

# COMPETENCIAS PROFESIONALES Y ORGANIZACIONALES EN EL PROCESO DE DESARROLLO DE SOFTWARE

Albeiro Cuesta Meza  
Docente Investigador Grupo Ingeniería de Software UAM  
Universidad Autónoma de Manizales  
Manizales, Colombia  
albeirocuesta@hotmail.com

Marcelo López Trujillo  
Profesor Asociado Universidad de Caldas  
Manizales, Colombia  
mlopez@ucaldas.edu.co

Luís Joyanes Aguilar  
Catedrático Departamento de Lenguajes e Ingeniería del Software Universidad Pontificia de Salamanca  
Madrid, España  
luis.joyanes@upsam.net

## RESUMEN

El proceso de desarrollo de software sigue evolucionando con la aparición de componentes, sistemas multi-agentes, ontologías, computación orientada al servicio, mejores niveles de confiabilidad (confidencialidad, seguridad) e incremento de la calidad; todo esto sobre el paradigma de la Web 2.0 y la Web 3.0. Las competencias personales y organizacionales que se presentan en este artículo, están pensadas desde la gestión de conocimiento y la complejidad del negocio de software en la era de la economía del conocimiento.

**Palabras Claves:** Proceso de desarrollo de software, competencias, gestión de conocimiento, software como servicio, web 2.0, inteligencia colectiva, Industria del Software.

## 1. INTRODUCCIÓN

La industria del software y su ciclo tradicional de desarrollo de software con: la planificación, el análisis de requerimientos, el diseño y desarrollo, la implementación, la integración y las pruebas, la evaluación, la entrega y el soporte; tiene nuevos retos marcados por la pérdida de competitividad, la desalineación entre el estado del arte y la práctica, los nuevos paradigmas de la provisión de servicios (Software as a Service) y del cambio radical en el desarrollo de los sistemas de software mediado por la aparición de componentes, sistemas multiagentes, fábricas de software, desarrollo de software dirigido por modelos, líneas de producto de Software y por servicios certificados.

Estos cambios están apalancados por otras variaciones como la forma de hacer negocios *just in time*, el manejo oportuno y seguro de la innovación, las oficinas móviles y virtuales, el *offshoring and outsourcing*, la capacidad de adaptarse rápidamente a las necesidades de los clientes, el modelo de software libre, los ecosistemas digitales y los ambientes colaborativos. El desarrollo de software además de estar cada vez más orientado al cliente, debe reducir tiempos de desarrollo sin afectar la calidad, el desarrollo debe adoptar más valor al producto final, aprovechar la innovación como parte de desarrollo y responder a un mercado y entorno de desarrollo globalizado [1].

Los casos de éxito de la industria del software en el ámbito mundial tienen que ver también con políticas públicas locales, regionales y nacionales; con procesos de calidad sobre estándares internacionales como CMMI; con estrategias y avances en investigación e innovación; con servicios conexos de comercialización, mercadeo, soporte y mejora; y con el aprovechamiento de talento humano altamente calificado y competente para la industria.

El concepto de competencia personal y organizacional para este entorno debe ser asumido desde una perspectiva compleja, como un saber hacer, razonado para enfrentarse a la incertidumbre. De esta manera, las competencias no podrían abordarse únicamente como comportamientos observables, sino como una compleja estructura de atributos necesarios para el desempeño en situaciones diversas, donde se combinan conocimientos (tácitos y explícitos), habilidades, actitudes y valores, con tareas que se tienen que desempeñar en determinadas situaciones.

Para lograr el propósito de este artículo presentamos el nuevo marco del proceso de desarrollo de software y el concepto y propuesta de competencias para la industria de software.

## 2. LOS CAMBIOS EN EL DESARROLLO DE SOFTWARE

El desarrollo de software es una actividad intensiva en capital humano, más que el desarrollo de otras tecnologías, este desarrollo se basa en los equipos humanos de trabajo y en menor medida, en inversiones tangibles; Esta situación implica la aparición de un modelo horizontal de división del trabajo que se superpone parcialmente con el vertical clásico. A cada integrante del equipo de trabajo se le va a exigir un conocimiento mayor sobre la totalidad del desarrollo (enfoque sistémico) aunque su actividad enfatice aspectos concretos (especialidad).

Las consecuencias no solamente tienen que ver con una estructura específica del talento humano propuesto, sino que el equipo humano formado por especialistas con formación y actividades más horizontales, requiere de técnicas de gestión y de interacciones distintas y una formación permanente. El objetivo ideal estriba en conseguir que el equipo funcione globalmente como lo haría un experto que tuviera todos los conocimientos requeridos.

El proceso de desarrollo ha estado vinculado a los modelos de gestión como CMMI, ISO e ITIL; a los modelos de gerencia de proyectos como PMI, IPMA, GPMA; a los marcos para el desarrollo como RUP, AUML, XP, SCRUM, MSF; todos estos estándares en permanente evolución y adaptación [2]. También ha estado ligado al estado del arte de la ingeniería del software, desde los modelos estructurados, orientados a los objetos, a los aspectos o a los componentes, hasta las propuestas de agentes y sistemas multi-agentes y ontologías de la Web 2.0 y la Web 3.0.

La Web 2.0 es la combinación de herramientas y tecnologías bajo una estrategia de negocios colaborativa y de inteligencia colectiva soportada en tendencias sociales y culturales que conduce a la creación individual y colectiva de contenidos en Internet. Las organizaciones que desarrollan software son comunidades con una estructura social y una cultura organizacional que pueden usar las wikis<sup>1</sup> para generar ambientes de colaboración y resolver problemas; usar los blogs<sup>2</sup> para motivar a los grupos de interés e informar de las dimensiones del proceso; ampliar la inteligencia colectiva a lo largo del

proceso de desarrollo permitiendo la participación en el proceso de aliados del negocio, ciudadanos al azar, empleados en formación o diversos grupos de interés (*Stakeholders*).

La Web 2.0 permite colaborar en el mejoramiento del proceso usando los wikis y los blogs para el manejo de las lecciones aprendidas, difusión de técnicas y *tips*, socialización de las mejores y las peores prácticas. Además, permitiendo la participación de actores externos al proceso de desarrollo se puede anticipar a las pretensiones siempre cambiantes de los clientes y el mercado.

La Web 3.0 es la Web semántica que mantiene los principios de las Web que la preceden, como son los principios de descentralización, compartición, compatibilidad, máxima facilidad de acceso y contribución, o la apertura al crecimiento y uso no previstos de antemano [3].

En este entorno es clave alcanzar un entendimiento entre las partes que han de intervenir en la construcción y explotación de la web: usuarios, desarrolladores y programas de muy diverso perfil. La Web 3.0 está ligada a la inteligencia computacional como base para cumplir el propósito de la inteligencia colectiva.

El software como servicio es una forma de ofrecer aplicaciones a través de Internet como una evolución del ASP (*application server provider*), como un software basado en Web, software *on-demand* o software albergado. Independientemente de su nombre, las aplicaciones SaaS se ejecutan en los servidores del proveedor SaaS. El proveedor es el encargado de gestionar el acceso a la aplicación, incluyendo los aspectos relacionados con la seguridad, la disponibilidad y el rendimiento.

Las ventajas se dan por la posibilidad de que el usuario únicamente se tiene que preocupar por usar el servicio, sin requerir licencia por número de usuarios, ni costos de implementación, reduciéndose el tiempo de implementación; el soporte técnico se realiza en línea por lo que su costo y tiempo de respuesta es menor y el usuario recibe de forma transparente actualizaciones, configuraciones de servidor.

Este modelo se ha dado en consonancia con la evolución del Internet y de tecnologías de software como objetos, ambientes, componentes, fábricas, agentes y ontologías.

El mismo servicio de hosting, de hospedaje de aplicaciones ha pasado del modelo de pagar por una licencia a perpetuidad y pagar por sus actualizaciones a pagar por uso al mes, pagar por transacción o pagar por almacenamiento.

---

<sup>1</sup> Puede ser internas o externas apoyadas en herramientas como Twiki, mediawiki, pbwiki, wikipatters, entre otros.

<sup>2</sup> Pueden ser internos o externos apoyados en microblogs (twitter) y de herramientas para manejo de comunidades y directorios, creación y alojamiento y rankings.

### 3. COMPETENCIAS

Las competencias tienen diferentes nociones desde diversas disciplinas como la lingüística, la filosofía, la sociología, la psicología cognitiva y laboral [4]. Para concebir las competencias del talento humano y de la organización involucradas en el proceso de desarrollo de software se deben tener en cuenta los planteamientos realizados desde la academia (formadora de profesionales de ingeniería de software) y del sector productivo: casas de software, clúster de software, parques tecnológicos de software, departamentos o unidades de desarrollo de software.

Teniendo en cuenta que las competencias son estructuras complejas de procesos que las personas ponen en acción-actuación-creación para resolver problemas y realizar actividades (de la vida cotidiana y del contexto laboral profesional), orientadas a la construcción y transformación de la realidad [5].

Las competencias integran el saber conocer (observar, analizar, comprender y explicar), el saber hacer (desempeño basado en procedimientos y estrategias), el saber estar (participación y trabajo colaborador) y el saber ser (auto motivación, iniciativa, liderazgo y creatividad); teniendo en cuenta los requerimientos específicos del entorno, las necesidades personales y las condiciones de incertidumbre, con autonomía intelectual, conciencia crítica y espíritu de reto, asumiendo al mismo tiempo las consecuencias de sus actos y buscando el bienestar humano.

Las competencias, en definitiva, están constituidas por procesos subyacentes (cognitivo-afectivos), así como también por procesos públicos y demostrables, que permitan elaborar algo de sí para los demás con rigurosidad [6].

Las competencias organizacionales que se deben desarrollar son las de comunicación, cooperación, aprender colectivamente la experticia y representaciones compartidas para captar con mayor precisión las preferencias y necesidades de los usuarios, para detectar (vigilancia e inteligencia) las amenazas, presiones y movimientos de la competencia, para hacer prospectiva en nuevos productos, servicios e ingresos, para maximizar la eficiencia operacional y de costos, para participar en la definición de requisitos regulatorios y legislativos, para reducir el ciclo de tiempo del mercado y del aprovechamiento de las oportunidades.

Para ACM e IEEE las competencias están asociadas a las disciplinas: ciencias de la computación, ingeniería de computadores, tecnologías de la información y de la comunicación y sistemas de información e ingeniería del software.

Las competencias personales de la ingeniería del software tienen que ver con integrar soluciones de TIC y procesos empresariales para satisfacer las necesidades de información de las organizaciones; permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente. Determinar los requisitos de los sistemas de información y comunicación de una organización y ser activos en su especificación, diseño e implementación; comprender los principios y prácticas de las organizaciones, de forma que sean capaces de ejercer como enlace entre las comunidades técnica y de gestión dentro de una organización; y con construir, configurar, gestionar, mantener y evaluar servicios y sistemas de información.

Las competencias transversales tienen que ver con habilidades de análisis, síntesis y evaluación, trabajar de forma autónoma y con capacidad de adaptación, espíritu innovador y emprendedor, trabajo en equipo, comunicación eficaz a nivel oral y escrito, conocimiento y uso de otros idiomas y actuar con compromiso ético y social.

Para que las personas involucradas en el proceso de desarrollo de software potencien su perfil personal y organizacional deben incorporar competencias sociales, cognitivas, interpersonales, creativas, sistémicas y científico-tecnológicas.

Las competencias sociales tienen que ver con la visión compartida, la convivencia democrática, el uso sostenible de los recursos, la sensibilidad cultural y el pensamiento emocional. Las cognoscitivas con la percepción y la acción, el juicio crítico y la capacidad de análisis y síntesis, las interpersonales con el compromiso ético, el reconocimiento y respeto a la multi-culturalidad y la diversidad, la capacidad crítica y autocrítica, y las habilidades interpersonales. Las creativas con el pensamiento analógico, los lenguajes visuales y la solución de problemas. Las sistémicas con el pensamiento complejo, la auto-gestión y la capacidad para integrarse, aprender a conocerse y comunicarse con expertos en distintos contextos. Las científico-tecnológicas con aprender a estar, aprender a ser, aprender a hacer y aprender a aprender.

Para valorar estas competencias en los profesionales de ingeniería del software, se propone recopilar evidencias y realizar un dictamen de estas evidencias a partir de criterios preestablecidos, con el propósito final de valorar como mejorar la competencia. Sus elementos son: las competencias, las pautas de evaluación, la idoneidad y las evidencias de la competencia.

Las evidencias de estas competencias están asociadas a pruebas escritas abiertas o cerradas, ensayos y pruebas orales; evidencias del ser: registro de comportamientos y actitudes, auto-valoración, representaciones colaborativas y cooperativas; evidencias del hacer a través de audios y

videos, testimonios, registro de observaciones (para lo que esta siendo muy útil el software social). Y evidencias del producto o servicio de software: creaciones, informes finales, documentos digitales y programas objeto.

#### **4. CONCLUSIONES**

El desarrollo de software orientado al servicio es una respuesta a los cambios radicales del proceso de desarrollo; a la rapidez y el vértigo de la competitividad y la productividad; al reto de una sociedad del conocimiento incluyente y de beneficio social, cultural y económico para todos.

Sus posibilidades están estrechamente ligadas a los nuevos paradigmas de las redes sociales y la inteligencia colectiva; a las competencias personales y organizacionales que tienen que ver no solo con las competencias generales asociadas a la ingeniería del software sino también con las competencias sociales, cognitivas, sistémicas, creativas e interpersonales del ciudadano del siglo XXI inmerso en la sociedad del conocimiento.

Las competencias y su evaluación se deben personalizar para cada uno de los roles de la industria del software, ya sea el programador, el analista, el documentalista, el de pruebas, el arquitecto de bases de datos, el arquitecto de redes, el de soporte, el de calidad o el líder; o también por los roles de gestión de proyectos, proceso de desarrollo, aseguramiento de la calidad, comercialización y mercadeo, soporte y servicio, investigación e innovación.

#### **5. REFERENCIAS**

- [1] L. Joyanes, Las redes sociales: de la mensajería instantánea a los WebLogs, Utopía y Sociedad, Universidad Pontificia de Salamanca, Madrid, 2004, pp. 24-32.
- [2] Universidad EAFIT, Prácticas y herramientas que apalancan el proceso de desarrollo de software, Medellín, 2006.
- [3] P. Castells, La Web semántica, Revista de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, Madrid, 2006.
- [4] M. Lopez, A. Cuesta y L. Joyanes, Ciudad-región Conocedora, Centro Editorial de la Universidad de Caldas, 2008.
- [5] R. Gallego, Competencias cognoscitivas. Un enfoque epistemológico, pedagógico y didáctico, Cooperativa Editorial Magisterio, Bogotá, 1999.
- [6] C. Fernández, M. Salinero, Las competencias en el marco de la convergencia europea: Un nuevo concepto para el diseño de programas educativos, Encounters on Education, 2006, pp. 131-153.