios de trabajo, construir, lanzar y depurar aplicaciones. A través de este se puede compartir artefactos con el equipo de desarrollo, construir versiones del código y personalizar fácilmente la experiencia del programador.[12]

Hablando más técnicamente, Eclipse se basa en un mecanismo para descubrir, integrar y ejecutar módulos llamados plugins.[12]

1.7.3. Plugins: Adobe Flash Builder y Subclipse

Adobe Flash Builder 4

Adobe Flex Builder es una herramienta de desarrollo basada en Eclipse muy productiva que incorpora las siguientes funciones: códigos inteligentes, depuración interactiva estratificada, además del diseño visual del aspecto y comportamiento de la interfaz de usuario de las Aplicaciones Enriquecidas de Internet (RIA).[23]

Cuenta con poderosas herramientas de codificación que incluye editores de MXML, ActionScript y CSS, color de sintaxis, cumplimentación de informes, contracción de código, depuración interactiva estratificada y muchas más características.[23]

Subclipse

Subclipse es un plugin para el entorno de desarrollo de Eclipse, el cual proporciona una serie de herramientas que interactúan con el sistema de control de versiones Subversion, permitiendo al usuario tener sus proyectos en un repositorio común al equipo de desarrollo.[7]

Es uno de los plugin más utilizados para integrar Subversion con Eclipse, incluso más que el plugin oficial.[7]

Es un plugin muy útil para el desarrollo colaborativo, en el que intervienen un conjunto de desarrolladores trabajando sobre el mismo proyecto, poniendo a disposición del equipo de desarrollo facilidades para el trabajo en equipo.[7]

1.7.4. Framework:Flex SDK 4

Flex es un framework altamente productivo de código abierto utilizado para crear Aplicaciones Enriquecidas de Internet. Los desarrolladores pueden crear aplica-

ciones web más atractivas e intuitivas que permite a los clientes, socios y empleados administrar e interpretar los datos con mayor eficiencia. Los resultados que obtienen son incremento de la productividad y mayor satisfacción en los usuarios.[17]

Flex proporciona un lenguaje basado en estándares modernos y modelos de programación que soportan patrones de diseño comunes. Además, incluye una biblioteca rica en componentes con más de 100 componentes probados y componentes extensibles de interfaz de usuario para crear RIA, así como un depurador interactivo de aplicaciones Flex.[17]

1.7.5. Lenguajes de programación: ActionScript 3.0

ActionScript es el lenguaje de programación de la plataforma Adobe Flash. Originalmente se hizo para que los desarrolladores programaran interactivamente. Permite una programación eficiente en aplicaciones de plataforma flash por todo, desde animaciones simples hasta complejas, ricas en datos e interfaces gráficas.[28]

Adobe declara en su página oficial los beneficios más significativos de utilizar ActionScript: [28]

- Lograr un mayor rendimiento.
- Aprovecha las nuevas APIs de Flash Player.
- Aprovecha las bibliotecas de la comunidad y APIs.
- Solución de problemas en el código más fácil.
- Desarrolla contenidos para múltiples plataformas.

1.7.6. Sistemas de gestión de base de datos: MySQL y PostgreSQL

MySQL

Se ha convertido en la base de datos de código abierto más popular debido a su alto rendimiento, alta fiabilidad y facilidad en su uso. Además, es la base de datos de elección para una nueva generación de aplicaciones basadas en la pila LAMP (Linux, Apache, MySQL, PHP / Perl / Python.)[29]

Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo, incluyendo Facebook, Google, Adobe, Alcatel Lucent y Zappos se basan en MySQL para ahorrar tiempo y dinero en sus grandes volúmenes de sitios web, los sistemas críticos de negocio y paquetes de software.[29]

MySQL se ejecuta en más de 20 plataformas, incluyendo Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX, que le da el tipo de flexibilidad para un mayor control. Además, ofrece una amplia gama de herramientas de base de datos, asistencia técnica, capacitación y servicios de consultoría para obtener el éxito.[29]

PostgreSQL

PostgreSQL es un sistema de base de datos objeto-relacional de código abierto muy poderoso. Cuenta con las de 15 años de desarrollo activo y arquitectura probada, ganándose una sólida reputación de fiabilidad e integridad de datos.[27]

Se ejecuta en todos los principales sistemas operativos incluyendo Linux, UNIX y Windows. Se incluyen la mayoría de los tipos de datos (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, entre otros.) Además, cuenta con integración de llaves foráneas y procedimientos almacenados(en varios idiomas).[27]

Cuenta con interfaces nativas de programación para C / C + +, Java,. Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y una documentación excepcional.[27]

Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede acomodar. Hay sistemas activos de PostgreSQL en entornos de producción que manejan más de 4 terabytes de datos.[27]

PostgreSQL ha ganado los elogios de sus usuarios y el reconocimiento de la industria, incluyendo el Premio de Linux New Media a la mejor base de datos del sistema y cinco veces ganador del Premio de elección de los Editores de Revistas Linux para el mejor DBMS (sistema de gestión de base de datos).[27]

1.7.7. Herramientas de modelado 3D: Blender

Blender ofrece en un paquete un amplio espectro de funcionalidad para el modelado, texturizado, iluminación, animación y post-procesado de vídeo. Por medio de su arquitectura abierta, ofrece interoperabilidad entre plataformas, extensibilidad, una superficie de apoyo increíblemente pequeña y un flujo de trabajo altamente integrado. Blender es una de las aplicaciones Open Source de gráficos 3D más populares del mundo.[4]

Blender puede usarse para crear visualizaciones 3D, tanto imágenes estáticas como videos de alta calidad, mientras que la incorporación de un motor 3D en tiempo real permite la creación de contenido interactivo que puede ser reproducido independientemente.[4]

Blender tiene su propio motor de juegos interno que le permitirá crear aplicaciones interactivas en 3D. El motor de juegos de Blender (BGE) es una poderosa herramienta de programación a alto nivel. Está enfocada principalmente al desarrollo de juegos pero puede usarse para crear software de tipo interactivo, tales como recorridos arquitectónicos en 3D o investigación.[4]

Características principales:

- Paquete de creación totalmente integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D, incluyendo modelado, mapeado, texturizado, animación, simulación de partículas y otros, scripting, render, composición, post-producción y creación de juegos.
- Multiplataforma, con una interfaz basada en OpenGL, lista para ser usada en todas las versiones de Windows, Linux y OS X.
- Arquitectura 3D de alta calidad permitiendo un rápido y eficiente desarrollo.

1.7.8. Herramientas de modelado: Visual Paradigm y Rational Rose

Visual Paradigm

Visual Paradigm es una herramienta de diseño UML y herramienta CASE UML diseñada para la ayuda al desarrollo de software. Soporta los principales estándares de la industria tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI, entre otros. Además, ofrece un completo conjunto de herramientas de los equipos de desarrollo de software necesario para los requisitos de la captura, software de planificación, la planificación de controles, el modelado de clases y modelado de datos.[32]

Genera documentos de diseño de software actualizados en formato PDF, MS Word y en formato HTML de forma automática. Además, permite la ingeniería inversa y directa en las de 10 lenguajes de programación y genera base de datos de los diagramas entidad-relación.[32]

Visual Paradigm para UML apoya la generación de bases de datos para todas las bases populares en el mercado, como MySQL, MS SQL Server, Oracle, Sybase,

PostgreSQL, Derby, Informix, Firebird y SQLite.[32]

Rational Rose

Rational Rose es una herramienta de diseño de software destinado para el modelado visual y componentes para la construcción de aplicaciones de software a nivel empresarial. Un diseñador de software lo utiliza para crear visualmente el marco para una aplicación mediante el bloqueo de las clases con los, los elementos de casos de uso, los objetos y los mensajes / relaciones en un diagrama de secuencia utilizando los símbolos de arrastrar y soltar. Rational Rose documentos el diagrama que se está construyendo y luego genera el código en la elección de los diseñadores de C + +, Visual Basic, Java, Oracle, CORBA o lenguaje de definición de datos.[34]

Una de sus características más resaltantes es que proporciona un desarrollo iterativo o evolutivo ya que la nueva aplicación se puede crear en los escenarios con la salida de una iteración convirtiéndose en la entrada del siguiente.[34]

Ofrece capacidades de análisis de calidad del código y la generación de código, con configurables en modelos con las capacidades de sincronización de código. Incluye un modelado Web Add-In, que permite la visualización, modelado y herramientas para el desarrollo de aplicaciones web. Además, proporciona modelado UML para diseñar bases de datos, con la capacidad para representar la integración de los datos y requisitos de las aplicaciones a través de diseños lógicos y físicos.[16]

1.7.9. Servidores de Aplicaciones:Apache

Apache es un esfuerzo por mantener y desarrollar un servidor HTTP libre para los sistemas operativos modernos incluyendo UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sintonía con los actuales estándares de HTTP.[21]

Es un potente servidor web, flexible y altamente configurable ya que proporciona el código fuente completo. Constantemente se encuentra en desarrollo ayudado por la extensa comunidad de usuarios que detectan errores, crean parches y brindan nuevas ideas.[21]

Apache permite crear con facilidad páginas protegidas por contraseña con gran cantidad de usuarios sin que se atasque el servidor. Provee respuestas personalizadas para errores y problemas. Además, cuenta con un potente motor de que se puede utilizar para resolver la mayoría de los problemas de manipulación de URL.[21]

1.7.10. Consideraciones sobre las herramientas

El mejoramiento continuo de las herramientas en el desarrollo de software se hace totalmente necesario para lograr la optimización de cualquier proceso y aumentar los niveles de productividad del equipo de desarrollo. Quien maneja las herramientas y en general la organización, debe dominar la utilidad de cada herramienta, los mecanismos para implantarla y la evaluación de las mismas.

La variedad de herramientas que coexisten hace difícil elegir entre ellas cual es mejor, por tanto esta selección debe basarse en sus ventajas y en la facilidad que tengan los desarrolladores de adaptarse a ellas sin poner en riesgo la productividad

y tiempo de entrega del proyecto. Como resultado del análisis anterior se identifica un catálogo de posibilidades y las ventajas de su uso, resolviendo cual utilizar después de caracterizar el dominio de la aplicación.

1.8. Conclusiones parciales

En el capítulo se han identificado los elementos fundamentales para desarrollar la propuesta de personalización del proceso de desarrollo para las RIA: los modelos, las metodologías que guían el proceso de desarrollo y las herramientas del ambiente de desarrollo para crear las aplicaciones de la línea de producción. Logrando de esta forma que el tiempo de desarrollo disminuya, que aumente la organización y control durante el proceso y consecuentemente el aumento en la calidad del producto.

Los elementos a tener en cuenta para la personalización del proceso de desarrollo son los que se mencionan a continuación:

- Los modelos a tener en cuenta tienen la característica de ser incrementales y basados en componentes.
- La metodología que guía el proceso de desarrollo es AUP que constituye una metodología ágil bien detallada basada en RUP.
- El ambiente integrado de desarrollo está compuesto por una gama de herramientas que los desarrolladores pueden utilizar. Se identifican herramientas de programación, plugins, framework, gestores de base de datos, herramientas de modelado en 3D, servidor de aplicaciones y algunas herramientas de gestión y soporte.

Sin embargo, el estudio de las metodologías demuestra que no hay una opción que lo reúna todo y se considera la posibilidad de combinarlas a pesar de que la base sea Proceso Unificado Ágil el cual nos permite seguir un orden lógico de las actividades, los roles, las tareas y los artefactos de entrada y salida. Esto garantiza el uso de buenas prácticas y el desarrollo de flujos de trabajo organizado, estructurado y disciplinado dando la oportunidad de que el proyecto tenga la capacidad de repetirlos y mejorarlos.

Capítulo 2

Solución técnica

Introducción

En el siguiente capítulo se describen los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación. Además se realiza una reseña de la Universidad de las Ciencias Informáticas (UCI), por ser el ambiente donde se implementará este proceso, referenciando las políticas y normas que influyen en la propuesta. Se presenta la propuesta de proceso de desarrollo, identificando los principales elementos que lo componen.

2.1. Métodos, procedimientos y técnicas utilizadas

En la investigación, como métodos científicos de investigación se han utilizado los Métodos Teóricos, específicamente el método histórico-lógico y el analítico-sintético.

Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e

hipótesis de investigación. Se analizó el cambio de visión referente al tema de procesos de desarrollo desde su inicio hasta la actualidad con análisis histórico-lógico. El método analítico-sintético permitió la selección de los modelos de proceso de desarrollo, las metodologías de desarrollo y herramientas utilizadas a lo largo del desarrollo de software en proyectos informáticos que construyen Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

La investigación continúa con la utilización de métodos empíricos como la entrevista y la consulta a expertos con el objetivo de conocer como es el flujo de desarrollo en el ambiente de proyecto. Los mismos fueron vitales para el diagnóstico de la organización, para establecer los elementos necesarios para la lógica del modelo, avalar los conceptos que se manejan en la investigación, medir el alcance y la importancia que tiene la temática. Captar la información cualitativa y cuantitativa del fenómeno, conocer los criterios sobre la forma en que se organiza y se lleva a cabo la producción de software en la UCI, así como las posibles soluciones que se proponen en la investigación, para ello se entrevistaron y encuestaron personas involucradas en la producción de software en la UCI, con competencia en el tema.

2.1.1. Entrevista # 1

La población a estudiar fue el personal involucrado en los proyectos de producción de software de la UCI y la unidad de estudio el proceso de producción en los proyectos de desarrollo de software de la facultad 4. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación que fue entrevistar personas que tienen experiencia en la producción y que han tenido que enfrentar problemáticas que le han permitido ir madurando.

Se seleccionaron directivos institucionales, jefes de centro, líderes de proyectos y desarrolladores de proyectos de las diferentes líneas de producción de la Universidad, teniendo en cuenta la cantidad de proyectos de la universidad la muestra para la investigación es mayor del 20 %.

Se realizó con el objetivo de identificar el grado de conocimiento de los involucrados de la situación problemática y del problema, validar la propuesta de solución y recopilar elementos a tener en cuenta en la solución. En el Anexo 6 se puede ver el diseño de la entrevista.

2.1.2. Entrevista # 2

La población a estudiar fueron desarrolladores de software de la UCI y la unidad de estudio el flujo de desarrollo en los proyectos de construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional. Se seleccionaron desarrolladores vinculados a la producción de estas aplicaciones.

Se realizó con el objetivo de conocer como era el flujo de desarrollo aplicado para la construcción de estas aplicaciones y recopilar elementos a tener en cuenta en la solución. En el anexo 7 se puede ver el modelo de la entrevista aplicada a los desarrolladores de software de la UCI.

2.1.3. Método para la evaluación del proceso de desarrollo propuesto. Consulta a expertos

Encuesta #1

La población a estudiar fue el personal involucrado en los proyectos de producción de software de la UCI y la unidad de estudio el proceso de producción en los proyectos de desarrollo de software de la UCI. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación que fue entrevistar personas que tienen experiencia en la producción y que han tenido que enfrentar problemáticas que le han permitido ir madurando.

Se seleccionaron 12 directivos institucionales, 4 jefes de centro, 9 líderes de proyectos y 15 desarrolladores de proyectos de las diferentes líneas de producción de la Universidad, teniendo en cuenta la cantidad de proyectos de la universidad la muestra para la investigación es mayor del 20 %.

El instrumento de medición utilizado fue la encuesta en la modalidad de cuestionario donde se realizaron preguntas a través del correo electrónico. Se realizó con el objetivo de validar las características del proceso de desarrollo propuesto, recopilar elementos a tener en cuenta para su evolución y realizar comparaciones a través de procedimientos estadísticos.

En la encuesta se evaluaron las características del proceso de desarrollo, las mismas fueron:

Características	Subcaracterísticas
Organización del proceso y las personas	Definición de Roles y Responsabilidades Definición del flujo de trabajo
Entendible	Facilidad de aprendizaje Capacidad de ser analizado
Fiable	Madurez Establecimiento de la revisión y control del proyecto Tolerancia a fallos
Visible	Generación de documentación
Apoyado por Herramientas CASE	Soporte Interactivo para el procesos Facilidad en revisión de aplicaciones
Rapidez	Obtención ágil de resultados Gestión de tiempo
Aceptabilidad	Ayuda Facilidad de uso Utilización de recursos

A la hora de elaborar la encuestas se combinaron tipos de preguntas. La mayoría fueron semicerradas pues se tiene el interés de conocer la información cuantitativa pero también de saber la opinión del tema, así como involucrar y motivar a los encuestados en la solución. En el Anexo 8 se puede ver el modelo de la encuesta aplicada.

2.2. Método para la evaluación técnica del proceso de desarrollo propuesto

Con el objetivo de realizar la evaluación técnica del proceso de desarrollo propuesto se utilizó el método de evaluación técnica multicriterios basado en los aspectos cualitativos evaluados por expertos, documentado en el Curso Básico de Gestión

de Proyectos del DrC. Rolando Alfredo Hernández León. Este método permite realizar un estudio de expertos que permita tomar decisiones para aceptar o no el proyecto de acuerdo con los criterios técnicos sobre el mismo.[22]

Puede calcularse el número de expertos necesarios, utilizando un método probabilístico y asumiendo una ley de probabilidad binomial mediante la siguiente expresión:

$$n = \frac{p*(1-p)*k}{i^2}$$

Donde:

i-Nivel de precision deseado.

p-Proporción estimada de errores de los expertos.

k-Constante asociada al nivel de confianza elegido.

Luego se realiza un proceso de selección de los expertos. La calidad de los expertos influye decisivamente en la exactitud y fiabilidad de los resultados y en ello interviene la calificación técnica, los conocimientos específicos sobre el objeto a evaluar y la posibilidad de decisión entre otros.

En este método se utilizó un conjunto de evaluadores, los cuales son expertos en diferentes temas asociados a la investigación, tienen grados científicos y a su vez son investigadores que pertenecen al sistema y que participan en convocatorias a diferentes eventos de la rama o especialistas que intervienen en el proceso productivo de software. Para la selección de los evaluadores se tuvo en cuenta su especialidad, grado científico y curriculum.

Se elaborará un conjunto de preindicadores los cuales fueron sometidos al criterio de los evaluadores, después de recibidas las opiniones y llegado a un consenso se elaboraron los criterios que fueron utilizados para la evaluación. Se definieron cuatro grupos de criterios los cuales son:

- Criterios de mérito científico.
- Criterios de evaluación.
- Criterios de generalización.
- Criterios de impacto.

Se le asigna un peso relativo a cada grupo de criterios de acuerdo con las características de la propuesta, el cual fue:

- Grupo 1...20
- Grupo 2...35
- Grupo 3...20
- Grupo 4...25

Los evaluadores estudiaron la temática a evaluar, se les entregó el Libro de Proceso de Desarrollo y dos modelos. El modelo 1 tiene como objetivo principal que valoren el peso relativo de cada criterio (Ver anexo 9) y el modelo 2 para realizar una evaluación cualitativa de cada criterio con una escala del 1-5 y la apreciación cualitativa con una clasificación final del proyecto en excelente, bueno, aceptable, cuestionable y malo (Ver Anexo 10). Con la información recibida dispusieron de un tiempo determinado.

Después, a través del peso relativo otorgado a cada criterio por el trabajo de expertos y la calificación cuantitativa realizada por los evaluadores, se podrá elaborar un índice de aceptación, lo que unido a la evaluación cualitativa y la evaluación final dada por los evaluadores, permitirá determinar la calidad del modelo.

Los criterios de evaluación por clasificación fueron:

Grupo No.1: Criterios de mérito científico.

- Calidad de la investigación.
- Novedad científica.
- Valor científico de la propuesta.
- Aporte científico.

Grupo No.2:Criterios de evaluación.

- Satisfacción de las necesidades de la producción.
- Garantía de los principios básicos de la IS.
- Uso de estándares de calidad.
- Necesidad de efectividad en el proceso.

Grupo No.3:Criterios de generalización.

- Atractividad para su uso.
- Adaptabilidad a proyectos en la misma línea de desarrollo.

Grupo No.4:Criterios de impacto.

- Repercusión en entidades que desarrollan software.
- Organización del proceso de producción.
- Posibilidad de aplicación.

Después de recibir los valores del peso relativo de cada criterio se construye la tabla que se muestra en el Anexo 11. Luego es necesario verificar la consistencia en el trabajo de expertos (ver Anexo 12), para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado(X^2).

Se sigue el procedimiento siguiente:

Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.

Para cada criterio se determina ΣE que representa la sumatoria del peso dado por cada experto.

Se calcula el peso medio de cada criterio ($M\Sigma E$) y se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión.

$$S = \Sigma(\Sigma E - \Sigma \Sigma \frac{E}{C})^2$$

Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W).

$$W = S/E^2(C^3 - C)/12$$

El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real.

$$X^2 = E(C - 1)W$$

El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas.

Si se cumple:

$$X^2_{real} < X^2(\alpha, c - 1)$$

Existe concordancia en el trabajo de expertos.[22]

Si existe consistencia en el trabajo de expertos el peso de cada criterio se calcula promediando lo que cada uno de ellos le asignó a cada criterio entre 100, pero si no existe concordancia se hace necesario repetir el trabajo de expertos o eliminar a los expertos que afectan la concordancia.

Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la tabla de calificación de cada criterio(Ver Anexo 13) y se determina el índice de aceptación del proyecto.

$$IA = P * C/5$$

IA:Índice de aceptación.

P:Peso de los criterios.

C:Criterio promedio concedido por los expertos.

Si:

$IA > 0,7$ Existe alta probabilidad de éxito.

$0,7 > IA > 0,5$ Existe probabilidad media de éxito.

$0,5 > IA > 0,3$ Probabilidad de éxito baja.

$0,3 > IA$ Fracaso seguro.

2.3. Entorno de implantación: Universidad de las Ciencias Informáticas

Tradicionalmente la universidad ha generado conocimientos pero es muy difícil que se aplique este conocimiento durante el aprendizaje, provocando un distanciamiento con la realidad necesaria de producción. Este distanciamiento impide la

realización de investigaciones que respondan a las necesidades de la industria. Por otra parte a la industria le es muy difícil investigar por depender de un mercado que exige tiempo, costos y calidad. La búsqueda de soluciones ha generado una combinación de universidad-industria.

La Universidad de las Ciencias Informáticas tiene la característica de tener vinculado la docencia con la producción, lo que permite a los estudiantes poder consolidar la teoría dada en clases y a su vez enfrentarse a las características de un desarrollo real de productos informáticos, con clientes reales y requisitos que cumplir. La producción está organizada en 7 facultades las cuales conforman 12 Centros de Desarrollo y Producción asociados a líneas temáticas.

Uno de los centros establecidos en la universidad es el Centro de Calidad para Soluciones Tecnológicas responsable de normar los elementos para la producción, asegurar la calidad de los proceso y productos, y desarrollar las pruebas de liberación y aceptación de las soluciones informáticas para su entrega al cliente.

Entre los elementos normados existe el Expediente de Proyecto del Proceso de Mejora versión 3.2 y los Proceso y Guías. El primero busca establecer y estandarizar la documentación de los productos en la Universidad para su desarrollo, así como el mantenimiento y reutilización en desarrollos posteriores. En el mismo se recogen los elementos técnicos, de gestión y soporte. El segundo se dirige a establecer e institucionalizar las buenas prácticas de la ingeniería de software en los proyectos productivos.

2.4. Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Antes de la definición del ciclo de vida que tendrá la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos, se debe recordar que el proceso estará guiado por la metodología AUP (una versión ágil de la metodología RUP) por tanto muchos de los elementos que se definen coincidirán con RUP siempre adaptado a las necesidades y características del proyecto. Además se tomara en cuenta los procesos y guías establecidos en Calisoft para los proyectos en el proceso de mejoras.

Definir el proceso productivo para el desarrollo del producto, es el objetivo fundamental de este epígrafe. El flujo de trabajo comprende el conjunto de actividades, los roles, las tareas, los artefactos de entrada y salidas para guiar la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Ciclo de Vida

El ciclo de vida de los productos, subsistemas o componentes contienen los flujos organizados consecutivamente: Estudio Preliminar, Requisitos, Análisis y Diseño, Implementación, Pruebas, Despliegue y Soporte.

Los flujos se desarrollarán en las etapas definidas por la metodología de desarrollo Proceso Unificado Ágil: Inicio, Elaboración, Construcción y Transición. Teniendo en cuenta que es un proceso iterativo e incremental constará con una iteración en la etapa de Inicio, una iteración en la etapa de Elaboración, n iteraciones en la etapa de Construcción y dos iteraciones en la etapa de Transición.

Paralelamente se incorporan las áreas de proceso que conforman el nivel 2 de CMMI establecido en el Proceso de Mejora, ellos son: Administración de Requisitos, Planeación de Proyecto, Monitoreo y Control del Proyecto, Administración de Acuerdo con Proveedores, Medición y Análisis, Aseguramiento de Calidad a Producto y Proceso y Administración de Configuración. Sin embargo, basado en la metodología que guía este proceso de desarrollo se le agrega el Ambiente, quedando de la siguiente manera:

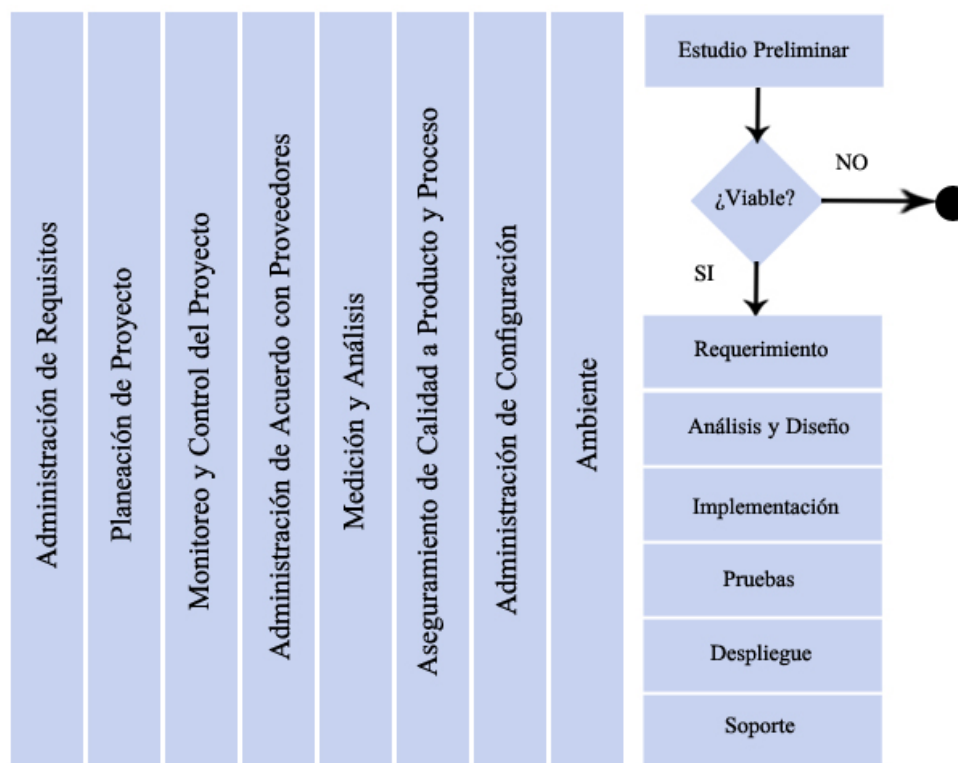


Figura 2.1: Ciclo de Vida.

A continuación se describen las principales actividades, los roles y los artefactos a desarrollar en cada uno de los flujos, el resto de los elementos se describen en los anexos que se referencian. Los artefactos corresponden con el Expediente de Proyecto versión 3.3 que establece la Dirección de Calidad de la UCI.

2.4.1. Estudio Preliminar

En este flujo se realiza un estudio inicial de la organización del cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo. Un equipo compuesto por el Jefe de Proyecto, el Ingeniero de Proceso, el Arquitecto de Software, el Analista y el Administrador de Calidad realiza las siguientes actividades:

- **Entender las necesidades del cliente:** incluye identificar las demandas del cliente, recoger información sobre el escenario, describir los elementos del escenario y la relación con el actor, manejar las dependencias entre los elementos, evaluar los resultados y elaborar el informe técnico y el proyecto técnico.
- **Analizar las necesidades del cliente:** incluye encontrar los términos comunes, determinar las áreas a priorizar dentro del escenario, desarrollar las reglas de comportamiento de los elementos del escenario y refinar informe y proyecto técnico.
- **Realizar estimaciones de tiempo, esfuerzo y costo:** definir los paquetes funcionales del sistema, determina la complejidad de los paquetes funcionales (grande, mediano o pequeño), determinar cantidad y complejidad de los factores y obtener reporte de las estimaciones de tiempo, esfuerzo y costo generado por la herramienta del Método de Estimación Calisoft.

- **Supervisar el alcance del proyecto:** incluye definir el contexto inicial del proyecto, analizar la arquitectura, analizar la tecnología, decidir la estrategia para el desarrollo del proyecto, establecer un cronograma (WBS y Calendario), evaluar los resultados, desarrollar el Plan de Desarrollo de Software, actualizar el informe y proyecto técnico.

En los Anexos 14 y 15 se puede observar la descripción gráfica y textual de este flujo. Para mayor información leer el Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.4.2. Requerimientos

El objetivo principal de este flujo es desarrollar el modelo del sistema que se va a construir. Incluye un conjunto de casos de uso que responden a los requisitos funcionales del sistema y la especificación de los requisitos no funcionales.

En esta etapa se incorpora al equipo de desarrollo el diseñador gráfico el cual será el encargado de definir las pautas de diseño, la arquitectura de información y ayuda a validar los requisitos del producto.

Las actividades principales son:

- **Identificar actores y casos de uso:** incluye describir la funcionalidad de la aplicación, definir qué debe manejar la aplicación, definir quién y cómo será la interacción con esta y crear la Especificación de Requisitos de Software.
- **Priorizar los casos de uso:** incluye definir los casos de uso que representan la funcionalidad central y definir los casos de uso que tienen gran influencia en la definición de la arquitectura de software.

- **Detallar los casos de uso:** consiste en describir el flujo de eventos al detalle y crear especificación de casos de uso.
- **Modelar y crear el prototipo de interfaz de usuario:** construir un modelo de la interfaz de usuario que de soporte a las actividades de los usuarios y satisfagan los requerimientos de usabilidad.

En los Anexos 16 y 17 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.4.3. Análisis y Diseño

El propósito de este flujo de trabajo es transformar los requerimientos en un modelo de la aplicación a desarrollar, definir una arquitectura robusta para la aplicación y adaptar este diseño a un entorno de implementación. Los roles que intervienen son el Arquitecto de Software, el Diseñador de Sistema y Diseñador de Base de Datos, los cuales realizan las siguientes actividades:

- **Analizar la arquitectura:** incluye definir la arquitectura candidata y definir los patrones arquitectónicos a utilizar.
- **Analizar los casos de uso:** incluye identificar las clases, distribuir el comportamiento de los casos de uso por las diferentes clases y especificar las responsabilidades, atributos y asociaciones de cada clase.
- **Diseñar los casos de uso:** consiste en refinar y distribuir los requerimientos como métodos entre las distintas clases.

- **Diseñar las clases:** el objetivo es asegurar que las clases proveen del comportamiento especificado en los casos de uso.
- **Diseñar la base de datos:** incluye asegurar el almacenamiento eficaz de los datos persistentes y definir el comportamiento de la base de datos.

En los Anexos 18 y 19 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.4.4. Implementación

El objetivo inicial de este flujo construir el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Además, completa el desarrollo del sistema basado en la línea base de la arquitectura. El Arquitecto de Software, el Ingeniero de Proceso, el Diseñador Gráfico y el Programador realizan según sus responsabilidades las siguientes actividades:

- **Estructurar el modelo de implementación:** incluye ajustar los subsistemas de implementación, definir las importaciones de cada subsistema de implementación, actualizar la vista de implementación y evaluar el modelo de implementación.
- **Implementar elementos de diseño:** inicia con la preparación de la implementación (comprender el problema y configuración del entorno de desarrollo). Además, es necesario transformar el diseño en implementación, evaluar la implementación y retroalimentar al diseño de lo que se obtuvo.

- **Creación de escenas y escenarios tridimensionales:** guiados por el prototipo de interfaz de usuario y los casos de uso se modela tridimensionalmente los requisitos analizados.
- **Planificar la integración del sistema:** es necesario identificar los subsistemas, definir una serie de compilación (para integrar el sistema de forma incremental) y evaluar el plan de compilación de integración.
- **Integrar el sistema (escenas, escenarios y funcionalidades):** incluye aceptar subsistemas y producir compilaciones intermedias. Además, será necesario promocionar las líneas bases.

En los Anexos 20 y 21 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.4.5. Pruebas

Las pruebas se centran principalmente en la evaluación o la valoración de la calidad del producto. Existen dos etapas de pruebas llamadas pruebas internas y pruebas de liberación.

Pruebas Internas.

Es el proceso mediante el cual se establece la existencia de errores en la aplicación por el equipo de desarrollo. Los roles que intervienen durante este flujo son el Diseñador de Pruebas, el Ingeniero de Procesos, el Analista y el Administrador de la Calidad.

Las principales actividades que se realizan son:

- **Definir enfoque de prueba:** incluye definir la estrategia de prueba y las técnicas específicas que se utilizarán. Su objetivo es describir los trabajos de cada técnica y los tipos de pruebas, y establecer una arquitectura candidata para el sistema de automatización de pruebas.
- **Definir requisitos para la valoración:** incluye identificar requisitos de valoración y rastreabilidad, identificar las restricciones que afectaran la implementación de los requisitos, considerar las estrategias posibles y definir los requisitos de configuración de las herramientas de soporte que llevan el proceso de valoración.
- **Estructurar la implementación de la prueba:** incluye examinar el enfoque de prueba, los elementos de prueba destino y las necesidades de valoración. Además, se debe examinar los mecanismos de comprobabilidad y modelar visualmente la arquitectura de implementación de la prueba.
- **Ejecutar el conjunto de aplicaciones de prueba:** incluye configurar el entorno de prueba, planificar el conjunto de aplicaciones de prueba y evaluar la ejecución.
- **Evaluar las pruebas:** incluye examinar todas las incidencias de las pruebas, crear las solicitudes de cambio, analizar y evaluar el estado de la aplicación, hacer una valoración de la experiencia del desarrollo, hacer una valoración de riesgos técnicos y de los resultados de las pruebas y evaluar y verificar sus resultados. Se obtiene como resultado una minuta de la reunión y las solicitudes de cambio.

En los Anexos 22 y 23 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los

artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Pruebas de Liberación.

Es el proceso mediante el cual se establece la existencia de errores en la aplicación por el Laboratorio de Pruebas de Calisoft. El Jefe de Proyecto es el encargado de llenar las planillas para las pruebas de liberación y enviar la solicitud al Laboratorio de Pruebas.

2.4.6. Despliegue

Durante este flujo se procede a la entrega de la aplicación, así como su instalación, configuración, prueba y puesta en marcha en el entorno real del cliente. Las pruebas de esta fase son pruebas de aceptación y pruebas piloto. Además, se realiza capacitación al cliente para asegurar los conocimientos mínimos para la manipulación de la aplicación. Los roles que ejecutan las tareas son el Implantador de Soluciones, el Jefe de Proyecto, el Arquitecto de Software y el Programador.

Las actividades principales durante este flujo son:

- **Desarrollar el plan de instalación:** incluye planificar cómo empaquetar el software, planificar cómo distribuir el software, planificar cómo instalar el software y proporcionar ayuda y asistencia a los usuarios.
- **Desarrollar los materiales de instalación:** incluye desarrollar los scripts de instalación, desarrollar los archivos de configuración y desarrollar instrucciones de instalación.
- **Ejecutar la instalación:** incluye preparar el ambiente para las pruebas de

aceptación, realizar las pruebas de aceptación y preparar el registro de revisión y de defectos del documento.

En los Anexos 24 y 25 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.4.7. Soporte

Durante este flujo y por un tiempo limitado se ofrece un servicio para resolver conflictos y problemas de usabilidad y rendimiento del software entregado al cliente, suministrándole actualizaciones y parches a errores. El Especialista de Soporte, el Ingeniero de Proceso y el Programador realizan las siguientes actividades de acuerdo con las respectivas responsabilidades asignadas:

- **Preparar ambiente de soporte:** incluye preparar la base de conocimiento para el soporte y preparar el ambiente para recibir solicitudes de soporte del cliente.
- **Responder solicitud de reporte:** Evaluar solicitud de soporte, responder solicitud, recoger errores o inconformidades de los usuarios con la aplicación y actualizar la base de conocimientos.
- **Realizar pruebas de soporte:** consiste en determinar si se ha solucionado la problemática encontrada en la aplicación.

En los Anexos 26 y 27 se puede observar la descripción gráfica y textual de este flujo. Para conocer las responsabilidades de cada rol, las actividades que realiza y los

artefactos que genera dirigirse al Libro de Proceso de Desarrollo para Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

2.5. Conclusiones Parciales

En el presente capítulo quedaron descritos los elementos que componen un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos. Se describieron los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación.

Además, se expuso la propuesta de proceso de desarrollo permitiendo de esta forma ubicar al equipo de desarrollo en las actividades que debe realizar durante cada etapa del ciclo de vida, los roles que intervienen y las respectivas responsabilidades.

El proceso de desarrollo que se describe en este capítulo desde el punto de vista teórico instrumenta los principios de industrialización de la producción a través de la definición y establecimiento de todos sus procesos, provocando altos niveles de reutilización y productividad, acortando el tiempo de desarrollo en la construcción de este tipo de aplicaciones y garantizando el uso de las buenas prácticas de la Ingeniería de Software. Permite realizar procesos organizados, estructurados y con un nivel de disciplina alto, donde el equipo de desarrollo tiene la capacidad de repetirlos y mejorarlos.

Capítulo 3

Evaluación de la propuesta

3.1. Introducción

En el presente capítulo se describe como se evaluaron las características de un proceso de desarrollo propuesto para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos. La evaluación nos permite mediante un método establecido conocer que tan eficiente resulta que el equipo de desarrollo se guíe por el proceso de desarrollo propuesto. Además, se describe el método de implantación del proceso el cual es guiado por las 10 prácticas generales establecidas por CMMI nivel 2 con el fin de institucionalizar un proceso gestionado.

3.2. Análisis de los resultados de la aplicación de las encuestas y las entrevistas

Se entrevistó según lo planificado a un total de 40 personas, directivos de la UCI, especialistas externos de Softel y Calisoft, líderes de proyecto, jefes de centro y desarrolladores.

La entrevista arrojó que se tiene conciencia de la situación actual de la producción y que hay gran interés por parte de los desarrolladores y directivos para la búsqueda de soluciones. Consideran que se puede trabajar mucho más en la planificación y en la definición de un proceso de desarrollo pues aunque dos proyectos que trabajan en una misma línea de producción usen la misma metodología para guiar el proceso lo aplican de manera totalmente diferente. Otro elemento que se debe fortalecer es la gestión de la calidad, el tiempo, costo y los recursos y el uso de estándares, así como la reutilización de código.

La mayoría de los entrevistados aunque no conocían las características de un Libro de Proceso de Desarrollo al hacerles la presentación del mismo plantearon que contar con un proceso de desarrollo establecido para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos pudiera cambiar el entorno productivo, elevar la calidad del software y la productividad del equipo de desarrollo.

Entre los elementos que más se mencionaron que no debía faltarle a un proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos es que permitiera la gestión de la calidad, la definición de roles ajustados a las necesidades y que permita la definición de las bases tecnológicas.

Las encuestas se les realizaron a 9 líderes de proyecto, 4 jefes de centro, 12 directivos y 15 desarrolladores, a un total de 40 personas de 11 proyectos de todas las líneas de producción y facultades de la universidad. En la figura 3.1 se muestra el gráfico de pastel que representa la composición de los encuestados.

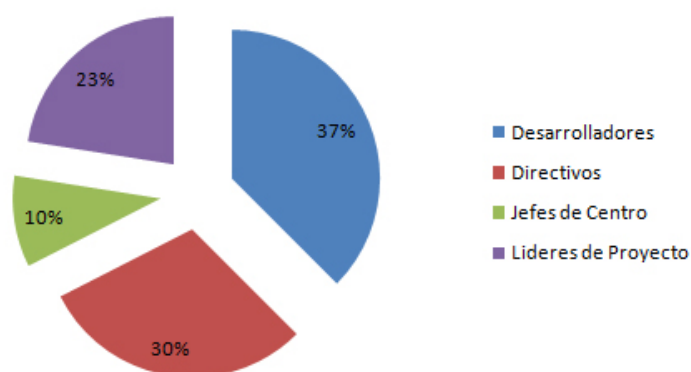


Figura 3.1: Composición de los encuestados.

Los resultados arrojaron que un 83% de los encuestados conocen realmente las responsabilidades del rol que desempeñan en su proyecto. Los desarrolladores con un 67% de conocimiento fueron los del por ciento mas bajo de conocimiento sobre su rol, problema que influye a la hora de realizar las actividades durante el ciclo de vida del software.

El conocimiento del flujo de trabajo es crítico por lo que se considera que no puede existir una buena planificación a nivel de persona y de equipo lo que repercute negativamente en la eficiente planificación del proyecto. Estos datos se muestran en la figura 3.2. En los casos de los líderes de proyecto el 67%, los desarrolladores el 53%, los jefes de centro el 50% y los directivos el 58%; el resultado mas bajo lo presenta los líderes de proyectos que son los que deberían controlar, exigir y dar el ejemplo en el uso de estándares.

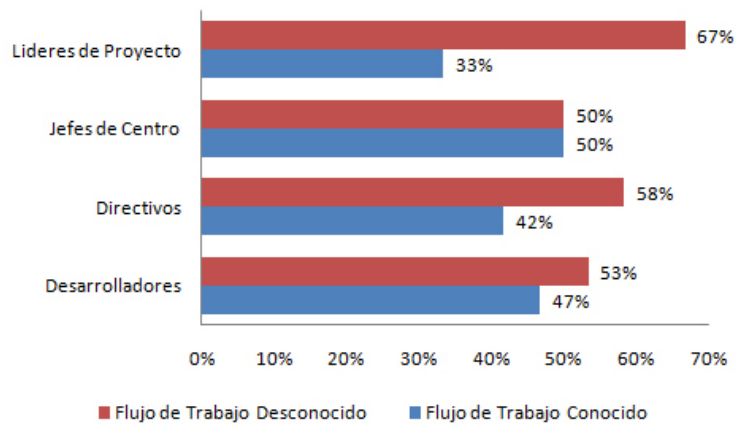


Figura 3.2: Conocimiento del flujo de trabajo.

Después de explicar y mostrarles el Libro de Proceso de Desarrollo a los evaluadores, se le realizó tratamiento estadístico a los resultados, obteniendo la evaluación de las características básicas de un proceso de desarrollo de software. Los resultados obtenidos fueron:

Entendible

El proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos resultó para el total de los encuestados entendible en un 83%. Se demostró que con una buena planificación y definición del ciclo de vida del software los líderes de proyecto guían mejor la construcción de estos ya que lograron comprenderlo en un 89%. Ningún grupo de entrevistados mostró datos negativos por encima del 50%, lo que nos asegura que la media de desarrolladores y personal involucrado en la producción de software comprende el proceso definido. En la figura 3.3 podemos visualizar los resultados.

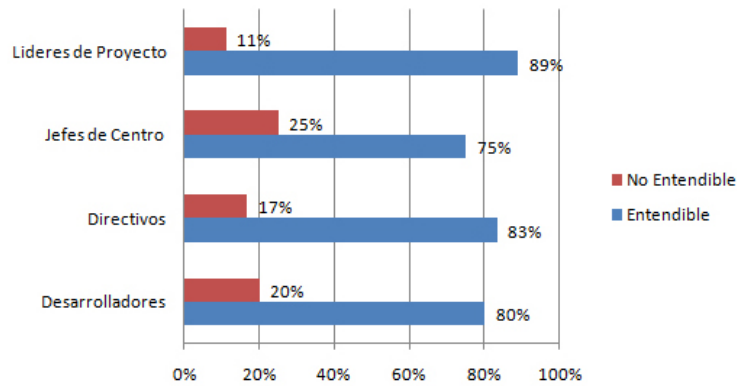


Figura 3.3: Análisis de Comprensión del Proceso de Desarrollo.

Fiable

La fiabilidad en el proceso de desarrollo propuesto fue evaluada de positiva por un 70 % del total de los encuestados, lo que nos demuestra que a pesar de necesitar una evolución en la etapa de mejora de proceso, más de la mitad plantea que el proceso está diseñado para que los errores puedan evitarse o prevenirse antes de que afecte la calidad del producto. En la figura 3.4 se pueden observar los resultados.

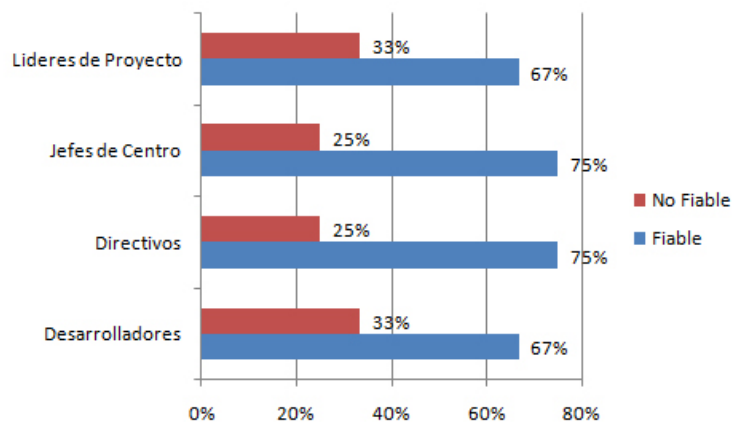


Figura 3.4: Análisis de Fiabilidad del Proceso de Desarrollo.

Visible

De un total de 40 evaluadores solo el 13 % considera que el proceso de desarrollo no es visible en todas sus etapas, es decir que no produce resultados claros en cada hito. Más específicamente se puede observar en la figura 3.5 que el 89 % de los líderes de proyecto consideran que el proceso de desarrollo propuesto genera resultados visibles, los desarrolladores en un 93 %, los directivos en un 83 % y los jefes de centro en un 75 %, para hacer un 88 % del total de encuestados (ver figura 3.5).

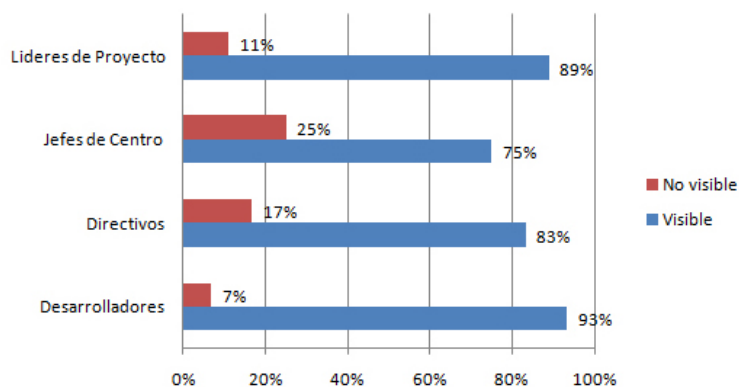


Figura 3.5: Análisis de Visibilidad del Proceso de Desarrollo.

Apoyado por Herramientas CASE

El proceso de desarrollo cuenta con una evaluación del 83 % en respuestas positivas en cuanto a la característica básica de ser apoyado por herramientas CASE. Los evaluadores consideran que en la propuesta de proceso de desarrollo se aprovechan las ventajas de las herramientas CASE y sobre todo se utilizan en beneficio para la obtención de una aplicación con calidad. Hay que tener en cuenta que las respuestas negativas no superan el 50 %, por tanto, puede concluirse que el proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos esta apoyado por las herramientas CASE. Ver figura 3.6 para mas detalles.

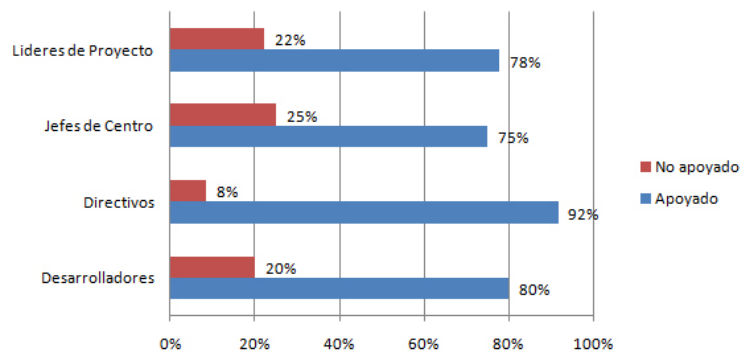


Figura 3.6: Análisis del Apoyo por herramientas CASE del Proceso de Desarrollo.

Rapidez

El 87 % de los desarrolladores entrevistados consideran que la propuesta de proceso de desarrollo agiliza la producción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos. A su misma vez, los jefes de centro resumen en un 50 % su evaluación, aclarando que no es el único factor que influye en finalizar en tiempo los productos después de tener las especificaciones del cliente. En la figura 3.7 se pueden observar más detalles.

Solo un 23 % del total considera que no se agiliza la construcción de aplicaciones siguiendo el proceso propuesto y teniendo en cuenta que es mucho menos del 50 % del total de entrevistados se resume que en gran medida influye en la rapidez de construcción.

Aceptabilidad

El proceso de desarrollo para lo construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos resultó para el total de los encuestados aceptable en un 75 %. Se demostró que los líderes de proyectos utilizarían una guía de planificación para desarrollar el proceso de desarrollo evaluando de positiva la aceptabilidad en un 89 %, los jefes de centro en un 50 %, los

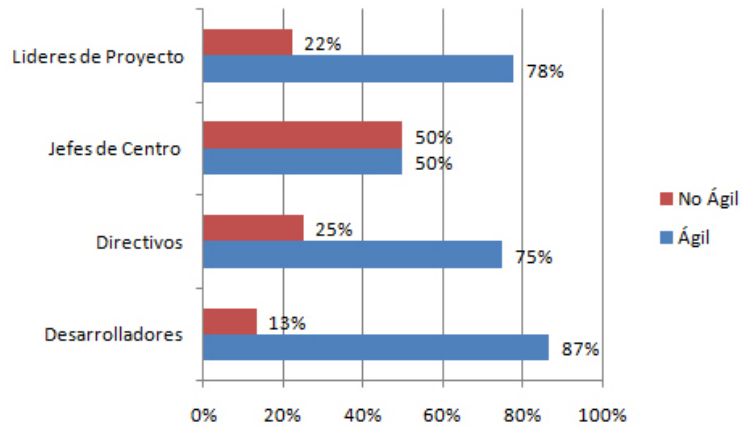


Figura 3.7: Análisis de la Agilidad del Proceso de Desarrollo.

directivos en un 58 % y los desarrolladores en un 87 % de su total.

En la figura 3.8 se puede observar cómo los valores negativos no sobrepasan el 50% en ninguno de los grupos de evaluadores, dando la aprobación total para plantear que el proceso de desarrollo propuesto cumple con la característica de ser aceptable por el equipo de desarrollo y personal involucrado en la producción de software.

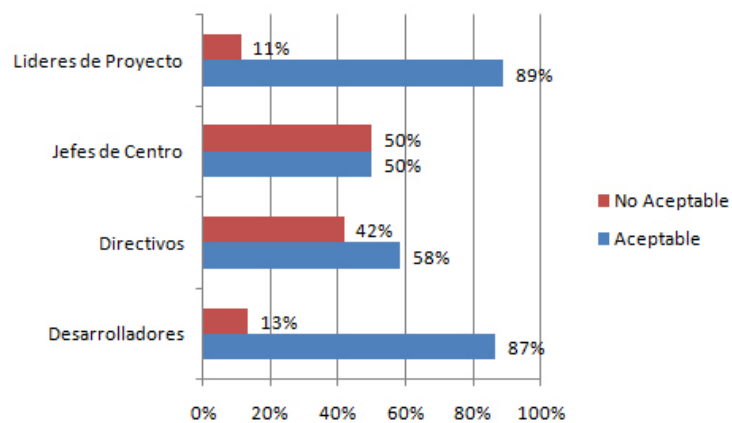


Figura 3.8: Análisis de la Aceptabilidad del Proceso de Desarrollo.

3.3. Análisis de los resultados de evaluación técnica del proceso de desarrollo

Para organizar la evaluación por el comité de expertos se calculó la cantidad de evaluadores utilizando la expresión:

$$n = \frac{p*(1-p)*k}{i^2}$$

Para los valores de $i=0.10$, $p=0.01$, $K=6.6564$.

El número de expertos consultados fue 7.

Posteriormente los expertos emitieron su juicio para darle peso a los indicadores. Se elaboró la tabla de los valores del peso relativo de cada criterio (Ver Anexo 28) y la tabla para el cálculo de la concordancia (Ver Anexo 29).

El resultado del cálculo fue:

$$X2_{real} < X2(\alpha, c - 1)$$

Si $1-\alpha=0.99$

Donde α es el error

$$\alpha=0,01$$

$$17,7888192 < 27,688$$

De la aplicación de esta prueba pudo concluirse que la concordancia en el juicio emitido por los expertos es significativa. Por último se construyó la tabla de calificación de cada criterio (Ver Anexo 30).

El Índice de Aceptación fue de 0,71357143.

$IA > 0,7$ Existe alta probabilidad de éxito.

$0,7 > IA > 0,5$ Existe probabilidad media de éxito.

$0,5 > IA > 0,3$ Probabilidad de éxito baja.

$0,3 > IA$ Fracaso seguro.

Por lo que la probabilidad de éxito es alta.

La mayoría de los encuestados les parece atractivo, de mucha actualidad e interesante, con posibilidades de aplicación en los proyectos con el mismo perfil de desarrollo dado que se definen elementos que deben ser tenidos en cuenta y buenas prácticas que deben ser generalizadas.

Algunos recomiendan su publicación, y su aplicación para confrontarse con la realidad de forma que se pudieran corregir las posibles insuficiencias, que la práctica, como criterio de la verdad, reclama. Plantean que el proceso de desarrollo propuesto puede contribuir al cumplimiento de plazos, presupuestos y exigencias de calidad como se expresa en el contenido.

Los encuestados recomiendan profundizar en temas del proceso de desarrollo propuesto tales como:

- **Métricas e indicadores:** identificar posibles métricas y/o indicadores que se deberían tener en cuenta para garantizar el éxito o fracaso del proceso de desarrollo definido.
- **Flexibilidad:** adaptar el proceso de desarrollo a cualquier proyecto que se dedique a la producción de software.
- **Cómo disminuir los costos de producción:** qué necesidades de organización se requieren para conseguirlo.

3.4. Elementos de implantación del proceso de desarrollo

A la hora de implantar el proceso de desarrollo en un proyecto real con el objetivo de medirlo y mejorarlo en caso de ser necesario, se propone el cumplimiento de una serie de elementos o pasos que garantizan la institucionalización del proceso, término acuñado en las prácticas generales de CMMI para el nivel 2.

Institucionalizar un proceso implica que el proceso está arraigado a la forma en que se realiza el trabajo y existe un compromiso y una consistencia para realizar el proceso. Las prácticas generales constan de 10 pasos los cuales se irán cumpliendo paulatinamente para implementar el proceso de desarrollo guiados por el Libro de Proceso de Desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

3.4.1. Prácticas Generales para CMMI nivel 2

Establecer una política organizacional

Se debe establecer y mantener una política de la organización para planificar y realizar el proceso de gestión integrada de proyecto. Los directivos del proyecto son los responsables de definir las expectativas de la organización para el proceso y hacer visibles estas expectativas a la organización.

Planificar el proceso

Se debe determinar que se necesita para ejecutar el proceso y alcanzar los objetivos establecidos, desarrollar una planificación para la ejecución del proceso y obtener la aceptación de la planificación por parte de los involucrados.

Establecer una planificación incluye la documentación de la planificación y la descripción del proceso, el mantenimiento de la planificación incluye su actualización para reflejar las acciones correctivas, los cambios a los requisitos o los cambios a los objetivos.

Proveer recursos

Se debe asegurar que los recursos necesarios para llevar a cabo el proceso definido en la planificación estarán disponibles cuando sean necesarios.

Asignación de responsabilidades

Indicar qué personas serán las responsables de que el proceso sea llevado de forma correcta conforme a lo planificado, estas personas tendrán que tener la autoridad suficiente y deberán aceptar dichas responsabilidades.

Formación del personal

Se debe asegurar que las personas posean los conocimientos necesarios para llevar a cabo la correcta ejecución del proceso. Se deben realizar talleres y cursos para introducir los elementos que conforman el proceso de desarrollo y crear conciencia referente a la importancia de su aplicación.

La formación da soporte a la realización satisfactoria del proceso estableciendo una comprensión común del proceso e impartiendo las habilidades y conocimientos necesarios para realizarlo.

Gestionar la configuración

El propósito de esta práctica genérica es establecer y mantener la integridad de los productos de trabajo designados del proceso a lo largo de su vida útil.

Se debe incorporar las buenas prácticas del área de proceso de CMMI que está definiéndose como parte del proceso de mejora para mantener la integridad de los datos y del producto. Se van a establecer diferentes niveles de control, se contará con políticas de seguridad de la información y con un repositorio donde se localizará la última versión del producto de trabajo en uso, al igual que otros documentos de interés para el proyecto.

Identificar e involucrar a las personas previstas

El propósito de esta práctica genérica es establecer y mantener la participación prevista de las partes interesadas durante la ejecución del proceso.

Monitorizar y controlar el proceso

Se debe ejecutar la monitorización y seguimiento del proceso, para tomar las medidas correctivas en caso que sea necesario. La monitorización se ejecutará principalmente mediante la creación de indicadores. Sin embargo, no existen indicadores definidos en la universidad para evaluar la calidad del proceso de desarrollo a corto plazo. En el proyecto se debe analizar las métricas que se desean medir, se recomienda que sean sobre los errores detectados antes de la entrega del software, los defectos detectados, el esfuerzo humano y el tiempo consumido.

Evaluar objetivamente su cumplimiento

El objetivo principal es asegurarse que el proceso se está ejecutando según lo planificado y de forma correcta con lo establecido en el proceso organizacional.

Revisar el estado con los superiores

Consiste en hacer el reporte del estado del proceso a los mandos superiores. Revisar con el nivel directivo las actividades, el estado y los resultados del proceso de gestión integrada de proyecto, y resolver los problemas.

3.5. Conclusiones Parciales

En el presente se propone una implantación orientada a la institucionalización del proceso de desarrollo guiado por las prácticas generales propuestas por CMMI para el nivel 2.

Se analizaron los resultados que se obtuvieron de la aplicación de los métodos, identificando los problemas en la producción de software en la UCI y demostrando las deficiencias existentes en la producción, particularmente los errores en el conocimiento de los flujos de trabajo, los roles y las responsabilidades.

Se realizó la evaluación técnica del modelo a través de un comité de experto que arrojó una probabilidad de éxito alta.

El modelo contribuyó a la definición del flujo de proceso, los roles y las responsabilidades a partir de una metodología.

Conclusiones

El proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos:

- Contribuyó a la definición de las fases, los roles y las responsabilidades a partir de una metodología.
- Mejoró la planificación y conocimiento de las actividades a partir del Libro de Proceso de Desarrollo.
- Contribuyó a realizar procesos organizados, estructurados y disciplinados, donde el proyecto tenga capacidad de repetirlos y mejorarlos.

Se realizó la evaluación técnica del modelo a través de un comité de expertos que auguró una probabilidad alta de éxito. Se propuso un método de implantación del proceso de desarrollo para la construcción de Aplicaciones Enriquecidas de Internet que incluyan escenarios tridimensionales interactivos.

Por tanto se concluye que el objetivo se cumple.

Recomendaciones

Los objetivos del trabajo no abarcan todos los elementos a definir; los cuales son amplios y diversos. Por lo que se propone:

- Definir posibles métricas y/o indicadores que se deberían tener en cuenta para garantizar el éxito o fracaso del proceso de desarrollo definido.
- Aplicar, evaluar y revisar la implantación del proceso de desarrollo en proyectos reales.
- Realizar un estudio de factibilidad económica que cuantifique la factibilidad de la propuesta.

Bibliografía

- [1] Adobe Systems Incorporated. Adobe systems incorporated. <http://www.adobe.com/es/resources/business/richinternetapps/>, 2011.
- [2] K. Beck and C. Andres. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2004.
- [3] B. Boehm, A. Egyed, J. Kwan, D. Port, A. Shah, and R. Madachy. Using the winwin spiral model: A case study. *Computer*, 31(7):33–44, 2002.
- [4] A. Brito. *Blender 3D-Guia do Usuário*. Novatec Editora, 2006.
- [5] J Butler. *Rapid Application Development in Action*. 1995.
- [6] M.E. Ciolli. del documento: Testing de migración de aplicaciones distribuidas a entornos web. 2007.
- [7] Inc. CollabNet. <http://subclipse.tigris.org/>, 2011.
- [8] M.J. Davis. Process and Product: Dichotomy or Duality? *ACM SIGSOFT Software Engineering Notes*, 20(2):18, 1995.
- [9] R. FIGUEROA, C. Solís, and A.A. Cabrera. Metodologías tradicionales vs. metodologías ágiles. *Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación*.(En línea), Disponible en: <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato.doc>.

- [10] M. Fowler. Is design dead? *SOFTWARE DEVELOPMENT-SAN FRANCISCO*, 9(4):42–47, 2001.
- [11] Rossi Gustavo y Sanchez-Fiegueroa Fernando Fraternali, Piero. Rich Internet Applications. *IEEE Computer Society*, 2010.
- [12] Eclipse Foundation. <http://www.eclipsefaq.org/>, 2011.
- [13] Elisa Gallo. <http://www.esi.es/Berrikuntza>, 2006.
- [14] V.S. Hernández. La Industria Del Software. Estudio A Nivel Global Y América Latina. *Observatorio de la Economía Latinoamericana*, (116), 2009.
- [15] M. Hirsch. Making RUP agile. In *OOPSLA 2002 Practitioners Reports*, page 1. ACM, 2002.
- [16] IBM. <http://www-01.ibm.com/software/awdtools/developer/rose/enterprise/index.html>, 2011.
- [17] Adobe Systems Incorporated. <http://www.adobe.com/products/flex/overview/framework>, 2011.
- [18] Project Management Institute. A guide to the project management body of knowledge: PMBOK Guide. Project Management Institute Newtown Square PA, USA, 2004.
- [19] I. Jacobson, G. Booch, and J. Rumbaugh. *El proceso unificado de desarrollo de software*. Addison Wesley, 2000.
- [20] C. Larman. *Agile and iterative development: a manager's guide*. Prentice Hall, 2004.
- [21] B. Laurie and P. Laurie. *Apache: the definitive guide*. O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2002.
- [22] DrC. Rolando Alfredo Hernández León. Curso básico de gestión de proyectos. 2005.

- [23] C. Lozano and P. Ramiro. Análisis comparativo de los Frameworks Adobe Flex, Java Rich Faces y Extjs para el desarrollo de aplicaciones enriquecidas en internet (RIA). 2010.
- [24] J. Martin. *Rapid application development*. Macmillan Publishing Co., Inc. Indianapolis, IN, USA, 1991.
- [25] S. McConnell and Inc Books24x7. *Code complete*, volume 2. Microsoft Press Washington, 2004.
- [26] J.A. McDermid. *Software engineer's reference book*. Butterworth-Heinemann Newton, MA, USA, 1991.
- [27] B. Momjian. *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley, 2001.
- [28] C. Mook. *Essential actionscript 3.0*. O'Reilly, 2007.
- [29] AB MySQL. <http://www.mysql.com/why-mysql/>, 2006.
- [30] O.Ñierstrasz, S. Gibbs, and D. Tschritzis. Component-oriented software development. *Communications of the ACM*, 35(9):160–165, 1992.
- [31] T. O'Reilly. What is web 2.0. *Design patterns and business models for the next generation of software*, 30:2005, 2005.
- [32] Visual Paradigm. <http://www.visual-paradigm.com/product/vpuml/>, 2011.
- [33] Roger Pressman. *Ingeniería de software: un enfoque práctico*. McGraw Hill, 2002.
- [34] T. Quatrani. *Visual modeling with Rational Rose 2000 and UML*. Addison-Wesley Longman Ltd. Essex, UK, UK, 2000.
- [35] Claudia García Romero. *El modelo de capacidad de madurez y su aplicación en empresas mexicanas de software*. PhD thesis, Ingeniería de Sis-

- temas Computacionales. Departamento de Ingeniería en Sistemas Computacionales. Escuela de Ingeniería, Universidad de las Américas- Puebla, 2001.
- [36] K. Schwaber and M. Beedle. *Agile software development with Scrum*, volume 18. Prentice Hall Upper Saddle River, NJ, 2002.
- [37] SIDAR. <http://www.sidar.org/>, 2011.
- [38] Magic Software. <http://www.magicsoftware.com.ar/novedades.php>, 2009.
- [39] I. Sommerville. *Ingeniería del software*. Pearson Educación, 2005.
- [40] P.M.I.G. Standard. Guía de los Fundamentos de la Dirección de Proyectos. *Norma Nacional Americana ANSI/PMI*, pages 99–001, 2004.
- [41] Eduardo A. Rodríguez Tello. Unified modeling language. <http://www.tamps.cinvestav.mx/ertello/swe/sesion03.pdf>, 2008.