



## Extracto del libro en formato digital “calidad tradicional y de software”

Msc. Alejandro Bedini G.  
abedini@ind.utfsm.cl

### 1. Calidad de software

Sin lugar a dudas la década del 90 se caracterizó por la palabra **CALIDAD**, normalmente... sin importar a qué elementos, entes y áreas se hace referencia; gerencial, objetos de consumo, servicios, administración, gestión, atención al público, universidades, hospitales, líneas aéreas, gobiernos, bancos (la lista continua). Las ciencias no escapan de este “fenómeno”: medicina, economía, ingeniería, biología, sociología, etc. En definitiva todos dicen poseerla y dominarla, pero cómo conocer y dominar un concepto tan amplio, subjetivo y muchas veces ambiguo, que ni siquiera la mayoría de las personas podrían dar una definición exacta de ¿qué es calidad?

La Ingeniería de Software no escapa de esta realidad pero sin lugar a dudas se enfrenta con muchos más obstáculos para poder dominar la calidad en comparación con otras ciencias. Para numerar solo algunos:

- **No existe una definición estándar y universal de qué es calidad.**  
En realidad algunos organismos e instituciones como ISO, IEEE, SEI, etc. brindan definiciones aceptables pero no son homogéneas, dando como resultado que cada profesional utilice su propia versión de calidad.
- **La calidad debe satisfacer a una amplia gama de entes relacionados pero no mutuamente excluyentes:**  
Clientes, procesos, organismos, productos (documentos, aplicaciones, mediciones, ...)
- **Crear cultura de calidad.**  
Esto implica un compromiso constante, tedioso, costoso, y a largo plazo por parte de la organización y las personas que lo componen.

Por tal motivo es necesario hacer una pausa y ver en más detalle los puntos que encierra este concepto y qué alternativas existen para alcanzarlo.



## ¿ Qué es calidad ?

El término calidad es ambigualmente definido y pocas veces comprendido, esto se debe porque:

- La calidad no es una sola idea, es un concepto multidimensional;
- La dimensión de calidad incluye el interés de la entidad, el punto de vista de la entidad, y los atributos de la entidad;
- Por cada concepto existen diferentes niveles de abstracción;
- Varía para cada persona en particular.

Podemos clasificar a la calidad bajo dos puntos de vista, **usual y profesional**.

### **Punto de vista usual**

La calidad contiene características intangibles, términos como alta, baja, y buena calidad son utilizados sin intentar definirlos.

### **Punto de vista profesional**

Juran(1970) definió la calidad como “Adaptabilidad de uso”, esto implica dos parámetros: calidad de diseño y calidad de conformidad. Es decir, adaptable a la necesidad de los usuarios.

Crosby(1979) definió la calidad como la “conformidad con los “requerimientos.”

Como ocurre con el concepto general de calidad no existe una única definición de **calidad del software**, según la IEEE:

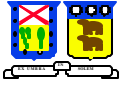
- La calidad debe ser **medurable**;
- La calidad debe ser **predecible**.

*Los factores deben ser:*

- **Ausencia** de defectos;
- **Satisfacción** del usuario;
- **Conformidad** con los requerimientos.

Sin lugar a dudas el factor inherente sobre calidad de software es la ausencia de defectos, este factor usualmente se expresa de dos maneras: **tasa de defecto** (número de defectos por KLOC, Puntos de Función, otra unidad) y **confiabilidad** (número de fallas por  $n$  horas de operación, tiempo medio entre fallas, otra probabilidad de libre de fallas por unidad de tiempo).

La satisfacción del usuario usualmente es medida por porcentaje de **satisfacción** o **insatisfacción**. Para evitar prejuicios se utiliza las técnicas de estudio o encuesta ciega (el entrevistador no sabe quién es el cliente, y el cliente no sabe a qué empresa representa el entrevistador).



## Características de la calidad del software

Según la norma **ISO 9126** las características son **portabilidad, eficiencia, confiabilidad, usabilidad, funcionalidad y mantenibilidad.**

Algunas empresas definen sus propios factores/atributos de calidad de software, por ejemplo en el caso de **IBM** se enfoca hacia:

Capacidad(funcionalidad), *usabilidad*, performance, confiabilidad, instalación, *mantenibilidad*, documentación/información, servicio, y “totalidad”.

En el caso de **Hewlett-Packard** se enfoca hacia:

Funcionalidad, *usabilidad*, confiabilidad, performance, servicio.

## 2. Marcos de trabajo

Los **MT** corresponden a estructuras escritas de una idea y/o conjunto de metas para facilitar a una organización la aplicación de las mismas. Es decir, mediante los MT se permite que todo el personal de una organización *se dirija* en la misma “**dirección**”.

En el área de Ingeniería de Software se puede catalogar a los MT por el propósito que cumplen, como se muestra en la Tabla, más de una categoría puede corresponder a un MT.

1. Estándares y Guías
2. Modelos de mejoramiento de procesos y métodos internos de evaluación
3. Pautas de selección de 3° empresas o contratistas
4. Premios de Calidad
5. Modelos de ciclo de vida de software
6. Modelos de ingeniería de sistemas

### Categorías de los Marcos de Trabajo.

La finalidad de los marcos de trabajo es la de **mejorar los procesos de software, brindar pautas para efectuar evaluaciones de la unidad informática, determinar la potencialidad y la performance de sus procesos, y la madurez de la organización.** En algunos MT estos ítemes se encuentran más explícitos y por consiguiente ocupan un área central en el mismo.

En definitiva los MT buscan:

- **mejorar** los **procesos** de software,
- **aumentar** la **productividad** y la **calidad**,
- y **disminuir** el **costo**.

Es conveniente definir los elementos que manejan los MT vistos en el párrafo anterior, dado la diversidad de “versiones” de muchos de los términos involucrados y con el fin de evitar ambigüedades.



**Proceso de software:** Es un conjunto de actividades, métodos, prácticas y transformaciones que el personal usa para desarrollar y mantener el software, y los productos asociados (planificación del proyecto, diseño de documentos, código, casos de prueba, manuales de usuario, entre otros).

**Capacidad / Potencialidad (*capability*) de un proceso de software:** Describe el rango de resultados esperados que se pueden llevar a cabo siguiendo un proceso de software. Una capacidad o potencial del proceso de software de una organización es un modo de predecir el resultado más probable del siguiente proyecto de software que se emprenda.

**Performance de un proceso de software:** Representa los resultados actuales logrados habiendo seguido un proceso de software. La performance de un proceso, por tanto, se enfoca en resultados logrados, mientras que el potencial de un proceso se centra en resultados esperados.

**Madurez de un proceso de software:** Se refiere a un proceso específico que está explícitamente definido, administrado, medido, controlado, y es efectivo.

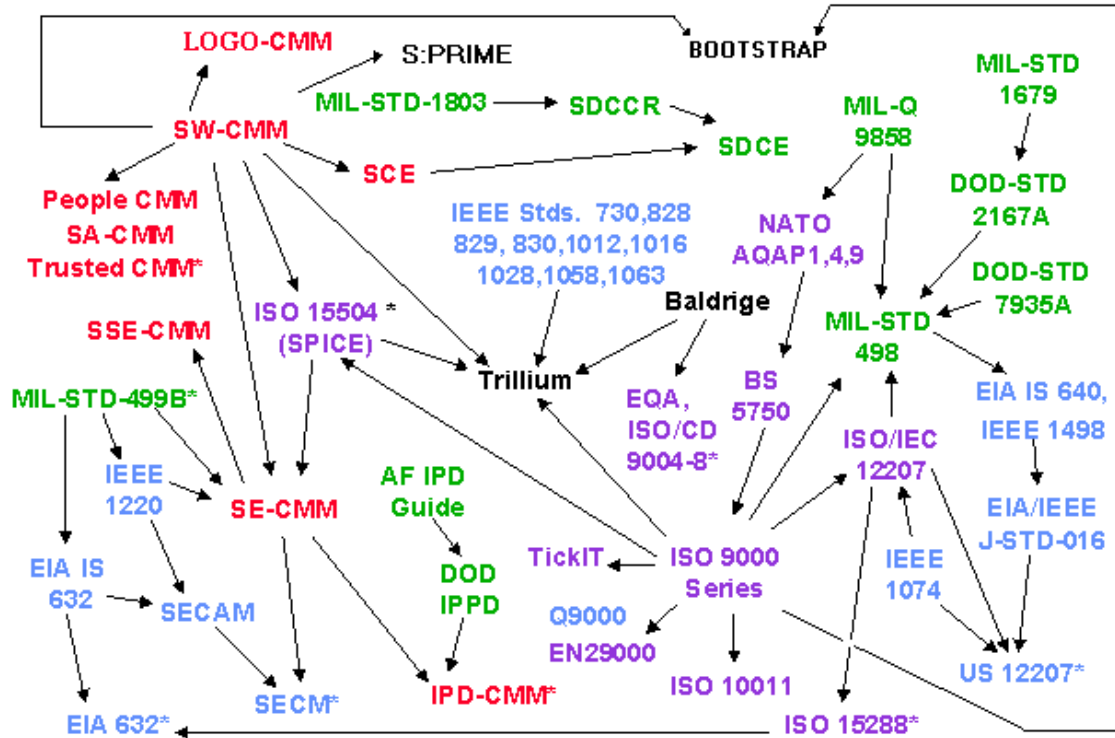
**Madurez:** Implica la potencialidad de poder crecer e indica tanto la riqueza de un proceso de software de una organización como la consistencia con que se aplica en proyectos toda la organización.

**Organización inmadura:** Los procesos de software generalmente se improvisan, esto incluye la posibilidad que, aún especificados los procesos, ellos no se desarrollen en forma rigurosa.

**Organización madura:** Posee la habilidad en toda su organización para administrar tanto el desarrollo como la mantención de proyectos.

### ¿ Qué marco de trabajo se debe aplicar ?

En la figura, aparecen los principales MT (algunos de próxima aparición) y las relaciones o

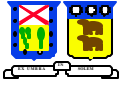


\* No publicado

herencias entre algunos de ellos.

Color	Significado del color
Rojo	Modelo de capacidad de madurez
Verde	Documento gubernamental o militar de U.S
Lila	Estándar internacional
Azul	Desarrollado por una organización profesional (mayoritariamente de U.S.A.)
Negro	Otros

Los principales marcos de trabajo por categoría.



Sería muy difícil y complejo plasmar y tratar todos los marcos de trabajo que existen actualmente o en período de estudio. La sola observación es indicativa de lo complejo que es discernir entre las variadas alternativas.

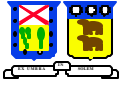
*Otras incógnitas que se derivan y cuya respuestas no son fáciles:*

- ¿Cuál MT se debe aplicar a una organización ?
- ¿ Qué características presentes en los MT hay que tomar en cuenta para seleccionar un MT ?
- ¿ Se deben aplicar en forma estricta y completa los MT o hay que ajustarlos de acuerdo a las características de la unidad informática en cuestión ?

Lamentablemente no existe una fórmula mágica que permita responder en forma taxativa estas incógnitas. En los próximos párrafos hay algunos puntos que ayudan a conseguir las respuestas y aspectos a considerar de los MT.

### **Aspectos a considerar de los MT**

1. Los MT nos permiten determinar “**dónde estamos**” hacia “**dónde queremos ir**” y “**cómo**” podemos hacerlo.  
(en la mayoría de los casos “cómo” no es muy explícito y concreto, es un “cómo” muy **difuso**)
2. Según un artículo de la *IEEE Computer*, los MT describen “el recipiente, los utensilios y las técnicas de cocina, asumiendo que la torta **sabrá bien**”.
3. El mismo artículo indica que la mayoría de los MT el **grado de bjetividad** es puramente **subjetivo** en comparación con el grado ideal que es **objetivo**.
4. Los MT permiten (y de hecho se hace) adaptar los mismos de acuerdo a las características de la unidad informática. (Recursos, experiencias, objetivos, etc.) **Lo que no se indica es cómo hacerlo y cuáles puntos hay que adaptar**.
5. La **etapa más difícil** y traumática por el cual muchas organizaciones no se adhieren hacia un MT **es el inicio**. Si se habla de modelos de madurez correspondería al nivel 0 (varios modelos) ó 1 (CMM®). Es **difícil justificar** la relación **costos-beneficios** a los gerentes para promover el uso de un MT.  
*Hay que tomar en cuenta que el esfuerzo de las evaluaciones no supere a las del desarrollo.*
6. **Incoherencia** de los MT de **proponer evaluaciones en los primeros niveles** si la mayoría de las organizaciones inmaduras no existen algunos de los procesos o prácticas que indican los MT para evaluar en los niveles iniciales.
7. Algunos expertos critican a los MT que contienen **evaluaciones**, el hecho de que estos son **derivaciones artificiales**.



Es decir, cuando una organización se compara a sí mismo con el modelo de evaluación, se están **comparando prácticas del mundo real con una lista idealizada**.

8. La mayoría de los MT considera necesario un conjunto de equipos especiales como por ejemplo una unidad de Aseguramiento de Calidad de Software (SQA), costo que muchas empresas pequeñas de la realidad observada normalmente no pueden afrontar.
9. La mayoría de los MT considera que el **personal** posee un **alto conocimiento y experiencia** de ciertas prácticas propuestas por él, como así también de conceptos generales. En la experiencia reflejada en este trabajo y en la encuesta, hay muchos profesionales que tienen solo vagas nociones de ingeniería de software.
10. La mayoría de los MT considera el **empleo de herramientas especiales**. Para el caso de las PyMES el uso de estas implica un **alto costo en recursos**, (costo de compra, entrenamiento del personal, tiempo, etc.) **estos factores** influyen en su uso.
11. La mayoría de los estudios sobre los resultados de mejoramientos de procesos están basados sobre el análisis de unos **pocos puntos de datos**. (data points) Hay muy pocos puntos de datos analizados sobre los costos actuales de los programas de mejoramiento de procesos. Por consiguiente los resultados positivos arrojados por el uso de estos programas son datos **“tentadores”** más que **hechos establecidos**.
12. No olvidar que **los procesos** involucrados en los MT son elementos que **ayudan a las soluciones** más que una solución en sí misma.
13. La finalidad directa o indirecta de los MT es de aumentar la calidad del software, pero la gran mayoría **no define en sus documentos ¿qué es calidad de software?**
14. Debido a la alta exigencia y presión para el mejoramiento de procesos, es muy factible “padecer” lo que se denomina **“parálisis de procesos”**. Este término fue definido por **Yourdon**, y se refiere al efecto sobre el equipo del proyecto que es abrumado por la nueva tecnología y gradualmente terminan por perder todo el tiempo en:
  - a) Tratar de entender la nueva tecnología;
  - b) Argumentar los méritos de la nueva tecnología, o
  - c) Tratar que trabaje o funcione la nueva tecnología.**Esta parálisis puede causar a los equipos olvidar que se desarrolla software más que procesos**. Parte de este problema se puede atribuir el no entender qué son procesos de software.

**Aclaración:** El motivo de la sección, no es el mostrar aspectos negativos de los marcos de trabajo, sino, exponer detalles a tomar en cuenta en el momento de aplicar un MT, sobre todo en el área de las unidades informáticas inmaduras/ limitadas.



### **Conclusiones sobre los marcos de trabajo**

Sin lugar a dudas que la decisión por parte de una organización de aplicar un MT trae aparejado muchos beneficios, por ejemplo:

- Disminución de costos y tiempo
- Aumento de la productividad y de la calidad de los productos
- Coordinar al personal hacia unas metas/objetivos con una misma “visión”.

**Pero, cuánto es esa disminución de costo, en qué porcentaje se puede expresar, cómo varía este valor por cada MT.** Son preguntas que todavía no tienen una respuesta directa y formal. Algunos motivos de esta situación:

- Los MT prácticamente son “nuevos”, los más antiguos tienen un promedio de existencia 7 a 10 años.
  - *Se necesita tiempo para observar los resultados y analizarlos -*
- Son muy pocas las organizaciones que publican los resultados obtenidos al utilizar un MT en particular y que se preste para dicho estudio.
  - *A quién le gusta mostrar sus defectos y errores -*
- La tecnología y herramientas (análisis, diseño, métricas, etc.) informáticas evolucionan más rápidamente en comparación con la actualización de los MT.
  - *¿Será una carrera perdida? O ¿Será cierto que los MT son totalmente independientes a esta realidad? -*
- Todavía no se ha masificado el uso de los MT.
  - *La decisión de adoptar un MT implica estudios de factibilidad. Todavía no hay una exigencia de mercado importante para certificarse en un MT en particular. -*

Actualmente los profesionales e investigadores están enfrentando estos problemas y los comentados anteriormente para conseguir un conjunto de soluciones posibles.

Quizás se está muy cerca de hallar estas soluciones pero existe un obstáculo por soslayar, y es el de crear un método dentro o anexo a los marcos de trabajo que permita: **“ Si una unidad informática está certificada y/o se encuentra en un nivel de madurez en particular determinado por un MT, determinar cuál nivel de madurez le correspondería con respecto a otro MT y poder certificarse, o tendría que cumplir otros requisitos para hacerlo.”**



### 3. Tabla comparativa

**Frameworks:** Son los MT denominados “pilares” en los cuales se basa el modelo propuesto.

**Tipo de Framework:** Nos indica si es un modelo o estándar, y el área central de trabajo. Por ejemplo, **SPICE** (la versión actual a partir de 1999 se llama **SPICE99 o ISO/IEC TR2 15504**, futuro estándar 15504) su área central es el de evaluar los procesos más que el de determinar la madurez, a pesar que posee mecanismos para hacerlo.

**Persona Interna / Externa:** Indica si es posible aplicar el MT a través del personal interno a la unidad informática, o es necesario un asesor directo externo para aplicar y/o certificar bajo la norma o modelo.

**Equipos Especiales:** Son los equipos especiales necesarios para aplicar el MT, y definidos por el marco de trabajo en particular. Los equipos entre paréntesis, el MT solo los menciona como opcional, relativo, o comparativo.

**Número de Páginas:** Indica el total de páginas del documento. Con respecto a ISO 9001, se indica TickIT, por ser una adaptación directa de ISO 9001 aplicado al software.

**Costo:** Es el costo por una copia impresa del MT, sin incluir los gastos de envío. La referencia “Gratis” indica la posibilidad de obtenerlo a través de la red Internet.

**Alcance:** El alcance del MT, si cubre niveles organizacionales, de procesos y proyectos.

**Focalizado:** Casi todos están focalizados en el ámbito de procesos.

**Niveles y Prácticas:** Son los niveles (*Trillium* lo denomina Escala) y prácticas que utiliza el MT. En ISO 9001 solo existen elementos de calidad.

**Número de Empresas:** Es la cantidad aproximada de empresas y/u organizaciones que han aplicado y continúan haciéndolo, un MT en particular.



**Tabla Comparativa de los Marcos de Trabajo Pilares**

<i>Frameworks</i>	<i>Tipo de Frameworks</i>	<i>Tiempo</i>	<i>Persona Interna/ Externa</i>	<i>Equipos especiales</i>	<i>Nº páginas</i>	<i>Costo</i>	<i>Alcance</i>	<i>Focalizado</i>	<i>Niveles y prácticas</i>	<i>Nº Empresas</i>
<b>CMM®</b> Vers. 1.1	Modelo de madurez	1-2 nivel, aprox. 2 ½ años.	Interna y Externa	SQA, SEPG, SCM	561	U\$ 49,44 Gratis Web	Organización Desarrollo Software	Procesos de software	5 Niveles de madurez 18 KPA	782
<b>LOGOS-CMM</b> Vers. 1.0	Modelo de madurez	No indicado	Interna y Externa	SQA, SEPG, SCM “reducidos”	450	U\$ 75 Web solo el artículo	Organización Desarrollo Software	Procesos de software	5 Niveles de madurez ?? KPA	200 Org. ( Etapa validación)
<b>IDEAL<sup>SM</sup></b> Vers. 1.6	<i>Administrador</i> mejoramiento de procesos	Relativo Plan SPI 1 a 5 años	Interna	SEPG, MSG, TWG	236	Gratis Web	Organización Desarrollo Software	Procesos de software	5 Fases 47 Tareas	??
<b>SPICE (ISO TR2 15504)</b> Vers. 1.00 “Borrador”	Modelo de madurez y evaluación de procesos	Relativo	Interna	<i>Sponsor, owner, assessor (SQA)</i>	474	U\$ 50 a 100 Gratis Web	Software Producción <i>Organización</i>	Procesos de software	5 Categorías de procesos 29 Procesos 200 práct. base 6 Niveles de madurez	Fase I 35 Fase II 250 Fase III 314
<b>ISO 9001</b> Vers. 1994	Estándar Internacional de calidad “tradicional”	Relativo aprox. 2 a 4 años	Interna y Externa	<i>Managers</i>	75 (TickIT)	U\$ 50 a 100	Producto Producción Organización	Calidad de los procesos	Consta de 20 elementos de calidad	8.100 a 13.500 SW Tickit >1.500 (Mundo)
<b>Trillium</b> Vers. 3.1	Modelo de madurez	18 meses	Interna	<i>Quality managment, TQM, (SQA)</i>	130	Gratis Web	Empresas (Telecomunicaciones)	Procesos	5 Niveles 8 áreas de capacidad 27 guías dtdas. 508 prácticas	< 100



## 4. Primeros pasos

Es necesario tener en claro ciertos aspectos antes de ver con detenimiento los modelos y normas pilares, y el modelo propuesto.

Por tal motivo a continuación se reseñan ciertos aspectos importantes indispensables para el entendimiento de los modelos de proceso y madurez.

### Mejoramiento de los procesos de software

Un paso importante en el direccionamiento de los **problemas del software** es tratar las tareas del software como un **proceso** que puede ser controlado, medido e improvisado. Para este propósito se define un proceso como un conjunto de tareas, que adecuadamente ejecutadas producen el resultado esperado. Claramente, un proceso de software completamente efectivo debe considerar las relaciones de los requerimientos de las tareas, las herramientas y métodos utilizados, el perfil, experiencia y motivación del personal involucrado.

*Para mejorar las capacidades del software, las unidades informáticas deben realizar 6 pasos:*

1. Comprender el estado actual del desarrollo de los procesos.
2. Desarrollar una visión de los procesos anhelados.
3. Establecer una lista de las acciones del mejoramiento de los procesos requeridos en orden de prioridad.
4. Producir un plan que acompañe las acciones requeridas.
5. Asegurar los recursos para ejecutar el plan.
6. Comenzar otra vez con el paso 1.

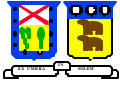
Para mejorar la organización, es conveniente tener una visión clara de los objetivos principales y algún método de medición de los progresos alcanzados.

### Las personas, en la optimización de los procesos

Claramente, cualquier proceso de software depende de la calidad de la/s persona/s que lo desarrolla.

La optimización de los procesos provee un ambiente de disciplina para trabajar de manera más formal. La disciplina debe ser manejada con cuidado, ya que, es fácil que se convierta en autoritarismo. La diferencia entre un ambiente con disciplina y otro con autoritarismo, es que el ambiente disciplinado controla el entorno y los métodos para especificar un estándar, mientras que un ambiente autoritarista define la forma actual de trabajo.

La disciplina es requerida en grandes proyectos de software para asegurar, por ejemplo, que todo el personal involucrado utilice la misma convención, no dañar los productos de otros equipos, y sincronizar apropiadamente el trabajo.



## **Principios de los procesos de cambios del software**

Los seis principios que rigen los procesos de cambio de los procesos de software son:

- I. *Los mayores cambios en la organización y a su vez en los procesos de software deben comenzar en la jerarquía más alta.* El apoyo del líder de la administración es requerido para que lance los cambios, administre los esfuerzos, los recursos y las prioridades de los cambios.
- II. *Todos deben estar involucrados.* La Ingeniería de Software es un esfuerzo de equipo, y todo aquel que no participe en el mejoramiento perderá los beneficios y siempre inhibirá todo progreso.
- III. *Cambios efectivos requieren un objetivo y conocimiento de los procesos actuales.* Para usar un mapa uno debe saber dónde está.
- IV. *Los cambios son continuos.* Los mejoramientos de los procesos de software no es un solo esfuerzo; involucra un continuo estudio y crecimiento.
- V. *Los cambios de los procesos de software no son retenidos sin un concienzudo esfuerzo y refuerzo periódico.* Es necesario introducir los cambios en forma cuidadosa, acompañado de un monitoreo constante para controlar la entropía.
- VI. *Los cambios en los procesos de software requieren inversión.* Estos necesitan: un planteamiento, personal dedicado, administración del tiempo e inversión de capital.

## **Ciertas creencias erróneas o mitos sobre los procesos de software**

Los cambios efectivos dependen del realismo. Sin ninguna apreciación del estado actual de los procesos de software y una visión futura realista, es fácil sucumbir en cambios “utópicos” en base a esquemas muy optimistas. Algunas de las creencias erróneas o mitos sobre los procesos de software son:

- ❑ *Comenzar cuando los requerimientos estén bien definidos o claros.* Es muy común pensar que los requerimientos son un trabajo del cliente y que el desarrollo no comienza hasta que estos no estén bien definidos.
- ❑ *Si pasa la etapa de testing, está todo bien.* Si los estudios existentes sobre esta errónea creencia no prueban esta falsedad, nada lo hará.
- ❑ *La calidad de software no se puede medir.* Si no medimos la calidad de nuestros productos, solamente podemos entender vagamente que tan bueno o malo son.



Actualmente existen muchas propuestas para medir la calidad de los procesos y de los productos.

- *Los problemas son técnicos.* A pesar de nuevos y mejores lenguajes, herramientas, y entornos, los problemas de costo, calendario y calidad permanecen.
- *Necesitamos el mejor personal.* Desde que los profesionales de software cometen algunos errores, ciertas personas sienten erróneamente que hay que culparlos y censurarlos.
- *La administración del software es diferente.* Aún cuando es un campo único y nuevo, los métodos tradicionales de administración pueden y deben ser usados.

## Estrategias para aplicar los cambios en los procesos de software

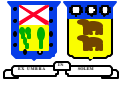
La mayoría de las organizaciones deben efectuar muchos mejoramientos, estas organizaciones necesitan por ejemplo, cambiar el sistema de administración, los procedimientos, las herramientas y los métodos, e incluso la estructura organizacional, la implementación de los cambios necesarios debe ser manejada con cuidado. Mientras que el personal de software da la bienvenida a concertar esfuerzos para mejorar sus procesos, los cambios deben ser implementados adecuadamente o estos generarán resistencias.

## Campeones, sponsors (*Sponsors*) y agentes

*Campeones* son aquellos que inician los cambios en los procesos. Ellos conducen una atención administrativa a los sujetos y elementos, obtiene el consentimiento del sponsor, y establece la credibilidad para que el programa de los cambios sea lanzado. El campeón se enfoca en los objetivos y metas, está atento para pasar los obstáculos, y rehusa darse por vencido cuando las cosas están difíciles. **Varios estudios indican que las nuevas ideas sin campeones tienen poca probabilidad de ser implementadas.**

El rol del administrador *senior* es igualmente crucial, alguien con autoridad necesita reconocer el valor del trabajo y respaldarlo/*sponsor*. Esto se realiza brindando recursos y respaldo oficial.

El paso siguiente es encontrar los *agentes* que lideren la planificación e implementación de los cambios, deben reunir los recursos, asignar los trabajos, y llamar al administrador *senior* por ayuda cuando sea necesario.



Existen varios puntos clave para elegir a los agentes:

1. Los agentes deben mostrar entusiasmo sobre liderar los cambios de los procesos.
2. Los agentes deben ser técnica y políticamente capaces de entender los problemas y asegurar que la solución efectiva sea implementada.
3. Los agentes necesitan el respeto de las personas con las cuales se relacionan.
4. Los agentes deben ser soporte y confidente de los administradores, de lo contrario no actuarán con la seguridad necesaria para obtener gran cooperación y aceptación en el grupo de trabajo.

### Los elementos del cambio

Los tres elementos claves de la efectividad del cambio son **planificación, implementación, y comunicación**. La etapa de la planificación involucra a toda persona representativa de cada grupo afectado, de esta forma se asegura realizar un plan competente y aceptado con más rapidez. Una vez elaborado el plan, hay que tener cuidado en el momento de aplicarlo, es mejor realizar pequeños esfuerzos de prueba para minimizar el riesgo de aplicación y vislumbrar algunos problemas. Una vez superado estas mini-pruebas se sigue con la aplicación en forma dinámica.

### ***Evaluación de los procesos de software***

La evaluación de los procesos de software ayuda a la organización a mejorar mediante la identificación de los problemas críticos y estableciendo prioridades de mejoramiento. Los objetivos básicos de las evaluaciones son:

- aprender cómo la organización trabaja
- identificar los mayores problemas
- enriquecer las opiniones de los líderes de los procesos de cambios.

### Visión general de las evaluaciones

Las evaluaciones de los procesos de software no son una auditoria, pero sí una revisión de la organización y de sus procesos de software para advertir a la gerencia y a los profesionales el cómo poder mejorar sus operaciones.

El propósito de las evaluaciones es el de identificar las áreas con altas prioridades de mejoramiento y el de proveer guías sobre cómo realizar dichas mejoras. Las evaluaciones están basadas en el principio que los administradores locales y los profesionales quieren mejorar sus operaciones y sus necesidades primarias son las de tener una guía de qué hacer y cómo hacerlo.



## ***Fases de las evaluaciones***

Las evaluaciones son típicamente conducidos en tres fases: preparación, evaluación, y recomendación. Durante la preparación, el administrador senior comienza a comprometerse con el proceso, participa de las decisiones personalmente, y toma en consideración las acciones resultantes de las recomendaciones o si fuera el caso explica por qué no se pueden aplicar. La fase uno termina en uno o dos días de entrenamiento al equipo de las evaluaciones.

La fase dos se realiza *in situ* en el periodo de evaluación. Esta actividad comúnmente toma varios días, incluso dos o más semanas, dependiendo del tamaño de la organización y la técnica de evaluación utilizada. Se concluye con un reporte preliminar de lo encontrado a la administración local.

En la fase tres el reporte preliminar y las recomendaciones son presentadas a los administradores locales. Luego, un grupo de acción local es armado para planificar y aplicar las recomendaciones. La organización que realiza la evaluación puede proveer asistencia durante este periodo y como consecuencia puede participar en el seguimiento de las evaluaciones.

## **Cinco principios de las evaluaciones**

Como en todas las actividades, los requerimientos básicos para una buena evaluación son un equipo competente, con liderazgo, y una organización cooperativa. Sin embargo ciertas consideraciones hay que tener en mente, estas son:

1. La necesidad de un modelo de procesos como una base para las evaluaciones
2. El requisito de la confidencia
3. Compromiso de los gerentes/administradores
4. Una actitud de respeto por los puntos de vista de las personas cuando la organización comienza a ser evaluada
5. Realizar acciones orientadoras

### **Guía para seleccionar a los miembros del equipo de evaluación.**

#### **CADA MIEMBRO DEL EQUIPO DE EVALUACION DEBE:**

- Tener entre 8 y 10 años de experiencia profesional en software
- Ser respetado en la organización
- Poseer habilidad de relacionarse con el personal
- Ser cooperador con el equipo
- Haber tenido entrenamiento con el equipo



## Reglas básicas del equipo

Es deseable el tener un conjunto de reglas para los equipos que realizan las evaluaciones. En caso de que exista un equipo externo los líderes tanto del equipo externo como del interno se reúnen para escribir acuerdos y reglas de cooperación. Los aspectos que se deben cubrir dentro del acuerdo son:

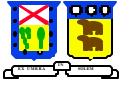
- Los resultados de las evaluaciones se manejarán confidencialmente en secreto por los miembros de los equipos.
- El administrador principal está de acuerdo en participar en la apertura y clausura de las reuniones.
- En adición de los miembros regulares, el administrador principal acceda asignar uno o dos profesionales locales para que manejen los preparativos de las evaluaciones y el seguimiento de las acciones del plan de trabajo. Ellos serán miembros a tiempo completo del equipo de evaluadores.
- El comité de administradores desarrolla e implementa el plan de acción en respuesta a las recomendaciones de los evaluadores. Cuando un plan de acción es considerado no apropiado, las razones de ello se exponen al equipo.
- El administrador principal está de acuerdo en designar una persona para que desarrolle los planes de acción. Es posible que esta persona sea miembro del equipo.

## Entrenamiento del equipo

Cuando el equipo ya este formado, los miembros acceden a participar en entrenamientos de revisiones y reuniones. Típicamente el líder del equipo conduce el programa de dos a tres días de duración. Este programa familiariza a los miembros del equipo con el proceso de evaluación y ayuda a construir un equipo cohesivo.

*Un típico programa de entrenamiento incluye los siguientes pasos:*

1. Bosquejo del calendario y objetivos de las evaluaciones.
2. Los principios de las evaluaciones son revisados, junto con el modelo de proceso de software que se utiliza como marco o referencia.
3. Los miembros de la organización concienzudamente esbozan la misión de la organización, la estructura de administración, y su historia reciente.
4. Las guías de las evaluaciones son discutidas a todo el equipo y se le solicita la firma de su aprobación.



5. Un ejercicio de construcción de equipos es realizado para asistir al grupo en el desarrollo de un soporte efectivo y cooperativo modo de operación.
6. Es cubierto el plan detallado del periodo de las evaluaciones, incluyendo los propósitos de cada sección, quienes participa sus roles. El plan cubre:
  - Los tópicos de cada sección
  - Discusión de las asignaciones de los líderes
  - Quién anota “lo encontrado” y Cómo
  - Cómo y cuándo las conclusiones del equipo serán conseguidas
  - Quiénes y por cuándo cualquier reporte y presentación son preparados y expuestos.
7. Cuando sea necesario, porción de estos procesos son repasados hasta que los miembros del equipo se sientan confortables con sus roles.
8. Planificación en el sitio o en el terreno: La tarea final del entrenamiento es la de trabajar (sin detalles) con un caso real, los miembros de la organización resumen las actividades claves del proyecto, y el equipo discute y acepta los mejores proyectos para ser realizados.

## **Bibliografía**

Extracto del libro en formato digital “calidad tradicional y de software”

Autor: Alejandro Bedini G.

abedini@ind.UTFSM.cl

Depto de Industrias, Universidad Técnica Federico Santa María, Chile