La ingeniería del software en Estados Unidos y Canadá

Walter Hugo Arboleda Mazo
Grupo de Investigación en Ingeniería Aplicada

Resumen

La ingeniería del software es una de las áreas de la ingeniería que aportan al resto de las disciplinas mediante la creación de aplicaciones con calidad y siguiendo buenas prácticas para el desarrollo de estas; ha estado mejorando con la aparición de nuevas metodologías, lográndose hoy que Estados Unidos y Canadá sean lideres en día a nivel mundial en metodologías, permitiendo que las empresas tengan cada día un mejor producto para realizar sus procesos de negocio.

Términos clave: ingeniería del software, calidad, buenas prácticas, metodología.

Abstract

The software engineering is one of engineering areas which gives to all disciplines a meaningful help, with the software manufacturing using quality and using best practices developing it; the software design have been improving with the new methodologies, all this have made that United States and Canada, become methodology worldwide leaders, permitting to enterprises do their business processes with a better product.

Keywords: software engineering, quality, best practices, methodology.

INTRODUCCIÓN

La ingeniería del software es un área de las ciencias de la computación, la cual se encarga de la creación de software que es utilizado en diversas fábricas y entornos como el comercial, militar, salud, bancario, negocios, académico, científico, entre otros. En sus más tempranos inicios en los años 1950, la ingeniería del software no contaba con el apoyo ni la seriedad que requiere esta vital área por parte de los gobiernos de los Estados Unidos y Canadá, generando enormes pérdidas por fallas y retrasos en los proyectos de software para diversas entidades

comerciales y gubernamentales. Es allí donde la comunidad que se encontraba trabajando en software decide reunirse en Alemania e Italia, para marcar un rumbo a la ingeniería del software, definir grupos de trabajo y formalizar el quehacer de los ingenieros de software, ya que al inicio no era vista como una profesión; además, quienes prestaban sus servicios en programación eran personas no capacitadas para ello y que no habían realizado estudios formales en ingeniería del software, generando todas las fallas posibles. Esta condición permitió que aparecieran organizaciones y estándares que han posibilitado la mitigación de la crisis del software y hacer de la ingeniería del software en Estados Unidos y Canadá, cada vez más una disciplina más seria y orientada hacia la producción de productos de alta calidad.

I. ORÍGENES DE LA INGENIERÍA DEL SOFTWARE

El origen de la ingeniería del software como palabra se atribuye a la OTAN, en los años 1968 y 1969, en los cuales de realizaron dos conferencias para abordar la "Crisis del Software", las cuales, al igual que hoy, buscaban abordar los problemas asociados a fallas en la entrega del software y mala calidad de este.

Para Boehm (1976), la ingeniería del software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software. A la vez, Zelkovitz, Shaw, & Gannon (1979) conciben la ingeniería del software como el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software.

En cuanto a los categorías de software Pressman (2003) establece que en la actualidad existen siete grandes categorías de software de computadora que presentan retos continuos para los ingenieros de software; estas son: software de sistemas, software de aplicación, software científico y de ingeniería, software empotrado, software de línea de productos, aplicaciones basadas en la web y software de inteligencia artificial.

Según Mahoney (1990) hablar de la historia de la ingeniería del software es remontarse a los años 1950, cuando se inició la comercialización de las computadoras y se empezó a obtener importantes avances en los microprocesadores, memorias y periféricos, tiempo en el cual antes de dicha década IBM no reconocía la programación de computadoras como un tipo de trabajo; de hecho, los programas eran objetos de producción por personas aficionadas que no eran científicos, matemáticos o ingenieros. Esta realidad originaba retardos en la entrega del software y fallas en su fabricación.

Por tanto, se define la ingeniería del software como un área de las ciencias de la computación, que permite la utilización de metodologías y técnicas que incluyen: gestionar proyectos de software, diseñar, desarrollar, documentar, probar, controlar la

calidad y el versionado del software, así como su mantenimiento, área en la cual participan los profesionales de la ingeniería del software y para lo cual usan computadoras cada vez más potentes. A pesar de que cada año se presentan conferencias y se publican nuevos adelantos e investigaciones en estas, para Sousa Gomes & da Mota Silveira Neto (2011) debido al incremento de los deseos de clientes y de las fábricas de software, la complejidad y el tamaño del software también han aumentado, por la cantidad de funcionalidades que se ponen en este, lo que hace que la presión a la comunidad de la ingeniería del software sea más social que técnica, la cual se limita a cumplir las expectativas de los clientes a como dé lugar, aspecto en el cual deben aportar cada vez más los investigadores e inventores de la ingeniería del software y las propias fábricas de software.

II. GÉNESIS DE LA HISTORIA DEL SOFTWARE

Para la fabricación de software se han utilizado varias metodologías desde la década de 1960 hasta la década del 2000; estas se han ido perfeccionando, permitiendo realizar software con altos niveles de calidad y rapidez en su ciclo de vida. Los primeros pasos en ingeniería del software en relación con diseño de software en los Estados Unidos fueron realizados por IBM, con Harlan Mills y Niklaus Wirth, partiendo en los años 1970 con Top Down Design, diseño que permite descomponer una aplicación en una serie de módulos y funcionalidades, permitiendo partir de lo global a lo detallado y funcional, con la creación de módulos, subprogramas, funciones y subrutinas, pero siempre conservando la programación estructurada.

En 1980, IBM continuó investigaciones en diseño de software y se logró con Grady Booch, crear una metodología que daba al diseño del software otro contexto y permitía el diseño de un programa mediante objetos y clases, lo mismo que su desarrollo, minimizando de este modo el entendimiento del código y los programas asociados a este, así como la cantidad de líneas de código de un programa, debido a la reutilización de código.

En 1990, James Rumbaugh, Ivar Jacobson y Grady Booch crean a UML en los laboratorios de IBM, logrando un consenso entre las fortalezas de las metodologías Object Oriented Software Engineering (OOSE), de Ivar Jacobson; OMT, de Rumbaugh; y la metodología Booch, de Booch, extrayendo lo más importante y complementario de estas metodologías, creando una mejor, que aportó tanto al diseño del software como a la facilidad para la programación de este.

A finales de 2005, Ivar Jacobson anunció el Essential Unified Process - EssUP, práctica totalmente orientada hacia el desarrollo de software que integra buenas prácticas de RUP, metodologías ágiles y CMMI, contribuyendo a la madurez, calidad y agilidad en los procesos del ciclo de vida del software, integrándose hoy esta metodología con herramientas como Microsoft Visual Studio Team System y Eclipse.

III. ÉPOCAS DEL DISEÑO DEL SOFTWARE

La industria del desarrollo de software en Estados Unidos y Canadá, ha estado estrechamente relacionada, a medida que las empresas canadienses han participado en los diferentes encuentros para mejora del software y los aspectos relacionados con este, así como han adoptado estándares y metodologías que se han desarrollado en los Estados Unidos.

Década de 1970	Diseño de software Top Down
Década de 1980	Diseño de software usando orientación a objetos
Década de 1990	Diseño de software mediante el Proceso Unificado UML (1995)

Tabla 1. Épocas del diseño de software en Estados Unidos y Canadá

Evolución del software en Estados Unidos y Canadá

Para Pressman (1997) el software ha evolucionado, estableciéndose cuatro bloques de tiempos en años de la siguiente forma: los primeros años (1950-1965), la segunda era (1965-1975), la tercera era (1975-1988), la cuarta era (1988-2000); estas estuvieron marcadas por la aparición de nuevas tecnologías y mejoras en software y hardware, así como cambios positivos en la infraestructura de red.

I. LA COMUNIDAD DE INGENIERÍA DEL SOFTWARE

The International Conference on Software Engineering (ICSE), es la conferencia más importante de Ingeniería del Software a nivel mundial, en esta se discuten los principales aspectos de la disciplina como se presentan nuevas propuestas e investigaciones en metodologías, estándares, lenguajes, métricas, calidad y gestión de proyectos de software; esta ha tenido 37 versiones, lo que muestra el interés de la comunidad de la ingeniería del software por encontrar nuevas formas de hacer mejor y más rápido productos de calidad, siendo el lider en toda america, Estados Unidos (18) seguido por Canada(2), lo que muestra el nivel de interes, aporte e investigacion en ingenieria del software en ambos paises.

Los primeros años (1950-1965)	Ejecución de procesos por lotes, sistemas no distribuidos y creación de software a la medida.
La segunda era (1965-1975)	Aparición de la programación multiusuario y en tiempo real, aparición de las bases de datos y la creación y venta de productos de software.
La tercera era (1975-1988)	Aparición de los sistemas distribuidos, incorporación de inteligencia al software, hardware de bajo costo y aumento en el consumo de software.
La cuarta era (1988-2000)	Equipos, personas, potentes, aplicación de las tecnologías orientadas a objetos, aplicación de redes neuronales e inteligencia artificial al software, computación en paralelo y crecimiento de las redes de computadoras.

Tabla 2. Evolución del software en Estados Unidos y Canadá

Keil-Slawik & Brennecke (1996) describen la ponencia de Martin Campbell-Kelly, Development and Structure of the International Software Industry,1950-1990, realizada en Dagstuhl, Alemania; el seminario History of Software Engineering en 1996 muestra que la industria del software en Estados Unidos desde 1950 hasta 1970 recibió poca atención por los pocos ingresos que aquélla representaba; fue solo después de 1970 cuando con la atención de IBM y los adelantos en las computadoras personales creció la industria del software; de esta forma los ingresos anuales de la industria fueron de la siguiente forma: 1970, US$1.2 billones; 1979, US$2 billones; 1982, US$10 billones; 1985, US$25 billones; y en 1990, US$100 billones); lo que llamó la atención de los economistas y el gobierno de Estados Unidos sobre los ingresos y el crecimiento de la industria del software.

En Estados Unidos, en 2010 según el Softwaretop100 (2013), entre las 25 empresas que más ingresos obtuvieron estuvieron Microsoft (1), IBM (2) y HP (3).

Este listado de las 25 empresas de software puede ubicar a los siguientes Estados entre los que más se generan ingresos por ingeniería del software, debido a la cantidad de empresas que tienen: California (13) y Nueva York (4); los demás que están en la lista solo poseen una sola empresa de software, ubicándose California y Nueva York en el primero y segundo lugares como Estados en los que más ingresos produce la ingeniería del software en Estados Unidos, lo que permite inferir los esfuerzos realizados en California y Nueva York por mantener la hegemonía a nivel nacional, así como la utilización de buenas prácticas, que permitan aumentar sus ingresos y la satisfacción de sus clientes.

Ranking 2010	Nombre de la compañía	Ciudad	Estado
1	Microsoft	Redmond	Washington
2	IBM	Armonk	Nueva York
3	Oracle	Redwood Shores	California
4	HP	Palo Alto	California
5	Symantec	Sunnyvale	California
6	Activision Blizzard	Santa Monica	California
7	Lockheed Martin	Bethesda	Maryland
8	Electronic Arts	Redwood	California
9	CA Technologies	Islandia	New York
10	Adobe	San Jose	California
11	EMC	Hopkinton	Massachusetts
12	SunGard	Wayne	Pennsylvania
13	Cisco	San Jose	California
14	Autodesk	San Rafael	California
15	BMC	Houston	Texas
16	Take-Two Interactive	Nueva York	NewYork
17	NCR	Duluth	Georgia

18	Intuit	Palo Alto	California
19	Synopsys	Mountain View	California
20	Citrix	Fort Lauderdale	Florida
21	VMWare	Palo Alto	California
22	Apple	Cupertino	California
23	SAS Institute	Cary	Carolina del Norte
24	Infor	New York	New York
25	Salesforce.com	San Francisco	California

Tabla 3. Ranking de las 25 empresas con mayores ventas en Estados Unidos en 2010

La empresa Branham Group, Inc. (2013) muestra en su ranking las primeras 25 empresas de software en 2012, en Canadá; es importante notar que las 10 primeras empresas están ubicadas por cantidad y provincia de la siguiente forma: Ontario, 6; Quebec, 2; y Britain Columbia, 2; de las seis que están en Ontario hay por ciudad: Waterloo, 2; Toronto, 2; Ottawa, 1; y Markham, 1, lo que puede mostrar el nivel de madurez de la ingeniería del software en la provincia de Ontario y las ciudades de Waterloo y Toronto, así como sucede en las ciudades de Palo Alto, San Jose, Redmond, Armonk, Redwood Shores y Sunnyvale en el Estado de California, en Estados Unidos.

En Canadá, según el listado de Branham Group, Inc. (2013), por provincia en la cual se desarrolla software, lo que representa una cantidad de fábricas de software, se tiene el siguiente ranking: Ontario (8),Quebec (7), Britain Columbia (7) y Alberta (3); a diferencia de los Estados Unidos, en Canadá por provincia no existe una concentración de fábricas de software como sucede en Estados Unidos en los Estados de California (13) y New York(4), lo que puede mostrar una representación y desarrollo más uniforme geográficamente en Canadá con respecto a Estados Unidos, donde las principales empresas de software están en California.

Ranking 2012	Nombre de la empresa	Ciudad	Provincia
1	OpenText	Waterloo	Ontario
2	Constellation Software	Toronto	Ontario
3	Mitel	Ottawa	Ontario
4	Points.com	Toronto	Ontario
5	Enghouse	Markham	Ontario
6	Descartes Systems Group	Waterloo	Ontario
7	ACCEO Solutions	Montreal	Quebec
8	VendTek Systems	Port Coquitlam	Britain Columbia
9	Logibec Groupe Informatique	Montreal	Quebec
10	Vision Critical	Vancouver	Britain Columbia
11	Amaya Gaming Group	Pointe-Claire	Quebec
12	Absolute Software	Vancouver	Britain Columbia

13	iQmetrix	Vancouver	Britain Columbia
14	Computer Modelling Group	Calgary	Alberta
15	PointClickCare	Mississauga	Ontario
16	Redknee	Mississauga	Ontario
17	Mediagrif Interactive Technologies	Longueuil	Quebec
18	Solium Capital	Calgary	Alberta
19	Zedi (technology segment)	Calgary	Alberta
20	GIRO	Montreal	Quebec
21	TIO Networks	Vancouver	Britain Columbia
22	TECSYS	Montreal	Quebec
23	Versatile Systems	Vancouver	Britain Columbia
24	StarDyne Technologies	Kelowna	Britain Columbia
25	Averna	Montreal	Quebec

Tabla 4. Ranking de las 25 empresas con mayores ventas en Canadá en 2012

Según Pressman (1997) la industria del software ya es la cuna de la economía del mundo. Las decisiones de empresas como Microsoft en los Estados Unidos, arriesgan billones de dólares. Para que esto no suceda la industria de la ingeniería del software se preocupa por aplicar estándares y metodologías que ayuden a minimizar las pérdidas ocasionadas por fallas y retrasos en la industria del software, los cuales aún se continúan presentando.

Para Pressman (1997) es clara la madurez presentada por los Estados de California y Nueva York en Estados Unidos, así como en las provincias de Ontario, Quebec y Britain Columbia, en las cuales se presentan altos niveles de ventas de software, las cuales directamente influyen en la economía de dichas regiones.

Pfleeger (2001) establece que los factores que han cambiado el desarrollo del software han sido: la tecnología de objetos, las limitaciones del modelo en cascada, el tiempo de respuesta en el mercado, el cambio en la computación de escritorio a computación en red, los cambios en la economía, los cambios en la interfaz gráfica y la mejora en la conectividad de las redes, lo que conlleva que todos estos factores presentados en los entornos estadounidense y canadiense, a que en ambos países se dé un mayor nivel de investigación, desarrollo y cambio constante alrededor de metodologías y estándares usados la ingeniería del software y la industria asociada a esta, en forma más frecuente que en otros países del mundo.

Para Raccoon (1997) la economía del software ha dependido siempre de la economía del hardware; así, en cada nueva generación se construyen nuevos computadores que llegan a un nuevo tipo de usuarios y programadores, lo que directamente cambia la economía y define nuevas necesidades de hardware y software en términos de la percepción de los usuarios y la economía de estos.

Metodologías de desarrollo de software

Heymans & Trigaux (2003) sostienen que la ingeniería de líneas

de producción nació en la década de 1980 como una teoría económica para incrementar la escala de la economía; en nuestros días estos conceptos son aplicados por empresarios e investigadores para ayudar a incrementar la calidad del software y reducir los costos de este; es allí donde se apalancan los estándares y metodologías modernas usados en la ingeniería del software.

A continuación se hace un resumen de los modelos usados en la ingeniería del software:

Modelo en Cascada: este modelo fue primeramente abordado por Winston Royce en 1970, siendo la metodología usada para el desarrollo de software, incluyendo el Departamento de Defensa de los Estados Unidos; su descripción incluye de forma secuencia el análisis de requerimientos, diseño del sistema, diseño del programa, codificación, pruebas unitarias y de integración, pruebas de sistema, pruebas de aceptación, operación y mantenimiento; se caracteriza porque se debe realizar de forma secuencial cada fase, estableciéndose dependencia entre una y otra; por ende, se necesitan la definición de las actividades y su correcto control.

Modelo prototipado: este modelo está caracterizado por la entrega de la implementación de ciertas funcionalidades clave de un sistema de información, de manera que clientes, desarrolladores y testers verifiquen rápidamente el cumplimiento de las principales funcionalidades.

Modelo incremental: permite entregar el software de forma más rápida, de modo que el usuario tenga una versión en producción y se tenga un incremento de este en desarrollo constantemente, el cual se convierte en la próxima versión del software que será utilizado por los clientes.

Modelo en espiral: según Pressman (2003), el modelo en espiral propuesto por Barry Boehm en 1988, se trabaja por medio de incrementos, los cuales se inician desde el análisis y el diseño, presentándose en cada iteración versiones más completas del sistema diseñado; este modelo se centra en la entrega de un producto operacional, el cual se va haciendo más terminado de iteración a iteración, siendo el cliente el centro de este modelo.

Rapid Application Development (RAD): desarrollado por James Martin en la década de 1980 en IBM y publicado en 1991; reduce el ciclo del software, limitándose a la planeación de requisitos del sistema, interacción con los usuarios para hacer un diseño del sistema centrado en sus necesidades; en la construcción también participan los usuarios para ver cómo va siendo elaborado el sistema basado en las requisitos, y finalmente se hacen las pruebas y la capacitación de usuarios. Este modelo está caracterizado porque se reduce el diseño, lo que exige que los equipos de trabajo involucrados tengan experiencia.

Finalmente, el uso de estas metodologías en la industria del software de Estados Unidos y Canadá, ha hecho que se disminuyan los problemas presentados por la "Crisis del Software" y que aumente los ingresos por desarrollo de software, al igual que la satisfacción de los clientes, quienes prefieren las metodologías

más rápidas y que los involucren en el proceso; y desde el punto de las empresas de software aumentan la productividad.

Hablar de la historia de la ingeniería del software en Estados Unidos y Canadá para Mahoney (2004), es hablar de los principios científico administrativos de Frederick Winslow Taylor, ya que los encargados de los proyectos de ingeniería del software solo se basaban en dichos principios; después aparecieron los enfoques de la ingeniería industrial aplicados al proceso de desarrollo del software, para alcanzar mayores resultados a menor costo y con mejor calidad, siendo esta la realidad de las diversas metodologías existentes en la ingeniería del software.

Sobre la actualidad de la ingeniería del software y las metodologías y estándares usados, Müller (2006) afirma que los sistemas informáticos de hoy, incluyen compleja infraestructura y operan en ambientes complejos y heterogéneos. La proliferación de dispositivos móviles, el incrementado espectro de usuarios y la emergente economía de la información web, hacen de la ingeniería del software un desafío creciente para la fabricación de software de calidad y seguro; metodologías tradicionales Top Dow, usadas en ingeniería del software, son insuficientes para afrontar dicha complejidad en el software y la evolución de los problemas asociados a este.

Para (Yang & Mei, 2006), el desarrollo de la ingeniería del software ha estado acompañada del gran y rápido crecimiento de la tecnología y la industria del software en las pasadas cuatro décadas. Todo esto muestra a la ingeniería del software como una disciplina independiente y su desarrollo depende de los esfuerzos cooperativos entre el gobierno, las fábricas de software, la academia y las empresas.

Para (Yang & Mei, 2006), el desarrollo de la ingeniería del software ha estado acompañada del gran y rápido crecimiento de la tecnología y la industria del software en las pasadas cuatro décadas. Todo esto muestra a la ingeniería del software como una disciplina independiente y su desarrollo depende de los esfuerzos cooperativos entre el gobierno, las fábricas de software, la academia y las empresas.

Shepperd (2003) asevera que la década de 1980 a 1990 estuvo caracterizada por el aprendizaje autónomo de la ingeniería del software, de una forma empírica, lo que marcó un hito en el crecimiento de la industria del software, en el cual se tenía interés y se aprendía de diseño, especificaciones, planes de prueba y manuales de usuarios. Aunque los tiempos cambian y ahora las fábricas de software certifican a sus ingenieros, también es importante acceder a los recursos y compartir con organizaciones como The International Software Engineering Research Network (ISERN).

Año	Metodología ágil
1990	Adaptative Software Development ASD
1991	Rapid Application Development RAD
1992	SCRUM
1993	Lean Software Methodology
1994	Crystal Clear
1996	eXtreme Programming XP
1997	Feature Driven Development Methodology FDD
2002	Agile Unified Process AUP
2005	Essential Unified Process EssUP
2008	Kanban

Tabla 5. Nuevas metodologías á giles entre 1990 y 2008

Campbell-Kelly (2007) señala que es posible que se esté observando en estos momentos una convergencia entre los negocios de información y las empresas de software, migrando la unión de estos dos tipos de empresas a servicios para los usuarios, haciendo difícil para los usuarios la distinción de fábrica de software o empresa que presta servicios en la internet, además de que en 20 o 30 años, con esta unión se generará un cambio tan grande en la forma de vivir, como el logrado con el aporte de la máquina de vapor, el coche o el avión.

CONCLUSIONES Y RECOMENDACIONES

La crisis del software es una realidad diaria, cada día el software obtiene nuevas funcionalidades que hacen que este sea cada vez más complejo y difícil de realizar, aunque existan frameworks que aceleran el trabajo de programadores, analistas, diseñadores, directores de proyectos y testers; siempre es necesario el abordaje de certificaciones en ingeniería del software, garantizando con ello la minimización de fallas y errores de los productos de software, asegurando una alta calidad y satisfacción de los clientes. Desde el apoyo de los gobierno de los Estados Unidos y Canadá, se debe propiciar un mayor encuentro y patrocinio para la investigación de la industria del software, haciendo que trabajen juntos empresas, fábricas de software, centros y grupos de investigación en ingeniería del software, academia y gobierno. Se nota en presente estudio el afán por el desarrollo de software cada vez más rápido y de mejor calidad, mediante la utilización de metodologías ágiles y estándares de calidad empresariales.

REFERENCIAS

B. G., Inc. (s.f.). www.branham300.com. Recuperado el 4 de marzo de 2013, de http://www.branham300.com: http://www.branham300.com/index.php?year=2013&listing=5

Boehm, B. (1976). Software engineering. IEEE Transaction on Computers, 1.226-1.241).

Boehm, B. (2006). A view of 20th and 21st Century software engineering. ICSE'06. Shanghai, China: ACM.

Campbell-Kelly, M. (2007). The history of software. IEEE Annals of the History of Computing. IEEE Computer Society.

Ebert, C. (2008). A brief history of software technology. IEEE Computer Society.

Goth, G. (2008). The Big Bang: 25 years of software history. IEEE Software. IEEE.

Heymans, P. A. & Trigaux, J.-C. (2003). Software product lines: State of the art. Belgica: Namur.

ICSE, I. - T. (s.f.). http://www.icse-conferences.org. Recuperado el 20 de abtil de 2013, de www.icse-conferences.org: http://www.icse-conferences.org

Keil-Slawik, R. & Brennecke, A. (1996). History of software enginnering. Seminar 9635. Dagstuhl.

Mahoney, M. S. (1990). The roots of software enginnering. Princeton: Princeton University.

Mahoney, M. S. (2004). Finding a history for software engineering. IEEE Annals of the History of Computing. IEEE Computer Society.

McClure, R. M. (2001). Sofware engineering, report on a conference sponsored by the NATO Science Committee, 27th to 31th October1969. Rome, Italy: Scientic Affairs Division NATO.

McClure, R. M. (2001). Sofware engineering, report on a conference sponsored by the NATO Science Committee, 7th to 11th October 1968. Garmisch, Germany: Scientic Affairs Division NATO.

McDonald, C. (2010). From art to Engineering discipline? A history of US military software development standards, 1974–1998. IEEE Annals of the History of Computing. IEEE Computer Society.

Müller, H. A. (2006). Bits of history, challenges for the future and autonomic computing technology. Proceedings of the 13th Working Conference on Reverse Engineering (WCRE'06). IEEE.

Pfleeger, S. L. (2001). Software engineering,theory and practice. Prentice Hall, Pearson.

Pressman, R. (1997). Software engineering. McGraw-Hill.

Pressman, R. (2003). Software engineering. McGraw-Hill.

Raccoon, H. B. (1997). Fifty years of progress in software engineering.

Shepperd, M. (2003). Empirically-based software engineering. The European

Journal for the Informatics Professional, Software Engineering State of the Art. EUCIP.

Softwaretop100. (s.f.). The top 100 software companies in the United States. Recuperado el 23 de febrero de 2013, de Softwaretop100.org: http://www.softwaretop100.org/the-top-100-software-companies-in-the-us

Sommerville, I. (2006). Software engineering. Pearson, Addison-Wesley.

Sousa Gomes, J. & da Mota Silveira Neto, P. A. (2011). 25 years of software engineering in Brazil : An Analysis of SBES History. 25th Brazilian Symposium on Software Engineering. IEEE.

Yang, F. & Mei, H. (2006). Development of software engineering: Co-operative efforts from academia, government and industry. ICSE. Shanghai, China: IEEE.

Zelkovitz, M., Shaw, A. & Gannon, J. (1979). Principles of software engineering and design. Prentice-Hall.

Fecha de recepción: 3 de febrero de 2014

Fecha de aprobación: 18 de febrero de 2014

Walter Hugo Arboleda Mazo

Ingeniero de Sistemas de la Universidad de San Buenaventura, Medellín, 2000.

Especialista en Redes Corporativas e Integración de Tecnologías, de la Universidad de San Buenaventura, Medellín, 2005. Candidato a Magíster en Ingeniería de la Universidad EAFIT, Medellín. Director Académico del Nodo de RENATA en Antioquia – Red RUANA. Coordinador del Centro de Investigación de Ingeniería de la Corporación Universitaria Adventista. Líder del Grupo de Investigación en Ingeniería Aplicada –UNAC. 15 años de experiencia docente en varias universidades de Medellín.

Correo electrónico: warboleda@unac.edu.co