

Fundamentación de la Línea de Investigación en Ingeniería de Software

Resumen

La ingeniería de software, a pesar de su historia reciente, ha permitido que el desarrollo de software pase de ser una actividad de nivel casi personal a una actividad de tipo industrial. A lo largo de la investigación realizada se buscó identificar las organizaciones a nivel mundial, los modelos y estándares que rigen la ingeniería de software, a fin de que la línea de investigación que lleva el mismo nombre tenga referentes claros para futuros trabajos de investigación en dicha área, a cargo del programa de Tecnología en Sistemas de la Corporación Universitaria Adventista.

Contenido de la investigación

El crecimiento acelerado de la tecnología ha generado al mismo tiempo nuevas posibilidades como nuevos problemas. De acuerdo con la famosa Ley de Moore la tecnología

ha seguido mejorando en aspectos como el costo, niveles de miniaturización y capacidad de operaciones por segundo.

Claramente los aspectos en mención han favorecido el desarrollo de aplicaciones informáticas más avanzadas y complejas que en anteriores años. Por otro lado, todo este desarrollo tecnológico vertiginoso generó problemas adicionales a la inherente complejidad encontrada en el desarrollo de un producto de software. Entre estos problemas se encuentra el hecho de que los usuarios esperan que los nuevos aplicativos informáticos aprovechen herramientas y desarrollos tecnológicos que para ellos son transparentes pero que al momento de desarrollar un software implican grandes desafíos a nivel de estructura. La complejidad inherente a la construcción del software se visualiza fácilmente en los altos índices de

proyectos de software que no son terminados, otros que son terminados pero no de acuerdo con los cronogramas, requisitos y presupuestos acordados; y finalmente un pequeño grupo que logra cumplir con todos los aspectos acordados con el cliente. De acuerdo con Standish Group en su Chaos Report de 2003, el 15 % de proyectos eran cancelados, el 51 % sufrían alguna desviación (costo, tiempo, funcionalidad) y el 34 % eran exitosos. Al revisar reportes similares en años anteriores el porcentaje de proyectos exitosos era mucho menor que el actual, que alcanza un 35 % aproximadamente, pero sigue siendo un porcentaje de éxito bajo.

Hacia el año 1968 la OTAN acuña el término «Crisis del Software» debido a lo mencionado previamente. Es necesario definir entonces el término «ingeniería de software». Entre muchas definiciones se tiene la generada por la IEEE: «Estudio y aplicación de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería de software».

A partir de la «creación» de la ingeniería de software se hace necesario también definir algunos elementos como:

Áreas de conocimiento que la comprenden.

Procesos que intervienen en el desarrollo, mantenimiento y operación del software.

De las mejores prácticas, extraer modelos para el desarrollo de software.

Criterios unificados para tareas de requisitos, pruebas y gestión de la configuración, entre otras.

Ahora bien, todos estos elementos son definidos por diferentes organizaciones a nivel mundial y sus modelos estándares, tales como:

IEEE – Computer Society: Institute of Electrical and Electronics Engineers. Esta sociedad está dedicada a generar teorías avanzadas, prácticas y tecnologías de procesamiento de información de manera específica en la Computer Society. Esta organización responde a la primera pregunta sobre las áreas que componen la ingeniería de software a través del documento SWEBOK (Software Engineering Body of Knowledge).

ISO: La Organización Internacional para la Estandarización tiene como función principal buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional. El estándar actual es ISO 9001 de manera general, y su aplicación al proceso de desarrollo de software es dada por ISO 90003. Por otra parte tiene el estándar ISO/IEC 9126 para evaluación de la calidad de un producto software. Otros estándares son ISO/IEC 12207 e ISO/IEC TR 15504.

SEI: El Software Engineering Institute es financiado por el Departamento de Defensa de Estados Unidos y administrado por la

Universidad Carnegie Mellon. Es un referente en ingeniería de software por realizar el desarrollo del modelo SW-CMM (1991), que ha sido el punto de arranque de todos los que han ido formando parte del modelo que ha desarrollado sobre el concepto de capacidad y madurez, hasta el actual CMMI (Capacity Maturity Model Integrated).

ISACA e ITGI: Information Systems Audit and Control Association e IT Governance Institute. Ambas organizaciones han generado un modelo para la administración de tecnologías de la información y las comunicaciones. Su modelo estrella es COBIT.

OGC: Office of Government Commerce. Esta organización generó el modelo para administración de la tecnología llamado ITIL.

PMI: Project Management Institute. Este Instituto genera modelos y buenas prácticas para la administración de proyectos de software. Su modelo estrella es PMBOK (Project Management Body of Knowledge).

Las anteriores son solo algunas de las organizaciones que fortalecen el área de ingeniería de software integrando el ejercicio académico de grandes instituciones de educación superior a nivel mundial y empresas con altos niveles de calidad, por medio de la implementación de «buenas prácticas» en un proceso de desarrollo de software. Volviendo a las preguntas que requieren ser contestadas para este campo relativamente nuevo de la Ingeniería

de software, se tienen las siguientes relaciones con modelos y estándares internacionales de calidad:

Definición de las áreas: SWEBOK.

Definición de los procesos dentro del ciclo de vida del software: ISO/IEC 12207.

Modelos basados en mejores prácticas: CMM / CMMI - ISO/IEC TR 15504

Criterios de calidad para las áreas definidas en ISO/IEC 12207: IEEE 830 - IEEE 1362 - ISO/IEC 14764.

Son algunos de los referentes.

SWEBOK da el primer paso necesario para constituir la ingeniería del software como profesión: la delimitación del cuerpo de conocimiento que comprende la profesión. El proyecto parte de la suposición de que es necesario establecer cuál es el cuerpo de conocimiento que deben asimilar los ingenieros del software, y en su desarrollo ha agrupado este conocimiento en 10 áreas:

Requisitos

Diseño

Construcción

Pruebas

Mantenimiento

Gestión de la configuración

Gestión

Procesos

Herramientas y métodos Calidad.

Estas áreas son la esencia de la ingeniería de software; no obstante, no se especifican herramientas, lenguajes de programación y asuntos que dependen de la empresa desarrolladora

de software y sus proyectos en particular.

La forma de desarrollar software debe estar enmarcada también en un paradigma de desarrollo de software, entre los cuales se encuentran:

Lineal o secuencial

Cascada

Espiral

Incremental

Evolutivo

Prototipazo

Proceso unificado de desarrollo de software

Programación extrema.

Cada uno de estos paradigmas tiene sus particularidades y tipos de proyectos en los cuales sea más aplicable uno que otro o la combinación de algunos de ellos. Por encima de estas particularidades se cuenta con modelos para el mejoramiento del proceso de desarrollo de software tales como:

CMMI

Modelo IDEAL

MOPROSOFT

SPICE

Cada uno de estos modelos contempla los procesos con sus roles, artefactos y actividades necesarias para garantizar la calidad del proceso y por ende del producto. El más utilizado en la actualidad es el modelo CMMI del SEI. Además, estos modelos se encuentran en permanente adaptación a las necesidades de la empresa en general.

Uno de los aspectos más complejos es exteriorizar y formalizar el análisis y el diseño para cada proyecto de software, puesto que generalmente era

conocimiento tácito y como tal dependía enormemente del personal que pertenecía al equipo de desarrollo, dificultando tareas como modificación y actualización del software y por supuesto, la coordinación de equipos de desarrollo grandes. Para solucionar este problema, se usa el UML (Lenguaje Unificado de Modelamiento), el cual permite que equipos de desarrollo puedan entenderse en su interior e incluso con desarrolladores que no estuvieron involucrados en un proyecto específico.

Para concluir, es importante establecer cuáles son las áreas que la ingeniería de software contempla. De acuerdo con SWEBOK se tienen 10 áreas de conocimiento necesarias para el proceso de desarrollo de software, éstas son:

1. Ingeniería de requisitos: es la relacionada con la gestión de las necesidades de los clientes (requisitos). Tiene las siguientes subáreas:

- a. Principios de los requisitos de software
- b. Procesos de requisitos
- c. Elicitación de requisitos
- d. Análisis de requisitos
- e. Especificación de requisitos
- f. Validación de requisitos
- g. Consideraciones prácticas

2. Diseño de software: orientada a la definición de la arquitectura y componentes del software. Tiene las siguientes subáreas:

- a. Principios de diseño de software

- b. Aspectos claves en el diseño de software
 - c. Arquitectura y estructura de software
 - d. Evaluación y análisis de la calidad del diseño de software
 - e. Notaciones de diseño de software
 - f. Métodos y estrategias para el diseño de software.
3. Construcción de software: se refiere a la codificación del software a partir del diseño usando lenguajes de programación con herramientas pertinentes. Tiene las siguientes subáreas:
- a. Principios de construcción de software
 - b. Dirección de construcción
 - c. Consideraciones prácticas.
4. Pruebas de software: se refiere a la verificación dinámica del comportamiento del software usando casos de prueba. Tiene las siguientes subáreas:
- a. Principios de pruebas de software
 - b. Niveles de prueba
 - c. Técnicas de prueba
 - d. Medidas relacionadas con pruebas
 - e. Proceso de pruebas.
5. Mantenimiento del software: se relaciona con las modificaciones hechas al software una vez que esté ya está en operación. Tiene las siguientes subáreas:
- a. Principios de mantenimiento de software
 - b. Aspectos claves en el mantenimiento del software
 - c. Proceso de mantenimientos
 - d. Técnicas para el mantenimiento.
6. Gestión de la configuración de software (SCM): se relaciona con la sistematización de la identificación de cambios en los artefactos generados en cada fase con el fin de garantizar la rastreabilidad de los mismos. Tiene la siguientes subáreas:
- a. Gestión del proceso de SCM
 - b. Identificación de la configuración de software
 - c. Control de la configuración de software
 - d. Registro de estado de la configuración de software
 - e. Auditoría de la configuración de software
 - f. Entrega y gestión de versiones de software.
7. Gestión de la ingeniería de software: está orientada a la gestión cuantitativa del proceso. Tiene las siguientes subáreas:
- a. Inicio y definición del alcance
 - b. Planeación del proyecto de software
 - c. Pronunciamento del proyecto de software
 - d. Repaso y evaluación
 - e. Cierre
 - f. Gestión de la Ingeniería de software.
8. Proceso de ingeniería de software: se relaciona con todo el proceso desde el comienzo hasta el fin. Tiene las siguientes subáreas:
- a. Implementación y cambio del proceso
 - b. Definición del proceso
 - c. Valoración del proceso

- d. Medidas del producto y del proceso
9. Métodos y herramientas de ingeniería de software: busca garantizar la calidad del proceso. Tiene las siguientes subáreas:
- a. Principios de calidad de software
 - b. Proceso de gestión de la calidad de software
 - c. Consideraciones prácticas.
10. Disciplinas relacionadas con ingeniería de software: en esta área se definen las disciplinas que comparten límites con la Ingeniería de software. Estas disciplinas son:
- a. Ingeniería de computadores
 - b. Ciencia de los computadores
 - c. Administración
 - d. Matemáticas
 - e. Gestión de proyectos
 - f. Gestión de la calidad
 - g. Ergonomía de software
 - h. Ingeniería de sistemas.

Conclusiones

Tal como se ha visto, la ingeniería de software es un campo muy amplio que está en constante cambio a medida que la tecnología y las necesidades de los clientes hacen más complejo el desarrollo de aplicaciones informáticas. La investigación les permitirá a los futuros investigadores tener un referente claro sobre el área en mención, de manera que cualquier aspecto que se quiera estudiar tenga unos parámetros claros y definidos. Por otra parte, este trabajo de Fundamentación de la Línea de Investigación en Ingeniería de Software es un trabajo continuo debido a los cambios permanentes que se dan en ella; por ello se plantea la necesidad de fortalecer los semilleros de investigación en el programa de Tecnología en Sistemas por medio de los cursos relacionados con el área, a fin de que la experiencia investigativa de los estudiantes sea amplia, y la actualización del documento de *Fundamentación* sea una realidad permanente.