

Método para el desarrollo de proyectos de software

Resultados preliminares

Lucy Nohemy Medina Velandia

Universidad Sergio Arboleda, Bogotá, Colombia, lucy.medina@usa.edu.co

Álvaro Escobar Escobar

Universidad Sergio Arboleda, Bogotá, Colombia, alvaro.escobar@usa.edu.co

Andrés Abel Arenas Prada

Universidad Sergio Arboleda, Bogotá, Colombia, andres.arenas@usa.edu.co

RESUMEN

La presente investigación versa sobre un método propio para desarrollar software a partir de las metodologías más reconocidas y se realiza desde dos enfoques distintos y complementarios:

El primero, relacionado con conocer las metodologías, métodos, arquitecturas, lenguajes, sistemas operativos y bases de datos que más utilizan las empresas desarrolladoras de software en Bogotá para construir sus proyectos y dar solución a problemas empresariales.

El segundo, dirigido a determinar el conocimiento y uso de las metodologías que los estudiantes de noveno y décimo semestre del Programa de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda, para saber si están seguros al iniciar su proyecto de grado de qué modelo o metodología pueden utilizar y, de no ser así, cómo los afecta.

Luego de organizados y analizados los resultados de los dos enfoques, se revisa la armonización de lo que los estudiantes deben haber aprendido y lo que necesitan las empresas. En consecuencia, será necesario, por una parte, adecuar el Plan de Estudios del programa de Ingeniería de Sistemas y Telecomunicaciones, y, por la otra, diseñar y construir un método que sea aplicable tanto por empresas como por estudiantes, en su gestión de desarrolladores de software.

En este documento se plasman no solo los resultados de la investigación que hasta la fecha se tienen, sino también la descripción de las actividades pendientes.

Palabras Claves: Métodos, Metodologías, Ingeniería de Software, Paradigmas, Calidad

ABSTRACT

The present investigation concerns a method for developing software itself from the most recognized and methodologies is made from two different and complementary approaches.

The first, related to known methodologies, methods, architectures, languages, operating systems and databases that use the software developer in Bogota to build projects and solving business problems. The second, to determine the awareness and use of methodologies that students in ninth and tenth semester of the Program of Systems Engineering and Telecommunications of the Sergio Arboleda University, to see if they are sure to begin your project model or what level of methodology can be used and, if not, how it affects them.

Then organized and analyzed the results of the two approaches, we review the alignment of what students should have learned and what business needs. Therefore, it will be a need to adapt the curriculum of the program of Systems Engineering and Telecommunications, and, secondly, to design and construct a method that is applicable both for businesses and students, in their management of developers software.

This paper will reflect not only the research results to date are, but also a description of pending activities.

Keywords: Methods, Methodologies, Software Engineering, Paradigms, Quality

1. INTRODUCCIÓN

Aunque Edsger Dijkstra fue el primero en mencionar el término Ingeniería de Software, quien lo hizo público fue Fritz Bauer en la Primera Conferencia sobre desarrollo de software celebrada en Garmisch, Alemania, en octubre de 1968. A partir de este año, crece la demanda de software cada vez más complejo, pero una vez se habían desarrollado suficientes programas, se empezó a evidenciar que surgían diversos problemas, como por ejemplo, el tiempo que se tomaba un desarrollo de software, los costos, los errores con que se entregaban las aplicaciones, el mantenimiento de los códigos, la falta de seguimiento en el desarrollo y la no elaboración de un cronograma. Todo lo anterior asociado a la inmadurez del sector informático el cual dependía casi en su totalidad sólo de la parte electrónica, además de no existir métodos claros ni mucho menos recursos; es a partir de todos estos problemas que Edsger Dijkstra denominara a esta época (entre 1965 a 1985) “La Crisis del Software”.

Los anteriores problemas fueron algunas de las razones por las que los estudiosos propusieron la aplicación de métodos rigurosos que guiaran el desarrollo de software. Fue entonces cuando surgió la Ingeniería de Software, la que, a través de métodos, modelos, la inclusión de la calidad y una buena porción de administración, empezó a orientar el desarrollo de proyectos para que cada vez se lograran mejores productos.

Fue así como se empezaron a definir científicamente las siguientes características esenciales:

1. **Metodologías.** Se componen de una serie de pasos, como identificación de necesidades, planificación, calidad, análisis, diseño, codificación, pruebas y mantenimiento de productos.
2. **Herramientas.** Son sistemas automáticos que apoyan la implementación de la metodología utilizada. Las herramientas pueden ser CASE (Computer-Aided Software Engineering), control automático de software, módulos reusables, bases de datos que sirven para realizar seguimiento de errores o para el registro.
3. **Procesos.** Son los pasos en que se aplican la metodología, los artefactos entregados en cada fase, la calidad en los procesos y productos a través de controles, así como la verificación tanto de hitos como del progreso del proyecto.

De esta manera, a través de la Ingeniería de Software y la regulación que se le ha dado al desarrollo de productos de software a partir de los nuevos métodos y herramientas formales, ha sido posible resolver algunos problemas, dado que aún persisten otros cuya solución aún no se ha logrado, problemas estos que se refieren a la administración, la calidad, la productividad y el fácil mantenimiento, que, en general, se ven reflejados en la necesidad de mejorar la eficiencia de la producción.

El desarrollo de software debe tener un proceso coherente y sistemático que lo conduzca, es a lo que comúnmente se le denomina metodología, vista por (Marcos, 2005) como, “Una metodología de desarrollo debe tomarse como una guía, pero no como algo rígido; debe adaptarse para cada utilización de la misma, del mismo modo que el método de investigación.”, también asegura que, “al iniciar un desarrollo de software se decide el paradigma metodológico (estructurado, orientado a objetos) y la metodología concreta a seguir”. Lo anterior indica que si en el proceso de desarrollo del software no se tiene la guía con la que se va a trabajar (tanto método como metodología), el proceso posiblemente no dé los resultados esperados o la entropía no le permita avanzar en su desarrollo, de ahí surgen algunas opiniones sobre el fracaso de muchos proyectos que se inician o de la cantidad de correcciones que se deban efectuar con los respectivos retrasos que ello proporciona y la baja cantidad de proyectos culminados con éxito. Los proyectos fracasados no necesariamente se deben a las fallas técnicas, sino

más bien, a la carencia de una buena metodología o modelo que lo conduzca. Por lo anterior, sería fantástico que los estudiantes desde sus cátedras se concientizaran de la buena práctica del uso de métodos y metodologías recubiertas de calidad para desarrollar proyectos de software y que se acostumbraran desde allí a utilizarlas adecuadamente.

El presente trabajo se desarrolla en seis fases, de las cuales se obtienen resultados que están permitiendo la respuesta a los dos problemas planteados:

De una parte se plantea. ¿De qué manera la falta de una metodología o modelo para el desarrollo de software definida explícitamente y orientada a la realización de sistemas informáticos para micro, pequeña, mediana o grande empresa, afecta el cumplimiento del cronograma, aumentan los costos, esfuerzos y tiempo en la búsqueda de algún método o metodología que se ajuste al desarrollo de un proyecto de software? y de otra, ¿Cómo afecta a los estudiantes que inician el planteamiento de proyectos de software, la falta de un método o de una metodología específica para concebirlo, avanzar en él o madurarlo?

Alguno de los resultados obtenidos hasta este momento luego del análisis de datos hecho tanto a las respuestas de los desarrolladores empresariales como de los estudiantes de la Universidad Sergio Arboleda (USA), es el refuerzo que debe darse al Plan de Estudios de la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la USA, por cuanto, una cosa es lo que las empresas están requiriendo y algunas otras las que están recibiendo los estudiantes; aprovechando que en este momento se está llevando a cabo un proceso de autoevaluación en la Escuela de Ingeniería, el Comité Curricular está trabajando en dicha reforma. De otra parte, es evidente el poco formalismo que las empresas desarrolladoras de software siguen en el desarrollo de proyectos, debido a que, en primer lugar, algunas de ellas sólo utilizan la metodología RUP, independiente de tratarse de empresas micro, pequeñas, medianas o grandes empresas, en segundo lugar, otras empresas no utilizan ninguna metodología base para que los proyectos sean uniformes y conduzcan a la calidad del software producido, lo anterior indica que, existe un nicho en la industria del software para promover el resultado de esta investigación.

Luego de realizar el análisis de la información, es decir, al terminar la Fase 3, se procederá a elaborar los lineamientos, pautas o criterios que sirvan como soporte en la construcción del método para desarrollar software, Fase que es el corazón de esta investigación y que conducirá a obtener los resultados esperados.

Para tener una mayor claridad en este escrito, se resalta que las referencias que se hagan a los diferentes tipos de empresas denominadas según su forma jurídica atendiendo a su titularidad y a la dimensión de las mismas, en Colombia, según las leyes 905 y 504 de Mipymes, hacen alusión a esta clasificación según su dimensión, en las cuales se tienen en cuenta como indicadores principales: el capital propio y el número de trabajadores. Por lo anterior, la magnitud de las empresas se delimita de la siguiente manera:

Microempresa si posee menos de 10 trabajadores.

Pequeña empresa: si tiene menos de 50 trabajadores.

Mediana empresa: si tiene un número entre 50 y 250 trabajadores.

Gran empresa: si posee más de 250 trabajadores.

De otra parte, según la titularidad del capital, pueden ser:

Empresa privada: si el capital está en manos de particulares,

Empresa pública: si el capital y el control está en manos del Estado.

Empresa mixta: si la propiedad es compartida.

Este artículo se diseñó de tal manera que a través de él se traten las diferentes fases en que fue dividido el proyecto, se mencione lo conseguido en cada una de ellas, se clarifique en qué punto se encuentra el proyecto y se indique qué falta por hacer.

1.1 Metodología de desarrollo

El tipo de investigación escogido para formalizar este proyecto fue la Aplicada, y se utilizó por cuanto se busca y se consolida el saber, se analiza información escrita sobre los diferentes métodos y metodologías de desarrollo de software y a partir de ellas se crea una propuesta metodológica, dirigida a la fabricación de software para empresas micro, pequeñas, medianas o grandes y al uso que los estudiantes de la Facultad de Ingeniería de Sistemas le darán.

Teniendo estas premisas, se describe en seguida la metodología que conduce el proyecto.

Descripción de la metodología.

Fase 1. Fase Exploratoria

Se realizó una revisión de literatura, en donde se consultó, se hizo la selección y se obtuvo de forma selectiva la bibliografía y materiales útiles sobre las diversas metodologías, métodos y autores que enseñan a desarrollar proyectos de software.

Fase 2. Fase de descripción y explicación

Se revisaron los diferentes modelos y metodologías utilizadas en el desarrollo de software, se identificaron y clasificaron cronológicamente algunas que servirían como base para realizar las encuestas y ser tenidas en cuenta por su importancia en el desarrollo del proyecto.

Fase 3. Fase de Encuestas

Se identificó el sector objetivo de la industria del software de la ciudad de Bogotá, así como la población objetivo estudiantil de la Universidad Sergio Arboleda para aplicar las encuestas orientadas a conocer las dificultades que se presentan al emprender un proyecto de software, bien sea dirigido al sector empresarial, como en un proyecto de grado.

Se prepararon los formatos para la aplicación de las encuestas teniendo en cuenta los elementos propuestos en el problema de investigación y lo arrojado por el análisis de la Fase 2.

Se aplicaron las encuestas a la población objetivo, se realizó el análisis de los datos y se tuvieron en cuenta las metodologías que más utilizan los desarrolladores de software en Bogotá-Colombia.

Según los modelos y metodologías más utilizadas en las empresas encuestadas, se realizó el análisis de dichos métodos.

Fase 4. Fase de construcción de la propuesta

Con base en la comparación y análisis de resultados de las Fases 2 y 3, se están construyendo los lineamientos, pautas o criterios para el desarrollo de software, que servirán como soporte en la construcción del método, resultado y aporte de la investigación, para el desarrollo de proyectos de software que se ajuste a micros, pequeñas, medianas y grandes empresas, así como para la realización de proyectos en la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda.

Preparar un documento resultado con la propuesta para el desarrollo de proyectos de software que se ajusten a micros, pequeñas, medianas, grandes empresas y a la realización de proyectos en la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda.

Fase 5. Fase de documentación

Preparar un documento general con la descripción del trabajo adelantado, la especificación de la metodología propuesta y el aporte que desde la línea de Desarrollo de Software se difundirá entre las empresas desarrolladoras de software y los estudiantes que quieran acoger o aceptar la metodología en la realización de sus proyectos de software.

Fase 6. Fase de difusión

Presentar los resultados obtenidos en eventos académicos, así como escribir y publicar artículos resultado de la investigación

2. Resultados parciales encontrados en el desarrollo de la Investigación

Al terminar la Fase 1, se identificaron plenamente las teorías y fundamentos que direccionan y apoyan la investigación. Entre los fundamentos sobresalientes que se tuvieron en cuenta está la mirada que IEEE hace sobre la Ingeniería de Software, la posición que (Pressman, 2006) plasma en su libro “Ingeniería de software, un enfoque práctico”, así como la de (Sommerville, 2005) en su libro Ingeniería de software, (Craig, 2000) en su texto UML y Patrones, (Weitzenfeld, 2006) en su libro Ingeniería de Software Orientada a Objetos con UML-Java e Internet, el instituto: Software Engineering Institute (SEI) y otros muchos pensadores, que han permitido a los desarrolladores o ingenieros de software adquirir bases para culminar sus proyectos.

Teniendo como referente a Roger Pressman, se obtuvo el siguiente diseño para iniciar y centrar el estudio en una de las capas que interesa en este proyecto.



Figure 1: Capas de la Ingeniería de Software

En la figura se aprecian las diferentes capas en que la Ingeniería de Software ha sido dividida. Esta distribución, permitió centrar la investigación en la capa de **Los Métodos**, pues es allí en donde se llevan a cabo las actividades del ciclo de vida de un proyecto.

La capa **Calidad**, es la que permea todo el proceso de software y es la base en su desarrollo, sin descuidar ninguna de las etapas que lo componen, pues la calidad deberá estar presente desde la concepción de la idea del proyecto hasta más allá de su culminación.

En la capa **Herramientas**, se considera si para el desarrollo del proyecto se utilizarán herramientas automáticas o semi-automáticas, dependiendo de la magnitud del proyecto y de los lineamientos que al respecto indique el gestor del mismo, además, proporciona un soporte para el proceso y los métodos.

La capa **El proceso**, permite que las actividades que se siguen en un desarrollo de software sean estables, estén controladas y organizadas, por ello, para la Ingeniería de software se han construido una serie de pasos organizados y predecibles que han sido llamados Proceso de software, estos pueden ser evaluados y medidos a raíz de mecanismos de evaluación que permiten determinar el grado de madurez del proceso, pero a medida que el producto crece en su desarrollo los mejores indicadores de la eficiencia del proceso usado son: la calidad, el tiempo y la viabilidad del producto a largo plazo. En esta capa es en donde se realiza control a los métodos técnicos y se generan algunos artefactos o productos como por ejemplo, documentos, datos, reportes, modelos, entre otros. En consecuencia, es aquí donde se define la aplicación de métodos técnicos, se dan los resultados del trabajo, se establecen hitos, se asegura la calidad y se gestiona el cambio, además, se define un marco de trabajo para un conjunto de áreas clave de proceso. Según (Pressman, 2006), el **proceso de software** se define como “marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad”. Esta definición reafirma lo indicado, que la calidad es la estructura que permanecerá en el desarrollo de un proyecto desde su inicio.

En cuanto a la capa de interés para este proyecto **Los Métodos**, que se desarrolla en la Fase 2, y en la cual se llevan a cabo las actividades comunes del ciclo de vida de un proyecto de software, como la comunicación, el análisis de requisitos, el modelado del diseño y del sistema en general, la construcción de los programas, se hacen los testing y el soporte, en sí, indican cómo construir técnicamente el software.

Al centrar la atención sobre esta capa, se tuvieron en cuenta métodos o modelos que han tenido representación importante en el mundo del desarrollo del software y que vale la pena su estudio, entre los que se encuentran los modelos: En cascada, también llamado modelo lineal secuencial o ciclo de vida clásico. Los modelos incremental, prototipos, espiral, modelo V, diente de Sierra, desarrollo concurrente, técnicas de 4ª generación, DRA, XP, paradigma orientado a objetos, desarrollo basado en componentes, desarrollo orientado a aspectos, modelo de métodos formales, proceso unificado, paradigma orientado a servicios, Microsoft Solution Framework (MSF).

Como resultado, se obtuvo un listado de los modelos más representativos, que sirven como base para ser tenidos en cuenta tanto en las encuestas realizadas a los desarrolladores de las empresas, así como de los estudiantes de noveno y décimo semestre de la Universidad Sergio Arboleda.

En la Fase 3, se tomó como población objetivo para realizar las encuestas a empresas bogotanas que desarrollaran software; no se consideró si utilizaban para ello, software libre o propietario, tampoco interesó si se trataba de empresas desarrolladoras oficiales o privadas. El total de personas encuestadas fue de 50, que trabajan en diferentes compañías.

De las encuestas realizadas tanto a estudiantes como desarrolladores de software, se tomaron los modelos Cascada, DRA, Incremental, prototipos, espiral, paradigma orientado a servicios, el proceso unificado, XP, paradigma orientado a objetos y el MSF como los modelos más utilizados y los cuales servirán como base de la investigación.

Para el análisis de estos modelos, se tuvieron en cuenta variables relacionadas con, la pureza del modelo (si está combinado con otro), el tipo de ciclo de desarrollo que utiliza, el enfoque de construcción del modelo, cuándo se debe o no utilizar, el tiempo aproximado para el desarrollo del proyecto utilizando el modelo, las actividades que se cumplen en el modelo, el tipo de trabajo que permite, en qué tipos de proyectos se utiliza, lo que más consume el modelo y si el cliente puede o no ver los avances del proyecto.

Al analizar los resultados de las encuestas elaboradas a los 50 desarrolladores de software de las diferentes empresas, se nota que gran parte de ellos (78.27%) estarán de acuerdo en utilizar el método para desarrollar software propuesto en esta investigación y de aplicarlo en sus respectivas empresas. A su vez, el 84.79% estarían dispuestos a asistir a eventos que a partir de esta investigación resulten. Lo anterior infiere que al concluir este proyecto, gran parte de las empresas serían receptoras de los resultados para formalizar su trabajo con el método propuesto.

Al final de la Fase 3, también se obtuvieron resultados que han servido para iniciar el robustecimiento del Plan Curricular de la Facultad de Ingeniería de Sistemas, así como tener los indicadores claros sobre lo que la industria está demandando de la Academia, se conocieron además cuáles son realmente los modelos y metodologías que actualmente utilizan las empresas desarrolladoras de software. Estos resultados preliminares, permiten tener en claro que academia y empresa deben estar siempre de la mano y que la academia deberá formar a sus estudiantes de acuerdo a los requerimientos que el mercado exige, el divorcio entre estos dos entes no puede darse en ningún momento.

A continuación se presentan algunos resultados que se han obtenido al terminar la Fase 3:

Resultado del Análisis de Datos de las Empresas

Como resultado del análisis de datos, encuesta aplicada a una muestra de cincuenta empresas desarrolladoras de software en Bogotá - Colombia, se distingue que el total de las personas que contestaron dicha encuesta fueron en un 70.08% hombres, y el total de empresas a las que se les aplicó la encuesta fue del 100% privadas. Gran parte de ellas, empresas medianas (58.69%); cabe resaltar que muy pocas empresas micros (4.34%) desarrollan software en Bogotá, pero también son pocas las consideradas grandes empresas que elaboran software (10.86%), mientras que, la pequeña empresa cuenta con una representación del 20.08% en el mercado encuestado.

En cuanto al sector que se dirigen las empresas desarrolladoras de software para satisfacer necesidades, es claro que el 82.60% está orientado a las grandes empresas, el 41.30% a empresas medianas, el 17.39% a las pequeñas empresas y el 15.231% a las empresas micro.

Aunque el portafolio de servicios de las empresas encuestadas para satisfacer necesidades del mercado es muy diverso, se distingue que el Desarrollo de Software a la mediada (73.91%) cubre gran parte de este mercado, el 45.65% se orienta a la consultoría, el 32.60% al mantenimiento de software, el 30.43% a la gerencia de proyectos de desarrollo de software, el 23.91% a la administración de IT y al aseguramiento de calidad, el 21.73% a la inteligencia de negocios y sólo el 13.04% orienta sus servicios a la seguridad informática.

Los principales sectores a los que dirigen la actividad del desarrollo de software las empresas encuestadas, son en su orden: al sector bancario el 43.47%, al estatal un 39.13%, a la industria el 34.78%, al financiero el 32.60%, a los sectores salud y asegurador (empresas aseguradoras) un 17.39%, mientras que a los sectores educativo y de seguridad social sólo los cubren en un 15.21%. Es evidente que estos dos últimos sectores serán eventualmente tenidos en cuenta para orientar proyectos desde la academia, puesto que sus necesidades son muy grandes y la oferta puede tener una gran respuesta de su parte.

Algunas de las respuestas a las preguntas realizadas a las empresas encuestadas están permitiendo tomar algunos correctivos y lineamientos claros para proporcionar al sector empresarial egresados con conocimientos que realmente les dé un valor agregado en su vida profesional, cabe destacar que al hacer referencia a los encuestados sobre los lenguajes de programación que más utilizan para desarrollar software se encuentran el Java con un 73.91%, XML con el 52.17%, PHP con un 45.65%, VB. Net con el 43.47%, C# con 39.13%, DHTML con 34.78%. Lenguajes como Delphi, Power Builder, Perl, Cobol, OWL, RDF, tienen una baja utilización, pero no se descartan estos para que sean presentados a los estudiantes como electivas en su plan de estudios.

Otro de los elementos que la Facultad de Ingeniería de Sistemas y Telecomunicaciones tendrá en cuenta para la propuesta del nuevo plan curricular según las respuestas que sobre este tema se dieron, es el tipo de plataforma que se deberá implantar dentro de los esquemas de las asignaturas orientadas a tal fin, es así que la plataforma .NET con un 69.56%, la J2EE con un 65.21%, deberán formar parte de los contenidos de las materias. Mientras que las plataformas, SAP, AS/400, S/390 y otras no revisten gran utilización en el mercado.

En cuanto a las bases de datos, se requiere reforzar en el plan curricular la enseñanza de la herramienta Oracle, utilizada para la gestión de bases de datos, pues la evidencia muestra que en la industria es utilizada por un 71.73% de las empresas encuestadas, el SQL Server, otro sistema de gestión de bases de datos relacionales (SGBD) lo usan el 60.86%, seguidos por MySQL con un 39.13% y PosgresSql con 19.56%. Otros sistemas como SyBase, DB2, Access, Informix, SQLite, Apache Derby, Paradox, tienen una representación muy baja en el desarrollo de software, dentro de las personas encuestadas.

En referencia a los Sistemas Operativos utilizados en las empresas, se encuentran en su gran mayoría utilizando el Windows con el 82.60%, seguido por Linux con el 50% de uso, al contrario de lo que muchos de los ingenieros imaginamos, los desarrolladores encuestados no utilizan el Unix sino en un 15.21% y esto tiene su explicación, debido a que sólo el 58.69%, de las empresas encuestadas son medianas empresas y únicamente el 10.86% son grandes, puesto que este es un sistema utilizado sobre todo en computadoras grandes, se caracteriza por permitir tiempo compartido y trabajo simultáneo de varios usuarios, cosa que en el tipo de empresas utilizado no lo requieren.

Al iniciar un proyecto de desarrollo de software en las empresas, sólo el 60.86% de los encuestados saben qué metodología utilizar, el 32.60% no lo saben, esto da un buen margen para promover el resultado de esta investigación entre esta población que pierde tiempo al considerar qué método seguir. Sólo el 50% de ellas, utilizan metodologías conocidas y el 41.30% tienen la propia que han establecido de la fusión de otras conocidas, como por ejemplo: Xp y Rup, otros utilizan certificaciones ISO, otros emplean guías establecidas por la empresa y otros se rigen por lo que el cliente exija.

En cuanto a las metodologías utilizadas para el desarrollo de software, un 45.65% se guían por RUP, un 43.47% por el análisis y diseño orientado a objetos, un 15.21% por XP y el 2.17% por MSF. Cuando se hace referencia a los paradigmas de desarrollo de software gran parte de las personas encuestadas no los emplean (45.65%), pero un

41.30% sí lo hacen, esto da a entender que gran parte de las empresas no tienen parámetros fijos, ni estándares de calidad establecidos para el desarrollo de software. Algunas de dichas empresas aplican el modelo de prototipos (11%), el modelo en espiral y desarrollo por etapas en un 8%, el modelo en cascada, el desarrollo iterativo y creciente, y el RAD (Rapid Application Development) en un 3%, lo que da a entender que sin importar el tipo de desarrollo que se haga, adoptan métodos que como en el cascada deben ordenarse rigurosamente las etapas del ciclo de vida del software, de tal manera que el inicio de cada una de ellas, se debe esperar a que finalice la inmediatamente anterior. Algunos autores critican dicho paradigma, porque acarrea mayores costos y esfuerzos. Como anteriormente se mencionó en este escrito, se ratifica con las cifras que, RUP, es usada de forma indiscriminada, pues se dirige a todo tipo de desarrollo, sean estos, micro, pequeños, medianos o grandes, no obstante, la misma metodología RUP aconseja sea utilizada en grandes desarrollos.

Para el modelado de las aplicaciones, el UML es el más utilizado, pues un 82.60% de los encuestados modela el software por medio de dicho lenguaje, y sólo el 4.34% lo hacen con Enterprise Architect y el 2.17% con Arquitectura Dirigida por Modelos.

Cabe resaltar que se elaboraron ciertas preguntas que conducen a los investigadores a revisar qué tanta aceptación tendría el resultado del presente trabajo, para lo que un 69.56% de los desarrolladores estaría en disponibilidad de utilizar el producto de esta investigación y el 65.21% de las empresas como tales también lo acogerían. En cuanto a la divulgación que se realizará del resultado de este proyecto, un 69.56% de los desarrolladores asistirían a eventos que se realicen dentro de la facultad para conocer dicho método, una vez se logre el objetivo.

Resultado del Análisis de Datos de los Estudiantes de la Universidad Sergio Arboleda (Semestre 9)

La población escogida dentro de la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda, fueron los semestres de 9 y 10, debido a que en el semestre 9, los estudiantes presentan el anteproyecto de grado y es precisamente allí en donde proponen el método que seguirán en el desarrollo del proyecto y en el semestre 10, desarrollan el mismo.

El número de estudiantes de noveno semestre fueron de 10, población total, que está cursando dicho semestre, de los cuales el 80% son hombre y 20% mujeres.

A la pregunta sobre los modelos o paradigmas que conocen los estudiantes para desarrollar proyectos de software, un 90% afirman que distinguen el modelo en espiral, el 80% el modelo en cascada, clásico o lineal secuencial, desarrollo iterativo y creciente o interactivo incremental. El extreme programming en un 80%. El modelo WinWin en un 20% y los modelos prototipos, por etapas y desarrollo concurrente solo en un 10%. Estas respuestas alertan, para que los estudiantes dentro de sus asignaturas conozcan otros elementos que en un momento dado les puede servir en su vida profesional.

Definitivamente, la metodología RUP es la que más conocen los estudiantes (90%), seguida del análisis y diseño orientado a objetos con un 60%. De otra parte el UML es reconocido por todos los estudiantes como herramienta de modelado para sus aplicaciones de software.

Preocupa que un 60% de los estudiantes de noveno semestre no tienen idea de la metodología que aplicarán en el proyecto de grado, esto conduce a que ellos se retrasen en tiempo para entregar su anteproyecto (60%), en el avance del proyecto (50%), en la maduración de la idea (40%), en la consecución o asignación de un director del proyecto (30%) y en la concepción plena del proyecto (20%).

El 100% de los estudiantes de noveno semestre, les complace que se esté adelantado una investigación que conduzca a la propuesta de un método para desarrollar software, teniendo en cuenta si se trata de empresas micros, pequeñas, medianas o grandes y el 90% de ellos la aplicarían, el 70% asistiría a eventos diversos que se programarán para socializar el resultado de la investigación.

Resultado del Análisis de Datos de los Estudiantes de la Universidad Sergio Arboleda (Semestre 10)

La población total de estudiantes de décimo semestre fue de 13 personas, de las cuales 84.61% fueron hombres y 15.39% mujeres.

A la pregunta sobre los modelos o paradigmas que conocen para desarrollar proyectos de software, un 100% conocen el modelo en cascada, clásico o lineal secuencial, el 92.31% modelo en espiral. El extreme programming en un 53.84%. El modelo WinWin en un 30.76% , el modelo prototipos y el desarrollo iterativo y creciente o interactivo Incremental en 69.23%, modelo por etapas el 61.54% , el modelo V en 38.46% , el desarrollo concurrente y DRA en 30.76% , mientras que el modelo diente de sierra y técnicas 4G los conocen 23.08% y 15.38% respectivamente. A diferencia de los estudiantes de noveno, en este semestre, conocen un poco más cantidad de paradigmas para desarrollo de software, pero deberían tener referentes de todos ellos, puesto que ya están listos para salir a enfrentarse a la vida profesional, y es responsabilidad de la academia conducirlos para que, mientras estén en sus aulas facilitarles el camino para su autoaprendizaje sobre los temas que competen a su carrera.

Al igual que los estudiantes de noveno, el 92.31% de ellos conocen la metodología RUP, seguida del análisis y diseño orientado a objetos con un 84.61%. De otra parte el UML es reconocido por todos los estudiantes como herramienta de modelado (igual que los estudiantes de noveno semestre).

Es también motivo de preocupación que los estudiantes que están a unos pocos meses de graduarse como profesionales, un 46.15% no saben aplicar la metodología propuesta en el anteproyecto y el 15.38% no aplican la metodología que propusieron en el mismo. Mientras que, el 30.77% aplican sólo parte de la metodología propuesta en el anteproyecto. Lo anterior trae como consecuencia que los estudiantes se retrasen en tiempo para entregar su proyecto de grado (76.092%), en el avance del proyecto (53.85%), en la maduración de la idea (38.46%), en la consecución o asignación de un director del proyecto y en la concepción plena del mismo en un 30.77%.

El 92.30% de los estudiantes de décimo semestre, les complace que se esté adelantado una investigación que conduzca a la propuesta de un método para desarrollar software, teniendo en cuenta si se trata de empresas micros, pequeñas, medianas o grandes y el 84.61% % de ellos la aplicarían, el 92.31% asistiría a eventos diversos que se programarán para socializar el resultado de la investigación.

Luego del análisis que se hicieron a los datos de las empresas Vrs. estudiantes, se da el reporte a la Decanatura del programa con el objeto de tomar los correctivos necesarios, lo que conduce a que aprovechando el proceso de autoevaluación del programa se decidiera replantear el plan de estudios del programa incluyendo gran parte de las conclusiones expuestas en este escrito.

Una conclusión inevitable es que el sector de las empresas denominadas micro, está un tanto desprotegido y se visualiza como un buen nicho para enfocar en el futuro el desarrollo de proyectos académicos.

3. Trabajo que queda por completar

Son seis las fases que componen el desarrollo del proyecto, en este momento se está trabajando sobre la Fase 4, que con base en la comparación y análisis de resultados de las Fases 2 y 3, se están construyendo los lineamientos, pautas o criterios para el desarrollo de software, que servirán como soporte en la construcción de la metodología para el desarrollo de software.

Se piensa además construir la propuesta metodológica, resultado y aporte de la investigación, para el desarrollo de proyectos de software que se ajuste a micros, pequeñas, medianas y grandes empresas, así como para la realización de proyectos en la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda.

Se pretende también, preparar un documento resultado con la propuesta para el desarrollo de proyectos de software que se ajusten a micros, pequeñas, medianas, grandes empresas y a la realización de proyectos de grado

o de cursos intermedios en la Facultad de Ingeniería de Sistemas y Telecomunicaciones de la Universidad Sergio Arboleda.

En la Fase 5, se preparará un documento general con la descripción del trabajo adelantado, se especificará la metodología utilizada y el aporte que desde la Línea Desarrollo de Software de la Universidad Sergio Arboleda se hará a las empresas desarrolladoras de software, así como a los estudiantes que inicien o estén trabajando su proyecto de grado y quieran utilizar la metodología planteada.

En la Fase 6 o fase de difusión, se presentarán los resultados obtenidos en eventos académicos dentro y fuera de la Universidad Sergio Arboleda y se escribirán artículos que den cuenta de los resultados de la investigación.

REFERENCIAS

Larman, C. (2000). *UML y Patrones*, 1ª edición, Prentice Hall, México

Marcos, E. (2005). *“Investigación en Ingeniería del Software vs. Desarrollo Software”*, Grupo KYBELE, Universidad Rey Juan Carlos.

Pressman, R.S. (2006). *Ingeniería del Software, un enfoque práctico*, 6ª edición, Mcgraw-Hill, México.

Software Engineering Institute (SEI), <http://www.sei.cmu.edu/publications/documents>

Sommerville, I.(2005). *Ingeniería del software*, 7ª edición, Pearson-Addison Wesley, España

Weitzenfeld, A. .(2006). *Ingeniería de software orientada a objetos con Uml, java e internet*, Thomson, México

Autorización y Renuncia

Los autores autorizan a LACCEI para publicar el escrito en los procedimientos de la conferencia. LACCEI o los editores no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.