

SWEBOK

1. INTRODUCCIÓN

La IEEE Computer Society y la Association for Computing Machinery han trabajado en un proyecto conjunto para desarrollar una guía del Cuerpo de Conocimientos de la Ingeniería de Software (SWEBOK). Considerando que un cuerpo de conocimientos es un paso esencial hacia el desarrollo de una profesión debido a que representa un acuerdo general con respecto al contenido de la disciplina.

SWEBOK busca aglutinar en un solo texto las competencias que debiese tener todo ingeniero de software para desempeñarse competentemente en el mercado. Es un proyecto para clasificar y definir todo lo que es Ingeniería de Software (IS), pero antes de llegar a ésta guía fueron 5 años de trabajo. La idea fue que los expertos en IS del mundo dieran sus opiniones sobre la disciplina, sus fortalezas, debilidades y diferencias y para ello fue necesario llegar a un consenso. Estas ideas fueron canalizadas por un grupo de editores, quienes añadieron sus comentarios y dieron vida a esta guía.

Sin este acuerdo, no hay ninguna manera de validar un examen de licenciamiento, tener un plan de estudios para preparar a los individuos para el examen, o formular un criterio para acreditar el plan de estudios.

2. OBJETIVOS Y PÚBLICO

El proyecto SWEBOK tiene cinco objetivos:

1. Identificar el contenido de la disciplina de la Ingeniería de Software.
2. Proveer acceso al cuerpo de conocimientos de la Ingeniería de Software.
3. Promover una visión uniforme y consistente de la Ingeniería de Software a nivel mundial.
4. Aclarar el lugar de la Ingeniería de Software con respecto a otras disciplinas tales como, ciencias de la computación, gestión de proyectos, matemáticas, etc.
5. Proveer una fundamentación para el desarrollo del currículum (programas universitarios) y material de certificación individual.

El producto del proyecto de SWEBOK no será el cuerpo de conocimiento en sí mismo, sino una guía hacia él.

El conocimiento ya existe; nuestra meta es obtener un acuerdo general en el subconjunto del centro de conocimientos caracterizando la disciplina de la ingeniería de software.

Para lograr estas metas, el proyecto está orientado hacia una variedad amplia de audiencias como ser: Organizaciones públicas y privadas. Ingenieros de software practicantes. Elaboradores de políticas públicas. Sociedades profesionales. Estudiantes de Ingeniería de Software y Educadores y formadores.

3. LA GUÍA

El proyecto comprende tres fases: Strawman, Stoneman, y Ironman. La fase Strawman se completó dentro de los nueve meses de iniciación del proyecto, y sirvió como modelo para organizar la guía SWEBOK. La Primavera del 2000 vio la realización de la versión Stoneman, la fase Ironman, se desarrolló durante dos o tres años. Siguiendo los principios de la fase Stoneman, Ironman se benefició de los análisis más en profundidad, de un proceso de revisión más amplio, y de la experiencia ganada del desarrollo de la fase Stoneman.

La Guía de SWEBOK organiza el cuerpo de conocimientos en varias Áreas de Conocimiento (AC). En total se tienen 10 ACs. Asimismo considera ocho disciplinas relacionadas. (Ver Tabla 1)

AREAS DE CONOCIMIENTO	DISCIPLINAS RELACIONADAS
Requerimientos de Software	Ingeniería de la Computación
Diseño de Software	Ciencias de la Computación
Construcción de Software	Gestión
Prueba del Software	Matemáticas
Mantenimiento del Software	Gestión de Proyectos
Gestión de la Configuración Software	Gestión de la Calidad
Gestión de la Ingeniería de Software	Ergonomía del Software
Proceso de Ingeniería de Software	Ingeniería de Sistemas
Herramientas y Métodos en Ingeniería de Software	
Calidad del Software	

Tabla 1

Organización jerárquica

La Guía de SWEBOK usa una organización jerárquica para descomponer cada AC en un conjunto de temas con dos niveles. La Guía trata temas seleccionados de una manera compatible con grandes escuelas de pensamiento y con fallas generalmente encontradas en industrias, literatura y normas de la ingeniería de software. Después de todo, el cuerpo de conocimientos se encuentra en los temas a los que hace referencia, y no en la propia Guía.

Valuaciones

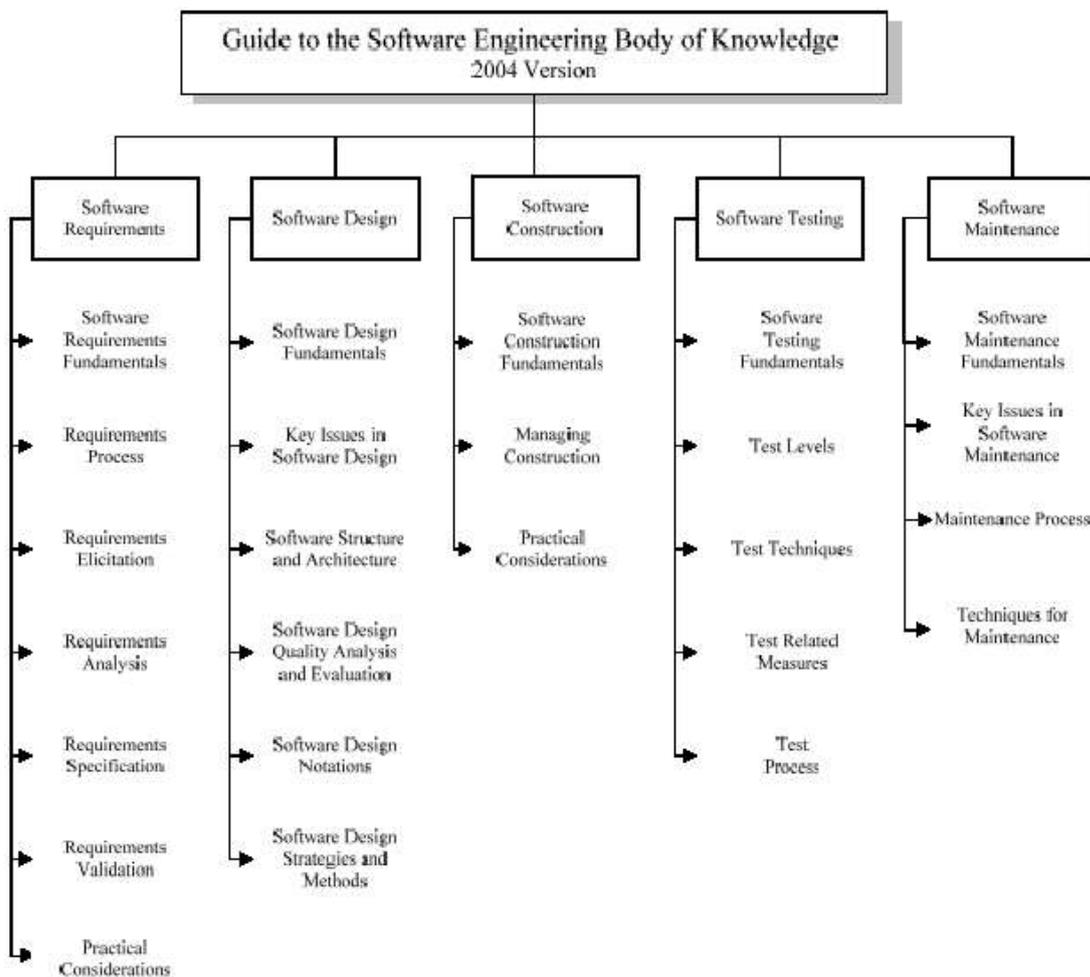
Como una ayuda, notable a diseñadores de planes de estudio, la guía también valua cada tema en un conjunto de categorías normalmente atribuidas a Benjamín Bloom. El concepto es que los objetivos educativos pueden ser clasificados en seis categorías representando el crecimiento a profundidad: conocimiento, comprensión, aplicación, análisis, síntesis, y evaluación

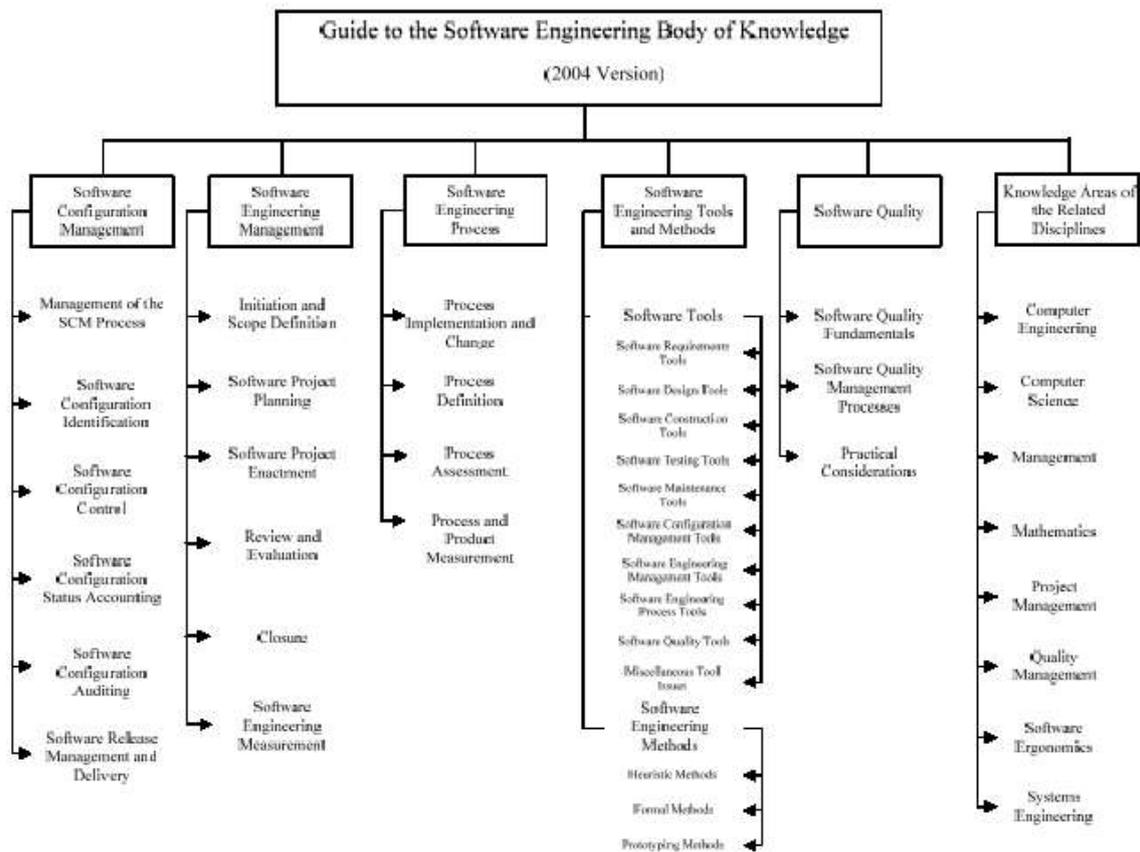
Áreas de Conocimiento y disciplinas relacionadas

Cada descripción de un Área de Conocimiento del SWEBOK también identifica la descripción de las disciplinas relacionadas con las ACs. Aunque éstas ACs se identifican sin descripciones o referencias adicionales, ayudando a los diseñadores de planes de estudio.

4. LAS ÁREAS DE CONOCIMIENTO

A continuación se describirá las 10 áreas de conocimiento y los tópicos importantes incorporados dentro de ellos.





4.1 REQUERIMIENTOS DE SOFTWARE

La subarea de análisis de requerimiento de software esta dividida en 5 subareas que corresponden a tareas que más que secuenciales suceden común e iterativamente. (ver figura).

Esta área trata los procesos de análisis de requerimientos para:

- ✚ detectar y resolver conflictos emergentes
- ✚ descubrir falencias del sistema
- ✚ cómo debería interactuar con su medio ambiente

4.2 DISEÑO DE SOFTWARE

Describe cómo el sistema se descompone y se organiza en componentes, y describe las interfaces entre estos componentes. Diseño también refina la descripción de estos componentes en un nivel de detalle conveniente por comenzar su construcción.

Tiene las siguientes subareas:

- Fundamentos de Diseño de Software
 - Proceso del Diseño de Software
 - Diseño arquitectónico
 - Diseño detallado

- Técnicas clave
- Abstracción
- Acoplamiento/Cohesión
- Descomposición y modularidad
- Encapsulamiento/ocultamiento de la información
- Separación de interfase e implementación
- Tópicos clave de Diseño de Software
 - Concurrencia
 - Control y Manejo de Eventos
 - Distribución de Componentes
 - Manejo de Errores y Excepciones y Tolerancia a fallos
 - Interacción y Presentación
 - Persistencia de Datos
- Estructura y Arquitectura de Software
 - Estructuras Arquitectónicas y Puntos de Vista
 - Patrones de Diseño
 - Familias de Programas y Frameworks
- Notaciones de Diseño de Software

**Descripciones Estructurales
(estáticas)**

- Diagramas de Clases y Objetos
- Diagramas de componentes
- Tarjetas de Clase-Responsabilidad-Colaborador
- Diagramas entidad-relación
- Diagramas de estructura de Jackson

**Descripciones de Comportamiento
(dinámicas)**

- Diagramas de actividad y transición de estados
- Diagramas de colaboración y secuencia
- Diagrama de flujo y de flujo de datos
- Diagramas y tablas de decisión
- Lenguajes de especificación formales
- Pseudocódigo

- Estrategias y Métodos del Diseño de Software
 - Estrategias Generales
 - Diseño Orientado a la Funcionalidad (estructurado)
 - Diseño Orientado a Objetos
 - Diseño Centrado en Estructuras de Datos
 - Diseño Basado en Componentes

4.3 CONSTRUCCIÓN DE SOFTWARE

La construcción de software es un acto fundamental de la ingeniería de software; programadores deben trabajar construyendo, lo que significa a través de codificar, la validación propia, y pruebas. Lejos de ser una traducción del mecanismo simple de traslado, el trabajo de construcción de software es uno de los problemas más difíciles de la ingeniería de software.

Los temas a ser tratados en esta KA, adoptan dos vistas complementarias de construcción de software. La primera vista comprende tres estilos de interfaces de construcción de software: lingüístico, matemático, y visual. Para cada estilo, los temas son listados según cuatro principios básicos de organización que fuertemente afecta a la construcción de software: complejidad reducida, diversidad anticipada, estructuración para validación, y el uso de normas externas.

4.4 PRUEBAS DE SOFTWARE

Las pruebas de software consisten en verificar dinámicamente la conducta del programa bajo un conjunto finito de casos de prueba y comparar los resultados con lo que se esperaba.

Esta área del conocimiento se divide en dos, la primera de las cuales es organizada conforme a las fases tradicionales para testeo de grandes sistemas de software. La segunda trata las pruebas para condiciones o propiedades específicas.

4.5 MANTENIMIENTO DEL SOFTWARE

El mantenimiento de software es definido como una modificación al producto de software después de corregir fallas, mejorar la actuación, otros atributos o adaptar el producto a otro ambiente modificado. Sin embargo, los sistemas de software son raramente completados y constantemente evolucionan con el tiempo. Esta área del conocimiento también incluye tópicos relevantes a la evolución del software.

La subarea de conceptos de mantenimiento define al mantenimiento, sus conceptos básicos, y cómo el concepto de evolución del sistema encaja en la ingeniería del software (ver figura). También explica las tareas que el mantenimiento implica. La subarea de actividades de mantenimiento y roles indica los tipos formales de mantenimiento y las actividades comunes.

Al igual que con el desarrollo de software, el proceso es crítico para tener éxito y un buen entendimiento del mantenimiento y evolución del software.

Proceso de Mantenimiento

Los Procesos de Mantenimiento nos indican las actividades necesarias y las entradas y salidas de estas actividades.



Fig. Procesos de Mantenimiento de Software

Actividades de Mantenimiento:

- ✚ Actividades Únicas: Transición, Modificación requerida, etc.
- ✚ Actividades de Soporte: Verificación y validación, etc.
- ✚ Actividades de Planificación de Mantenimiento
- ✚ Gestión de la Configuración del Software
- ✚ Calidad del Software

Técnicas de Mantenimiento:

- ✚ Compresión del programa
- ✚ Reingeniería
- ✚ Ingeniería Reversa

4.6 GESTIÓN DE LA CONFIGURACIÓN DE SOFTWARE

Se puede definir un sistema como una colección de componentes organizados para lograr una función específica o un conjunto de funciones. Una configuración del sistema es una función o característica física de hardware, firmware, software, o una combinación de estos como conjunto adelante en documentación técnica y logros en un producto.

La gestión de la configuración, entonces, es la disciplina que permite identificar la configuración a puntos discretos de tiempo para controlar sus cambios sistemáticamente y a mantener su integridad y trazabilidad a lo largo del ciclo de vida del sistema.

Los conceptos de gestión de la configuración se aplica a todos los artículos que requieren control de mando, aunque existen diferencias de aplicación entre la gestión de configuración de hardware y la gestión de configuración de software. Las actividades primarias de la gestión de configuración de software son la utilización de esta como el armazón por organizar y describir los temas de esta AC.

Los componentes de la gestión de configuración son:

- Identificación de configuración de software
- Control de cambios
- Estado de la contabilidad
- Auditorias
- Incumbencia en el ciclo de vida del software
- Relación con otras áreas del desarrollo de software

4.7 GESTION DE LA INGENIERIA DEL SOFTWARE

La área de gestión de la Ingeniería del Software consiste en:

-  Gestión del Proceso,
-  Subareas de Medida

La *área de gestión del proceso* considera la noción de gestión bajo la coordinación de tópicos, estándares de desarrollo e implementación, delimitación del proyecto y equipo de desarrollo.

Para la organización se considera los estados en el ciclo de vida del desarrollo del proyecto: Iniciación y definición de alcance, planificación, ejecución, revisión y evaluación, y conclusión.

La *área de medida* considera cuatro tópicos: metas del programa de medida, selección de medidas, colección de datos, y modelo de desarrollo. Los primeros tres tópicos tienen que ver con la teoría y propósito de la medida, tales como escala y selección de la misma. El cuarto tópico concierne el uso de los datos y el conocimiento para construir modelos.

4.8 PROCESO DE INGENIERIA DEL SOFTWARE

Esta área cubre la definición, implementación, medida, gestión, cambio y mejoramiento del proceso de software. La primera subarea, conceptos básicos y definiciones, establece los temas de esta área del conocimiento y terminología.

Tanto el propósito y métodos para definir el proceso de software, como las definiciones de proceso de software existente son descritos en el subarea del proceso de definición. Los tópicos de esta subarea son tipos de definiciones de

procesos, modelos de ciclo de vida, notaciones para definiciones de procesos, métodos de definición de procesos y automatización.

La subarea del proceso de evaluación describe los logros para el análisis cuantitativo y cualitativo de los procesos de software. La medición juega un importante rol en el proceso de evaluación; sin embargo, la metodología en el proceso de medida es el primer tópico de alcance esta área.

4.9 HERRAMIENTAS Y MÉTODOS DE LA INGENIERIA DE SOFTWARE

Esta Área del Conocimiento contiene 3 subareas:

- ✚ Métodos de Desarrollo,
- ✚ Herramientas de Software y
- ✚ componentes de integración (ver figura).

Los *Métodos de Desarrollo* determinan una estructura en el desarrollo de software y actividades de mantenimiento con el objetivo de hacer las tareas sistemática y exitosamente. Los *Métodos* usualmente proveen una notación y un vocabulario, además de procedimientos para tareas identificables, y una guía para chequear los procesos y el producto a la vez.

El SWEBOK divide esta subarea en tres tópicos principales relacionados: *Métodos heurísticos*, *Métodos formales*, y *Métodos prototipados*.

Las *Herramientas de software* están basadas en herramientas de computadora para asistir los procesos de la IS. Estas herramientas a menudo son diseñadas para soportar métodos particulares. Como los métodos, su intención es hacer el desarrollo más sistemáticamente, y también varían en su alcance, desde abarcar tareas individuales hasta el ciclo de vida completo.

La subarea de *componentes de integración* se divide en tópicos: componentes individuales, modelos de referencia que describen como los componentes pueden ser combinados, y el tópico de reuso.

4.10 CALIDAD DEL SOFTWARE

Producir productos de calidad es la llave para la satisfacción del cliente. Esta área del conocimiento contiene el conocimiento relacionado a la calidad del software y a las actividades de verificación y validación.

La meta de la ingeniería del software es un producto de calidad, pero la calidad por si misma puede tener varios significados. A pesar de la diferente terminología, hay cierto consenso acerca de los atributos que definen la calidad del software.

Estas definiciones proveen la base de conocimientos desde la cual cada producto de calidad es individualmente planeado, construido, analizado, medido y mejorado.

Los procesos de verificación y validación permiten ver la calidad del producto.

Esta Subarea se divide en 4 tópicos principales: definición de análisis de calidad, proceso de planeación, actividades y técnicas para el análisis de calidad, y medidas.

5. USO Y APLICACIONES DE LA GUÍA SWEBOK

- Industria y gobierno
 - Descripción de empleos (Bombardier Transport)
 - Contratación
 - Creación de equipos de proyectos
 - Planificación de carreras (Construx)
 - Negociación de contratos
 - Política gubernamental (Turquía)

- Desarrollo profesional
 - Formación interna, "corporate universities" (SAP)
 - Concepción de cursos
 - Auto-valoración
 - Auto-formación

- Educación
 - Concepción y valoración de currículo (CC2001, ETS, Iceland, Monash)
 - Acreditación (Japón)
 - Concepción y valuación de cursos Ø (Arizona State, ETS)

- Conferencias: tema y referencia
 - América del Norte
 - Europa
 - Australia, Nueva-Zelanda, Argentina, ..
 - América del Norte
 - Europa
 - Asia
 - América del Sur

- Investigación: publicaciones
 - Estados Unidos: U. California, Clamson U., Kentucky U., Denver U., Alabama U.
 - Reino Unido: Sutherland U. , Brighton U. , Aberdeen U., Sheffield U.
 - Holanda: T. U. Delft, T.U. Eindhoven, Twente U.
 - España: U. Polytechnica Catalunya

- Alemania: T.U Chemnitz, U. Hannover
- Dubai, Finlandia, Nueva-Zelanda, Canadá

6. CONCLUSIONES

Swebok, finalmente, es lo que es aceptado por todos los especialistas y hay también prácticas especiales, por ejemplo, para sistemas embebidos o para sistemas de salud. Este trabajo se divide en el software ingenieril y diversas áreas de conocimientos, como Mantenimiento y Testing y otras disciplinas como: Ingeniería en Computación, Ciencia de la Computación, Management, Matemáticas, Gestión de Proyectos de Software, Ingeniería en Sistemas, y otras que pueden ser usadas, pero que no son Ingeniería de Software propiamente tales.

El trabajo se editó el año 2005 y hoy es una referencia, pues su aplicación se da en innumerables escenarios: Industria, Gobierno, empleo, contratación de personal, negociación, planificación de carreras, educación, conferencias, investigación, publicaciones, etc. Lo novedoso de la guía, es que es absolutamente gratuita, y aunque está en formato duro, se puede descargar de la Internet en www.swebok.org. Hoy se está tratando de traducirlo al francés y al castellano, pero ya tiene sus ediciones en japonés y chino.

7. BIBLIOGRAFIA

1. www.swebok.org
2. <http://www.poppendieck.com/papers/Architecture.PDF>
3. <http://martinfowler.com/bliki/Swebok.html>
4. http://robertlevy.net/2003_06_22_archive.aspx#105652424204797983
5. <http://blackbox.cs.fit.edu/blog/kaner/archives/000056.html>