

## **PROCESOS DE SOFTWARE**

En un mundo de cambios constantes y competencia global, las organizaciones de desarrollo de software son presionadas a alcanzar mayor eficiencia con menores costos. Para poder lograr este objetivo, es necesario adoptar una forma de trabajo que permita entender, controlar, comunicar, mejorar, predecir y certificar el trabajo realizado.

Actualmente existe una gran diversidad de opciones relacionadas con procesos de desarrollo. Constantemente se escuchan diferentes acrónimos como CMM, CMMI, RUP, ISO, PSP, TSP, etc., que causan confusión, principalmente debido a la mala interpretación de los mismos.

Revisemos entonces los principales procesos de desarrollo y modelos más utilizados al momento, así como los estándares relacionados.

### **¿Por qué contar con un proceso de software?**

Hasta hace poco tiempo, la producción de software era realizada con un enfoque artístico, a diferencia de un enfoque industrial. Ante la constante presencia de proyectos fallidos, y con el objetivo de mejorar la calidad de los productos, en los últimos años las organizaciones introdujeron los métodos de ingeniería de software (Ver Fundamentos – Desarrollar software es mucho más que programar, pág. 42).

A partir de estos, se formalizó el enfoque de ingeniería de producto para desarrollar software. Factores como la globalización han obligado a las organizaciones a contar con marcos de trabajo que las ayuden hacer las cosas de la manera más eficiente. Fue entonces que se incorporó la ingeniería de procesos al desarrollo de software.

### **Proceso**

Antes de definir lo que es un proceso de desarrollo de software, entendamos lo que es un proceso. Una definición sencilla de proceso es "serie de acciones que conducen a un final". Esta definición parece coincidir con las ideas generales de la gente sobre procesos, pero deja muchas preguntas abiertas. ¿El proceso es la forma en que la organización opera —desde mercadotecnia hasta recursos humanos— o es la forma en que un desarrollador diseña, produce código, o prueba el software? ¿El proceso se refiere a administración, ingeniería, o ambas? ¿El proceso implica demasiada documentación y nos abstiene de desarrollar el producto objetivo?

La respuesta a éstas puede variar dependiendo de la perspectiva. Sin embargo, siempre que para alcanzar algún fin deseado necesitemos ejecutar una serie de acciones, y estas acciones tengan cierto orden, dependencias, roles responsables, resultados, tiempos de ejecución y herramientas de apoyo, estaremos hablando de procesos, que pueden ser predefinidos y personalizados.

### **Proceso de software**

La meta de la ingeniería de software es construir productos de software, o mejorar los existentes; en ingeniería de procesos, la meta es desarrollar o mejorar procesos.

Un proceso de desarrollo de software es un conjunto de personas, estructuras de organización, reglas, políticas, actividades y sus procedimientos, componentes de software, metodologías, y herramientas utilizadas o creadas específicamente para definir, desarrollar, ofrecer un servicio, innovar y extender un producto de software.

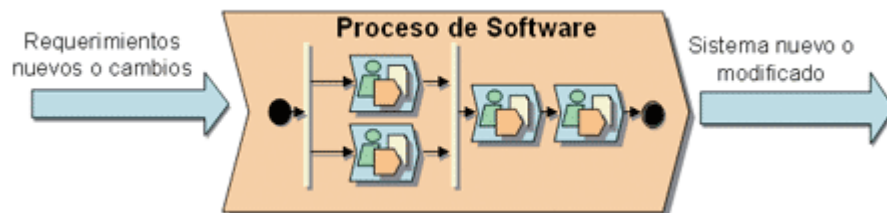
Un proceso de software efectivo habilita a la organización a incrementar su productividad al desarrollar software:

- Permite estandarizar esfuerzos, promover reuso, repetición y consistencia entre proyectos.
- Provee la oportunidad de introducir mejores prácticas de la industria.
- Permite entender que las herramientas deben ser utilizadas para soportar un proceso.
- Establece la base para una mayor consistencia y mejoras futuras.

Un proceso de software mejora los esfuerzos de mantenimiento y soporte:

- Define cómo manejar los cambios y liberaciones a sistemas de software existentes.
- Define cómo lograr la transición del software a la operación, y cómo ejecutar los esfuerzos de operación y soporte.

Necesitamos un proceso de software cuya funcionalidad esté probada en la práctica, y personalizado para que cumpla con nuestras necesidades específicas.



### Elementos Típicos del Proceso de Software

Actividad	Definen las acciones que se llevan a cabo en un momento dado del desarrollo de software.
Flujo de Trabajo	Colección estructurada de actividades y elementos asociados (artefactos y roles), que producen un resultado de valor.
Rol	Son responsables por llevar a cabo las actividades del proceso, pueden ser personas o herramientas.
Producto o Artefacto	Son las entradas y salidas de las actividades, pueden ser de diferentes tipos, como documentos, modelos, componentes, planes, reportes, etc.
Disciplina	Conjunto integrado por actividades relativas a una rama particular de conocimiento. Ej. Análisis y diseño.

### Diversidad en Modelos

Actualmente existe una gran variedad de modelos para procesos de software. Podemos entenderlos más fácilmente si los clasificamos en dos tipos: genéricos y específicos.

Modelos Genéricos	Modelos Específicos
Abarcan todos los procesos relacionados con el desarrollo de software	Enfocados a la ingeniería de productos de software
CMM – modelo de madurez de capacidades – estándar de facto	UP – proceso de desarrollo
CMMI – modelo integrado	RUP – proceso de desarrollo
ISO 9001-2000 – sistema para administración de la calidad - estándar	PSP – enfocado en individuos
ISO/IEC 15504 – marco para evaluación de procesos de software – en vías de ser estándar, por ahora reporte técnico	TSP – enfocado en equipos (incluye PSP)
MoProSoft – modelo de procesos para la industria de software en México – en vías de ser norma mexicana	
<ul style="list-style-type: none"> <li>• Nos dicen qué debemos hacer</li> <li>• Se deben usar como referencia para definir procesos en una organización y para autoevaluación.</li> <li>• Medio para evaluar que tan bien o mal esta la organización.</li> </ul>	<ul style="list-style-type: none"> <li>• Nos dicen el cómo debemos hacer las cosas</li> <li>• Se usan como guía para ejecutar proyectos</li> </ul>

Revisemos estos modelos para entender mejor su objetivo y estructura. Primero conoceremos modelos genéricos como CMM e ISO, y posteriormente estudiaremos modelos específicos para software.

### **CMM (Capability Maturity Model) - Modelo de Madurez de Capacidades**

Marco que describe elementos clave de procesos efectivos de software. Creado por el Software Engineering Institute (SEI) en conjunto con Carnegie Mellon University. La primera versión se publicó en 1994.

CMM describe un camino evolutivo en 5 niveles de mejora de procesos para lograr su madurez. Cubre prácticas de planeación, ingeniería y administración del desarrollo y mantenimiento de software.

Niveles de Madurez	Áreas clave del proceso
<b>Nivel</b>  <b>1 – Inicial</b> Proceso caótico, impredecible. El éxito depende del esfuerzo heroico de individuos.	Ninguna
<b>2 – Repetible</b> Institucionalizar procesos efectivos de administración de proyectos de software, que permiten a las organizaciones repetir prácticas exitosas desarrolladas en proyectos previos.	<ul style="list-style-type: none"> <li>• Administración de Requerimientos.</li> <li>• Planeación de Proyecto de Software.</li> <li>• Seguimiento y Control del Proyecto de Software.</li> <li>• Administración de Subcontratos de Software.</li> <li>• Aseguramiento de Calidad de Software.</li> <li>• Administración de Configuración de Software.</li> </ul>
<b>3 – Definido</b> El proceso estándar para desarrollar y mantener software en la organización esta documentado, incluyendo procesos de administración e ingeniería de software, y estos procesos están integrados.	<ul style="list-style-type: none"> <li>• Enfoque en Procesos de la Organización.</li> <li>• Definición de Procesos de la Organización.</li> <li>• Programa de Capacitación.</li> <li>• Administración Integral de Software.</li> <li>• Ingeniería de Productos de Software.</li> <li>• Coordinación Intergrupala.</li> <li>• Revisiones entre Colegas.</li> </ul>
<b>4 – Administrado</b> Se establece un conjunto de metas cuantitativas para medir el nivel de calidad y desempeño de los proyectos y del proceso organizacional.	<ul style="list-style-type: none"> <li>• Administración cuantitativa de procesos.</li> <li>• Administración de la calidad del producto de software.</li> </ul>
<b>5 - Optimizado</b> No es simplemente detectar y resolver defectos, sino prevenirlos y evitarlos al implementar actividades proactivas. Mejora continua de procesos.	<ul style="list-style-type: none"> <li>• Prevención de defectos.</li> <li>• Administración de cambio de tecnología.</li> <li>• Administración de cambio de procesos.</li> </ul>

## ISO 9001-2000

ISO (International Standards Organization) en 1987 crea la norma ISO 9000, conjunto de estándares que establecen los requerimientos para la gestión de los sistemas de calidad. ISO 9000:2000 está formado por:

- ISO 9000 Fundamentos y Vocabulario.
- ISO 9001 Requisitos.
- ISO 9004 Recomendaciones.

La parte de Requisitos - ISO 9001:2000, está estructurado en 8 secciones:

1. Alcance.
2. Normas para la Consulta.
3. Términos y Definiciones.
4. Sistema de Gestión de la Calidad.
5. Responsabilidad de la Dirección.
6. Gestión de los Recursos.
7. Realización del Producto.
8. Medida, Análisis y Mejora.

Aunque ISO 9001:2000 no otorga un estándar específico para sistemas de desarrollo de software, es decir, no abarca todos los procesos relacionados con el desarrollo de software,

muchas organizaciones de software han optado por gestionar su sistema de calidad en base a este estándar, y obtener una certificación reconocida de manera internacional.

### **CMMI (Capability Maturity Model Integration) - Modelo de Madurez de Capacidades Integrado**

El proyecto de CMMI fue concebido como una iniciativa para reunir los diferentes CMMs en un conjunto de modelos integrados, más consistentes entre ellos. Los modelos fuente que sirvieron como bases incluyen: CMM Software, CMM Ingeniería de Sistemas, y CMM Desarrollo Integrado de Producto.

CMMI proporciona una guía para desarrollar procesos, que además ayuda a evaluar la madurez de la organización o capacidad de un área de procesos. CMMI incluye los procesos de ingeniería de software e ingeniería de sistemas.

El modelo está representado de forma continua y escalonada. Contiene 22 áreas de procesos. Cada área de proceso está formada por: Objetivos específicos, Prácticas específicas, Objetivos genéricos, y Prácticas genéricas.

### **CMMI Modelo Continuo**

Está formado por 5 niveles de capacidad del proceso:

0. Incompleto
1. Desempeñado
2. Administrado
3. Definido
4. Administrado cuantitativamente
5. Optimizado

Y por cuatro categorías de áreas de procesos: Administración de Procesos, Administración de Proyectos, Ingeniería, Soporte. Estas categorías agrupan a las diferentes áreas de proceso, dividiéndolos en procesos básicos y avanzados.

### **CMMI Modelo Escalonado**

El modelo escalonado, al igual que CMM, describe un camino evolutivo en 5 niveles de madurez de procesos, además integra nuevas áreas de proceso.

La selección entre el modelo escalonado y el continuo depende del objetivo de la organización, además de tener que considerar la situación (si ya se tiene CMM, cultura en procesos, etc.). Por ejemplo, si el objetivo es llevar a la organización a cierto nivel de capacidad, deberá seleccionarse la forma escalonada; en cambio si el objetivo es mejorar cierto proceso, será mejor seguir la forma continua.

### **Algunos Beneficios de CMMI vs. CMM**

Algunos de los beneficios que han experimentado las organizaciones que pasan de CMM a CMMI son los siguientes:

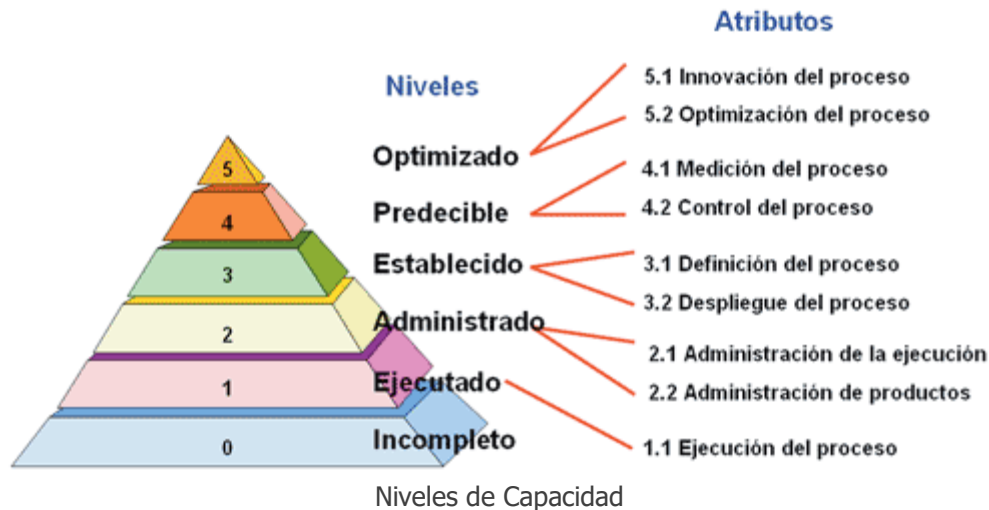
- Mejor alineación a objetivos de negocio.
- Mejor visibilidad hacia las actividades de ingeniería, con el objetivo de asegurar que el producto o servicio cumple las expectativas del cliente.
- Aprovechamiento de mejores prácticas adicionales (e.j., medición, riesgo, administración, y manejo de proveedores).
- Acoplamiento más estrecho con estándares de ISO.

### **ISO/IEC 15504**

Estándar internacional que ofrece un marco para la evaluación de procesos. Fue iniciado en 1991 como el proyecto SPICE (Software Process Improvement and Capability dEtermination). La versión de Reporte Técnico fue aceptada y publicada en 1998, enfocada únicamente en procesos de software.

En el transcurso de su desarrollo ha evolucionado, de ser un modelo de referencia de buenas prácticas de software, para convertirse en un marco de trabajo para evaluación de múltiples modelos (de software o no). Su dirección actual es poder aplicarse a múltiples disciplinas y permitir a las diferentes comunidades definir sus propios procesos específicos, modelos de referencia o buenas prácticas. ISO 15504 está en vías de convertirse en estándar internacional, y se espera que esté completo para 2006. Esta versión está compuesta de cinco documentos, a diferencia del reporte técnico que consta de nueve partes (Ver gráfica 1 - Estructura de documentos).

La parte 2 de este estándar es de especial interés, ya que es la que define como se realiza una evaluación. Establece requisitos tanto para modelos de procesos de referencia como para los métodos de evaluación sin establecer alguno en particular.



- 0. Incompleto.- Falta de cumplimiento del proceso.
- 1. Realizado.- Genera los productos de trabajo esperados.
- 2. Administrado.- Proceso y productos administrados y controlados.
- 3. Establecido.- Proceso definido para la organización y utilizado adecuadamente.
- 4. Predecible.- El proceso opera dentro de los límites estadísticos establecidos.
- 5. Optimizado.- El proceso mejora continuamente.

Las organizaciones de software no tienen que seleccionar entre 15504 y su modelo actual. El rol de 15504 es proveer un marco de trabajo en el que los modelos y métodos de evaluación puedan existir de una manera armónica. No está diseñado para ser utilizado solo. La selección radica en decidir si el ajustarse a 15504 es importante para el negocio (¿El negocio compete en el mercado global?, ¿Provee software a algún cliente que requiera 15504?, ¿Existen varios modelos de evaluación en la organización?). De ser así, deberán elegir modelos que se ajusten a 15504.

## CMMI Modelo Escalonado

### Nivel de Madurez

**1 – Inicial**  
Madurez de un proceso caracterizada por resultados impredecibles.

**2 – Administrado**  
Madurez del proceso caracterizada por el desempeño repetible del proyecto. El foco clave del proceso esta en actividades y prácticas a nivel proyecto.

**3 – Definido**  
Madurez del proceso caracterizada por mejorar el desempeño del proceso dentro de una organización.

**4 – Administrado Cuantitativamente**  
Caracterizada por mejorar el desempeño organizacional.

**5 - Optimizado**  
Madurez del proceso caracterizada por un desempeño organizacional rápido y configurable, Mejora continua y cuantitativa de procesos.

### Áreas de proceso

Ninguna

- Administración de Requerimientos.
- Planeación de Proyectos.
- Monitoreo y Control de Proyectos.
- Administración del Acuerdo con el Proveedor.
- Administración de Configuración.
- Aseguramiento de Calidad de Proceso y Prod.
- Mediciones y Análisis.

- Desarrollo de Requerimientos.
- Solución Técnica.
- Integración de Producto.
- Verificación, Validación.
- Enfoque Organizacional en Proceso.
- Definición de Procesos de la Organización.
- Capacitación Organizacional.
- Administración Integral de Proyectos.
- Administración del Riesgo.
- Análisis de Decisión y Resolución.

- Desempeño de Procesos Organizacionales.
- Administración Cuantitativa de Proyectos.

- Innovación y Despliegue Organizacional.
- Análisis Causal y Resolución.

## Compatibilidad entre Modelos

### ISO/IEC 15504

- Evaluación de procesos de software.
- En vías de ser estándar.
- Dirección amplia.
- Niveles de capacidad.
- Evaluación de capacidades por proceso individual.
- Guía para realizar la evaluación.
- Toma como referencia ISO 9001 y CMMI.

### CMMI

- Modelo de madurez de procesos de software.
- Modelo – estándar de facto.
- Dirección detallada.
- Pasos progresivos (niveles) - Escalonada.
- Categorías de procesos - Continua.
- Guía para institucionalización e implementación.
- Modelo de evaluación será conforme al modelo de evaluación de 15504.

### ISO 9001-2000

- Sistema de Gestión de la Calidad.
- Estándar internacional.
- Dirección amplia.

- Un conjunto de requerimientos a ser cubierto.
- No hay lineamientos para su implementación.
- Usado como referencia en actividades de gestión de calidad por CMMI y 15504.

Con eso concluimos nuestra revisión de modelos genéricos. A continuación veamos los modelos específicos para software.

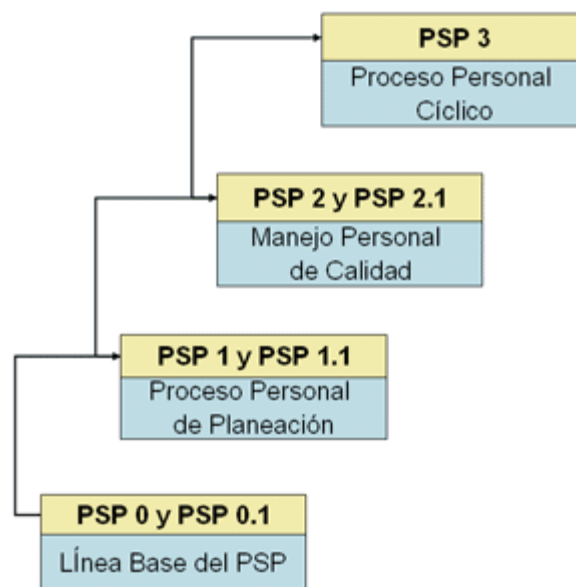
### **PSP – Personal Software ProcessSM**

Personal Software Process (PSP) es un proceso diseñado para ayudar a los ingenieros de software a controlar, manejar y mejorar su trabajo. PSP está basado en una motivación: La calidad de software depende del trabajo de cada uno de los ingenieros de software. Debido a que los costos de personal constituyen 70% del costo del desarrollo de software, las capacidades y hábitos de trabajo de los ingenieros determinan en gran manera los resultados del desarrollo de software.

Basado en prácticas encontradas en CMM, el PSP puede ser usado por ingenieros para estructurar y disciplinar el desarrollo de software. El ingeniero de software podrá planear mejor el trabajo, conocer con precisión el desempeño, medir la calidad de productos, y mejorar las técnicas.

PSP puede ser aplicado en:

- Desarrollo de programas.
- Definición de requerimientos.
- Documentación.
- Pruebas de sistemas.
- Mantenimiento de sistemas.



### **TSP - Team Software Process**

Team Software Process (TSP) es un marco para el desarrollo de software que pone igual énfasis en el proceso, producto y trabajo en equipo. Al igual que PSP, TSP fue propuesto por Watts Humphrey.

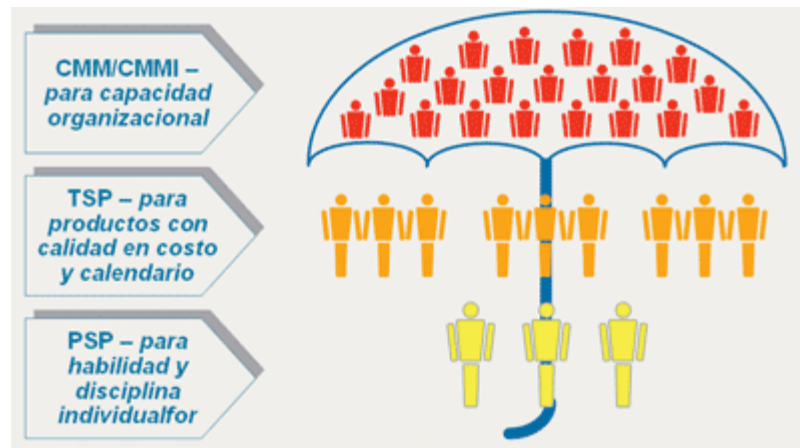
TSP se basa en PSP, y se fundamenta en que el software, en su mayoría, es desarrollado por equipos, por lo que los ingenieros de software deben primero saber controlar su trabajo, y después saber trabajar en equipo. TSP le enseña a los ingenieros a construir equipos



autodirigidos y desempeñarse como un miembro efectivo del equipo. También muestra a los administradores como guiar y soportar estos equipos.

Estrategia de TSP

- Proveer un proceso sencillo basado en PSP.
- Desarrollar productos en varios ciclos. Ciclo de TSP: Lanzamiento, Estrategia, Plan, Requerimientos, Diseño, Implementación, Pruebas, Postmortem.
- Establecer medidas estándares para calidad y desempeño.
- Proveer definiciones de roles, y evaluaciones de rol y de equipo.
- Requiere disciplina de proceso.
- Provee guía para manejo de problemas de trabajo en equipo.



## RUP

El Rational Unified Process (RUP) es un proceso de software desarrollado y comercializado por Rational Software (ahora parte de IBM). RUP está diseñado alrededor de seis mejores prácticas para el desarrollo de software:

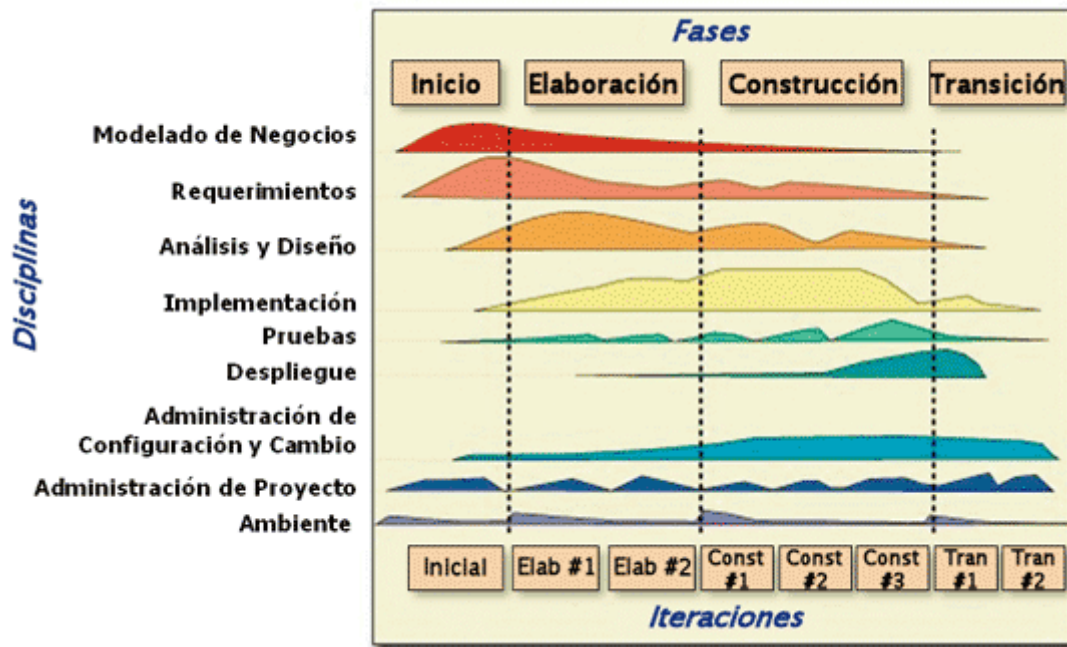
- Desarrollar de manera iterativa.
- Administrar los requerimientos.
- Utilizar arquitecturas basadas en componentes.
- Modelar el software visualmente.
- Verificar la calidad de manera continua.
- Controlar los cambios.

En sí, RUP es una guía que define roles, actividades, flujos de trabajo y lineamientos para ejecutar proyectos de software de acuerdo a estas mejores prácticas.

RUP organiza los proyectos de software en dos dimensiones: la del tiempo y la de las actividades. En base al tiempo, los proyectos se dividen en cuatro fases secuenciales:

1. Concepción – Definición de alcance, identificación de riesgos.
2. Elaboración – Resolución de riesgos, establecimiento de arquitectura.
3. Construcción – Generación del producto.
4. Transición – Disponibilidad a la comunidad de usuarios finales.

Las actividades se organizan en nueve diferentes disciplinas que son ejecutadas durante las diferentes fases.



En realidad RUP es un framework (marco de trabajo) que pretende ser personalizado o configurado para organizaciones y proyectos específicos. RUP no se puede aplicar de la misma forma en todos los proyectos de una organización. Es por esto que pretender seguir RUP a través de ir cumpliendo con la lista de artefactos que define, es una estrategia poco efectiva. Lo que las organizaciones deben hacer es entender la razón de ser de RUP – las prácticas citadas anteriormente – y en base a esto aplicar lo que decidan que es conveniente para cada área o proyecto específico.

RUP es una instancia particular del Proceso Unificado, definido por Ivar Jacobson, Grady Booch y James Rumbaugh en el libro "The Unified Software Development Process" de 1998. Adicionalmente existen otras instancias de este proceso, tales como el Proceso Unificado Mejorado (Enhanced Unified Process), el cual agrega soporte multiproyectos y fases y disciplinas para el mantenimiento y retiro de sistemas de software.

#### Referencias:

- Proceso de Desarrollo de Software, Hanna Oktaba, Facultad de Ciencias – UNAM
- ¿Por qué usar MoProSoft como modelo de procesos de referencia?, Hanna Oktaba, [www.amcis.org.mx](http://www.amcis.org.mx)
- Software Engineering Institute – Carnegie Mellon University, [www.sei.cmu.edu](http://www.sei.cmu.edu)
- Diplomado en Calidad de Software, AMCIS 2002 – [www.amcis.org.mx](http://www.amcis.org.mx)
- Process Diversity in Software Development – IEEE Software, julio-agosto 2000