

Estructura organizativa y metodología para el desarrollo de software en la “Fábrica de Software” de la ESPE Extensión Latacunga”

López Chico Ximena del Rocío

Montaluisa Pilatásig Edgar Fabián

Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército Extensión Latacunga

Resumen.- *La Ingeniería de Software establece el proceso que guía el desarrollo del producto software, el mismo que es aplicado por las empresas dedicadas a la industria del software en las que se cuentan con unidades operativas dedicadas al soporte en Tecnologías de Información. La ESPE Extensión Latacunga cuenta en su estructura con la unidad de Tecnologías de Información y Comunicación UTIC's, la que se encarga de proporcionar el soporte para el correcto funcionamiento de los sistemas de información y la implementación de nuevos requerimientos.*

En este artículo se presenta la propuesta organizacional y metodología de desarrollo de software, que establece el marco de referencia para satisfacer los requerimientos internos y externos de la institución. Los resultados alcanzados en el estudio permiten concluir que su utilización es viable.

Abstract.- *Software engineering establishes the process that guides the development of the software product, the same being applied by companies engaged in software industry which have operational units dedicated to the support in Information Technology. The ESPE Extension Latacunga on its structure with the unit of Information Technology and Communication UTIC's, which is responsible for providing support for the proper functioning of information systems and the implementation of new requirements.*

In this article the organizational proposal and software development methodology, which provides the framework to meet internal and external requirements of the institution is presented. The results obtained in the study to conclude that its use is viable.

Palabras claves—Ingeniería Software, Software Engineering, Software Industry, Software - Methodology

I. INTRODUCCIÓN

La Ingeniería de Software proporciona los lineamientos necesarios para guiar el proceso de desarrollo de software (Pressman, 2002), los mismos que han ganado importancia al insertarse las Tecnologías de la Información y la Comunicación en las empresas o instituciones gubernamentales en el Ecuador (Constitución de la República del Ecuador, 2008).

Estas tecnologías han propiciado que podamos realizar a través de internet parte o la totalidad de los procesos de negocio, tales como:

- Comprar o vender un producto.
- Recibir la factura de los servicios básicos domésticos.
- Consultar a un especialista médico sobre un determinado tratamiento en el ciclo de una enfermedad.
- Comunicarnos de forma visual con una persona, entre otras.

Este tratamiento que se da a la información ha marcado las capacidades productivas de la era moderna, y Ecuador no está al margen de este desarrollo, pues así se evidencia en la inclusión de la tecnología en las empresas públicas. (Ministerio de Coordinación de la Producción, Empleo y Competitividad, 2010)

En la actualidad nos vemos inmersos en la era de la información y el conocimiento, de tal forma que la economía, cultura y bienestar social se desarrollan alrededor de estas nuevas tecnologías (Unesco, 2002). La guía para el desarrollo del país se evidencia en este entorno, entre otras cosas hablamos de la educación en escuelas del milenio, el desarrollo de nuevas tecnologías de información (Ministerio de Coordinación de la Producción, Empleo y Competitividad, 2010); dejando de lado los viejos pilares de la economía, tierra y capital, propiciando en la actualidad el desarrollo del capital intangible, el capital humano.

Las pequeñas y medianas empresas que se han desarrollado influenciadas por el mercado a través del tiempo, no conocen su entorno, pues no lo planifican sino reaccionan con la demanda. Esto ha ocasionado que se pretenda que un profesional cumpla al mismo tiempo los papeles de diferentes actores del entorno inmediato a la carrera de su formación. Así lo evidenciamos un sin número de veces, pues nos encontramos con publicidad en donde se requiere llenar una vacante para un área específica de software, de sistemas, de desarrollo, o de tecnología de información, en donde a más de poseer la destreza principal del requerimiento y de formación profesional se exigen destrezas complementaria de otro perfil.

Cuando la división de trabajo no está caracterizada, magnifica uno u otro título profesional, y no es factible actuar

dentro de la administración de talento humano considerando sus habilidades, conocimiento y capacidades (Idalberto, 2008)

En este artículo se presenta el estudio realizado en dos etapas, la primera presenta la División del Trabajo en donde se analiza la información de la situación actual, se presenta el diseño organizacional y la Metodología de desarrollo, y como segunda etapa se describe la evaluación de la propuesta, por elementos exógenos al trabajo práctico efectuado.

II. DIVISIÓN DE TRABAJO

A. Análisis de la información

Las Instituciones y Empresas localizadas en el entorno inmediato y mediato (en la ciudad, en la región 3 o Unidades Militares) solicitan en forma frecuente apoyo para desarrollo de productos software, sin embargo la ESPE Extensión Latacunga, no posee el recurso humano técnico para afrontar este pedido. Analizando la situación actual podemos evidenciar que en el manual de puestos no existe la estructura para responder a estos requerimientos, y que por encontrarse en un lugar geográfico distinto debería contar con el personal que en los manuales se describen. Al contrario de esto la Extensión ha sido y es considerada como una oficina de atención para soporte.

El análisis de la información nos conlleva a afirmar lo siguiente:

- En el manual de puestos de la ESPE se encuentran descritos varios puestos que no existen en la Extensión.
- No existe un presupuesto con partidas asignadas para profesionales que ocupen los puestos de trabajo descritos en el manual.
- El requerimiento de personal para el desarrollo de aplicaciones de software no es permanente, sin embargo las solicitudes de requerimientos son frecuentes.
- La situación legal laboral y la generación de capital intangible no puede ser considerado como un gasto.
- La extensión tiene entre sus programas carrera la de Ingeniería de Software y ella tiene entre sus procesos de formación las pasantías, prácticas profesionales, transferencia de Tecnología, Vinculación entre otros.

Para lograr la atención requerida es necesario diseñar la división de trabajo del desarrollo de productos software, sin descuidar la Ingeniería de Software, el aspecto legal y la naturaleza propia de institucional.

B. Diseño de la Organización Estructural

En el caso de estudio, la ESPE Extensión Latacunga a sufrido cambios durante su desarrollo, afrontando un crecimiento significativo de puestos de trabajo, cada uno con un sistema computacional, la implementación de conexiones de red, requerimientos de software y sobre todo el cambio del mercado donde desarrolla sus actividades.

Para diseñar la estructura organizacional se partió de conocer la situación de la organización y su unidad operativa de desarrollo, denominada Unidad de Tecnologías de Información y Comunicación (UTIC's), sus prácticas, sus

resultados y sus experiencias.

En la institución se analizó, como los usuarios valoran la acción de la UTIC's, la legislación de pasantías para la formación de Ingenieros de Software y Tecnólogos en Computación (ESPE, 2008), la legislación de generación de puestos de trabajo y compromisos contractuales, la legislación de enfermedades profesionales y del mercado (IESS, 2010), las solicitudes de acción en esta temática (demanda) por parte de empresas e instituciones de la región.

Todo lo mencionado nos permite proponer la Estructura Organizacional asociada a los profesiogramas de cada uno de los puestos, habiéndose creados los siguientes:

- Organigrama Estructural.
- Organigrama Funcional.
- Organigrama Posicional
- Profesiogramas

La estructura mantiene una fuerte relación con la Metodología de desarrollo propuesta, y descrita en el siguiente apartado.

En la figura 1, se presenta el Organigrama Estructural, en el cual constan los niveles de dirección, asesoría, y operativos necesarios para soportar un crecimiento dinámico, proporcionando el marco de trabajo para los grupos de proyectos y la aplicación del proceso metodológico para el desarrollo del producto software.

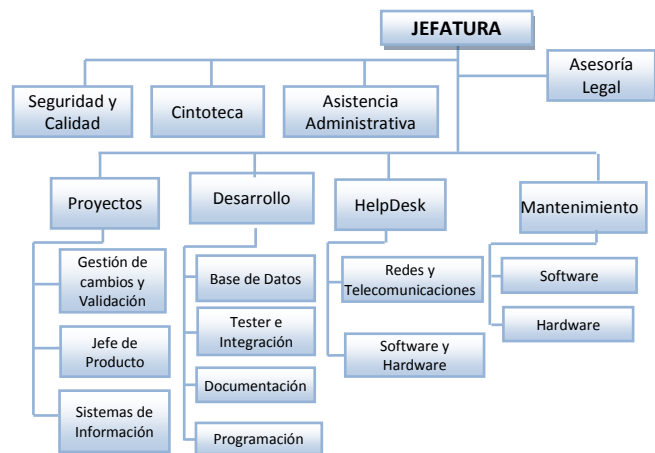


Fig. 1. Organigrama Estructural "Fabrica de Software"

Se han considerado aspectos como: la dirección de proyectos, soporte administrativo, soporte legal, aseguramiento de la calidad, la adecuada conservación de la información generada, el desarrollo del proyecto en su conjunto, y el soporte previo y posterior al producto desarrollado para el cliente.

El Organigrama Funcional descrito en las figuras 2.a. y 2.b. definen las actividades inherentes a cada puesto de trabajo que se propone en esta estructura. Estas actividades describen de forma específica el campo de acción de cada uno de los

profesionales que deban actuar bajo el rol asignado de acuerdo al desarrollo del proyecto. (Lopez & Montaluisa, 2015)

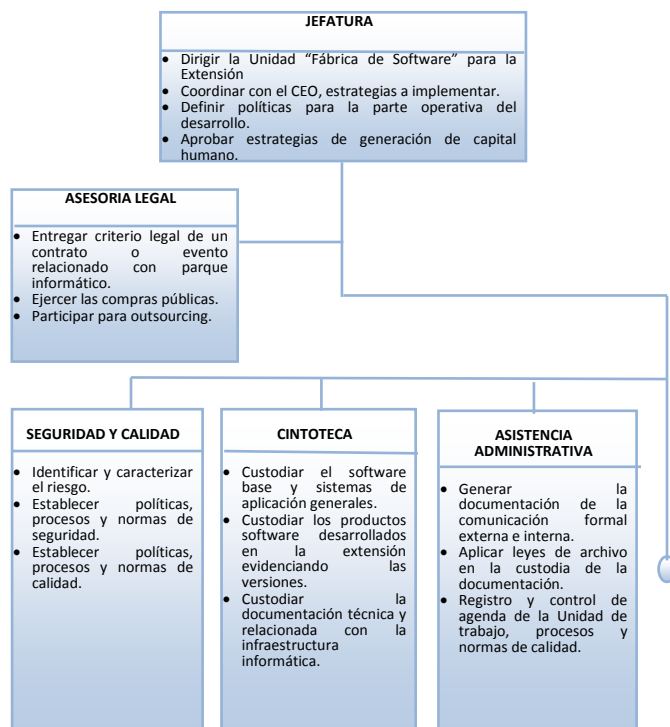


Fig. 2.a. Organigrama Funcional de la "Fábrica de Software"

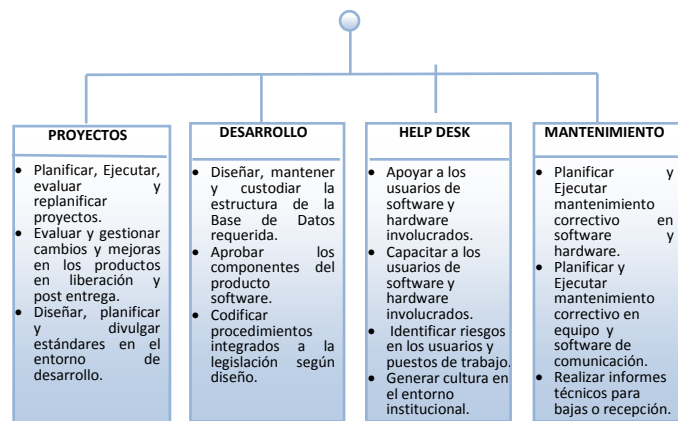


Fig. 2.b. Organigrama Funcional de la "Fábrica de Software"

Considerando que el organigrama posicional es un aspecto que depende de la alta gerencia, se excluye de este artículo pero se puede encontrar su descripción en el repositorio digital de ESPE en (Lopez & Montaluisa, 2015), de la misma forma que los profesiogramas.

Se considera para el orgánico posicional de la "Fábrica de Software", su característica dinámica, de tal forma que permita reclutar el personal para el puesto requerido, y únicamente algunos puestos sean considerados fijos. Se vigila la equidad de género y que no exista sobredimensionamiento del perfil.

La división de trabajo se lo expresa utilizando los profesiogramas diseñados para cada puesto de trabajo, su objetivo es representar en forma íntegra la definición del puesto de trabajo, así como su misión, el estándar básico de las competencias institucionales, conductuales y cognitivas contratables por la institución en una persona que cumpla determinados indicadores que conlleven a un alto desempeño laboral y evolución del capital humano. Adicionalmente se ha incluido un área para definir los riesgos laborales generales a los que día a día este profesional está expuesto, así como las enfermedades profesionales que puede sufrir y detallar los exámenes médicos considerados como requisito para el ingreso. (Lopez & Montaluisa, 2015)

En torno a los riesgos laborales que se registran, se enmarca dentro de la ley: El artículo 326 numeral 5 de la constitución que dice "toda persona tendrá derecho a desarrollar sus labores en un ambiente adecuado y propicio, que garantice salud, integridad, seguridad, higiene y bienestar". Se registran los riesgos percibidos en cada uno de los puestos de trabajo en cumplimiento del artículo 12 de la Resolución 390 (IESS, 2010). Los mismos que se establecen en la Resolución C.D. No. 333 Artículo 9 - 4.2 (IESS, 2010), y en los requisitos básicos del reglamento interno de seguridad y salud de trabajo en el capítulo IV artículos del 19 al 24.

Los exámenes médicos son aquellos que se hallan relacionados con enfermedades laborales a cada uno de estos puestos de trabajo, los mismos que deben realizarlos al ingreso a la dependencia patronal y periódicamente para mantener la salud laboral en cumplimiento del artículo 12 y 14 de la Decisión 584 del Instrumento Andino de Seguridad y Salud en el Trabajo (IESS, 2010), así como en los requisitos básicos del reglamento interno de seguridad y salud de trabajo en el capítulo VII artículo 29 literal a y b.

C. Metodología de Desarrollo

La metodología de desarrollo software organiza el proceso como un conjunto de fases que de forma ordenada proporcionan los lineamientos y el entorno adecuado para el desarrollo del producto software (Pressman, 2002). La propuesta consta de varios procesos que orientan el desarrollo del producto software, el primer proceso que se ejecuta al iniciar el proyecto y que no deberá exceder de 3 días, el segundo un proceso iterativo el cual dará como resultado el desarrollo de versiones de software con operatividad autónoma que tendrá una duración de 2 a 4 semanas, y el tercer proceso donde el usuario final podrá usar el sistema para desarrollar sus actividades administrativas y se proporcionará la capacitación en un tiempo total de 3 a 5 días.

Las fases de la Metodología de desarrollo de Software propuesta se esquematizan en la figura 3 y son las siguientes:

- Formalización de la Necesidad
- Requerimientos
- Diseño
- Implementación
- Pruebas
- Puesta en Marcha

La metodología propicia el desarrollo en un entorno en el cual los requerimientos se presentan de forma urgente y en la

cual el contacto con el dueño del proceso suele ser dificultoso por cuestiones de la organización, los procesos internos y la urgencia de contar con la herramienta de software que ayude a acelerar el trabajo de forma automatizada.



Fig. 3. Metodología de desarrollo de la “Fábrica de Software”

1) Formalización de la Necesidad

La fase de Formalización de la Necesidad constituye el primer proceso de la metodología que se ejecuta por única vez al inicio del proceso metodológico.

Esta fase determina el inicio del proyecto, propiciada por la iniciativa de desarrollo de una aplicación de software como respuesta a una necesidad. Considerando las circunstancias que rodean el entorno en el cual se originan estas necesidades, en esta fase se establece el inicio del proyecto de desarrollo mediante una disposición que es propiciada por la unidad operativa requirente, esta necesidad se modela mediante un pedido formal que se denomina “Formalización de la Necesidad”. La Formalización de la Necesidad establece los responsables y una primera indagación de lo requerido, citando las leyes, reglamentos y procesos establecidos que respalden dicha necesidad.

La Formalización de la Necesidad tendrá como finalidad dar inicio formalmente al proceso de desarrollo de software, comprometer a los actores a aportar en todas las fases requeridas para el desarrollo, e identificar el entorno en el cual se desarrollará y se ejecutará la aplicación de software desarrollada.

En esta fase se establecerá un cronograma de entregas parciales que determinará el alcance del proyecto y de las versiones de software a desarrollarse, estas versiones tendrán autonomía y que en su conjunto conformarán la aplicación de software resultante del proyecto.

Se propicia garantizar los compromisos adquiridos tanto por parte del desarrollador como del requirente.

2) Requerimientos

La especificación de requisitos de software describe la necesidad expresada por el usuario como una unidad capaz de ser modelada y convertida en un componente del producto software (IEEE, 830-1998). La fase de Requerimientos es la

segunda fase de este proceso metodológico, y se constituye como un componente que es parte del proceso repetitivo para el desarrollo de las versiones de software. Esta fase se ejecuta cada vez que se repita el proceso de desarrollo de una nueva versión, en esta fase se detalla la descripción formal de los requerimientos logrando plasmar el alcance del sistema.

Se considera en la metodología que el desarrollo del proyecto se ejecuta en un entorno complejo que dificulta la obtención de los requerimientos y debido al corto tiempo que se establece para el desarrollo de los proyectos se propone la esquematización de historias de usuarios (ScrumManager, 2014) con la variante que el técnico responsable de la elaboración del sistema será el encargado de recopilar y describir en el formato la información necesaria para detallar el requerimiento y luego deberá confrontarlo con el usuario responsable del proceso a fin de legalizarlo. Para esta fase deberá aprovechar al máximo el tiempo disponible del personal operativo responsable de los procesos y recopilar la información necesaria aplicando técnicas como:

- Entrevistas al personal involucrado con los procesos a automatizar.
- Recopilación de documentos de apoyo y formatos que hayan sido establecidos para el trabajo diario.
- Observación de reglamentos e instructivos desarrollados para la ejecución de proyectos que están relacionados con el requerimiento

3) Diseño

La fase de diseño constituye la tercera fase y es parte del proceso iterativo de desarrollo de la aplicación de software, su ejecución se refleja en los modelos producto de los requerimientos establecidos para la versión de software en ejecución.

El proceso de diseño contemplará la esquematización de los requerimientos agrupados por funcionalidades, por consiguiente generará modelos parciales que provean funcionalidad independientes. La fase de diseño considerará los siguientes aspectos que deberán ser modelados:

Diseño de Datos.- El diseño establecerá el modelo de datos que dará soporte al sistema en desarrollo, se realizará mediante la aplicación de software relacionado directamente con el DBMS que se haya definido para el proyecto. Por ejemplo para MySQL se usará MySQLWorkbench.

Diseño de Interfaces.- El Diseño de interfaces se establecerá de acuerdo al Framework a utilizarse, se realizará un único diseño base para todas las interfaces de mantenimiento de las tablas en la cual se establecerá el posicionamiento para:

- Texto en el formulario
- Cajas de entrada de datos (texto, combos, casillas de selección, etc.)
- Botones de comando y enlaces de funcionalidad
- Gráficos y logos

Se personalizará aquellas interfaces que requieran de complejidad en la implementación y que difieren del esquema general de entradas y presentación de datos. El diseño de interfaces determina en forma gráfica, mediante esquemas, la

distribución y operatividad deseada de la interface, la que concordará con los requerimientos establecidos.

4) Implementación

La fase de Implementación es la cuarta fase del proceso iterativo de desarrollo y se enfocará en obtener código de forma rápida, con la mayor fiabilidad posible y con un alto grado de calidad.

Para lograr un estilo homogéneo y eficiente se deberá observar los diseños establecidos de interfaces y se utilizarán técnicas de programación que permitan acelerar la generación de código, tal como Frameworks (Mark Safronov, 2014) que posibiliten generar código fuente con una arquitectura de software MVC (Modelo Vista Controlador).

El objetivo al integrar el Framework y la arquitectura MVC es permitir un desarrollo rápido de cada módulo y las versiones establecidas en el cronograma de entregas, pasando de la necesidad al producto software de la forma más rápida posible y la entrega al usuario final luego de las pruebas y validaciones necesarias. La implementación generará versiones de software que serán autónomos y fácilmente acoplables al producto final.

El proceso para la generación de código establece seguir varios pasos de forma metódica, los que tomando como insumo los diseños de datos e interfaces generarán el código:

- Creación de Tablas de Datos en el DBMS
- Uso del generador de código
- Personalización del Modelo
- Personalización de la Vista
- Personalización del Controlador
- Pruebas Unitarias (Funcionalidad del código)

Para la implementación, así como para el resto de fases se recomienda el uso de herramientas de software dentro del marco que regula a las instituciones públicas. El decreto 1014 dispone el uso del Software Libre (Administración Pública, 2008), y en este marco podemos usar herramientas de desarrollo como NetBeans, así como herramientas de gestión de versiones como SubVersion (SVN).

5) Pruebas

La fase pruebas es la quinta fase y es parte del proceso iterativo de desarrollo. Una vez finalizada la generación del código se realizarán pruebas del software a fin de garantizar la calidad mediante la concordancia entre los requerimientos, el diseño y el producto resultante (Tuya, 2014).

Las pruebas se establecerán en 2 puntos a fin de minimizar los fallos residuales en el software:

Pruebas de Desarrollo.- Las realizará el técnico a cargo del desarrollo y se basa en examinar de forma detallada los objetos creados y su funcionalidad. Propicia la eliminación de errores a nivel de código la funcionalidad individual del módulo.

Pruebas de Función.- Las realizará un técnico a fin de encontrar residuos de fallos que no han sido detectados y también establecerá la correcta relación entre los módulos desde el punto de vista funcional del software, de esta manera

se asegura que la versión de software sea un elemento entregable al usuario final. En esta evaluación se contrapondrá los requerimientos establecidos para la versión de software contra el desempeño de la aplicación desarrollada. Conseguiremos entonces una versión que pasará a estado de producción y podrá ser implantada.

6) Puesta en Marcha

La fase de puesta en marcha se ejecutará al finalizar cada una de las versiones de software planificadas. La ejecución de esta fase requiere la puesta en marcha de la versión de software, realizando paralelamente la capacitación sobre las funcionalidades incluidas en la determinada versión. (Pressman, 2002)

Se orientará al usuario final a utilizar las funcionalidades implementadas y se determinará el grado de satisfacción sobre los requerimientos. Se realizará un Acta de Conformidad en la cual se indicará las funcionalidades implementadas en la versión y se registrarán las novedades suscitadas en la capacitación. Mientras se avanza en el proceso de Puesta en Marcha por parte de un técnico, el grupo de trabajo iniciará inmediatamente a trabajar en la nueva versión. Los fallos residuales detectados en la Puesta en Marcha serán incluidos como parte de la nueva versión a fin de viabilizar de forma fluida el desarrollo del software.

7) Artefactos y Manuales

Para lograr la operatividad de la metodología se plantea un conjunto de artefactos de apoyo, que permitirán documentar el proceso de desarrollo e integrar el esfuerzo de las personas involucradas en el proyecto. Estos artefactos se establecerán en cada una de las fases de la metodología y se encuentran detallados en el repositorio digital de la ESPE en Lopez&Montaluisa.

Para poder organizar los entregables que se definen en la metodología se ha establecido un árbol de directorios que contendrán cada uno de los entregables de forma organizada y permiten que todos los miembros del equipo de desarrollo sean capaces de encontrar fácilmente los artefactos que han desarrollado otros colaboradores.

- + Raíz [Repositorio de Proyectos de Desarrollo]
- Proyecto [Nombre Corto del Proyecto]
- Entregables [Documentos de Texto y Modelos]
- + Código [Versiones de Código Fuente]
- Sistema Versión 1.0 [Primera Versión]
- Sistema Versión 2.0 [Segunda Versión]
- Manuales [Versiones de Manuales]

III. EVALUACION DE LA PROPUESTA

Para evidenciar la usabilidad de la metodología propuesta se realizó una valoración de la viabilidad y flexibilidad de la misma, considerando un caso de prueba aplicado por los autores y la interacción de actores externos al grupo de desarrollo.

A. Caracterización de evaluadores

La población que participó en la evaluación fueron

personas involucradas en el desarrollo de software a medida, es decir personas que conocen y aplican diferentes metodologías dentro de una división de trabajo acorde a la exigencia que ha vivido nuestro país. Así como personas que conocen de la división de trabajo para lograr un ordenamiento departamental jerárquico y con acción de responsabilidad por proyecto.

B. Propuesta

La propuesta de trabajo está orientada en la hipótesis planteada y en las variables dependientes e independientes, identificadas en su momento, las mismas que se describen en la Tabla I.

TABLA I. VARIABLES DE LA PROPUESTA EN EVALUACIÓN

Hipótesis	Variables	Atributos	Valor
Es posible desarrollar la "Fábrica de Software" mediante el diseño organizacional y establecimiento de la Metodología de desarrollo.	Estructura Organizacional	Dinámica	Número de proyectos implementados por los creadores de la propuesta
		Viabilidad	
	Metodología de desarrollo	Flexibilidad	Número de proyectos implementados por personal diferente a los creadores de la propuesta

C. Artefactos de evaluación

La propuesta se evaluó mediante el análisis de los atributos de Viabilidad y Flexibilidad. Para el proceso de evaluación de la viabilidad de la metodología los creadores de la propuesta implementaron un proyecto de desarrollo de software. Para la evaluación de la flexibilidad se generan tres artefactos de evaluación, adaptados desde el Test de Usabilidad de Lewis y Sauro 2006, que permiten conocer el grado de aplicabilidad del proceso de desarrollo propuesto y sus artefactos. Además, se considera que la propuesta es flexible si ésta permite su implementación en al menos un proyecto distinto al que se utilizó en la evaluación de la viabilidad.

Por último se interpreta como la metodología se ajusta a la organización dinámica propuesta, sin existir gasto de capital innecesario en recurso humano y congruente a la demanda que satisfacemos.

D. Procedimiento

A diferentes grupos de desarrolladores se les entregó la documentación de la metodología, la que utilizaron para desarrollar un producto software. Posteriormente, cuando ya se desarrolló el producto software con el uso de la metodología, se efectuó una reunión de trabajo, en la cual se expone el trabajo efectuado por los desarrolladores, y ellos deben asignar un valor a cada una de las preguntas que se encuentran en los artefactos desarrollados para la evaluación.

E. Resultados

Como resultados de las encuestas contestadas por los desarrolladores, se evidencian los siguientes resultados:

1) VIABILIDAD

Los evaluadores discurren:

- Que la metodología contribuye al desarrollo del software.
- Ayuda a la validación por parte del usuario
- Existe comunicación por intermedio de los artefactos.
- Permite una planificación adecuada.
- Permite incrementar el capital intangible en marca y madurez

Teniendo como resultado que la propuesta es viable, ya que la aplicación de la metodología permite instanciar el proyecto de software "Proyecto INNOVATE"

2) FLEXIBILIDAD

El personal desarrollador adopta como suya la metodología presentada, por considerarla entendible y de fácil aplicación, sus instrumentos tienen coherencia y pertenencia.

- Consideran que la documentación es la necesaria.
- No requiere de capacitación personalizada del autor.
- Los entregables son consistentes.

El análisis de los datos procesados usando SUS Calculation, nos permite demostrar que la propuesta es flexible, ya que nos permitió instanciar dos aplicaciones, obteniéndose una calificación aceptable que se representa con un valor mayor a 70, valores que podemos observar en la Tabla II.

TABLA II. RESULTADOS DE LOS INSTRUMENTOS – FLEXIBILIDAD

PARTICIP.	PREGUNTAS										CALIFICACIÓN
	1	2	3	4	5	6	7	8	9	10	
1	4	2	4	3	4	2	4	2	4	2	72,5
2	4	1	4	3	4	2	4	1	4	1	80,0
3	5	2	5	2	5	4	5	1	5	4	80,0

El rango de aceptación está definido por parámetros de la tabla III, sobre un valor de 100.

TABLA III. RANGO DE ACEPTACIÓN DE FLEXIBILIDAD.

LIMITES	ESTADO
<50	No Aceptable
[50, 70]	Marginal
>70	Aceptable

3) RELACIÓN METODOLOGIA - ORGANIZACIÓN DE TRABAJO

La relación entre la división del trabajo propuesto, considerando que las personas evaluadoras son aquellas que tienen su experiencia profesional relacionada a un escalafón estructural, con la metodología adquiere una calificación superior a 56 considerándose como aceptable, como se muestra en la tabla IV.

TABLA IV. RESULTADOS DE LOS INSTRUMENTOS – RELACIÓN METODOLOGÍA ORGANIZACIÓN

PARTICIPANTE	PREGUNTAS								CALIFICACIÓN
	1	2	3	4	5	6	7	8	
1	5	5	5	5	5	5	5	5	80,0
2	4	3	5	3	4	4	4	3	55,0
PROMEDIO									67,5

El rango de aceptación está definido por los valores de la tabla V, sobre una calificación de 80.

TABLA V. RANGO DE ACEPTACIÓN – RELACIÓN METODOLOGÍA ORGANIZACIÓN

LIMITES	ESTADO
<40	No Aceptable
[40, 56]	Marginal
>56	Aceptable

Los resultados obtenidos de la evaluación de la propuesta se muestran en la tabla VI.

TABLA VI. RESULTADOS DE LA EVALUACIÓN

Hipótesis	Variables	Atributos	Valor
Es posible desarrollar la “Fábrica de Software” mediante el diseño organizacional y establecimiento de la Metodología de desarrollo	Estructura Organizacional	Dinámica	Aceptable
		Viabilidad	1 proyecto Aceptable
	Metodología de desarrollo	Flexibilidad	2 proyectos Aceptables

Se concluye que la metodología está enmarcada en los atributos que permiten la implementación en relación con la división de trabajo propuesta.

RECONOCIMIENTOS

A la Universidad de la Fuerzas Armadas que nos permitió desarrollar este proyecto, fortaleciendo nuestro sentido crítico y conocimientos.

REFERENCIAS

- Administración Pública. (10 de Abril de 2008). *Software Libre Mandato 1014*. Obtenido de <http://www.administracionpublica.gob.ec/software-libre/>
- Constitución de la República del Ecuador. (2008). Comunicación e Información Título II Sección III arte 16-19. Ecuador.
- ESPE, P. (2008). *Manual de Pasantías de la ESPE*. Sangolquí.
- Idalberto, C. (2008). *Gestión de Talento Humano*. DFMexico, Mexico, Mexico: Elsevier Editora Ltda.
- IEEE. (830-1998). *IEEE Recommended Practice for Software Requirements Specifications*. Obtenido de <http://standards.ieee.org/findstds/standard/830-1998.html>
- IESS. (2010). *Normativas de Seguridad y Salud en el Trabajo IESS*.
- Lopez, X., & Montaluisa, E. (Febrero de 2015). *Diseño de la Estructura Organizativa e Implementación de una Metodología para el desarrollo de Software en la “Fábrica de Software” de la ESPE Extensión Latacunga*. ESPE.
- Mark Safronov, J. W. (2014). *Web Application Development with Yii 2 and PHP*.
- Ministerio de Coordinación de la Producción, Empleo y Competitividad. (2010). *Agenda para la Transformación Productiva*. Quito, Pichincha, Ecuador.
- Pressman, R. s. (2002). *Ingeniería del software: Un enfoque práctico* (Quinta ed.).
- ScrumManager. (2014). *Gestión de Proyectos Scrum Manager*.
- Tuya, J. (2014). *ISO/IEC/IEEE 29119 El nuevo estándar internacional para pruebas de software*.
- Unesco . (2002).
- Vidmar, R. J. (Agosto de 1992). *On the use of atmospheric plasmas as electromagnetic reflectors*. *IEEE Trans. Plasma Sci. [Online]*. 21(3). pp. 876–880. Obtenido de <http://www.halcyon.com/pub/journals/21ps03-vidmar>