

DESARROLLO DE SOFTWARE, HACIA UNA PROFESIÓN



AUTOR

Henry Arguello Fuentes
Magíster en Potencia Eléctrica
Profesor Auxiliar
henarfu@uis.edu.co
COLOMBIA

AUTOR

Fernando Antonio Rojas Morales
Magíster en Ciencias Computacionales
Profesor Auxiliar
frojas@uis.edu.co
COLOMBIA

Fecha de Recepción del Artículo

Fecha de Aceptación del Artículo

Artículo Tipo 2

RESUMEN

Este trabajo de investigación describe un modelo de componentes para una profesión de ingeniería de software. El modelo se organiza en cuatro tipos de elementos: conocimiento documentado, educación, profesión y actividades, para cada componente se presenta una serie de aspectos que influyen en la consolidación de una profesión. El artículo se enfoca en la presentación de los componentes, entre los cuales se cuentan: el cuerpo de conocimientos, los estándares, las publicaciones, los aspectos curriculares, los programas educativos, aspectos relacionados con certificación y licenciamiento, y el código de ética de los ingenieros de software, con el fin de establecer un fundamento conceptual de múltiples referentes que permita visualizar como el desarrollo de software se ha venido consolidando como profesión. Como soporte se mencionan importantes organizaciones que han protagonizado el desarrollo de los componentes y se dan referencias sobre su contribución. Se pretende con este artículo ofrecer información suficiente para que los lectores posean elementos de juicio sobre como se constituye una profesión y puedan dar cuenta de cómo la actividad de desarrollo de software se consolida como una profesión de ingeniería de software.

PALABRAS CLAVES

Desarrollo de Software
Modelo de profesión
Ingeniería de Software

ABSTRACT

In this research, it is described a model of components of a profession of software engineering. The description of the model is organized in four kinds of elements: documented knowledge, education, profession and

activities. For every kind of element a group of components that makes influence in the consolidation of a profession is presented. The paper aims the presentation of the components for the software engineering profession. Some of them are: body of knowledge, standards, publications, aspects of the curricula, education programs, some issues related with certification and licensing, and the code of ethic of software engineers. The main goal of the paper is to establish a conceptual foundation from different sources to bring a view of how the activity of software development has been concreting as a profession. As a support, important organizations that have worked in

the development of components are mentioned, and some references of its contribution are given. The paper aims to offer sufficient information to the reader, in order to know how a profession is built up, and to provide the certainty that software development is transforming itself in a real profession of software engineering.

KEYWORDS

Software Development
 Model of profession
 Software Engineering

INTRODUCCIÓN

La tendencia moderna del desarrollo de software es convertir esta actividad en una profesión de ingeniería. La realidad de software defectuoso evidencia una práctica más artesanal que profesional en la elaboración de productos de software [9]. Sin embargo, la incorporación del software de misión crítica en actividades humanas demanda calidad en los productos de software, que implica la necesidad desarrollar software dentro de los estándares y disciplinas de una profesión [14].

Debido al estado actual de innovación tecnológica, las habilidades artesanales no son suficientes para satisfacer la siempre creciente demanda de software de calidad [7]. Factores que dictan la necesidad de habilidades profesionales de ingeniería incluyen el tamaño y complejidad de los modernos sistemas con alto contenido de software, la variedad de destrezas requeridas para construir y mantener estos sistemas, y la continua expansión de la aplicabilidad del software [13].

Algunas organizaciones como el IEEE, la ACM, y ABET, referenciados ampliamente en el artículo, han estado uniendo esfuerzos con el propósito de consolidar el desarrollo de los diversos aspectos de una profesión de ingeniería de software, que lleve a sus practicantes a la elaboración de productos de software con la calidad suficiente para satisfacer las expectativas de sus usuarios, que sean desarrollados en el tiempo y con el presupuesto estimado.

Este artículo se organiza de la siguiente manera: se parte del modelo de componentes de una profesión propuesto por el Instituto de Ingeniería de Software (SEI), sobre esa base se adicionan algunos componentes y se agrupan de acuerdo a su naturaleza para proponer un modelo de profesión de ingeniería de software. Luego se expone ampliamente el estado de desarrollo de cada componente siguiendo el modelo propuesto, para terminar con algunas reflexiones sobre lo que esto implica.

1. MODELO DE COMPONENTES PARA UNA PROFESIÓN DE INGENIERÍA DE SOFTWARE

Partiendo de algunas definiciones de profesión, en las que se describen como características notables: la necesidad de conocimiento especializado, la preparación intensa, y el entrenamiento formal, Ford y Gibbs miembros del SEI, desarrollaron un modelo para establecer los componentes que caracterizan una profesión, validaron el modelo con profesiones ampliamente establecidas como medicina, leyes e ingeniería, estableciendo el estado de desarrollo de cada uno de sus componentes. Posteriormente realizaron un examen del modelo aplicándolo a la ingeniería de software para determinar el estado del arte de la profesión, su nivel de madurez, y predecir cual debería ser su desarrollo futuro [8].

El modelo de Ford y Gibbs, que describe dos niveles de componentes se presenta en la tabla 1.

Tabla 1. Modelo de componentes de una profesión del SEI.

Nivel de Practicante	Nivel de Infraestructura
Profesionales	Educación profesional inicial
Conocimiento	Acreditación
Práctica profesional	Desarrollo de habilidades
	Certificación
	Licenciamiento
	Desarrollo profesional
	Código de ética
	Sociedad profesional

Aunque existen trabajos similares como [7] del 2006, y [13] del 2000, el modelo de Ford y Gibbs de 1996 es el trabajo más referenciado constituyéndose en un ícono con respecto a la formalización de la ingeniería de software como profesión.

Sobre esta base, y considerando los esfuerzos de la comunidad desarrolladora de software, se propone un modelo de componentes para la profesión de ingeniería de software que toma los componentes del modelo del SEI, adiciona algunos que se derivan de la investigación y los agrupa de acuerdo a su naturaleza en cuatro tipos: conocimiento, educación, profesión y actividades como se describe en la Tabla 2. El **conocimiento** define la ontología de la profesión, la **educación** se presenta como medio para la formación de profesionales, el componente **profesión** abarca aspectos del ejercicio profesional, y las **actividades** constituyen el componente dinámico que nutre su desarrollo. El **código de ética** que se refiere a la responsabilidad social de la profesión y se le considera, en este modelo, como componente transversal a todos los demás. En las secciones subsiguientes se presentan

referencias que permiten dar cuenta del estado de cada componente del modelo, evocando la validación del modelo de Ford y Gibbs [8].

Tabla 2. Modelo de componentes de una profesión de ingeniería de software

Tipo de Componente	Conocimiento documentado	Educación	Profesión	Actividades
Componentes	Cuerpo de conocimientos de la ingeniería de software (SWEBOK)	Modelos curriculares	Certificación	Conferencias y talleres
	Estándares	Programas educativos	Licenciamiento	
	Publicaciones	Acreditación de programas educativos	Sociedades de profesionales	
	Procesos de trabajo			
Código de ética				

2. CONOCIMIENTO DOCUMENTADO

En este tipo de componente se encuentran: el cuerpo de conocimientos de la ingeniería de software (SWEBOK), algunos estándares y publicaciones sobre tópicos relacionados con la profesión, y los procesos de trabajo.

2.1 Cuerpo de conocimientos de la ingeniería de software (swobok)

SWEBOK es un proyecto del Institute of Electrical and Electronic Engineers - Computer Society (IEEE-CS), que establece una línea de base para el cuerpo de conocimientos de la ingeniería de software y promueve el avance de la teoría y la práctica en este campo. El SWEBOK agrupa tres categorías de conocimiento:

- Generalmente aceptado
- Avanzado y de investigación
- Especializado

Como "conocimiento generalmente aceptado" se considera al conocimiento y las prácticas aplicables a la mayoría de los proyectos, la mayor parte del tiempo, y por lo cual existe un consenso general a causa de su valor y utilidad. El conocimiento avanzado y de investigación abarca nuevas prácticas innovadoras probadas y usadas solo por algunas organizaciones, y que se encuentran en desarrollo en organizaciones de investigación. El conocimiento especializado reúne las prácticas usadas solo para cierto tipo de software [17].

El SWEBOK incorpora las siguientes 10 áreas de conocimiento, que se presentan en la tabla 3, distribuidas en dos categorías, las orientadas al proyecto y las orientadas a la organización:

Tabla 3. Áreas de conocimiento del SWEBOK.

Áreas de conocimiento orientadas al proyecto	Áreas de conocimiento orientadas a la organización
Requerimientos de software	Administración de la configuración de software
Diseño de software	Administración de ingeniería de software
Construcción de software	Procesos de ingeniería de software
Prueba de software	Métodos y herramientas de ingeniería de software
Mantenimiento de software	Calidad de software

Además de las 10 áreas de conocimiento de la ingeniería de software, el SWEBOK menciona otras 10 áreas relacionadas, entre las cuales se encuentran:

- Ingeniería de computadoras
- Ciencias computacionales
- Administración
- Matemáticas
- Gestión de proyectos
- Gestión de calidad
- Ergonomía de software
- Ingeniería de sistemas

Después de cuatro fases, la última versión del SWEBOK, IRON MAN del 2004 se considera estable [17].

2.2 Estándares

El IEEE-CS posee un comité de estándares de ingeniería de software conformado por voluntarios, entre cuyos propósitos están [19]:

- Codificar las normas de las prácticas profesionales de ingeniería de software en estándares.
- Promover el uso de estándares de ingeniería de software entre clientes, practicantes y educadores.
- Armonizar el desarrollo de estándares de ingeniería de software a nivel nacional e internacional.

Actualmente existen más de 39 estándares aprobados que cubren una amplia variedad de tópicos en ingeniería de software. Estos comités se ocupan de la creación de nuevos estándares y la actualización de los existentes.

2.3 Publicaciones

Existen varias publicaciones arbitradas sobre ingeniería de software, entre las que se encuentran: IEEE Transactions on Software Engineering [6] y ACM Transactions on Software Engineering and Methodology [2]. La revista IEEE SOFTWARE [5] y un boletín de la ACM Software Engineering Notes [1] publican artículos donde se registran progresos logrados en la ingeniería de software. Algunas revistas sobre informática como [13] y [30] han dedicado ediciones completas a la ingeniería de software.

2.4 Procesos de trabajo documentados

El modelo de madurez de capacidad CMM, actualmente CMMi, desarrollado por el SEI, es un enfoque para el mejoramiento de procesos que provee a las organizaciones los elementos esenciales para la efectividad en sus procesos [29]. La organización internacional de estándares ha continuado el trabajo logrando la aprobación del estándar ISO 15504 (conocido como modelo SPICE: Software Process Improvement and Capability dEtermination) para el mejoramiento de procesos de software y la determinación de la calidad [22].

3 EDUCACIÓN

En este tipo de componentes se encuentran: los modelos curriculares, que dan lineamientos para la creación de programas formales; los programas educativos, que comprenden niveles de estudio desde pregrado hasta doctorados; y los esquemas de acreditación de programas educativos, que vienen desarrollándose con

el fin de proporcionar una base formal a los programas educativos que se crean en ingeniería de software.

3.1 Modelos curriculares

En el área de la educación se han desarrollado modelos curriculares que proponen claramente el establecimiento de programas de educación superior y posgrados enfocados en la ingeniería de software. El trabajo unido de IEEE-CS y ACM sobre "Modelo Curricular para la Computación" (CC) ha creado cinco volúmenes de recomendaciones curriculares en: ciencias computacionales (CC2001), sistemas de información (IS2002), ingeniería de computadoras (CE2004), Tecnologías de la Información (IT2006), e ingeniería de software (SE2004), documentos que se pueden encontrar en [28] y [4].

La construcción del volumen SE2004 se ha centrado en tres esfuerzos principales. El primero es el desarrollo de metas curriculares y una declaración de lo que cada graduado en ingeniería de software (SE) debería saber. El segundo es la determinación y especificación del conocimiento que debe incluirse en programas de ingeniería de software de pregrado, Software Engineering Education Knowledge (SEEK). El tercero es la construcción de un conjunto de recomendaciones, que describen como un currículo de ingeniería de software que incorpore el SEEK puede ser construido en diferentes contextos. El borrador de SEEK abarca las siguientes 8 áreas: fundamentos, requerimientos, diseño, construcción, mantenimiento, procesos, calidad y administración [18].

3.2 Programas educativos

Tradicionalmente, han sido graduados en ciencias computacionales quienes se han dedicado al desarrollo de software con las consecuentes debilidades en la aplicación de principios y técnicas de ingeniería, debido a que por tradición estos programas de estudio no incluyen esos tópicos. La similitud entre los programas educativos en ciencias computacionales e ingeniería de software es semejante a la que existe entre los programas de física e ingeniería eléctrica. Los programas científicos de investigación se preocupan con el descubrimiento de principios básicos, su comprensión y su transmisión, los de ingeniería se preocupan además, de la aplicación de estos principios básicos a la resolución de problemas de interés para la sociedad y al desarrollo de nuevas técnicas para resolver problemas a medida que se van presentando [12].

Esta situación ha generado la necesidad de establecer programas educativos formales a nivel de estudios

superiores en ingeniería de software. En los Estados Unidos los primeros programas de master en ingeniería de software fueron introducidos alrededor de 1980, en la Universidad de Seattle y en el Instituto Wang, el primer programa de pregrado en ingeniería de software fue introducido en el Instituto de Tecnología de Rochester en 1996, y el primer programa de doctorado fue introducido en la Universidad de Carnegie Mellon en el 2000 [10]. El departamento de ciencias computacionales de la Universidad de Sheffield en el Reino Unido ha ofrecido un programa de ingeniería de software de pregrado desde 1988. El Reino Unido y Canadá, así como países de Europa del Este, se han unido a la lista desarrollando un enfoque curricular particular partiendo de los modelos curriculares de IEEE-CS y ACM [21], [11].

3.3 Acreditación de programas educativos

Para garantizar la calidad de los programas educativos existen organismos como el Accreditation Board for Science and Engineering (ABET), que desarrolla criterios y acredita programas en ciencias e ingeniería computacionales en Estados Unidos y algunas localizaciones internacionales.

Para ABET: "la acreditación es usada para asegurar la calidad de las instituciones educativas y de sus programas. La acreditación es un proceso voluntario no gubernamental, de revisión de pares. Requiere de las instituciones o programas, mantener ciertos criterios o estándares definidos. Aunque en ocasiones se confunde con la certificación, en general, se acreditan programas e instituciones y se certifican personas". [3].

En este ámbito, se creó la comisión de acreditación en computación, Computing Accreditation Commission (CAC/ABET). CAC es responsable por la acreditación de programas de pregrado en ciencias computacionales e ingeniería de software [25].

4. PROFESIÓN

En este tipo de componentes se incluyen: la certificación, y el licenciamiento, componentes que impulsan el establecimiento de cualquier profesión, incluida la ingeniería de software; y las sociedades profesionales, que se constituyen en agentes del desarrollo de la práctica profesional.

4.1 Certificación

La certificación es un reconocimiento sobre el dominio o conjunto de habilidades que posee una persona en un área de estudio y los certificados son expedidos por

agencias certificadoras sobre la base de algún criterio objetivo; la mayoría de las veces son pruebas de conocimientos.

El IEEE-CS ha desarrollado un proceso para certificar desarrolladores de software como profesionales certificados en desarrollo de software, Certified Software Development Professional (CSDP) [24]. El sitio Web incluye referencias recomendadas, preguntas de ejemplo, una guía a la terminología en ingeniería de software, y una guía de recursos.

4.2 Licenciamiento

El licenciamiento es un proceso obligatorio administrado por una entidad gubernamental y es un componente clave para el ejercicio de la ingeniería de software como profesión. El estado de Texas en Estados Unidos aprobó un estándar de licenciamiento y ha estado otorgando licencias desde 1998 [16]. En el Reino Unido y Canadá se han estado otorgando licencias para el desempeño de profesionales en ingeniería de software [21].

4.3 Sociedades de profesionales

Las dos sociedades de profesionales más importantes que dan soporte a la ingeniería de software son la IEEE-CS y la Association for Computing Machinery (ACM). La IEEE-CS es una sociedad de interés especial del Institute of Electrical and Electronic Engineers (IEEE) [26]. Por su parte, la ACM incluye el Grupo de Interés Especial por la Ingeniería de Software, Special Interest Group on Software Engineering (SIGSOFT) [23].

5. ACTIVIDADES RELACIONADAS CON LA INGENIERÍA DE SOFTWARE

Este tipo de componente incluye: las conferencias, y los talleres, como actividades que promueven el intercambio de información referente a la profesión; fue en una conferencia de la NATO donde por primera vez se mencionó el término "Ingeniería de Software".

5.1 Conferencias y talleres de trabajo

Con el patrocinio conjunto de ACM-SIGSOFT, y IEEE-CS, se realiza desde 1975 el evento más importante a nivel mundial sobre el tema: la Conferencia Internacional de Ingeniería de Software. La última versión se realizó en Shanghai en el 2006 [27]. Estas dos organizaciones patrocinan numerosos talleres de trabajo y seminarios de interés especial sobre una gran variedad de tópicos de ingeniería de software [20].

6. CÓDIGO DE ÉTICA

El código de ética se ha descrito en el modelo como un componente transversal con la intención de que se le considere pertinente en todos los demás componentes. Un código de ética se elabora con el fin de blindar la práctica de una profesión. Los profesionales velan por su cumplimiento en su práctica y la de sus colegas, y quienes lo incumplan son sancionados. El código de ética inicia con la siguiente declaración: "Los ingenieros de software se comprometen con ellos mismos a hacer del análisis, la especificación, el diseño, el desarrollo, la prueba y el mantenimiento del software, una profesión respetada. En concordancia con su compromiso con la salud, la seguridad y la protección del público, los ingenieros de software se adherirán a los siguientes ocho principios". Los ocho principios especifican consideraciones éticas relacionadas con: el público, el cliente y empleador, el producto, el juicio, la administración, la profesión, los colegas y el profesional mismo.

El código ha sido aprobado conjuntamente por la ACM y la IEEE-CS como el estándar para la enseñanza y la práctica de la ingeniería de software [15].

7. CONCLUSIONES

En este trabajo se puede observar como la comunidad de desarrolladores de software, en respuesta a su compromiso social de satisfacer la creciente demanda de productos de software de calidad, converge hacia la constitución de una profesión de ingeniería mediante el desarrollo de los componentes propios de una profesión.

Sobre la base del modelo de componentes para una profesión del SEI, se ha propuesto un modelo que clasifica los componentes de la profesión de acuerdo a su naturaleza, proveyendo argumentos para concluir que, a pesar de que la ingeniería de software a menudo es considerada como una profesión 'joven e inmadura', existe un conjunto de evidencias concretas, como las presentadas en este documento, que indican que el desarrollo de software se encuentra en evolución hacia una profesión de ingeniería de software que se encuentra en proceso de maduración.

8. REFERENCIAS

- [1] ACM Software Engineering Notes. <https://oldwww.cs.aau.dk/research/IS/library/SENotes.html>
- [2] ACM Transactions on Software Engineering. <http://tosem.acm.org/>
- [3] Accreditation Board for Science and Engineering (ABET). http://www.abet.org/about_ac.html
- [4] Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series. <http://sites.computer.org/ccse/SE2004Volume.pdf>
- [5] IEEE SOFTWARE. <http://www.computer.org/software/>
- [6] IEEE Transactions on Software Engineering. <http://www.computer.org/tse/archives.htm>
- [7] Fairley, R. y Tripp, L. Software Engineering, moving from craft to profession, 2006.
- [8] Ford, G. & Gibbs, N. A Mature Profession of Software Engineering, SEI Report CMU/SEI-96-TR-004, Carnegie Mellon University, 1996.
- [9] Kaner, Cem, "Computer Malpractice," Software QA, Volume 3, no. 4, 1997, p. 23.
- [10] Lutz, M.J. and Bagert, D. "Guest Editors' Introduction: Software Engineering Curriculum Development". Software, IEEE Vol. 23, Issue 6, pp 16 – 18, Nov.-Dec. 2006.
- [11] Ndvrat, Pavol. Bielikova, Mdria. Software Engineering Education: Different Contexts, Similar Contents. ACM SIGCSE Bulletin, Volume 31, Issue 2 (June 1999), Year of Publication: 1999.
- [12] Parnas, David L. "Software Engineering Programmes Are Not Computer Science Programmes," IEEE Software, November/December 1999.
- [13] Sánz, L. El futuro de la ingeniería de software o ¿Cuándo será el software un producto de ingeniería? Novática, Mayo-Junio 2000.
- [14] Shaw, Mary. Prospects for an Engineering Discipline of Software, IEEE Software, v.7 n.6, p.15-24, November 1990
- [15] Short and long forms of the ACM/IEEE-CS code of ethics. <http://computer.org/certification/ethics.htm>
- [16] Simmons, D.B. "Software engineering licenses in texas". Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International pp 346 - 346, 26-29 Aug. 2002.
- [17] Software Engineering Body of Knowledge 2004 version. A project of the IEEE Computer Society Professional Practices Committee. <http://www.swbok.org>
- [18] Software Engineering Education Knowledge Area descriptions. <http://sites.computer.org/ccse/artifacts/KADescriptions>
- [19] Software Engineering Standards Committee of IEEE-CS. <http://standards.computer.org/sesc/>
- [20] Special Interest Group on Software Engineering (SIGSOFT). <http://www.sigsoft.org/>

- [21] Speed, John. What do you mean I can't call myself a software engineering. PE. Texas Board of Professionals Engineering. IEEE Software, v.16 n.6, p45-50, November 1999.
- [22] SPICE documents, V 2.0.<http://www.sqi.gu.edu.au/spice/>
- [23] The Association for Computing Machinery. <http://www.acm.org>
- [24] The Certified Software Development Professional program of the IEEE Computer Society. <http://computer.org/certification>
- [25] The Computing Accreditation Commission of ABET (CAC/ABET). <http://www.abet.org/cac1.html>
- [26] The IEEE Computer Society. <http://www.computer.org>
- [27] The International Conference on Software Engineering (ICSE-2006). <http://www.isr.uci.edu/icse-06/>
- [28] Computing Curricula, The Join Task Force for Computing Curricula. A Cooperative Project of The Association for Computing Machinery (ACM), The Association for Information Systems (AIS), and The Computer Society (IEEE-CS), 30 de Septiembre 2005. <http://www.computer.org/education/cc2005>
- [29] The Software Engineering Institute of Carnegie Mellon University. <http://www.sei.cmu.edu/cmml/>
- [30] Upgrade, the European Journal for the Informatics professional. Software Engineering, State of an Art. Vol. IV, No 4, Agosto 2003. <http://www.upgrade-cepis.org>

Nota: todas las referencias de Internet fueron verificadas durante el primer semestre de 2008.