

Presentación de la asignatura

Justificación

El concepto de Proceso de *software* es uno de los conceptos más abstractos en la historia de la Ingeniería de *software*. Muchas veces confundido como un proyecto informático o como un abstracto que pretende ser la base de una potencial gestión de conocimiento de los procesos de desarrollo informático. Desde que Watts Humphrey define más formalmente el proceso de *software*, se ha logrado concebir una cosmovisión del proceso de *software* más formal, siendo fijada como un proceso de continuo aprendizaje mediante el cual una organización mejora y se mejora a través de procesos adquiridos y/o sus propios procesos. Así se llega al proceso llamado CMM o *Capability Maturity Model* con sus modelos asociados a personas y adquisiciones. La versión actualizada del CMM, ICMMI o CMM *Integrated*, es actualmente un instrumento de la gestión de la ingeniería de *software* tradicional y formal.

No obstante, esta formalización ha hecho que su validez se pierda como "un proceso más a ejecutar" y no en su sentido organizador del trabajo informático sobre todos los artefactos informáticos que una organización utiliza hoy en día. Aún más, los adelantos en ingeniería organizacional y su pervivencia en las empresas hoy en día, hacen de que un enfoque de Proceso de *software* no se limite ni constriña a un conjunto de prácticas y al cumplimiento de un modelo, sino a un estado o proceder organizacional tendiente a gestionar el conjunto de proyectos, procesos y tareas asociadas al procesamiento de datos, de información y de conocimiento de una organización cualquiera.

En este sentido la asignatura presenta el proceso de *software* bajo una óptica de enclave organizacional que precisa una gestión completa, cabal e integral. Con relación a la Ingeniería de *software* como disciplina y al Ingeniero de *software* como profesional, el enfoque dado al concepto de Proceso de *software* resulta integrador con otros campos relacionados como la calidad y la gestión de riesgos, la gestión de proyectos y, la administración de sistemas tecnologizados al introducir conceptos de ventas versus diseño, selección de componentes o simplemente la constitución de una Oficina de Proyectos Informáticos.

En este sentido se introduce como término provocador el de Proceso de Negocio de *Software*, para encaminar al lector hacia esta nueva idea del alcance de un proceso de *software*, ya no como un conjunto de "cosas por hacer" para producir *software*, sino un conjunto de procesos claves y estratégicos que regulan todo el negocio adscrito a una producción de *software* lo cual no limita los procesos a la producción tradicional, sino que incluye toda la cadena de valor que aporta valor a

un *software* o desarrollo tecnológico cualquiera, lo cual incluye marketing, gerencia, investigación, etc.

Descripción de la asignatura

La asignatura revisa diversos conceptos asociados a la Ingeniería de *software* y a Proceso de *software* desde una óptica de unidad organizacional estratégica. Se pretende revisar una serie de instrumentos de gestión organización y administración empresarial en la esfera de la Ingeniería de *software* entendida como un negocio o unidad estratégica dentro de una empresa o en sí misma como una empresa.

La asignatura revisa diversos elementos vinculados tradicionalmente al proceso de *software* unido a otros propios de gestión y gerencia de proyectos para distinguir los procesos que serían clave al momento de sentar las bases de la idea de proceso de negocio de *software* y que en esencia define una empresa de *software*.

Objetivos

Objetivos generales

- Presentar el concepto y noción de Proceso de *software* entendido como una herramienta organizacional y un signo de madurez organizacional de unidades informáticas con el fin de interpretar el proceso de *software* como una unidad de negocios empresarial.
- Vincular el proceso de *software* a los tradicionales paradigmas de *software* ampliamente utilizados en la planificación de proyectos informáticos e igualmente se presenta asociado a conceptos de proyectos de *software*.
- Relacionar el proceso de *software* con la estructura de una oficina de proyectos como una instancia de gestión organizacional del conocimiento asociado a las actividades de *software* en una organización.

Objetivos específicos

- Conocer los fundamentos de la calidad y de la organización de la calidad en una organización (ISO 9000, 1400 y 1800) y en una unidad de calidad integral (SQA, SCM, SEPG) y situarlos en una óptica de estrategia organizacional.

- Comprender el rol organizacional de los modelos de mejora (CMM, SPICE, etc.), los conceptos de modelización, el rol de herramientas CASE centrado en el proceso, y dominar y generar métricas en ingeniería del *software*.

- Conocer las herramientas de decisión al momento de comprar o producir un *software*.

- Comprender los mecanismos que definen y regulan la relación entre clientes, consumidores, operadores y desarrolladores de *software* (*outsourcing*, deslocalización, etc.)
- Conocer sobre gestión de proveedores y la subcontratación, análisis de valor y pliego de condiciones.
- Comprender y conocer los procesos y herramientas de la evaluación de proyectos y la valorización de empresas.
- Interpretar la función del proceso de *software* como una unidad de negocio empresarial.

Organización del contenido y estructura del documento

Estos objetivos se desarrollan en los capítulos de la asignatura tal como se describe a continuación.

CAPÍTULO	OBJETIVO(S)	RESUMEN DEL CAPÍTULO	APORTACIÓN Y RESULTADO OBTENIDO
Capítulo 1	Conocer los conceptos y características esenciales de un proceso de <i>software</i> .	Revisión del concepto de proceso, ligado a la noción de calidad y riesgo como ejes del proceso de <i>software</i> incluyendo herramientas como SCM y SEPG, CASE, entornos de trabajo, Modelos de procesos de <i>software</i> y de mejora de la calidad (CMM, CMMI, SPICE, Trillium, etc.).	Se identifica la noción de proceso de <i>software</i> y sus rasgos distintivos.
Capítulo 2	Conocer los paradigmas y su relación con los procesos de <i>software</i> .	Revisión de los paradigmas de <i>software</i> y su relación con la Ingeniería de <i>software</i> , asimismo el cambio en los paradigmas bajo la dirección de un proceso de <i>software</i> .	Se relacionan los diversos paradigmas de desarrollo de <i>software</i> con los procesos de <i>software</i> lo cual relaciona la visión operacional con la estratégica.
Capítulo 3	Conocer el enfoque de proyectos y su relación con los procesos de <i>software</i> .	Revisión del proceso de <i>software</i> y su relación con la Gestión de Proyectos, y el cambio que se produce en el desarrollo de <i>software</i> desde la gestión de proyectos.	Se relaciona el enfoque de proyectos tradicional (diseño, gestión y dirección) con los procesos de <i>software</i> lo cual relaciona la visión de proceso organizacional con la visión de proceso asociada al <i>software</i> .
Capítulo 4	Reflexionar sobre la oficina de proyectos y los procesos de	Reflexión sobre una oficina de proyectos (y sus herramientas) como unidad estratégica organizacional y como base de	Se discute y plantea una unidad organizacional estratégica y gerencial para gestionar y administrar

	<i>software.</i>	una oficina de proyectos de <i>software</i> como motor del cambio organizacional.	proyectos de <i>software</i> desde la óptica de los procesos de <i>software</i> .
--	------------------	---	---

Capítulo 1 .- Conceptos y características esenciales de un proceso de *software*

OBJETIVO

- Conocer los conceptos y características esenciales de un proceso de *software*.

1.1. Concepto de proceso

A continuación se presentan diversas definiciones de Proceso:¹

- La norma internacional ISO-9001 define un proceso como "una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados".²
- Oscar Barros hace una importante distinción, al introducir el concepto de valor agregado en la definición de proceso, señalando que "un proceso es un conjunto de tareas lógicamente relacionadas que existen para conseguir un resultado bien definido dentro de un negocio; por lo tanto, toman una entrada y le agregan valor para producir una salida. Los procesos tienen entonces clientes que pueden ser internos o externos, los cuales reciben a la salida, lo que puede ser un producto físico o un servicio. Estos establecen las condiciones de satisfacción o declaran que el producto o servicio es aceptable o no" (Barros, 1994; pp.56).
- Thomas Davenport, uno de los pioneros de la reingeniería, señala que un proceso, simplemente, es "un conjunto estructurado, medible de actividades diseñadas para producir un producto especificado, para un cliente o mercado específico. Implica un fuerte énfasis en **cómo** se ejecuta el trabajo dentro de la organización, en contraste con el énfasis en el **qué**, característico de la focalización en el producto" (Davenport, 1993; pp. 5).
- Hammer (1996) por su parte, establece la diferencia sustancial entre un proceso y una tarea, señalando que una tarea corresponde a una actividad conducida por una persona o un grupo de personas, mientras que un proceso de negocio corresponde a un conjunto de actividades que, como un todo, crean valor para el cliente externo. Al hacer esta comparación, Hammer hace la analogía con la diferencia que existe entre las partes y el todo.
- Por su parte, Ould (1995) lista una serie de características que deben cumplir los procesos de negocio y que refuerzan la posición de Hammer; según este autor, un proceso de negocio contiene actividades con propósito, es ejecutado

colaborativamente por un grupo de trabajadores de distintas especialidades, con frecuencia cruza las fronteras de un área funcional, e invariablemente es detonado por agentes externos o clientes de dicho proceso.

¹ BARROS, Oscar. (1994). - *Reingeniería de Procesos de negocio*. Editorial Dolmen. Chile.

² ISO. (2000). *Norma Internacional ISO 9001 - Sistemas de gestión de la calidad - Requisitos*- Impreso en la Secretaría Central de ISO en Ginebra, Suiza.

1.1.1. Proceso de negocios

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. Las entradas son requisitos que deben tenerse antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes.

Es una colección de actividades estructurales relacionadas que producen un valor para la organización, sus inversores o sus clientes. Es, por ejemplo, el proceso a través del que una organización ofrece sus servicios a sus clientes.

Un proceso de negocio puede ser parte de un proceso mayor que lo abarque o bien puede incluir otros procesos de negocio que deban ser incluidos en su función. En este contexto un proceso de negocio puede ser visto a varios niveles de granularidad. El enlace entre procesos de negocio y generación de valor lleva a algunos practicantes a ver los procesos de negocio como los flujos de trabajo que efectúan las tareas de una organización.

Un subproceso es parte de un proceso de mayor nivel que tiene su propia meta, propietario, entradas y salidas. Las actividades son partes de los procesos de negocio que no incluyen ninguna toma de decisión ni vale la pena descomponer (aunque ello sea posible). Por ejemplo, "Responde al teléfono", "Haz una factura".

Un proceso de negocio es usualmente el resultado de una Reingeniería de Procesos. El modelado de procesos es usado para capturar, documentar y rediseñar procesos de negocio. Para aplicar los procesos se deben tener claras las tareas, una estructura jerárquica y una tendencia a la interacción y comunicación vertical.

1.1.1.1. Características de un proceso de negocio

Los procesos poseen las siguientes características:¹

- Pueden ser medidos y están orientados al rendimiento.
- Tienen resultados específicos.

- Entregan resultados a clientes o "*stakeholders*".²
- Responden a alguna acción o evento específico.
- Las actividades deben agregar valor a las entradas del proceso.

Los procesos de negocio pueden ser vistos como un recetario para hacer funcionar un negocio y alcanzar las metas definidas en la estrategia de negocio de la empresa. Las dos formas principales de visualizar una organización, son la vista funcional y la vista de procesos.

¹ DAVENPORT, Thomas. (1993). *Process Innovation*. Harvard Business School Press. USA.

² **Stakeholders**. Todo grupo o persona que puede afectar o se ve afectado por una organización o sus actividades. También, toda persona o grupo que puede ayudar a definir las proposiciones de valor de una organización.

1.1.1.2. Objetivos centrales de un proceso de negocio

Los objetivos centrales de un proceso de negocio son:

- Mejorar el entendimiento de una situación y comunicarla entre los diversos *stakeholders*.
- Utilizarlos como una herramienta para alcanzar las metas de un proyecto de proceso de desarrollo. No obstante, para que los procesos de negocio puedan cumplir con su objetivo, constantemente son sometidos a cambios debido a programas de mejora continua en las organizaciones.

1.1.1.3. Tipos de procesos de negocio

Existen tres tipos de procesos de negocio:¹

- **Procesos estratégicos** - Estos procesos dan orientación al negocio. Por ejemplo, "Planificar estrategia", "Establecer objetivos y metas".
- **Procesos centrales** - Estos procesos dan el valor al cliente, son la parte principal del negocio. Por ejemplo, "Repartir mercancías".
- **Procesos de soporte** - Estos procesos dan soporte a los procesos centrales. Por ejemplo, "contabilidad", "Servicio técnico".

¹ IBM. (1981). *Business Systems Planning-Information Systems Planning Guide*, GE20-0527-3. USA

1.1.1.4. Vista funcional vs vista de proceso

La **vista funcional** descansa en el organigrama de la empresa como modelo fundamental del negocio; las actividades que debe ejecutar la organización, para cumplir con su misión, se estructuran en conjuntos de funciones relativamente homogéneas (por ejemplo, todas las actividades que tienen que ver con las finanzas de la organización, se unen bajo un mismo "techo"). Y así, los recursos pertenecen a los departamentos y la especialización funcional y el expertizaje, son las principales consideraciones a la hora de formar los departamentos, los cuales se relacionan a través de una jerarquía de estructuras de autoridad. Los programas de mejoramiento se focalizan en aumentar la eficiencia y efectividad de las funciones y unidades organizacionales específicas, muchas veces suboptimizando a la organización como un todo holístico y descuidando aquello que debiera ser la primera prioridad de cualquier organización: satisfacer y en lo posible anticiparse a los requerimientos de los clientes.

En contraste, la vista de procesos se orienta al trabajo mismo que se debe desarrollar en la organización, para que el negocio funcione y entregue un producto o servicio, por el cual un cliente externo está dispuesto a pagar. La vista de procesos es una manera tan poderosa de visualizar y analizar un negocio, porque provee de la lógica con la cual los clientes lo miran; los clientes interactúan con la empresa, a través de los procesos del negocio, contratando un servicio, recibiendo dicho servicio, pagándolo y recibiendo atención de post venta. Cuando se entiende el negocio desde esta perspectiva, es posible evaluar el "valor agregado" del trabajo que aporta cada cual.

1.1.2. Proceso de *software*

Un proceso de desarrollo de *software* tiene como propósito la producción eficaz y eficiente de un producto *software* que reúna los requisitos del cliente. Dicho proceso, en términos globales se puede ver en la **figura 1.1**.

Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de *software* es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de *software* hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. A continuación se explican algunas particularidades asociadas al desarrollo de *software* y que influyen en su proceso de construcción.

Un producto *software* en sí es complejo, es prácticamente inviable conseguir un 100% de confiabilidad de un programa por pequeño que sea. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de las todas posibles situaciones de ejecución que se puedan presentar (entradas, valores de variables, datos almacenados, *software* del sistema, otras aplicaciones que intervienen, el *hardware* sobre el cual se ejecuta.).

Un producto *software* es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos *software* similares. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado en producto sino también durante el proceso de desarrollo.

Además, de las dos anteriores, siempre puede señalarse la inmadurez de la ingeniería del *software* como disciplina, justificada por su corta vida comparada con otras disciplinas de la ingeniería. Sin embargo, esto no es más que un inútil consuelo.

1.1.2. Proceso de *software*

Un proceso de desarrollo de *software* tiene como propósito la producción eficaz y eficiente de un producto *software* que reúna los requisitos del cliente. Dicho proceso, en términos globales se puede ver en la **figura 1.1**.

Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de *software* es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de *software* hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. A continuación se explican algunas particularidades asociadas al desarrollo de *software* y que influyen en su proceso de construcción.

Un producto *software* en sí es complejo, es prácticamente inviable conseguir un 100% de confiabilidad de un programa por pequeño que sea. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de las todas posibles situaciones de ejecución que se puedan presentar (entradas, valores de variables, datos almacenados, *software* del sistema, otras aplicaciones que intervienen, el *hardware* sobre el cual se ejecuta.).

Un producto *software* es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos *software* similares. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado en producto sino también durante el proceso de desarrollo.

Además, de las dos anteriores, siempre puede señalarse la inmadurez de la ingeniería del *software* como disciplina, justificada por su corta vida comparada con otras disciplinas de la ingeniería. Sin embargo, esto no es más que un inútil consuelo.

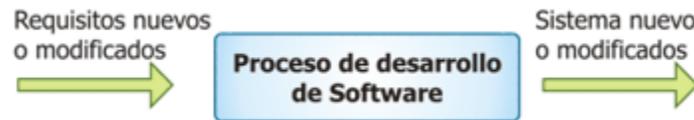


Figura 1.1: Proceso de desarrollo de *software*.

El proceso de desarrollo de *software* no es único. No existe un proceso de *software* universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de *software*.

A pesar de la variedad de propuestas de proceso de *software*, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos:

Figura 1.1: Proceso de desarrollo de *software*.

El proceso de desarrollo de *software* no es único. No existe un proceso de *software* universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de *software*.

A pesar de la variedad de propuestas de proceso de *software*, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos:

1.1.2.1. Actividades fundamentales del proceso de software

A continuación se describen las actividades fundamentales del proceso de *software*:

- Especificación de *software*: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el *software*.
- Diseño e Implementación: Se diseña y construye el *software* de acuerdo a la especificación.

- Validación: El *software* debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- Evolución: El *software* debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Pressman menciona un conjunto de "actividades protectoras", que se aplican a lo largo de todo el proceso del *software*. Ellas se señalan a continuación:

- Seguimiento y control de proyecto de *software*.
- Revisiones técnicas formales.
- Garantía de calidad del *software*.
- Gestión de configuración del *software*.
- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.
- Gestión de riesgos.

1.1.2.2. Elementos del proceso de *software*

Pressman caracteriza un proceso de desarrollo de *software* como se muestra en la **figura 1.2**. Los elementos involucrados se describen a continuación:

- **Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de *software*, con independencia del tamaño o complejidad.
- **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del *software*, hitos de proyectos, entregas y productos de trabajo del *software*, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de *software* y los requisitos del equipo del proyecto.
- **Las actividades de protección**, tales como garantía de calidad del *software*, gestión de configuración del *software* y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.



Figura 1.2: Elementos del proceso del *software*.

1.1.2.3. Relación entre los elementos del proceso de software

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de *software* es establecer las relaciones entre elementos que permitan responder **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo.

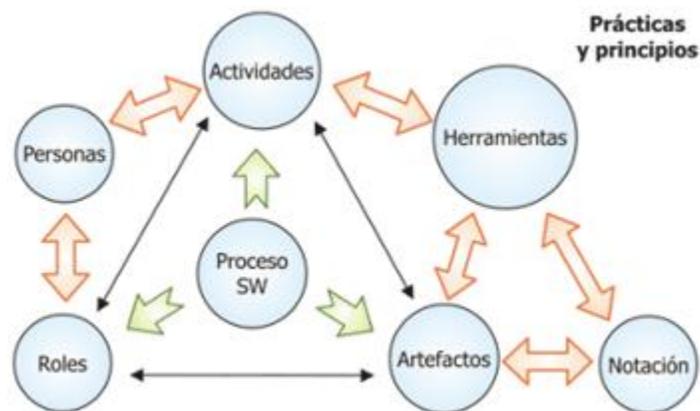


Figura 1.3: Relación entre elementos del proceso del *software*.

En la **figura 1.3** se muestran los elementos de un proceso de desarrollo de *software* y sus relaciones. Así las interrogantes se responden de la siguiente forma:

- **Quién:** Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.

- **Qué:** Un Artefacto¹ es producido por un Rol en una de sus Actividades. Los Artefactos se especifican utilizando Notaciones específicas. Las Herramientas apoyan la elaboración de Artefactos soportando ciertas Notaciones.

- **Cómo y Cuándo:** Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

¹ Un artefacto es una pieza de información que es producida, modificada o usada por el proceso, define un área de responsabilidad para un rol y está sujeta a control de versiones. Un artefacto puede ser un modelo, un elemento de modelo o un documento.

1.2. Proceso de negocio de *software* versus proceso de *software* de negocio

- Proceso de negocio de *software* es el proceso (organizacional) de un negocio cuyo eje de actividad es cualquier actividad del ciclo de vida del *software*: desde la concepción hasta la implantación llave en mano, pasando por comercialización, desarrollo, y/o estudio, etc. En otras palabras sería el proceso de negocio adscrito a la cadena de valor de un negocio de *software*, dejando claro que no necesariamente puede desarrollarse sólo al desarrollo de *software*, sino también a otras facetas a la actividad de negocio (por ejemplo, comercialización de *software*). Esto no excluye, que se incluyan actividades de negocio tradicional para soportar las actividades centrales del negocio (por ejemplo, contabilidad y finanzas del negocio). Por ejemplo: una empresa de benchmarking de *software* cuyo proceso de negocio de *Software* sería todas las fases de estudio, investigación, análisis y test de *software* y el proceso de negocio sería estas mismas fases más otras como las propias de gerencia y de administración general. Por ejemplo: una empresa que desarrolla *software* ERP, su proceso de negocio, sería la producción de *software* y -entre otras- las actividades de difusión y comercialización del ERP, y el proceso de negocio de *Software* serían los procesos claves de producción, como la customización al cliente de los ERP.

- Proceso de *software* de negocio es el proceso de *software* de un *software* de un negocio o en otras palabras el proceso de un desarrollo de *software* orientado a un negocio. Por ejemplo: el análisis, diseño, programación, implantación y puesta en marcha de un *software* en un empresa cualquiera.

1.3. La calidad y el riesgo como ejes del proceso

La identificación y gestión de los riesgos asociados a los requisitos del *software*, individuales y a grupos de ellos, desde la fase de ingeniería de requisitos puede permitir minimizarlos, evadirlos y controlarlos. El enfrentamiento proactivo de los riesgos que pueden afectar al desarrollo o a la calidad de los requisitos y las acciones para evitarlos, permitirían minimizar problemas que persisten en el desarrollo de *software*. Son de mayor importancia los riesgos asociados a las principales características de calidad de los requisitos.

La ingeniería de requisitos es un área de investigación que procura atacar un punto fundamental en el proceso, que es la definición de lo que se quiere producir. Jackson afirma que la ingeniería de requisitos se ubica en el punto de encuentro entre lo informal y lo formal del desarrollo de *software* (Jackson, 2001).¹

Para lograr producir aquello que el cliente requiere, en el plazo solicitado y ajustados al presupuesto asignado, se necesita desarrollar un proceso que incluya desde la etapa más temprana la gestión de los riesgos asociados a los requisitos, de forma que se contribuya al mejoramiento gradual del proceso de desarrollo y la gestión de un proyecto de *software* que logre la satisfacción del cliente en estas organizaciones.

¹ JACKSON, M. (2001). *Software Requirements and Specifications*. Addison-Wesley.

1.3.1. La gestión de riesgos en el ámbito del *software*

La gestión de riesgos en el ámbito del *software* procura formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de *software*, mediante la generación de principios y buenas prácticas de aplicación realista (Ropponen, 2000).¹ Hasta el momento se ha propuesto y utilizado diferentes enfoques de gestión del riesgo desde que Boehm (Boehm, 1988)² atrajo a la comunidad de ingeniería del *software* hacia la gestión del riesgo. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y sistemática métodos específicos para gestionar los riesgos en sus proyectos *software*.

En (Pressman, 2002)³ se presenta la definición de riesgo dada por Robert Charette en (Charette, 1989)⁴ donde plantea que en primer lugar, el riesgo afecta a los futuros acontecimientos. En segundo lugar, el riesgo implica cambios. En tercer lugar, el riesgo implica elección, y la incertidumbre que entraña esta. Cuando se considera el riesgo en el contexto de la ingeniería de *software*, los tres pilares de Charette se hacen continuamente evidentes. Es indiscutible que están presentes permanentemente las características de incertidumbre (acontecimiento

que caracteriza al riesgo y que puede o no ocurrir) y de pérdida (si el riesgo se convierte en una realidad ocurrirán consecuencias no deseables o pérdidas).

¹ ROPPPONEN, J.; LYYTINEN, K. (2000). *Components of Software Development Risk: Hot to address Them?* IEEE transactions on software engineering.

² BOEHM, B. (1988). *A Spiral Model of Software Development and Enhancement*. IEEE Computer. Vol. 21, # 5.

³ PRESSMAN, Roger S. (2002). *Ingeniería de Software. Un enfoque práctico*. Quinta edición. McGraw-Hill.

⁴ CHARETTE, R. N. (1989). *Software Engineering Risk Analysis and Management*. McGraw-Hill.

1.3.2. Categorías de riesgos

Están definidas las categorías de riesgos: los riesgos del proyecto, que amenazan el plan; los riesgos técnicos, que amenazan la calidad y la planificación temporal; y los riesgos del negocio, que amenazan la viabilidad del proyecto o del producto. Otra categorización a considerar a partir del conocimiento que se tenga de ellos: los riesgos conocidos (los que se descubren en las evaluaciones); los riesgos predecibles (se extrapolan de la experiencia) y los riesgos impredecibles (pueden ocurrir, pero es muy difícil identificarlos de antemano).

1.3.3. Estrategias frente al riesgo

También son claras las estrategias frente al riesgo. Por un lado están las reactivas, cuyo método es evaluar las consecuencias del riesgo cuando este ya se ha producido (ya no es un riesgo) y actuar en consecuencia. Este tipo de estrategias acarrea consecuencias negativas, al poner el proyecto en peligro. Y por el otro las preactivas, que aplican el método de evaluación previa y sistemática de los riesgos y sus posibles consecuencias, a la par que conforman planes de contingencias para evitar y minimizar las consecuencias. Consecuentemente, este tipo de estrategias permite lograr un menor tiempo de reacción ante la aparición de riesgos impredecibles.

De acuerdo con (Pressman, 2002), la administración o gestión de riesgos es un proceso iterativo que se aplica durante todo el proyecto y se desarrolla en cuatro etapas. Los resultados de la administración de riesgos deben ser documentados en un plan de administración de riesgos. La **figura 1.4** refleja el procedimiento.

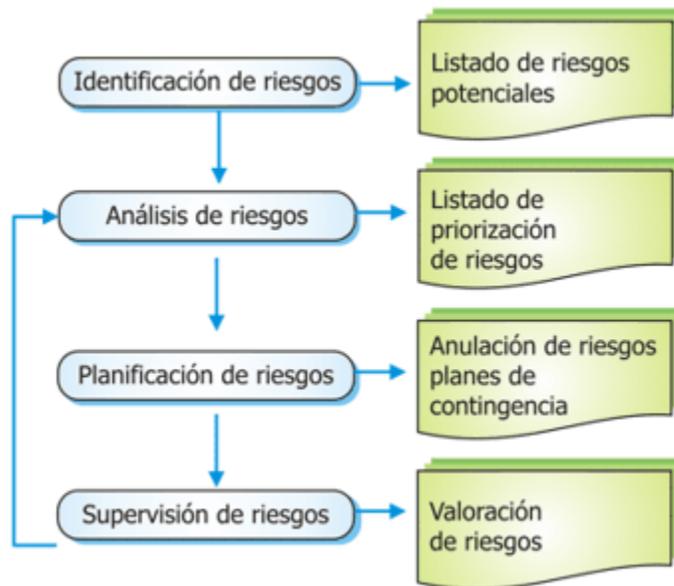


Figura 1.4: Procedimiento de gestión de riesgos.

1.4. Los procesos de apoyo organizacional: SCM y SEPG

A continuación se describen los siguientes procesos de apoyo organizacional: SCM *Software Configuration Management* y SEPG *Software Engineering Process Group*.

1.4.1. SCM - *Software Configuration Management*

SCM es la actividad que se aplica durante todo el proceso, desde el inicio del proyecto hasta que el mismo esté fuera de uso. Comprende básicamente la administración de cambios y control de versiones. Está orientada a identificar el cambio, controlar el cambio, garantizar su correcta implementación e Informar el mismo a los interesados.

1.4.1.1. Qué involucra SCM

A continuación se describen las actividades que involucra SCM:

a) **Identificación de los *artifacts*.**¹

Esta actividad responde el siguiente interrogante ¿Qué componentes quiero administrar?

- Objetos básicos: es un sólo *deliverable*.
- Objetos compuestos: se incluyen varios *deliverables*.

- Interrelaciones entre objetos.
- Evolución del objeto.

b) **Control de cambios.**

Procedimientos humanos y automáticos para el control de cambio, a continuación se describen el ciclo de vida de un cambio:

- Solicitud de cambio,
- evaluación,
- ejecución,
- control; y,
- nueva versión.

Debe asegurarse:

- Control de accesos: quiénes pueden modificar los componentes
- Control de sincronización: que no se pierdan cambios porque más de un recurso hace modificaciones sobre un mismo componente.

c) **Configuración del software.**

Es toda la información producida a lo largo del ciclo de vida. Pone énfasis en los cambios y presta especial atención a los cambios que se producen en un único *deliverable*.

d) **Línea base.**

Cualquier *deliverable* revisado y aprobado formalmente (está en la línea base) solo puede ser modificado bajo un procedimiento formal de "control de cambios".

e) **Control de versiones.**

Permite especificar configuraciones alternativas del SW mediante la selección de versiones adecuadas. Una nueva versión se define cuando se realizan cambios significativos en uno o más objetos. Existen herramientas de control de versiones que guardan la versión original más los cambios (deltas).

f) Auditoría de la configuración.

Debe asegurarse que el cambio ha sido implementado correctamente y debo saber en qué estado está cada componente.

g) Informes de estado.

Los informes de estado responden los siguientes interrogantes:

- ¿Cuándo se realizó el cambio?
- ¿Dónde se cambió un componente?
- ¿Quién es el autor del cambio?
- ¿Qué se cambió?
- ¿Cuál es el alcance del cambio?

h) Cambio - Cambiando el ciclo de vida.

A medida que se avanza a lo largo del ciclo de vida se debe poder restringir que tipo de cambios se puede realizar sobre el *software*. El ciclo de vida determina dos aspectos que influyen sobre el plan de SCM: cantidad de versiones que deberá administrar y etapas durante las cuales aceptará sean efectuados cambios sobre el producto en sí.

La rigidez de los controles será directamente proporcional al avance del proyecto en la fase. A medida que se acerca al final de la fase, se debe estar acercando a tener algún entregable listo (versión). Esto hace que las restricciones para modificar el *software* sean incrementadas para asegurar que no existan desvíos, solo los cambios de urgencia o muy bien justificados tendrán lugar en esta etapa. De la misma manera, en ciclos de vida con un alto grado de prototipación, se hará énfasis en la auditoría en lugar de restringir que el equipo pueda modificar el producto.

¹ Cualquier elemento de un producto de *software* que esté sujeto a cambios. Esto incluye código fuente, documentación, QA test plans, QA data, librerías, código objeto, etc. También es conocido en la terminología traducida que encontramos como "ítem de configuración".

1.4.1.2. Actividades esenciales de SCM

A continuación, se listan las actividades básicas de SCM:

- Identificar y almacenar *artifacts* en un repositorio seguro.

- Controlar y auditar cambios sobre los *artifacts*.
- Organizar *artifacts* en forma de componentes "versionados".
- Crear *baselines* para cada *milestone* (puntos de control) del proyecto.
- Registrar y hacer seguimiento a *change requests*.
- Organizar e integrar conjuntos consistentes de versiones a través de actividades.
- Mantener espacios de trabajo consistentes y estables.
- Soportar cambios concurrentes sobre *artifacts*.
- Integrar en forma temprana y frecuente.
- Asegurar que sea posible reproducir construcciones de *software*.

1.4.2. SEPG - Software Engineering Process Group

Es el punto focal de la organización de las actividades de mejora de procesos de *software*. Estas personas desarrollan principalmente las siguientes actividades: Realizan evaluaciones de la capacidad de organización, desarrollar planes para implementar las mejoras necesarias, coordinar la ejecución de esos planes, y medir la eficacia de estos esfuerzos. SEPGs exitosos requieren habilidades y conocimientos especializados de muchas áreas fuera de la ingeniería de *software* tradicional.¹

¹ FOWLER, Priscilla; RIFKIN, Stanley. (1990). *Software Engineering Process Group Guide*. CMU/SEI-90-TR-024. Carnegie Mellon University.
Enlace web: <http://www.sei.cmu.edu/library/abstracts/reports/90tr024.cfm>. [Leído: 20 de enero 2010, 12.00 h GMT+1].

1.4.2.1. Actividades SEPG

Las siguientes son algunas de las actividades del grupo de procesos de Ingeniería de *software*:¹

- Obtiene y mantiene el apoyo de todos los niveles de gestión.
- Facilita la evaluación de los procesos de *software*.

- Trabaja con los gerentes de línea, cuyos proyectos se ven afectados por cambios en las prácticas de ingeniería de *software*, proporcionando una amplia perspectiva de los esfuerzos de mejora y ayudarles a fijar las expectativas.
- Mantiene relaciones de colaboración de trabajo con los ingenieros de *software*, especialmente para obtener, planificar, e instalar nuevas prácticas y tecnologías.
- Los arreglos para cualquier capacitación o educación continua relacionados con mejoras en el proceso.
- Pistas, monitores, y los informes sobre el estado de los esfuerzos de mejora en particular.
- Facilita la creación y mantenimiento de las definiciones de procesos, en colaboración con los directores y el personal de ingeniería.
- Mantiene una base de datos de proceso.
- Proporciona consultoría de procesos para proyectos de desarrollo y de gestión.

¹ *Software Engineering Institute*. Carnegie Mellon. (2010). *Software Engineering Process Group Guide*. [en línea].
 Enlace web: <http://www.sei.cmu.edu/library/abstracts/reports/90tr024.cfm>. [Leído: 20 de enero 2010, 14.00 h GMT+1].

1.4.2.2. Enfoques SEPG

Cada SEPG tiene un enfoque y misión diferente. A continuación se describen algunos:

- "Trabajo" SEPG de que realmente desarrollar e implementar el proceso como un tipo de equipo de consultores internos.
- "Supervisión" SEPG que supervisará el proceso de La arquitectura, gestión de cambios, y dar prioridad a él (una especie de proceso de CCB).
- "Qué Deliberante" SEPG es el enfoque de proceso de debate y desarrollo de estrategia para un proceso de arquitectura y el despliegue.
- "Virtual" SEPG de que se componen de representantes de toda la organización que dedicar una cierta cantidad de tiempo para el esfuerzo y son responsables de la implementación y la formación de todos los demás en la organización.

1.5. Herramientas CASE y entornos de trabajo

Las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del *software* en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Sistema de *software* que intenta proporcionar ayuda automatizada a las actividades del proceso de *software*. Los sistemas CASE a menudo se utilizan como apoyo al método.

1.5.1. Definiciones CASE

Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

La sigla genérica para una serie de programas y una filosofía de desarrollo de *software* que ayuda a automatizar el ciclo de vida de desarrollo de los sistemas.

Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un potencial efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de *software* y las metodologías de desarrollo de *software* formales.

La realización de un nuevo *software* requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de *software*.

1.5.2. Herramientas CASE en función de las fases del ciclo de vida

Las herramientas CASE, en función de las fases del ciclo de vida abarcadas, se pueden agrupar de la forma siguiente:¹

- Herramientas integradas, I-CASE (*Integrated CASE*, CASE integrado): abarcan todas las fases del ciclo de vida del desarrollo de sistemas. Son llamadas también CASE *workbench*.

- Herramientas de alto nivel, U-CASE (*Upper CASE* - CASE superior) o *front-end*, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.
- Herramientas de bajo nivel, L-CASE (*Lower CASE* - CASE inferior) o *back-end*, dirigidas a las últimas fases del desarrollo: construcción e implantación.
- Juegos de herramientas o *Tools-Case*, son el tipo más simple de herramientas CASE. Automatizan una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

¹ SCRIB. (2009). *Ingeniería de Software*. [en línea].

Enlace web: <http://www.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>. [Leído: 6 de enero 2010, 12.00 h GMT+1].

1.5.3. Ventajas de la utilización de herramientas CASE

La mejor razón para la creación de estas herramientas fue el incremento en la velocidad de desarrollo de los sistemas. Por esto, las compañías pudieron desarrollar sistemas sin encarar el problema de tener cambios en las necesidades del negocio, antes de finalizar el proceso de desarrollo. También permite a las compañías competir más efectivamente usando estos sistemas desarrollados nuevamente para compararlos con sus necesidades de negocio actuales. En un mercado altamente competitivo, esto puede hacer la diferencia entre el éxito y el fracaso.

Las herramientas CASE también permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar. La introducción de CASE integradas está comenzando a tener un impacto significativo en los negocios y sistemas de información de las organizaciones. Con un CASE integrado, las organizaciones pueden desarrollar rápidamente sistemas de mejor calidad para soportar procesos críticos del negocio y asistir en el desarrollo y promoción intensiva de la información de productos y servicios. La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados y, en segundo término, el aumento de la productividad. Para conseguir estos dos objetivos es conveniente contar con una organización y una metodología de trabajo, además de la propia herramienta.

1.6. Modelos de procesos de *software* y de mejorar de calidad

A continuación se describen los siguientes modelos de proceso de *software* y de mejorar de calidad:

- CMM - *Capability Maturity Model*,
- CMMI - *Capability Maturity Model Integration*,
- SPICE; y,
- Trillium *Model*.

1.6.1. CMM - *Capability Maturity Model*

El Modelo de Capacidad y Madurez o CMM (*Capability Maturity Model*), es un modelo de evaluación de los procesos de una organización. Fue desarrollado inicialmente para los procesos relativos al desarrollo e implementación de *software* por la Universidad Carnegie-Mellon para el SEI (*Software Engineering Institute*). El SEI es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMM" es una marca registrada del SEI.¹

A partir de noviembre de 1986 el SEI, a requerimiento del Gobierno Federal de los Estados Unidos de América (en particular del Departamento de Defensa, DoD), desarrolló una primera definición de un modelo de madurez de procesos en el desarrollo de *software*, que se publicó en septiembre de 1987. Este trabajo evolucionó al modelo CMM o SW-CMM (CMM for *Software*), cuya última versión (v1.1) se publicó en febrero de 1993.

¹ SERGIO Fernández. (2003). *Ingeniería de Software*. [en línea]. Enlace web: <http://apuntes-utn.com.ar/apuntes/documento.aspx?IdDocumento=433&bajar=1>. [Leído: 6 de enero 2010, 14.00 h GMT+1].

1.6.1.1. KPA - *Key Process Area*

Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso (KPA - *Key Process Area*). Para cada área de proceso define un conjunto de buenas prácticas que habrán de ser:

- Definidas en un procedimiento documentado.

- Provistas (la organización) de los medios y formación necesarios.
- Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas).
- Medidas.
- Verificadas.

Cada nivel, menos el primero, tiene asociado un conjunto de KPAs (*Key Process Areas*) que definen un conjunto de objetivos a cumplir. Cada KPA incluye un número de prácticas claves que llevan a cumplir esos objetivos. No es necesario implementar todas las prácticas. Sí es necesario cumplir con todos los objetivos del nivel a acreditar y sus inferiores.

Así es como el modelo CMM establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables. Con la excepción del primer nivel, cada uno de los restantes niveles de madurez está compuesto por un cierto número de Áreas Claves de Proceso, conocidas a través de la documentación del CMM por su sigla inglesa: KPA.

Cada KPA identifica un conjunto de actividades y prácticas interrelacionadas, las cuales cuando son realizadas en forma colectiva permiten alcanzar las metas fundamentales del proceso. Las KPAs pueden clasificarse en 3 tipos de proceso: Gestión, Organizacional e Ingeniería.

Las prácticas que deben ser realizadas por cada Área Clave de Proceso están organizadas en 5 características comunes, las cuales constituyen propiedades que indican si la implementación y la institucionalización de un proceso clave es efectivo, repetible y duradero.

Estas 5 características son:

- Compromiso de la realización,
- La capacidad de realización,
- Las actividades realizadas,
- Las mediciones y el análisis,
- La verificación de la implementación.

1.6.1.2. Niveles de Madurez

La mejora continua del proceso está basada en una cantidad de pequeños pasos evolutivos, más que en innovaciones revolucionarias. El CMM provee un marco para organizar estos pasos en cinco niveles de madurez que establecen sucesivas bases para una mejora continua. Estos cinco niveles definen una escala ordinal para medir la madurez del proceso de desarrollo de *software* de una organización y para evaluar su capacidad de proceso de *software*. Los niveles también ayudan a una organización a la hora de definir prioridades en sus esfuerzos por mejorar. En la **tabla 1.1**, se muestra las características de los 5 niveles de madurez

A su vez estas Áreas de Proceso se agrupan en cinco "niveles de madurez", de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez.

Los niveles son:

a) **Inicial.** Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de *software*. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobrecostos. El resultado de los proyectos es impredecible.

b) **Repetible.** En este nivel las organizaciones disponen de unas prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente.

c) **Definido.** Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión por pares (peer reviews).

d) **Gestionado.** Se caracteriza porque las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El *software* resultante es de alta calidad.

e) **Optimizado.** La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

NIVELES	CARACTERÍSTICAS
Nivel 1: Inicial	<ul style="list-style-type: none"> - Sin proceso definido. - Sin estándares o si existen son ignorados. - Poca habilidad para estimar. - El éxito está basado en "héroes". - Poca visibilidad. - <i>Deadlines</i> más importantes que calidad. - Proyectos fuera de término y presupuesto.
Nivel 2: Repetible	<ul style="list-style-type: none"> - Administración y control de proyectos. - Se crean y documentan los procesos. - Se recolectan métricas y se realiza <i>tracking</i> del progreso. - Mejora la estimación. - Menos desvíos que en el Nivel 1.
Nivel 3: Definido	<ul style="list-style-type: none"> - Implementación de estándares, documentación de procesos para el desarrollo y mantenimiento. - Guías para la adecuación de los procesos en un proyecto específico. - Incremento de efectividad y reducción de costos y tiempos. - Se crea el <i>Software Engineering Process Group</i> (SEPG).
Nivel 4: Administrado	<ul style="list-style-type: none"> - El foco de este nivel es la medición. Implementación de un plan de métricas para evaluar el proceso de la organización. - La calidad del producto debe ser alta y predecible.
Nivel 5: Optimizado	<ul style="list-style-type: none"> - Prevención de los defectos. Seguimiento y análisis. - Mejora continua de los procesos (calidad, productividad y tiempos de ciclos). - Administración de la incorporación de nuevas tecnologías.

Tabla 1.1. Características de los niveles de madurez

1.6.1.3. Evolución de las organizaciones según CMM

Las organizaciones que utilizan CMM para mejorar sus procesos disponen de una guía útil para orientar sus esfuerzos. Además, el SEI proporciona formación a evaluadores certificados (*Lead Assessors*) capacitados para evaluar y certificar el nivel CMM en el que se encuentra una organización. Esta certificación es requerida por el Departamento de Defensa de los Estados Unidos, pero también es utilizada por multitud de organizaciones de todo el mundo para valorar a sus subcontratistas de *software*.

Se considera típico que una organización dedique unos 18 meses para progresar un nivel, aunque algunas consiguen mejorarlo. En cualquier caso requiere un amplio esfuerzo y un compromiso intenso de la dirección.

Como consecuencia, muchas organizaciones que realizan funciones de factoría de *software* o, en general, *outsourcing* de procesos de *software*, adoptan el modelo CMM y se certifican en alguno de sus niveles. Esto explica que uno de los países en el que más organizaciones certificadas existan sea India, donde han florecido las factorías de *software* que trabajan para clientes estadounidenses y europeos.

1.6.2. CMMI - *Capability Maturity Model Integration*

CMMI es un modelo de aseguramiento de la calidad que busca la mejora continua de las organizaciones mediante el análisis y re-diseño de los procesos que subyacen en la organización. Fue creado por el SEI (*Software Engineering Institute*) de la Universidad de Carnegie-Mellon y patrocinado por el Ministerio de Defensa de los Estados Unidos. Con el propósito de lograr la mejora de los procesos, CMMI provee:

- Una forma de integrar los elementos funcionales de una organización.
- Un conjunto de mejores prácticas basadas en casos de éxito probado de organizaciones experimentadas en la mejora de procesos.
- Ayuda para identificar objetivos y prioridades para mejorar los procesos de la organización, dependiendo de las fortalezas y debilidades de la organización que son obtenidas mediante un método de evaluación.
- Un apoyo para que las empresas complejas en actividades productivas puedan coordinar sus actividades en la mejora de los procesos.
- Un punto de referencia para evaluar los procesos actuales de la organización.

1.6.2.1. Antecedentes

CMMI v1.2 corresponde a la tercera versión entregable del modelo CMMI, posterior a las versiones 1.02 (primera versión año 2000) y 1.1 (año 2002). Las versiones previas sirvieron como retroalimentación para que los propios usuarios, evaluadores y evaluados hicieran acotaciones sobre posibles mejoras, las cuales fueron estudiadas, refinadas y algunas incluidas en la versión 1.2. CMMI v1.2 para desarrollo, que corresponde a una de tres constelaciones de prácticas, es una guía que ayuda a manejar, medir y monitorear procesos utilizados en el desarrollo de productos y servicios de una organización, y contiene prácticas ligadas a la administración de proyectos, administración de procesos, ingeniería y soporte. Las otras dos constelaciones son CMMI para Adquisición que provee una guía para liderar la adquisición informada y decisiva, y CMMI para Servicios que proporciona

una guía para la entrega de servicios a clientes internos y externos de la organización. Ambas constelaciones se encuentran aún en desarrollo.

Junto con CMMI se desarrolló y publicó el método de evaluación "*Assessment Requirements for CMMI (ARC)*" en el año 2000, el cual define los requerimientos considerados esenciales para realizar una evaluación de CMMI en una organización y "*Standard CMMI Appraisal Method for Process Improvement*", (SCAMPI), manual seguido por los evaluadores para medir el nivel de madurez de una organización. Estos dos documentos también se han actualizado como consecuencia de la retroalimentación de la comunidad involucrada en CMMI, generando la última versión 1.2 de SCAMPI y ARC ambas publicadas el año 2006.

1.6.2.2. Representaciones

La representación usada en CMMI entrega una guía para efectuar las actividades de mejora de los procesos y es utilizada en el método de evaluación. Según el modelo se tienen dos formas para mejorar. Una forma es mejorar un proceso específico o un conjunto de ellos usando la "representación continua" (*continuous representation*) y la otra es la mejora de la organización completa según los procesos definidos y ocupados usando la "representación escalonada" o por "etapas" (*Staged Representation*). En la **tabla 1.2** se muestran los niveles para estos dos tipos de representaciones.

1.6.2.2.1. Representación continua

La representación continua se focaliza en la mejora de un proceso o un conjunto de ellos relacionado(s) estrechamente a un área de proceso en que una organización desea mejorar, por lo tanto una organización puede ser certificada para un área de proceso en cierto nivel de capacidad. Existen seis niveles de capacidad por donde transitan los procesos asociados a un área de proceso y cada nivel es construido sobre el nivel anterior, es decir para que un proceso alcance un nivel de capacidad necesariamente debe haber alcanzado el nivel anterior.

	REPRESENTACIÓN CONTINUA	REPRESENTACIÓN ESCALONADA
	Nivel de capacidad	Nivel de madurez
Nivel 0	Incompleto	-
Nivel 1	Realizado	Inicial

Nivel 2	Manejado	Manejado
Nivel 3	Definido	Definido
Nivel 4	Manejado cuantitativamente	Manejado cuantitativamente
Nivel 5	Optimizado	Optimizado

Tabla 1.2. Niveles de representación continua y escalonada.

Los niveles de capacidad son:

- **Nivel 0 - Incompleto:** Un proceso es denominado "proceso incompleto" cuando una o más objetivos específicos del área de proceso no son satisfechos.

- **Nivel 1 - Realizado:** Un proceso es denominado "proceso realizado" cuando satisface todos los objetivos específicos del área de proceso. Soporta y permite el trabajo necesario para producir artefactos.

- **Nivel 2 - Manejado:** Un proceso es denominado como "proceso manejado" cuando tiene la infraestructura base para apoyar el proceso. El proceso es planeado y ejecutado en concordancia con la política, emplea gente calificada los cuales tienen recursos adecuados para producir salidas controladas; involucra partes interesadas; es monitoreado, controlado y revisado; y es evaluado según la descripción del proceso.

- **Nivel 3 - Definido:** Un proceso denominado "proceso definido" es adaptado desde el conjunto de procesos estándares de la organización de acuerdo a las guías de adaptación de la organización, y aporta artefactos, medidas, y otra información de mejora a los activos organizacionales.

- **Nivel 4 - Manejado cuantitativamente:** Un proceso denominado "proceso manejado cuantitativamente" es controlado usando técnicas estadísticas y otras técnicas cuantitativas. Objetivos cuantitativos para la calidad y realización del proceso son establecidos y usados como criterios para manejar el proceso.

- **Nivel 5 - Optimización:** Un proceso denominado "proceso optimización" es mejorado basado en el entendimiento de causas comunes de variación del proceso. Un proceso en optimización se focaliza en la mejora continua del proceso realizado a través de mejoras incrementales y usando innovación tecnológica.

1.6.2.2.2. Representación escalonada

En la representación escalonada o por etapas se ofrece un método estructurado y sistemático de mejoramiento de procesos, que implica mejorar por etapas o niveles. Al alcanzar un nivel, la organización se asegura de contar con una infraestructura robusta en términos de procesos para optar a alcanzar el nivel siguiente. Por lo tanto es una organización la que puede ser certificada bajo un nivel, en este caso llamado nivel de madurez. Según esta representación un nivel de madurez está compuesto por áreas de procesos ver **tabla 1.3**, en donde los objetivos asociados a ese nivel deben ser cumplidos para que la organización pueda certificarse en aquel nivel de madurez. Hay cinco niveles de madurez, los que son descritos a continuación en la **tabla 1.3**.

NIVELES	CARACTERÍSTICAS
Nivel 1: Iniciado	<p>La organización usualmente no provee un ambiente estable para soportar los procesos. Éxitos en estas organizaciones se debe a la competencia y esfuerzos heroicos de la gente dentro de la organización y no al uso de procesos probados. A pesar de este caos, organizaciones pertenecientes al nivel de madurez 1 con frecuencia producen productos y servicios que funcionan; sin embargo, ellos frecuentemente exceden sus presupuestos y no cumplen sus planes.</p> <p>Estas organizaciones son caracterizadas por la tendencia a no cumplir sus compromisos, al abandono de procesos durante tiempos de crisis, y a la incapacidad para repetir sus éxitos.</p> <p>El Nivel 1 está caracterizado además por la realización de trabajo redundante, por personas que no comparten sus métodos de trabajo a lo largo de la organización y cuando una persona clave en un área de negocio específica dentro de la organización se marcha, su conocimiento se va con ella y se pierde para la organización. Es claro que el Nivel 1 es uno donde ninguna organización quiere estar y donde por lo general la mayoría que no tiene sus procesos definidos se encuentra.</p>
Nivel 2: Manejado	<p>En el nivel de madurez 2 se ordena el caos. En el nivel 2 las organizaciones se enfocan en tareas cotidianas referentes a la administración. Cada proyecto de la organización cuenta con una serie de procesos para llevarlo a cabo, los cuales son planeados y ejecutados de acuerdo con políticas establecidas; los proyectos utilizan gente capacitada quienes disponen de recursos para producir salidas controladas; se involucran a las partes interesadas; son monitoreados, controlados y revisados; y son evaluados según la descripción del proceso.</p>

	<p>La disciplina del proceso reflejada por el nivel de madurez 2 ayuda a asegurar que existen prácticas y los proyectos son realizados y manejados de acuerdo a los planes documentados.</p> <p>En el nivel de madurez 2 el estado de los artefactos y la entrega de los servicios siguen planes definidos. Acuerdos son establecidos entre partes interesadas y son revisados cuando sea necesario. Los artefactos y servicios son apropiadamente controlados. Estos además satisfacen sus descripciones especificadas, estándares, y procedimientos.</p>
<p>Nivel 3: Definido</p>	<p>En el nivel de madurez 3, procesos son caracterizados y entendidos de buena forma, y son descritos en estándares, procedimientos, herramientas, y métodos. El conjunto de procesos estándares de la organización, los cuales son la base para el nivel de madurez 3, es establecido y mejorado continuamente. Estos procesos estándares son usados para establecer consistencia a través de la organización. Los proyectos establecen sus procesos adaptando el conjunto de procesos estándares de la organización de acuerdo a guías de adaptación.</p> <p>Una diferencia importante entre el nivel 2 y 3 es el alcance de los estándares: la descripción de procesos y los procedimientos. En el nivel de madurez 2, los estándares pueden ser un poco diferentes en cada instancia específica del proceso (por ejemplo sobre un proyecto particular). En el nivel de madurez 3, los estándares, descripción de procesos y procedimientos para un proyecto, son adaptados desde un conjunto de procesos estándares de la organización a un particular proyecto o unidad organizacional y así son más consistentes.</p> <p>Otra distinción crítica es que el nivel de madurez 3, los procesos son típicamente descritos más rigurosamente que en el nivel 2. Un proceso definido claramente plantea el propósito, entradas, criterios de entrada, actividades, roles, medidas, pasos de verificación, salidas y criterios de salida. En el nivel de madurez 3, procesos son manejados más proactivamente entendiendo las interrelaciones de las actividades y medidas detalladas del proceso, sus artefactos y sus servicios.</p>
<p>Nivel 4: Manejado cuantitativamente</p>	<p>En el nivel de madurez 4, la organización y proyectos establecen objetivos cuantitativos para medir la calidad y realización de los procesos y los usa como criterios en el manejo de ellos. Los objetivos cuantitativos son definidos en base a las necesidades de clientes, usuarios finales, organización, y actores de los procesos. La calidad y realización de procesos son entendidos en términos estadísticos y son manejados durante todo el ciclo de vida del proceso.</p> <p>Para subprocesos seleccionados, se recolectan y analizan estadísticamente medidas sobre la realización de procesos. Estas métricas son incorporadas en el repositorio de métricas de la organización para apoyar la toma de</p>

	<p>decisiones. Causas especiales de variación de procesos son identificadas y, cuando sea necesario, las fuentes de estas causas son corregidas para prevenir futuras ocurrencias.</p> <p>Una diferencia importante entre los niveles 3 y 4 es la capacidad de predicción de la realización del proceso. En el nivel de madurez 4, la realización de procesos es controlada usando técnicas estadísticas y cuantitativas, y el proceso es cuantitativamente predecible, en cambio en el nivel de madurez 3 la realización del proceso es sólo predecible cualitativamente.</p>
<p>Nivel 5: Optimizado</p>	<p>En el nivel de madurez 5, una organización mejora continuamente sus procesos basándose en el conocimiento de las causas comunes de variación inherente en los procesos. El nivel de madurez 5 se focaliza sobre la mejora continua de los procesos a través de mejoras continuas, incrementales y tecnológicas. Los objetivos de mejora cuantitativa de procesos para la organización son establecidos, continuamente revisados para reflejar cambios en los objetivos del negocio y usados como criterio en la mejora de procesos. Los efectos del empleo de las mejoras de procesos son medidos y evaluados contra los objetivos de mejora cuantitativa del proceso.</p> <p>Una diferencia importante entre el nivel de madurez 4 y 5 es el enfoque de la variación de los procesos. En el nivel de madurez 4, la organización está orientada a encontrar causas especiales de variación y proveer una predicción estadística de los resultados. Sin embargo, los resultados pueden ser insuficientes para alcanzar los objetivos establecidos. En el nivel de madurez 5 la organización está enfocada en las causas comunes de variación de procesos y modificar los procesos afectados para mejorar la realización de ellos y alcanzar los objetivos cuantitativos de mejora de procesos.</p> <p>Dado a que la organización con que se trabajará quiere certificarse en forma organizacional en Nivel de madurez 3, en adelante sólo se detallará el modelo según la Representación Escalonada.</p>

Tabla 1.3. Características de los niveles CMMI.

1.6.2.3. Estructura del CMMI

Un área de proceso es un conjunto de prácticas relacionadas que cuando son implementadas colectivamente, satisfacen un conjunto objetivos considerados importantes para mejorar esa área de proceso. Las áreas de proceso del modelo son 22. En la **tabla 1.4** se indica los nombres de las áreas de proceso junto con su abreviación. Cada una de ellas es implementada para alcanzar el nivel de madurez correspondiente y se agrupan de acuerdo a cuatro categorías:

Administración de Procesos, Administración de Proyectos, Ingeniería y Soporte. Este agrupamiento es realizado para mostrar cómo se relaciona cada área de proceso dentro de una categoría. Sin embargo, áreas de procesos de distintas categorías pueden encontrarse relacionadas, pero dado que en este documento se desarrollarán sólo áreas de procesos de una misma categoría (Ingeniería) estas relaciones se desprecian.

ÁREA DE PROCESO	CATEGORÍA	NIVEL DE MADUREZ
Análisis y Resolución Causales (CAR)	Soporte	5
Análisis y Resolución de Decisiones (DAR)	Soporte	3
Aseguramiento de la Calidad de Procesos y Productos (PPQA)	Soporte	2
Definición de Procesos Organizacionales +IPPD(OPD +IPPD)	Gestión de procesos	3
Desarrollo de Requerimientos (RD)	Ingeniería	3
Entrenamiento Organizacional (OT)	Gestión de procesos	3
Administración Cuantitativa de Proyectos (QPM)	Gestión de proyectos	3
Administración de Acuerdos con Proveedores (SAM)	Ingeniería	2
Administración de Requerimientos (REQM)	Gestión de proyectos	3
Administración de Riesgos (RSKM)	Soporte	2
Administración de la Configuración (CM)	Gestión de proyectos	3
Administración Integral de Proyecto + IPD (IPM+IPPD) 1	Gestión de proyectos	3

Innovación y Despliegue Organizacional (OID)	Gestión de procesos	5
Integración de Producto (PI)	Ingeniería	3
Medición y Análisis (MA)	Soporte	2
Monitoreo y Control de Proyecto (PMC)	Gestión de proyectos	2
Planificación de Proyecto (PP)	Gestión de proyectos	2
Procesos Orientados a la Organizacionales (OPF)	Gestión de procesos	3
Rendimiento de Procesos Organizacionales (OPP)	Gestión de procesos	4
Solución Técnica (TS)	Ingeniería	3
Validación (VAL)	Ingeniería	3
Verificación (VER)	Ingeniería	3

Tabla 1.4. Las áreas de proceso del modelo CMMI.

1.6.3. SPICE

Para que una organización mejore la calidad de sus productos debe tener un método probado, consistente y fiable para evaluar el estado de sus procesos y además, unos medios para usar los resultados de la evaluación como parte de un programa de mejora coherente. El proyecto internacional SPICE, llevado a cabo por la organización ISO, ha obtenido en su primera fase del proyecto un Informe Técnico Tipo 2 (ISO 15504) formado por un conjunto de documentos todos ellos bajo el título general de Evaluación del Proceso *Software*.

1.6.3.1. Necesidades y requerimientos para un estándar de evaluación de procesos de *software*

En Junio de 1991 el comité ISO/IEC JTC1/SC7 aprobó un estudio para que se investigara las necesidades y requerimientos para un estándar de evaluación de procesos *software*. Un año más tarde, se obtuvo como conclusión que existía un

consenso internacional para dicho estándar. En Junio de 1993 arrancó el proyecto SPICE con los objetivos de:

- Ayudar al proyecto de estandarización, en su etapa preparatoria, para desarrollar los borradores iniciales de trabajo.
- Realizar las pruebas de usuario, obteniendo datos de la experiencia que constituirán la base de la revisión del Estándar antes de emitirlo como *International Standard*.
- Crear el conocimiento del mercado y evolucionar el estándar.

El proyecto SPICE ha logrado ya el primer objetivo. Se ha obtenido el "*Technical Report Type 2*" que consta de las partes que muestra la **figura 1.5**.

El marco proporciona un enfoque estructurado de la evaluación de los procesos *software*, mediante el cual:

- Se anima a la auto-evaluación;
- Se aborda la idoneidad de la gestión de los procesos evaluados;
- Tiene en cuenta el contexto en el que operan los procesos evaluados;
- Produce un conjunto de valores del proceso (perfil del proceso) en vez de un resultado (válido/no válido).

Este marco es válido para todos los dominios de aplicaciones y tamaños de organización.

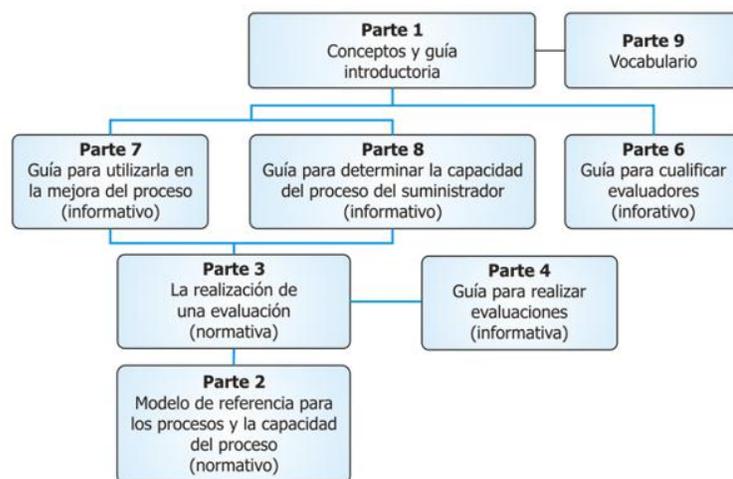


Figura 1.5: Informe Técnico: Componentes y las relaciones entre ellos.

El uso de la evaluación del proceso en una organización debería estimular principalmente a:

- Una cultura de mejora constante y al establecimiento de los mecanismos adecuados para soportar y mantener tal cultura;
- La ingeniería de procesos para cumplir los requisitos del negocio;
- La optimización de recursos.

Como resultado se obtendrán organizaciones con alta sensibilidad hacia el cliente y hacia los requisitos del mercado, minimizando los costes de sus productos y logrando una satisfacción del usuario final.

El Informe Técnico ha sido diseñado para satisfacer a tres usuarios diferentes: evaluadores, clientes y suministradores. Se muestra en la **tabla 1.5**.

USUARIOS	CARACTERÍSTICAS
Evaluadores	Un marco que define todos los aspectos para dirigir evaluaciones.
Clientes	<p>Un medio para determinar la capacidad actual y potencial de los procesos <i>software</i> del suministrador. Estos obtendrán los beneficios de:</p> <ul style="list-style-type: none"> - reducir incertidumbre para seleccionar suministradores <i>software</i>, al conocer los riesgos asociados con el contratista; - dotar de controles adecuados para contener el riesgo; - suministrar una métrica para elegir entre coste estimado del proyecto y capacidad de los suministradores que compiten.
Suministradores	<ul style="list-style-type: none"> - Un medio para determinar la capacidad actual y potencial de sus propios procesos <i>software</i>; - un instrumento para definir áreas y prioridades para la mejora del proceso <i>software</i>; - un marco que defina un mapa de caminos para la mejora del proceso <i>software</i>.

Tabla 1.5. Necesidades a satisfacer de los diferentes usuarios.

1.6.3.2. Beneficios de la estandarización de la evaluación de procesos

Los beneficios principales de un enfoque estandarizado para la evaluación del proceso son:

- Proporcionar un modelo para la mejora del proceso público y compartido;
- Conducir a un entendimiento común del uso de la evaluación del proceso para la mejora del proceso y la evaluación de la capacidad;
- Facilitar la evaluación de la capacidad en un concurso abierto;
- Realizar una revisión regular y controlada sobre la experiencia de la utilización;
- Será cambiado únicamente mediante el consenso internacional;
- Animar la armonización de los esquemas existentes.

1.6.3.3. Visión general

El marco para la evaluación de procesos *software* se puede utilizar por organizaciones implicadas en la planificación, gestión, monitorización, control y mejora de la adquisición, suministro, desarrollo, operación, evolución y soporte del *software*.

La evaluación del proceso examina los procesos utilizados por una organización para determinar si son efectivos para conseguir sus objetivos. Los resultados de la evaluación se pueden utilizar para conducir las actividades de mejora o para la determinación de la capacidad del proceso.

1.6.3.4. Elementos principales de SPICE

Los elementos esenciales para comprender SPICE son los siguientes: Los resultados de la evaluación del proceso se describe en un modelo de dos dimensiones: Dimensión del proceso y Dimensión de la capacidad. Esto es lo que se denomina arquitectura del modelo de referencia.

- **Dimensión del proceso:** que está caracterizado por los objetivos del proceso que constituye los elementos fundamentales a medir;

- **Dimensión de la capacidad del proceso:** que está caracterizado por una serie de atributos de proceso, aplicables a cualquier proceso, que representan características necesarias para gestionar y mejorar su capacidad de realización.

1.6.3.4.1. Dimensión del proceso

El modelo de referencia agrupa los procesos en categoría de procesos. Estos procesos se corresponden con los definidos en ISO 12207 *Software Life-Cycle Process*. Ver **tabla 1.6**. La descripción de cada proceso consta de:

- Una declaración del objetivo del proceso describiendo a un alto nivel los objetivos generales del proceso, y también una descripción en términos genéricos de los probables resultados de una implementación efectiva del proceso; y
- una o más observaciones proporcionando mayor información sobre los procesos y su relación a los procesos definidos en ISO 12207 y a otros procesos en este modelo de referencia.

1.6.3.5. Niveles de capacidad y atributos de proceso

La **figura 1.6** sintetiza la dimensión de la capacidad de proceso, indicando los atributos de proceso (PA) de cada nivel de capacidad. A continuación, se describe cada nivel. En la **tabla 1.7** se muestra los niveles de capacidad.

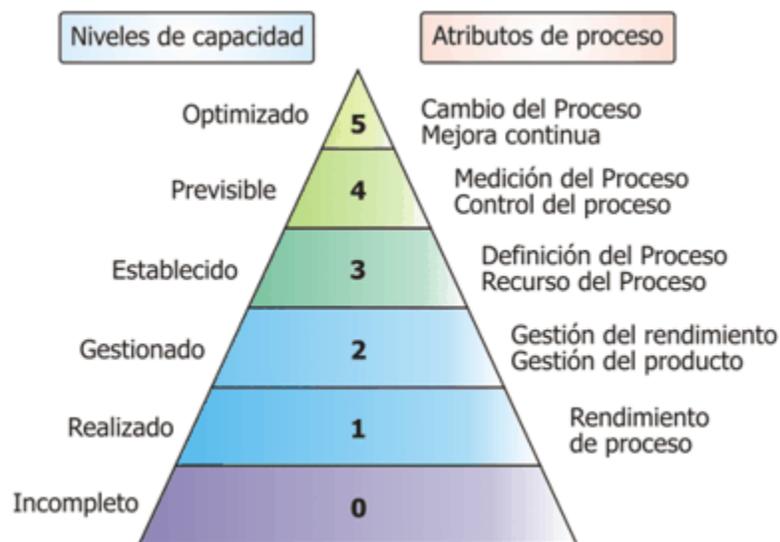


Figura 1.6: Dimensión de la Capacidad de Proceso.

NIVELES	CARACTERÍSTICAS
<p>Nivel 0. Proceso Incompleto</p>	<p>El proceso no está implementado o no logra conseguir su objetivo. No hay atributos en este nivel.</p>
<p>Nivel 1. Proceso Realizado</p>	<p>El propósito implementado logra su objetivo definido. P.A 1.1: Rendimiento del Proceso El proceso emplea un conjunto de prácticas, que son iniciadas por unos productos identificables y produce unos productos identificables, que satisfacen el propósito del proceso.</p>
<p>Nivel 2. Proceso Gestionado</p>	<p>El proceso <i>Realizado</i> entrega productos con una calidad aceptable en un margen de tiempo y necesidades de recursos definidos. PA 2.1: Gestión del Rendimiento La ejecución del proceso se gestiona para producir productos en un plazo de tiempo y con unos requisitos preestablecidos. PA 2.2: Gestión del Producto La ejecución del proceso se gestiona para producir productos que se documentan y se controlan satisfaciendo sus requisitos funcionales y no funcionales, de acuerdo con los objetivos de calidad del producto del proceso.</p>
<p>Nivel 3. Proceso Establecido</p>	<p>El proceso <i>Gestionado</i> se realiza utilizando un proceso definido basado en los principios de la ingeniería del <i>software</i>. PA 3.1: Definición del Proceso La ejecución del proceso utiliza una definición de proceso basada en un proceso estándar, que permite contribuir a los objetivos de negocio definidos en la organización. PA 3.2: Recursos del Proceso La ejecución del proceso utiliza eficazmente recursos humanos con las habilidades adecuadas y una infraestructura de proceso que contribuyen a los objetivos de negocio definidos de la organización.</p>
<p>Nivel 4. Proceso Previsible</p>	<p>El proceso <i>Establecido</i> se realiza constantemente dentro de los límites de control definidos para lograr sus objetivos. PA 4.1: Medición del Proceso La ejecución del proceso se soporta por los objetivos y mediciones que son utilizadas para asegurar que la implementación del proceso contribuye a la consecución de los objetivos. PA 4.2: Control del Proceso La ejecución del proceso se controla a través de la recopilación y análisis de mediciones para controlar y corregir, donde sea necesario, el rendimiento del proceso para lograr fiablemente los objetivos del proceso definidos.</p>

<p>Nivel 5. Proceso Optimizado</p>	<p>El proceso <i>Previsible</i> optimiza su rendimiento para satisfacer las necesidades de negocio actuales y futuras y logra repetidamente satisfacer sus objetivos de negocio definidos.</p> <p>PA 5.1: Cambio de Proceso Los cambios a la definición, gestión y rendimiento del proceso son controlados mejor para conseguir los objetivos de negocio de la organización.</p> <p>PA 5.2: Mejora Continua Los cambios a los procesos se identifican y se implementan para asegurar la mejora continua en el cumplimiento de los objetivos del negocio definidos de la organización.</p>
---	---

Tabla 1.7. Niveles de capacidad.

1.6.3.6. Atributos del proceso

Un atributo del proceso representa una característica medible de cualquier proceso. Los atributos de capacidad del proceso son los elementos básicos del esquema de evaluación.

Cada atributo se evalúa entre un rango de cuatro puntos:

- **N**= No conseguido. No hay evidencia de que se consigue el atributo definido.
 - **P**= Conseguido parcialmente. Se ha conseguido algo el atributo definido.
 - **L**= Bastante conseguido. Se ha conseguido significativamente el atributo definido.
 - **F**= Conseguido completamente. Se ha conseguido totalmente el atributo definido
- el nivel de capacidad se derivará de los valores de los atributos de los procesos.

1.6.3.7. Perfil del proceso

Una evaluación SPICE se realiza con el propósito de obtener un perfil de cada uno de los procesos (o instancias de proceso) dentro del alcance de la evaluación. Este perfil muestra la capacidad de la unidad organizativa para lograr el objetivo del proceso.

La evaluación examina a un número de instancias de proceso con el fin de obtener los datos necesarios para producir un perfil del proceso. Una instancia de proceso es una implementación particular de un proceso. Por ejemplo, para cada vez que

se realiza la prueba de un módulo del sistema, habrá una instancia de Realizar Prueba de Unidad. Las instancias de proceso examinadas durante la evaluación tiene que ser cuidadosamente seleccionadas para asegurar que la evaluación alcanzará su propósito y cubrirá su alcance.

Se evalúa cada instancia de proceso examinando sus atributos, y obteniendo como consecuencia un valor. Estos valores son decididos mediante el análisis de los indicadores asociados y juzgando su existencia. Las decisiones sobre la existencia de indicadores están basados en una objetiva evidencia, la cual es registrada para soportar y justificar los resultados de la evaluación.

El resultado básico de la evaluación es un conjunto de valores de los atributos de cada instancia de proceso. Éstos se pueden combinar para producir un nivel de capacidad para la instancia del proceso. Los valores para las distintas instancias del mismo proceso se pueden combinar para producir un perfil del proceso como unidad.

Se realizan dos tipos de juicio durante la evaluación de una instancia de un proceso. El primero es para ver si el proceso se realizó, para ello se examina si las prácticas y los tipos de productos esperados existen. Este juicio es la base del valor del atributo de Rendimiento del Proceso, examinando si la instancia está al nivel No realizado, o a uno de los niveles de mayor de capacidad.

El segundo juicio determina cómo de bien se gestiona ese proceso. Se examina la existencia de diferentes prácticas y productos con objeto de realizar los juicios sobre los diferentes atributos de capacidad. Se registran, para soportar los valores de cada atributo, la evidencia sobre la existencia de las prácticas y de sus características y productos.

Los valores de los atributos se sitúan en una escala de cuatro niveles de consecución: no (N), parcialmente (P), en su mayoría (L), completamente (F). La instancia del proceso se da en uno de estos valores según el logro de cada uno de los nueve atributos. Este conjunto formado por los nueve valores de atributos es el perfil de la instancia del proceso. Ver **tabla 1.8** que muestra los perfiles de la instancia del proceso.

NIVEL DE ATRIBUTO	ATRIBUTO	VALOR DE LA INSTANCIA A	VALOR DE LA INSTANCIA B
5	5.2	N	N
	5.1	N	N
4	4.2	P	N
	4.1	P	N
3	3.2	L	P
	3.1	L	P
2	2.2	L	L
	2.1	L	L
1	1.1	L	F
Nivel de Capacidad =		1	2

Tabla 1.8. Perfil de la Instancia del Proceso.

1.6.3.8. Ventajas y desventajas

SPICE ofrece una base para una evaluación muy detallada del estado actual del proceso de una organización. Por su gran nivel de descomposición de los procesos e indicadores, proporciona evaluaciones objetivas y con resultados repetibles, especialmente cuando es realizada por evaluadores entrenados y cualificados. El *European Software Institute* (ESI) ya ofrece cursos para ello.

Al disminuir la subjetividad se consigue reducir discordias sobre los resultados de la evaluación y a adoptar actitudes positivas de los equipos hacia la evaluación.

Por contra se requiere un gran esfuerzo para realizar las evaluaciones y por tanto un alto coste. Las evaluaciones se pueden llevar a cabo por personal interno de tal manera que se puedan ver reducidos estos costes. Es importante tener en cuenta que la evaluación no necesita abordarse a toda la organización, las evaluaciones SPICE se puede realizar únicamente en aquellos procesos que sean áreas de problema.

El modelo de referencia SPICE no contiene una estrategia de mejora del proceso. Esto puede verse como positivo o negativo dependiendo de lo que se quiera.

Pero la ventaja principal es que al disponer de un estándar internacional se pueden realizar comparaciones a nivel mundial entre evaluaciones en contextos similares.¹

¹ DE AMESCUA, Antonio; LLORENS, Juan; GARCÍA, Angel. *Knowledge Reuse Group*. (2009). *SPICE: Un marco para la evaluación de procesos de software*. [en línea]. Enlace web: www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc. [Leído: 6 de enero 2010, 15.00 h GMT+1].

1.6.4. Trillium *Model*

El modelo Trillium es un producto usado por Bell Canada para dar valor al desarrollo de un producto y apoyar las capacidades de proveedores de telecomunicaciones o productos basados en tecnologías de la información existentes o futuros (Trillium, 2000). El modelo ha sido diseñado para ser aplicado a sistemas de *software* 'empotrados' tales como sistemas de telecomunicaciones, no obstante buena parte del modelo puede ser aplicado a otros segmentos de la industria del *software* como sería el área de *Management Information Systems*.

1.6.4.1. Características del Modelo Trillium

Con respecto al CMM, el modelo Trillium, se distingue en que:¹

- define *roadmaps* en lugar de áreas clave; tiene una perspectiva orientada al producto, haciéndolo más general y de amplio uso;
- da amplia cobertura a los aspectos que inciden o impactan en la capacidad del proceso; y,
- se enfoca al cliente en lugar del desarrollo mismo.

Esto consigue que se definan prácticas que guían al proyectista sobre cómo conseguir lo que desea un usuario/cliente donde, en lugar de señalar determinadas metas que se deben alcanzar con ciertas prácticas de diseño, se buscan aquellas prácticas que habiliten la consecución de lo que desea el usuario.

¹ ESTAY-NICULCAR, Christian A. Dr. (2009). *Fundamentos de Gestión de Proyectos: De la Teoría de Proyectos a la Gestión de Proyectos según el PMBOK. Gestión de Proyectos Informáticos y Cambio*. [en línea]. Enlace web: http://www.inf.utfsm.cl/~lhevía/asignaturas/proy_ti/topicos/PMBOK-Apunte.doc. [Leído: 28 de febrero 2010, 15.00 h GMT+1].

1.6.4.2. Niveles de madurez

A semejanza del modelo CMM, el modelo Trillium presenta una escala de cinco niveles de madurez (Trillium, 2000):

- *Nivel 1*. Desestructurado. En este nivel el proceso de desarrollo es ad-hoc. Los proyectos frecuentemente no pueden satisfacer objetivos de calidad o de programación. El éxito posible se basa más en el trabajo de los individuos que en la propia estructura e infraestructura organizacional.
- *Nivel 2*. Repetible y orientado al proyecto. El éxito individual del proyecto se consigue a través de una férrea planificación y control de gestión del proyecto, dando especial énfasis a los requerimientos de gestión, técnicas de estimación y configuración del cambio.
- *Nivel 3*. Definido y orientado al proceso. Aquí los procesos son definidos y utilizados al nivel organizacional, no obstante se acepta que el proyecto sea adaptado a las circunstancias. Los procesos son controlados y mejorados. Se incorporan requerimientos ISO 9001 como procesos de entrenamiento y auditoría interna.
- *Nivel 4*. Gestionado e integrado. La monitorización y análisis del proceso es usado como mecanismo clave de mejora. Procesos de gestión del cambio y programas de prevención de defectos son integrados. Las herramientas CASE se integran dentro del proceso.
- *Nivel 5*. Completamente integrado. Metodologías formales son extensivamente usadas. Repositorios organizacionales son usados para soportar y mantener la historia del proceso de desarrollo.

1.6.4.3. Arquitectura Trillium Model

En la **figura 1.7¹** se muestra la arquitectura de Trillium, igualmente plantea una descomposición pero con una diferencia sustancial con el CMM: CMM: no se comienza la descomposición desde los niveles de madurez, sino desde ocho áreas de capacidad (*Capability Areas*), cada una de las cuales contiene varias *roadmaps* y estos últimos a su vez contienen prácticas (*Practices*), usados paulatinamente por niveles de madurez.

De esta forma, la arquitectura de Trillium se caracteriza por poseer (Trillium, 2000):

- *Capability Areas* (CA), que son áreas centrales de preocupación del modelo Trillium y que encuentran contenidas por prácticas;
- *Roadmaps* (RM), que son un conjunto de prácticas relacionadas, enfocadas sobre un área o necesidad organizacional, o un elemento específico, dentro del proceso de desarrollo del producto; y,
- *Practices*, que son las acciones a desarrollar para conseguir una mejor capacidad del proceso, cada una de las cuales se vincula a un nivel de madurez.



Figura 1.7: Arquitectura Modelo Trillium.

¹ Trillium, *Model Description*. [en línea]. (2003).

Enlace web: <http://www.sqi.gu.edu.au/trillium/t3modc3.html>. [Leído: 25 de febrero 2010, 16.00 h GMT+1]

Realizado por
Lic. Rafael Gonzalez Freites
rafaelfreites@hotmail.com
azua de Compostela
rep. dominicana