**UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**RESUMEN DEL CAPITULO 8 y 9**

**CURSO:**

**CALIDAD DE SOFTWARE Y USABILIDAD**

**PRESENTADO POR:**

**Ivan Antony Luque Suca**

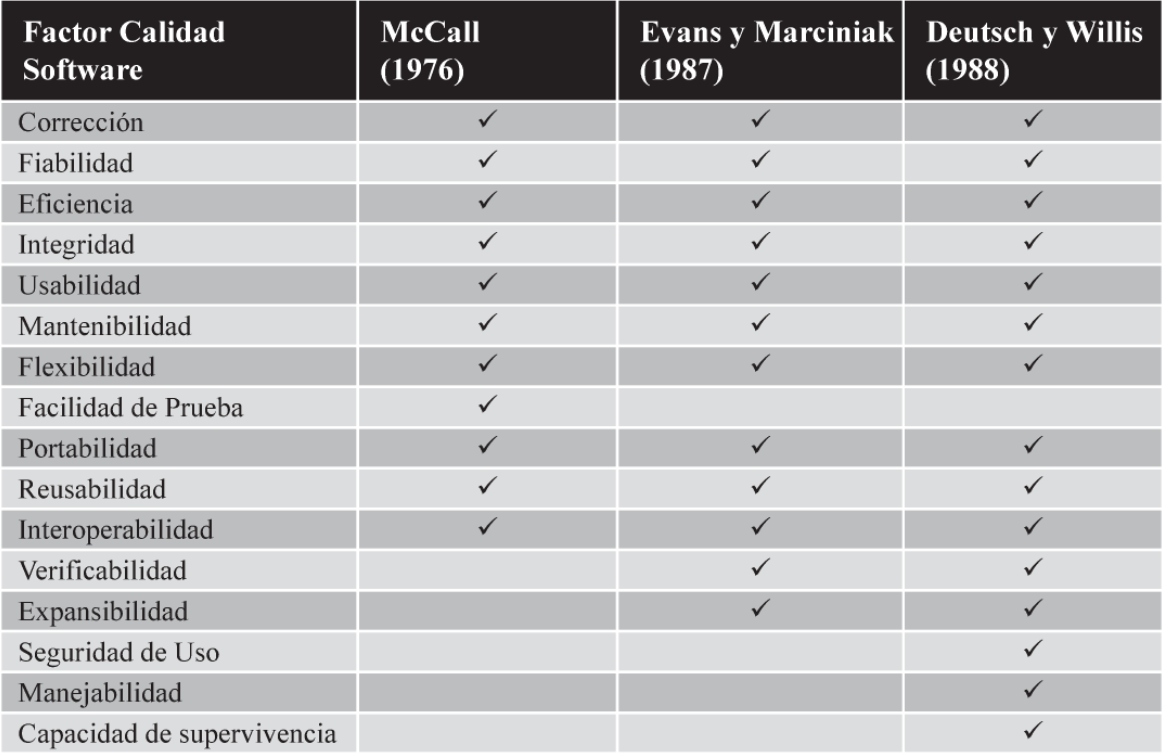
**DOCENTE**

**Ing. Fidel Ernesto Ticona Yanqui**

**SEMESTRE: IX**

**PUNO 2023 PERÚ**

**CAPITULO 8**



**Normas ISO sobre la calidad de software**

La familia ISO/IEC 25000 es la evolución de las normas ISO/IEC 9126 e ISO/IEC 14598.

* ISO/IEC 9126: abordaba el tema sobre la calidad del producto de software mediante 3 informes técnicos (métricas externas, internas y calidad en uso) en su nueva versión del 2001.
* ISO/IEC 14598: abordaba el tema de la evaluación del producto de software bajo 6 normas que son las siguientes: visión general, planificación y gestión, proceso para desarrolladores, compradores, evaluadores y por último la documentación de módulos de evaluación.

**Familia de Software ISO 25000**

La familia de normas ISO/IEC 25000 se organiza en seis apartados principales:

* **ISO/IEC 2500n – División de Gestión de Calidad:** Definen todos los modelos, términos y definiciones que referencia la familia ISO 25000.
* **ISO/IEC 2501n – División de Modelo de Calidad:** Presentan modelos de calidad para productos de software, sistemas, calidad en uso y datos
* **ISO/IEC 2502n – División de Medición de Calidad:** Referencia medición de la calidad de productos de software y sistemas, definiciones de medidas de calidad y guías prácticas para su aplicación.
* **ISO/IEC 2503n – División de Requisitos de Calidad:** especifican que requisitos de calidad pueden ser utilización en el proceso de licitación de requisitos de calidad de un producto a desarrollar.
* **ISO/IEC 2504n – División de Evaluación de la Calidad:** Proporcionan requisitos, recomendaciones y guías para la evaluación de productos.
* **ISO/IEC 25050-25099n – División de Extensiones:** Incluyen informes técnicos que abordan dominios de aplicación específicos.

**Normas sobre Gestión de Calidad (ISO/IEC 2500n)**

* ISO/IEC 25000: Visión general sobre su nueva serie conocida como SQuaRE.
* ISO/IEC 25001: Proporciona requisitos y guías para la función responsable.

**Normas sobre Modelado de la Calidad (ISO/IEC 2501n)**

De la familia ISO 25000, nos proponen 3 modelos de calidad que están compuestas por las normas ISO 25010 e ISO 25012. En donde, el modelo de calidad de producto de software, modelo de uso de sistema es de la ISO 25010 y modelo de calidad de datos es de la ISO 25012.

Además, la norma los clasifica las propiedades del software en: Inherentes y Asignadas.

**Normas sobre Medición de Calidad (ISO/IEC 2502n)**

* ISO/IEC 25020: presenta modelo de medición de calidad.
* ISO/IEC 25021: define un conjunto de medidas base y derivadas.
* ISO/IEC 25022: medidas de calidad en uso.
* ISO/IEC 25023: medidas para calidad de producto.
* ISO/IEC 25024: medidas de calidad de datos.

**Normas sobre Requisitos de Calidad (ISO 2503n)**

* ISO/IEC 25030: Nos da un análisis y definición de los requisitos de la calidad de software.

**Normas sobre Evaluación de Calidad (ISO 2504n)**

* ISO/IEC 25040: propone un modelo referencia.
* ISO/IEC 25041: define la estructura y contenido de un módulo de evaluación.
* ISO/IEC 25045: especifica la evaluación de la “recuperabilidad”.

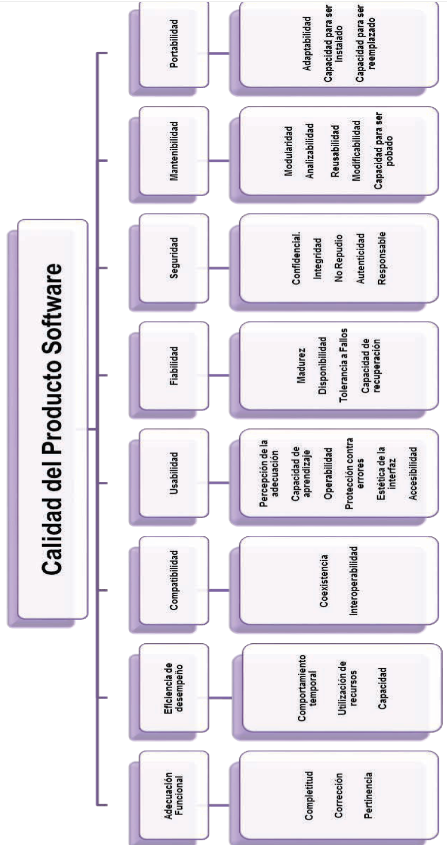
**Normas sobre Extensiones de SQuaRE**

* ISO/IEC 25051: define requisitos de calidad para productos COTS y documentación de pruebas.
* ISO/IEC TR 25060: define normas que documentan la especificación y evaluación de la usabilidad de los sistemas interactivos.

**Modelos de Calidad de Producto Software**

La norma ISO/IEC 25010 define dos modelos: modelo de calidad de producto y calidad en uso.

**Modelo de calidad de Producto**

****

**Modelo de calidad en Uso**

****

**Evaluación de la Calidad de Productos software**

* ISO/IEC 25040: propone un modelo de referencia para la evaluación, que considera tanto las entras de proceso de evaluación, las restricciones y los recursos disponibles, para obtener las correspondientes salidas.

**Tareas del proceso de evaluación**

* **Establecer los requisitos de la evaluación:** Establecer el propósito de la evaluación, obtener los requisitos de calidad del producto, identificar las partes del producto sobre las que se realiza la evaluación y definir el rigor de la evaluación.
* **Especificar la evaluación:** Seleccionar las métricas de evaluación, definir los criterios de decisión para las métricas y establecer los criterios de decisión para la evaluación.
* **Diseñar la evaluación:** Planificar las actividades de evaluación
* **Ejecutar la evaluación:** Realizar las mediciones, aplicar los criterios de decisión para las métricas y para la evaluación.
* **Concluir la evaluación:** revisar los resultados de la evaluación, crear el informe de evaluación, revisar la calidad de evaluación y realizar la disposición de los datos de la evaluación.

**Recursos para el proceso de evaluación**

Para el proceso de evaluación de la calidad del producto de software requiere una gran cantidad de recursos: herramientas de medición, recursos humanos especializados, un sistema de información para la evaluación del producto, etc.

De todos estos recursos es necesario contar con herramientas que faciliten tanto en la toma de decisiones de los criterios de medición y evaluación, para evitar errores humanos y facilitar la medición de grandes cantidades de información.

**Certificación de la calidad de los productos de Software**

En 2013 AENOR, incluye un modelo para el gobierno de las TIC con normas ISO, la certificación del propio producto de software en base a la familia de normas ISO/IEC 25000.

AENOR colabora con AQCLab que es el responsable de la evaluación técnica del producto software y AENOR, en base a los resultados de la evaluación, se encarga de la inspección.

Para el proceso de la certificación de calidad del producto software, se presentan 6 actividades:

* Solicitud de evaluación de la calidad de un producto
* Informe de evaluación de la calidad
* Solicitud de certificación
* Solicitud de informe de evaluación
* Resultados de la evaluación realizada
* Resultado y certificado de calidad de producto

CAPITULO 9

La ingeniería de Software se ha centrado en metodologías y lenguajes de programación, modelos de desarrollo y herramientas, fue necesario incluir determinadas áreas para la ingeniera de software, como las infraestructuras de gestión y organización, por lo que surge la denominada “ingeniería del software basada en el proceso”

Según ISO 1995 un **proceso** se define como un conjunto de actividades interrelacionadas que transforman entradas en salidas. Define quien lo está haciendo, el que, cuando y como alcanzar un determinado objetivo. Podemos encontrar diversas definiciones con respeto al proceso de software.

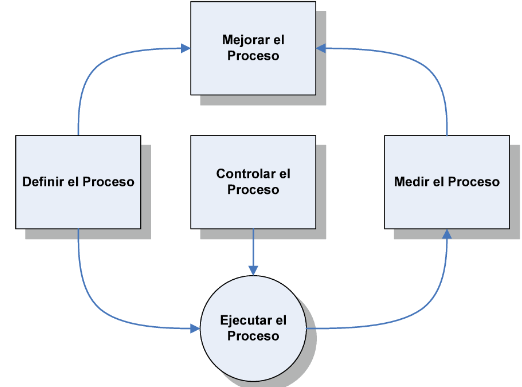
Según (Derniame 199) el proceso de software es un proceso con una naturaleza especial. Se determina de las siguientes características: es complejo, no es un proceso de producción típico, tampoco es un proceso de ingeniería **pura**, no es un proceso creativo y está basado en descubrimientos.

La necesidad de participación humana de forma creativa y la ausencia de acciones repetitivas hacen que ni el desarrollo y mantenimiento del software sean procesos de fabricación, es por eso que los procesos de software constan de dos subprocesos interrelaciones los cuales son: proceso de producción y de gestión.

Según Fuggeta, el proceso de software es un campo de estudio amplio y complejo, debido a las grandes cantidades de elementos relaciones y también a su diversidad, se pueden establecer categorías como las siguientes: tecnología de desarrollo software, métodos y técnicas de desarrollo de software, comportamiento organizacional y economía y marketing.

**Gestión de los Procesos Software**

Los requisitos de procesos de software son: produzcan resultados esperados, basados en una correcta definición y sean mejorados según a los objetivos. Para la aplicación de forma efectiva estos objetivos son necesario cuatro responsabilidades claves: **definir, medir, controlar y mejorar el proceso.**

****

**Modelado de los Procesos Software**

Para los procesos de software es necesario disponer de modelos de procesos que nos indiquen como hacer las cosas de una organización. Un modelo de procesos puede ser analizado, validad y simulado, si es ejecutable. En los modelos de procesos se pueden describir de una forma precisa los diferentes aspectos relacionados con los procesos software.

Los objetivos y beneficios que motivan la introducción de modelos de procesos son varios, podemos destacar algunas como: soporte y control de la gestión del proceso, soporte a la mejora.

Los **Lenguajes de Modelado de Procesos**, tienen como objetivo reportar de una forma precisa los diferentes elementos con un proceso software.

**Elementos del Proceso Software**

Los conceptos básicos relaciones con el proceso software, son los siguientes:

* **Actividad:** incorporan e implementan procedimientos, reglas y políticas.
* **Producto:** se denomina al conjunto de artefactos a ser desarrollados, entregados y mantenidos.
* **Recurso:** tenemos 2 recursos los desarrolladores y las herramientas de desarrollo
* **Rol y directivas:** las directivas son estructuradas que vienen en un manual, los roles van conjuntamente con los desarrolladores en donde cada uno tiene una tarea especifica.

**Clasificación de los Lenguajes de Modelado de Procesos**

Existen diferentes criterios para la clasificación de un LMP, la información de un modelo se estructura bajo diferentes puntos de vista como: **funcional, comportamental, organizacional e informativo.**

Otra posible forma de clasificación es **descriptiva (informales y formales) y prescriptivas (manuales y automáticos)**, pero en estas ya se distingue entre lenguajes.

Existen otras clasificaciones, en la que se distinguen en tres categorías en función del nivel de abstracción: PSL, PDL, PIL. Se propone una clasificación más genérica y menos centrada en los PSEE y se distingue entre LMP: **no ejecutable, simulados y ejecutable.**

**Metamodelos de proceso software**

Un metamodelo es un conjunto de clases a partir de las que se pueden crear fragmentos de metodologías completas y usables.

Existe metamodelos de procesos como **Gantt y PERT**, definen el mínimo numero de actividades necesarias de un proceso con determinado tiempo en los proyectos. También, hay propuestas de metamodelos de procesos con el objetivo de proporcionar un intercambio como: **PIF, PSL, UPM, CPR, ARIS, SPEARMINT y PROMENADE.**

**ISO/IEC 24744**

También conocido como **SEMDM**, establece metodologías de desarrollo software, incluyendo sus tres aspectos principales:

* El proceso a seguir
* Los productos utilizados y generados
* Las personas implicadas

**SEMDM** incorpora las diferentes ideas de los metamodelos más los resultados de investigación en el área facilitando:

* La comunicación entre los desarrolladores
* Ensamblaje de metodologías
* Creación de metamodelos de metodologías
* Comparación e integración de metodologías y metamodelos asociados
* Interoperabilidad de herramientas de soporte a modelado y metodologías

Las principales características del metamodelo SEMDM se resume en las siguientes:

* Integran los aspectos del proceso y modelado de producto en un único metamodelo.
* El metamodelo incluye constructores para modelas la capacidad de los procesos definidos.

**SPEM 2.0**

es un metamodelo para definir modelos de procesos de ingeniería de software, se considera un lenguaje general de modelado de procesos, pero orientado a los procesos software. Además de ser un metamodelo, es un marco conceptual proporcionando conceptos como, modelar, documentar presentar, publicar, gestionar, intercambiar y realizar métodos y proceso de software.

Representa procesos basado en tres elementos: rol, producto de trabajo y tarea, idea subyacente es que un modelo consiste, en decir quién (rol) realiza que(tarea) para, a partir de unas entradas (producto de trabajo) obtener unas salidas (producto de trabajo).

Con SPEM 2.0 podremos visualizar 4 escenarios fundamentales:

* Proporcionar una librería o repositorio de **contenido de método** reutilizables.
* Dar soporte al desarrollo, gestión y crecimiento de **procesos** software.
* Definir **configuraciones** de contenidos de método y procesos en función de las características específicas de los proyectos.
* Dar soporte a la **realización** de un proceso para llevara cabo los proyectos de desarrollo concretos.

A la hora de implementar una metodología con SPEM 2.0 se distinguen dos grupos: **contenido del método** (method content) y **definición de procesos** (process).

**Visión General del Metamodelo de SPEM 2.0**

SPEM 2.0 utiliza UML en donde adopta un enfoque orientado a objetos y esta organizado en 7 paquetes.

* ***Core:*** contiene clases y abstracciones que sirven de base para las clases de los demás paquetes.
* ***Process Structure:*** define proceso como una agregación, establece una EDT e incluye mecanismos para la reutilización mediante el ensamblado de procesos. Estas estructuras son utilizadas para representar procesos básicos de alto nivel.
* ***Process Behavior:*** es un proceso en donde extiende las estructuras del paquete anterior, donde en vez de incluir un mecanismo propio, opta por reutilizar los existentes.
* ***Managed Content:*** permite incorporar y gestionar descripciones en lenguaje natural, documentos y otras informaciones útiles para la comprensión por humanos, tiene una libertad total en donde combina modelos estructurales de proceso con contenidos de lenguaje natural.
* ***Method Content:*** se define elementos fundamentales en donde describen procesos, roles, tareas o productos de trabajo y las relaciones que tienen entre sí.
* ***Process with Methods:*** Es la combinación de procesos con métodos, cuando el contenido de método define técnicas y métodos para el desarrollo de software, es necesario definir los procesos de dichas técnicas y métodos en un ciclo de vida.
* ***Method Plugin:*** gestionan y diseñan bibliotecas con contenido de métodos y procesos, que sean mantenibles a gran escala, reutilizables y configurables.

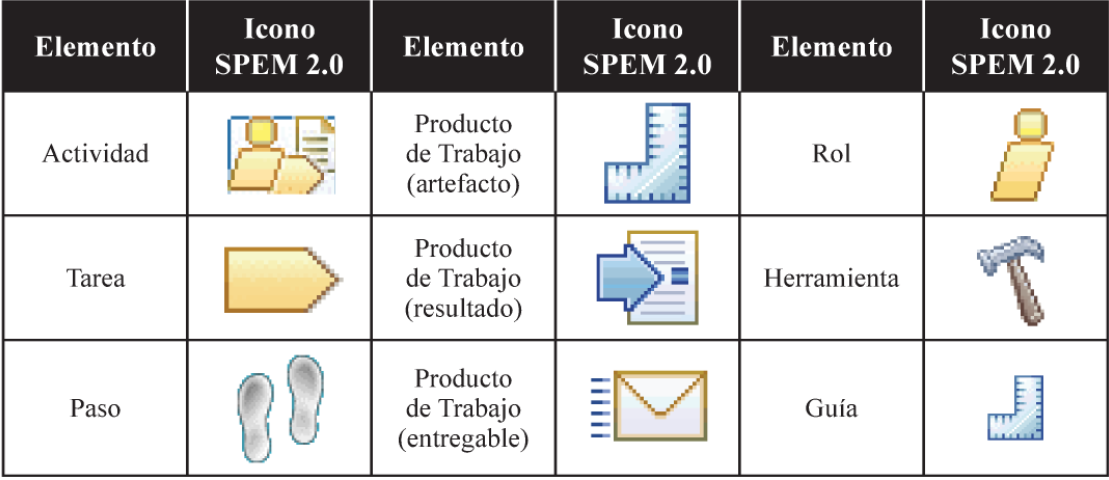
Al utilizar unos u otros paquetes, un ingeniero de procesos puede expresar sus procesos, los escenarios más habituales son:

* Core + Managed Content + Method Content
* Opción 1 + Process Structure + Process Behavior
* Los 7 paquetes completos.

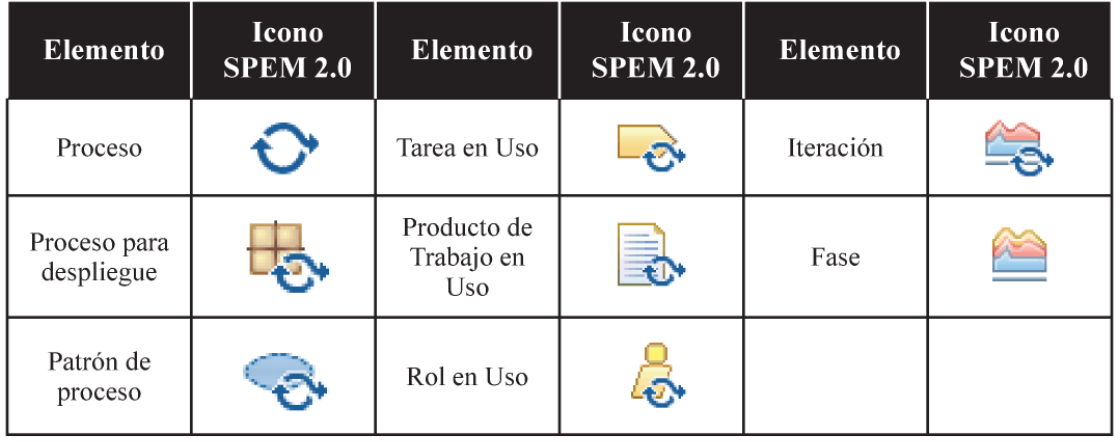
**Definición de modelos de Procesos con SPEM 2.0**

**Definición de Métodos con SPEM 2.0**

En la siguiente imagen describe los iconos de los principales constructores de contenido de método.

****

En la siguiente imagen describe los iconos de los principales constructores de procesos.



Hay cuatro tipos de elementos de contenido: Tarea, Rol, Producto de trabajo y Guía, también existe un quinto elemento con fines de clasificación y agrupación llamado Categorías.

Presentaremos unos constructores básicos:

* Tarea: definición de procesos y su granularidad es de pocas horas a pocos días.
* Paso: organiza las tareas en partes, un paso puede estar compuesto por otros pasos.
* Producto de trabajo: son los elementos del contenido del método que son utilizados, modificados y producidas por las tareas.
* Rol: define un conjunto de habilidades, competencias y responsabilidades de una persona.
* Cualificación: documenta aptitudes, habilidades o competencias requeridas para las tareas por parte de los roles.
* Herramienta: es una unidad que da soporte a los roles a la hora de realizar tareas.
* Categoría: permite agrupar cualquier número de elementos, puede anidarse una categoría estableciendo jerarquías de categorías.
* Guía: proporciona información adicional a cualquier elemento descriptible.
* Métrica: define una medición estándar para cualquier elemento descriptible de SPEM 2.0.

**EPF Composer**

**SEMAT**

Es una iniciativa que apoya un proceso con un enfoque más pragmático para redefinir la ingeniería de software, trata de resolver la falta de base teórica que se vieron en anteriores propuestas.

Propone **Essence**, define un núcleo y un lenguaje de soporte a la creación, uso y mejora métodos que sean escalables, extensibles y fáciles de usar. Su objetivo es que las personas puedan describir los aspectos esenciales que se puedan comparar, evaluar, adaptar, utilizar, simular y medir.

Los principales elementos de SEMAT son:

* Un método: no solo describe lo que supuestamente se debe hacer sino lo que también realmente se hace
* Una práctica: es un enfoque repetible para hacer algo con un objetivo.
* Essence Kernel: elementos esenciales en todos los métodos.
* Essence Language: lenguaje especifico de dominio para definir métodos, prácticas y kernels.

**Kernel**

Se organiza en tres áreas discreta de intereses, cada una enfocada en un aspecto específico.

* **Cliente:** uso y explotación del software a producir
* **Solución:** especificación y desarrollo del sistema de software
* **Esfuerzo**: relativa al equipo y la forma en que acomete su trabajo

Existen tres tipos de elementos que contiene el *kernel*:

* Alfas: proporcionan las descripciones necesarias para que los equipo desarrollen, mantenga y den soporte al software.
* Espacios de actividad: describen los retos que enfrenta cada equipo al momento de desarrollar, mantener y dar soporte a los sistemas.
* Competencias: Representan capacidades necesarias para realizar el trabajo requerido.

También tiene cinco niveles de realización para las competencias:

* *Assits*: demuestra un entendimiento básico.
* *Applies*: aplica los conceptos mas simples mediante la experiencia obtenida.
* *Master*: aplicar los conceptos mas complejos y trabajar sin supervisión.
* *Adapts*: razonar de forma critica el cuando y como aplicar los conceptos.
* *Innovates*: capaz de entender los conceptos a nuevos contextos e inspirar otros.

**Lenguaje Essence**

Es un lenguaje por una base estática para la descripción de los elementos anterior, para habilitar el uso y la adaptación de estos.

Los elementos (alfa, estado alfa, espacio de actividad y competencia), se usa para la describir el contenido del núcleo, también es suficiente para poder hablar sobre el estado, progreso y salud de un esfuerzo. Mientras que al usar con un kernel son abstractos.

**Extensiones**

El kernel de SEMAT puede ser extendido para dar soporte a distintas áreas.

Las siguientes extensiones son opcionales:

* Extensión de Análisis de negocio.
* Extensión de Desarrollo.
* Extensión de Gestión de Tareas.

**Comparativa SPEM-SEMAT**

Ambos auspiciados por OMG, se pueden ver algunas diferencias. En SPEM adopta más la filosofía de Ingeniería de Métodos que es mas optima para organizaciones que no quieren juntar la definición de métodos y los procesos para el desarrollo de un proyecto, cuenta con mayor madurez a la de SEMAT y cuenta con un soporte de herramientas de EFPC. En cambio, SEMAT se enfoca más en los métodos que en la definición como lo hace SPEM. Incorpora nuevos conceptos como *alphas*, además soporta una adopción incremental partiendo de un conjunto pequeño de elementos, lo que lo hace mas apropiado a organizaciones pequeñas.

**Entornos de ingeniería del software orientados al proceso**

**Introducción y características**

Los PSEE dan soporte a los procesos de ingeniería, mediante un modelo de procesos definido mediante un LMP adecuado. Los modelos asociados a PSEE los recursos, tareas y como se deben interactuar. **Motor del proceso** es un elemento clave, ya que se encarga de guiar y ayudar a las personas en el momento de realizar las tareas en las diferentes actividades de proceso, también automatiza las actividades que no requieran intervención humana. Sus elementos son: un intérprete del modelo de procesos, entorno de interacción del usuario y un repositorio.

**Clasificación de los PSEE**

Los LMP pueden adoptar los siguientes enfoques:

* Lenguaje basado en la programación
* Basados en reglas
* Basados en Autómatas Extendidos
* Multiparadigma

Otros aspectos claves son el tipo de soporte que ofrecen a los usuarios:

* Rol pasivo
* Guía activa
* Obligación
* Automatización

Un mismo PSEE puede adoptar distintas formas de soporte al usuario, también se pueden distinguir si son PSEE proactivos o reactivos.