

10 - Planeamiento del proyecto de software

Objetivo del proyecto de sw: Construir un sistema de sw que cumpla con los **costos**, **tiempos**, y **calidad**. Sin embargo, muchos proyectos fallan: Un tercio se desbocan con costos o tiempos superiores al 125 % de los estipulados. Las razones principales son:

1. Objetivos pocos claros
2. Mal planeamiento
3. Administración del proyecto sin metodología
4. Nueva tecnología
5. Personal insuficiente.

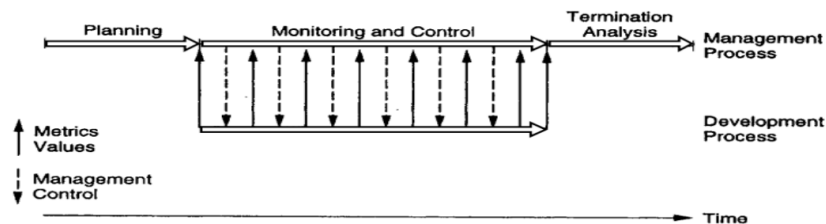
Todas ellas están relacionadas a la administración del proyecto. Vemos entonces que una administración del proyecto efectiva es clave para la ejecución exitosa del proyecto.

El proceso para la administración del proceso tiene 3 fases principales:

- planeamiento,
- seguimiento y control,
- y análisis de terminación.

El planeamiento es la actividad principal que **produce un plan**, el cual **forma la base del seguimiento**. El planeamiento se realiza antes de comenzar con el desarrollo propiamente dicho.

Requiere como entrada los **requerimientos** y la **arquitectura** (no es esencial que sean muy detallados). Durante el planeamiento se planean todas las tareas que la administración del proyecto necesita realizar. Durante el seguimiento y control, el plan es ejecutado y actualizado:



Tópicos más importantes:

1. Planeamiento del proceso.
2. Estimación del esfuerzo. Aquí está cocomo
3. Estimación de tiempos y recursos.
4. Plan para la administración de la configuración. Se toma poco
5. Planeamiento de la calidad.
6. Administración del riesgo. De lo que más se toma (es el que más se toma)
7. Plan para el seguimiento del proyecto.

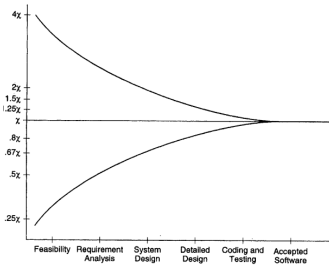
1 Planeamiento del proceso

Consiste en **planear cómo se ejecutará el proyecto**, esto incluye:

- determinar el **modelo** de proceso a seguir,
- **adecuarlo a las necesidades** del proyecto,
- definir las **etapas**,
- definir criterios de **entrada** y de **salida** en cada etapa,
- definir actividades de **verificación** a realizar en cada etapa,
- definir las **metas parciales** ("milestones") que se usan para analizar el progreso del proyecto.

2 Estimación del esfuerzo

Dado un conjunto de requerimientos es deseable/necesario saber **cuánto costará en tiempo y dinero** el desarrollo del sw. El esfuerzo se mide (usualmente) en **personas/mes**, considerando la recarga de costos por persona, la estimación del esfuerzo puede convertirse directamente en costo. La estimación del esfuerzo es clave para el planeamiento: de ello dependen los tiempos, costos, y recursos (humanos particularmente). Muchos problemas en la ejecución del proyecto se deben a una estimación inapropiada.



No hay una forma fácil de obtener la estimación. La estimación mejora a medida que se incrementa la información sobre el proyecto: las estimaciones más tempranas son más propensas a inexactitud que las avanzadas en el proyecto. Obtener una estimación con un error al 20% luego de haber especificado los requerimientos es bastante bueno.

El método COCOMO da una estimación con **error dentro del 20% en el 68% de los casos.**

Construcción de modelos

Un modelo intenta determinar la estimación del esfuerzo a partir de valores de ciertos parámetros (es decir, es una función), tales valores dependen del proyecto. Así, tener un modelo o procedimiento reduce el problema de estimar el esfuerzo del proyecto al de estimar/determinar ciertos parámetros claves del proyecto. El factor principal es el **tamaño del proyecto**. Estos parámetros deben poder medirse en etapas muy tempranas del proyecto. Dos enfoques: top-down y bottom-up.

Estimación top-down

Consiste en determinar el esfuerzo total, y luego calcular el esfuerzo de cada parte del proyecto. Por ejemplo, primero se **estima el tamaño global**, y luego se calcula:

$\text{esfuerzo} = a \times \text{tamaño}^b$, las constantes **a** y **b** se **determinan a través de análisis de regresión sobre proyectos pasados**. Los datos para la distribución del esfuerzo en cada fase a partir del total se obtienen de proyectos similares. El **tamaño** se mide en **KLOC**, el **esfuerzo** en **personas/mes**.

Estimación bottom-up

Pasos:

1. Identificar los **módulos** del sistema y clasificarlos como **simples**, **medios**, o **complejos**.
2. Determinar el **esfuerzo promedio de codificación** para cada tipo de módulo.
3. Obtener el **esfuerzo total de codificación** en base a la clasificación anterior y al conteo de cada tipo.
4. Utilizar la distribución de esfuerzos de proyectos similares para estimar el esfuerzo de cada tarea y finalmente el esfuerzo total.
5. **Refinar los estimadores anteriores** en base a factores específicos del proyecto.

Modelo COCOMO (COConstructive COst MOdel)

Es un método que sigue el enfoque top-down. Utiliza tamaño ajustado con algunos factores. Procedimiento:

1. Obtener el **estimador inicial** usando el tamaño (un número: persona/mes).
2. Determinar un conjunto de **15 factores** de multiplicación representando distintos atributos;
3. Ajustar el estimador de esfuerzo escalándolo según el factor de multiplicación final;
4. Calcular el estimador de esfuerzo de cada fase principal.

1. Obtener el estimador inicial

$$E_i = a \times \text{tamaño}^b$$

Se sigue la siguiente tabla:

Sistema	a	b
Orgánico (relativamente simple y desarrollado por equipos pequeños)	3.2	1.05
Semi-rígido	3.0	1.12
Rígido (más ambiciosos y novedosos, con restricciones estrictas impuestas por el entorno y altos requerimientos en aspectos como interface o confiabilidad)	2.8	1.20

2. Determinar los 15 factores f_k

Entre ellas:

- RELY: confiabilidad (software)
- STOR: limitaciones en el porcentaje del uso de la memoria. (hardware)
- PCAP: calificación de los programadores. (personal)
- TOOL: uso de herramientas de desarrollo de software. (proyecto)

Atributo	Muy bajo	Bajo	Normal	Alto	Muy alto	Extra alto
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1.62
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	

3. Ajustar el estimador de esfuerzo

$$\text{Esfuerzo} = E_i * \prod_{k=1}^{15} f_k$$

4. Calcular el estimador de esfuerzo de cada fase

Fase	Tamaño			
	Pequeño 2 KLOC	Intermedio 8 KLOC	Medio 32 KLOC	Grande 128 KLOC
Diseño del producto	16 %	16 %	16 %	16 %
Diseño detallado	26 %	25 %	24 %	23 %
Codificación y test de unidad	42 %	40 %	38 %	36 %
Integración y test	16 %	19 %	22 %	25 %

3 Planificación y recursos humanos

Planificación global

Depende fuertemente del esfuerzo estimado. Para una estimación dada, hay “cierta” flexibilidad, dependiendo de los recursos asignados. Ej.: un proyecto de 56 PM puede programarse en 8 meses (7 personas), 7 meses (8 personas), o 9 1/3 meses (6 personas). Hay que tener cuidado, las cuentas no son lineales.

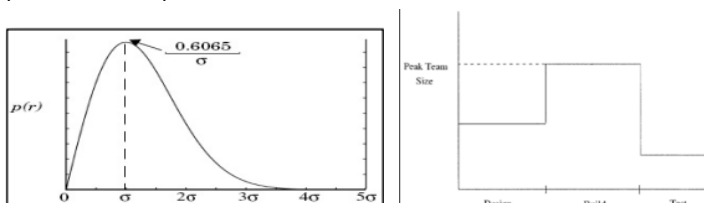
Un método es estimar el **tiempo programado del proyecto M (en meses)** como una función del esfuerzo en personas-mes. La función no es lineal y se determina analizando datos pasados.

- IBM: $M = 4.1 * \text{esfuerzo}^{0.36}$
- COCOMO: $M = 2.5 * \text{esfuerzo}^{0.38}$
- COCOMO II: $M = 3.67 * \text{esfuerzo}^{SE}$

“Rule of thumb” para chequear modificaciones: $M = \sqrt{\text{esfuerzo}}$

Luego hay que determinar la duración de cada meta parcial principal del proyecto.

La distribución de los recursos humanos no es homogénea, sigue aproximadamente una curva de Rayleigh, la cual es continua. Aunque la realidad es que los recursos humanos se distribuyen de manera discreta (escalonada).



Con esta curva y la distribución de esfuerzos, se puede determinar el tiempo de las metas parciales. La distribución de esfuerzo y la distribución de los tiempos en las fases son dos cosas distintas. Usualmente, la etapa de construcción conlleva el mayor esfuerzo, pero no necesariamente la mayor duración. COCOMO establece la distribución: Diseño - 19%, Programación - 62%, Integración - 18%.

Planificación detallada

Para alcanzar cada meta, muchas tareas deben llevarse a cabo.

Tareas de bajo nivel: aquellas realizadas por una persona en no más de 2 o 3 días.

La planificación consiste en decidir las tareas y asignarlas preservando siempre la planificación de alto nivel. Es un proceso iterativo: si no se pueden acomodar todas las tareas, se debe revisar la planificación global.

La planificación detallada no se realiza de manera completa al comienzo: ésta **evoluciona**. La planificación detallada es el documento más activo de la administración del proceso.

Cualquier actividad a realizarse debe quedar reflejada en la planificación detallada. Cada tarea tiene asignado nombre, esfuerzo, fecha, duración, recurso, etc. Hay un % de realización usado para seguimiento. La planificación detallada debe ser consistente con las metas: Las tareas son subactividades de las actividades a nivel de metas, así que el esfuerzo individual debe sumar apropiadamente preservando la duración total.

Estructura del equipo de trabajo

Para asignar las tareas en la planificación detallada es necesario un equipo de trabajo estructurado.

- La **organización jerárquica** es la más común: Hay un administrador de proyecto con la responsabilidad global (realiza diseño, asigna recursos, etc.), y tiene programadores, testers y administrador de configuración para ejecutar las tareas detalladas. Podrían existir otros roles, y una persona podría cumplir varios roles.
- Los **equipos democráticos** funcionan en pequeños grupos, y el **liderazgo es rotativo**.
- Hay una nueva alternativa utilizada para el desarrollo de grandes productos. Reconoce tres tareas principales: **desarrollo**, **testing** y **administración del programa**. Cada una de estas tareas tiene su **equipo** (se espera que los equipos de desarrollo y testing sean relativamente independientes) y cada equipo su **líder**, y todos reportan a un **líder general**. Es utilizado en grandes corporaciones desarrolladoras de software. Los administradores del programa proveen las especificaciones de lo que se debe construir y aseguran que el desarrollo y el testing estén apropiadamente coordinados.

4 Planeamiento de la administración de la configuración del software

Se deben identificar los ítems de configuración y especificar los procedimientos a usar para controlar e implementar los cambios de estos ítems. El planeamiento de la administración de configuración se realiza cuando el proyecto ha sido iniciado y ya se conoce la especificación de los requerimientos y el entorno de operación.

5 Planeamiento del control de calidad

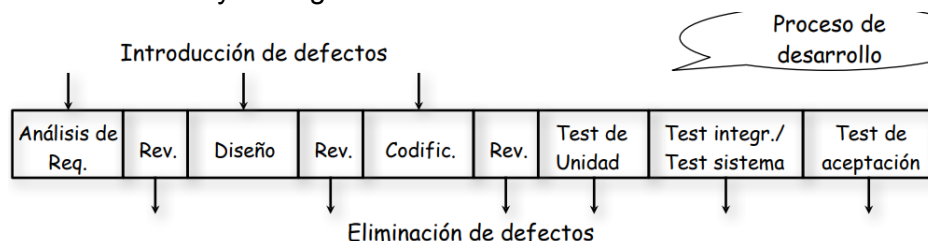
Objetivo básico: entregar un sw de alta calidad, la unidad de medida de calidad estándar es **densidad de defectos entregados** (i.e. en el sw entregado). Ej.: **#defectos/KLOC**.

Se considera **defecto** a algo que causa que el sw se comporte de manera inconsistente (con respecto a los requerimientos o necesidades del cliente).

Así, el **propósito del plan de calidad** es especificar las actividades que se necesitan realizar para **identificar y eliminar defectos** (incluyendo las herramientas y métodos que se usarán a tal efecto)

Introducción y eliminación de errores

El desarrollo de sw es una actividad altamente dependiente de personas, por lo que es propensa a errores. Los defectos se introducen en cualquier etapa. Como el objetivo de calidad es la baja densidad de defectos, los defectos deben eliminarse. Esto se realiza fundamentalmente mediante las actividades de control de calidad (QC) incluyendo revisiones y testing.



Enfoques para la administración del Control de Calidad

Enfoque ad hoc:

Se hacen tests y revisiones de acuerdo a **cuándo y cómo se necesiten**.

Enfoque de procedimiento

El plan define qué **tareas de control de calidad se realizarán y cuándo**. Las principales tareas del control de calidad son **revisión** y **testing**. Provee procedimientos y lineamientos a seguir en el testing y en la revisión. Durante la ejecución del proyecto se asegura el seguimiento del plan y los procedimientos.

Enfoque cuantitativo (!)

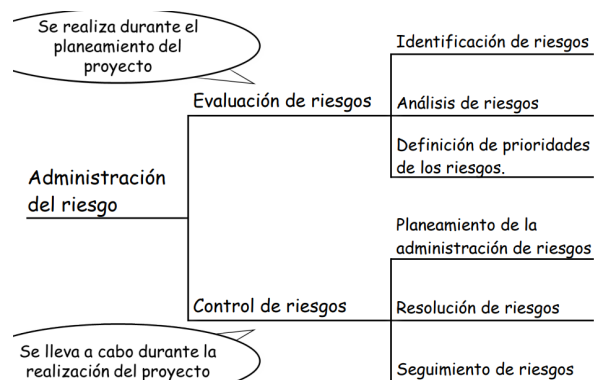
Va más allá de requerir que se ejecute el procedimiento. Analiza los datos recolectados de los defectos y **establece juicios sobre la calidad** (métricas, densidad de defectos). La información del pasado es muy importante, permite la predicción de defectos y compara la cantidad real de defectos contra la estimada. Parámetros claves: tasas de introducción y eliminación de defectos.

Plan de calidad

Establece qué actividades deben realizarse. El nivel del plan depende de los modelos de predicción disponibles. De mínima debe definir las tareas de QC que deben realizarse durante el proyecto, como por ejemplo qué documentos serán revisados, qué niveles de testing se realizarán, etc. También puede especificar los niveles esperados de defectos que cada tarea de QC debe encontrar, permite el seguimiento de la calidad del proyecto y depende de la disponibilidad de datos y modelos.

Administración de riesgos

Cualquier proyecto puede fallar debido a eventos no previstos. La administración de riesgo es un intento de minimizar las chances de fallas. **Un riesgo es cualquier condición o evento de ocurrencia incierta que puede causar la falla del proyecto**. Tales eventos no son comunes (ej.: no incluye movimiento de personal o cambio de requerimientos, los que deben ser tratados por la administración del proyecto.). El objetivo de la administración del riesgo: minimizar el impacto (en los costos, calidad y tiempos) de la materialización de los riesgos.



Evaluación del riesgo (parte de administración de riesgos)

Identificación del riesgo

Consiste en identificar los posibles riesgos del proyecto, es decir, aquellos eventos que podrían ocurrir y causar la falla del proyecto. La forma de hacerlo es mediante: listas de control, experiencias pasadas, brainstorming (“¿qué ocurriría si...?”), etc. **Los 10 factores de riesgo más importantes** son:

1. Personal: insuficiente o inapropiadamente entrenado.
2. Tiempos y costos irreales.
3. Componentes externas: incompatibles o de baja calidad.
4. Discrepancia con los requerimientos; “gold plating”.
5. Discrepancia con la interfaz del usuario.
6. Arquitectura, desempeño, calidad: inadecuada o insuficiente evaluación.
7. Flujo continuo en los cambios de requerimientos.
8. Software legado.
9. Tareas desarrolladas externamente: inadecuadas o demoradas.
10. Uso extremo de las capacidades de la ciencia de la computación.

Análisis de riesgos y definición de prioridades (!)

La cantidad de riesgos puede ser grande. Se deben **priorizar** para enfocar la **atención en las áreas de alto riesgo**. Para ello hay que establecer la probabilidad de materialización de los riesgos identificados y la pérdida que éstos originarían. Luego, se debe ordenar de acuerdo al “valor de exposición al riesgo” (RE):

RE = (probabilidad de la ocurrencia indeseada) * (impacto de la ocurrencia indeseada)

Es decir, RE es el valor esperado de la pérdida debido a un riesgo. Realizar planes para tratar con los riesgos de mayor RE.

Se clasifican las **probabilidades de ocurrencia** como **Bajas, Medias, o Altas**. Luego, clasificar los **impactos de los riesgos** como **Bajos, Medios, o Altos**. Identificar todos los riesgos AA y AM/MA, y enfocarse en ellos para mitigar riesgos. Esta técnica funcionará bien para la mayoría de los proyectos pequeños y medianos.

Control de riesgos

Si es posible evitarlo, hay que evitarlo. Por ejemplo, si usar nuevo hardware introduce riesgo, hay que evitarlo utilizando hardware ya verificado. En los otros casos, hay que planear y ejecutar los pasos necesarios para mitigar los riesgos: Definir las acciones a seguir en el proyecto de manera que, si el riesgo se materializa, su impacto sea mínimo. Esto involucra un costo extra. Ejemplos de mitigación de riesgo:

“Demasiados cambios de requerimientos”

- Convencer al cliente que los cambios de requerimientos tienen un **alto impacto en los tiempos**.
- Definir un procedimiento para cambios de requerimientos.
- Mantener el impacto acumulado de los cambios y hacérselo notar al cliente.
- Negociar **pagos del esfuerzo real**.

“Desgaste en el personal involucrado”

- Asegurarse que se asignan **múltiples recursos a áreas claves** del proyecto.
- Realizar actividades para la **integración del equipo** (team building sessions).
- **Alternar las tareas** entre los miembros del equipo.
- Mantener **recursos de apoyo** (backup) en el proyecto.
- Mantener documentación de los trabajos individuales.
- Seguir estrictamente el proceso de administración de cambios.

“Planificación de tiempos irreal”

- Negociar mejor planificación de tiempos.
- Identificar tareas paralelas.
- Tener los recursos listos de manera temprana.

- Identificar las áreas que pueden ser automatizadas.
- Si el camino crítico no cumple con los tiempos, renegociar con el cliente.
- Negociar pagos del esfuerzo real.

Plan de mitigación de riesgos

La mitigación de riesgos incluye los pasos a realizar (en consecuencia, a costo extra). Estos pasos deben planificarse en el tiempo y ejecutarse. Estos pasos son distintos de los que se deben realizar si el riesgo se materializa, los cuales se efectúan sólo si es necesario. Los riesgos deben revisarse periódicamente.

7 Planificación del seguimiento del proyecto

El plan de administración del proyecto es meramente un **documento** que sirve como **guía**, éste debe ejecutarse. Para asegurar que la ejecución se realiza como se planeó, ésta debe seguirse y controlarse. El **seguimiento** (monitoring) requiere de mediciones y métodos que las interpreten. El plan de seguimiento incluye: planificar qué medir, cuándo y cómo, y cómo analizar y reportar estos datos.

Mediciones

Principales medidas:

- **Tiempo** (cronograma): es la más importante de las medidas (¿se están cumpliendo los plazos?)
- **Esfuerzo**: es el principal recurso (¿se está dentro del presupuesto?); usualmente se sigue (track) a través de herramientas de reporte de esfuerzo.
- **Defectos**: determinan calidad; usualmente se siguen con herramientas de registros y seguimientos.
- **Tamaño**: mucha información se normaliza respecto al tamaño.

Seguimiento del proyecto (monitoring and tracking)

Objetivo: hacer visible la ejecución del proyecto de manera de realizar acciones correctivas cuando sea necesario con el fin de asegurar el éxito del proyecto.

Hay distintos niveles de seguimientos:

- nivel de actividad,
- reportes de estado,
- análisis de metas parciales.

Las mediciones proveen los datos para estos seguimientos.

Seguimiento a nivel de actividad:

- Asegura que cada actividad se realiza apropiadamente y a tiempo.
- Realizado diariamente por los administradores de proyecto.
- Una tarea realizada se marca con 100%; las herramientas pueden determinar el estado de las tareas de más alto nivel.

Reporte de estado:

- Usualmente se prepara semanalmente.
- Contiene: resumen de actividades completadas y pendientes (desde el último reporte); cuestiones que necesitan atención o deben ser resueltas.

Análisis de metas parciales:

- Se realiza una mayor revisión con cada meta parcial.
- Análisis de esfuerzos y tiempos reales vs. estimados.
- Si la desviación es amplia se deben tomar medidas correctivas.
- Revisión de los riesgos.
- El gráfico costo-tiempo de metas parciales es un método para capturar el progreso básico del proyecto.