10. Planeamiento del Proyecto del Software

Introducción

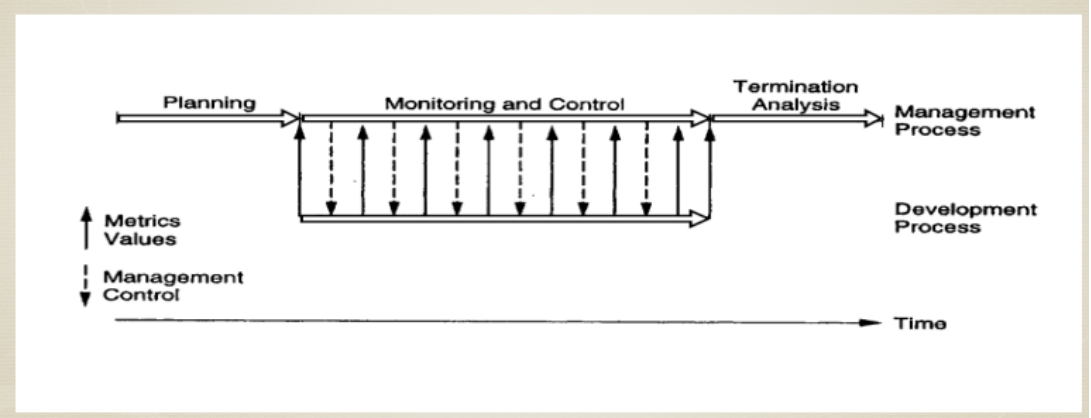
Muchos proyectos fallan a la hora de cumplir con costos o tiempos debido a objetivos pocos claros, mal planeamiento, administración del proyecto sin metodología, personal insuficiente, etc. Por eso, el planeamiento o administración de un proyecto efectivo es importante.

Veremos entonces como realizar una buena administración de un proyecto para evitar estos problemas.

Proceso de la administración del proyecto

Como vimos en la unidad 8 (Procesos de Software), el proceso de administración de un proyecto tiene tres fases.

1. **Planeamiento**: se deben asignar recursos. Las tareas claves son:
    - Estimar costos y tiempos
    - Seleccionar personal
    - Planear seguimiento
    - Planear control de calidad
2. **Seguimiento y control**: acompaña al desarrollo. Sus tareas son:
    - Seguir y observar parámetros como costo, tiempos, riesgo, etc.
    - Tomar acción correctiva si es necesario.
3. **Análisis de terminación**: se centra en analizar el desempeño del proceso de desarrollo y llegar a conclusiones como:
    - Qué errores se cometieron
    - Qué lecciones se aprendieron



Nos enfocaremos en la fase 1 de planeamiento:

Planeamiento

El planeamiento requiere como entrada los requerimientos y la arquitectura del SW. Se usarán para planear todas las tareas que se necesitan realizar. Éste tiene 7 puntos:

1. Planeamiento del proceso

Se plantea cómo se ejecutará el proyecto, incluyendo:

- Modelo de proceso a seguir
- Adecuarlo a las necesidades del proyecto
- Definir etapas, criterios de salida y entrada de cada una
- Actividades de verificaciones a realizar en cada etapa
- Definir metas parciales (”milestones”)

2. Estimación del esfuerzo usando modelos ⭐

Es deseable saber cuánto $$$ y tiempo costará el desarrollo de SW del proyecto. Para esto, estimamos el esfuerzo: se mide en personas/mes y se busca saber el costo $ por persona; muchos problemas en la ejecución de un proyecto se deben a una mal estimación de esfuerzo.

Para estimar el esfuerzo, se hace uso de **modelos** matemáticos que que buscan, a través de parámetros claves, estimar el esfuerzo requerido de un proyecto. Éstos modelos usan datos históricos y experiencia acumulada en proyectos anteriores para su estimación. Puntualmente, se basan en la idea de que ciertos factores influyen en el esfuerzo y pueden medirse en las primeras etapas del proyecto.

En particular, estos modelos pueden tener dos enfoques: top-down y bottom-up

Modelos para top-down

Primero se determina el esfuerzo total y luego el esfuerzo para cada parte del proyecto. Ejemplo:

1. Se estima el **tamaño global** tg
2. Se calcula **esfuerzo** = $a * tg^b$ donde a y b se determinan a través del análisis de regresión sobre proyectos pasados.
3. Se distribuye el esfuerzo en cada fase o componente del proyecto en función del tamaño estimado.

Modelo para bottom-up

1. La idea es identificar los módulos y clasificarlos como simples, medios e intermedios.
2. Se determina el esfuerzo promedio de codificación para cada tipo de módulo.
3. Se obtiene el esfuerzo total de codificación en base a la clasificación anterior y al conteo de cada tipo.
4. Se usa la distribución de esfuerzos de proyectos similares para estimar el esfuerzo de cada tarea y finalmente el esfuerzo total.
5. Se refinan los estimadores anteriores en base a factores específicos del proyecto.

Modelo COCOMO ⭐

Un modelo de estos para estimar el esfuerzo es el modelo top-down **COCOMO**, que tiene una estimación con un error dentro del 20% en el 68% de los casos.

El modelo **COCOMO** usa el tamaño global tg ajustado con algunos factores. El procedimiento de este modelo es: (estos procedimientos se toman en los parciales)

1. Obtener el **estimador inicial** E_i usando tg.

$$E_i = a * tg^b$$

donde a y b:

Sistema	a	b
Orgánico	3.2	1.05
Rígido	2.8	1.20
Semi-rígido	3.0	1.12

> 💡 No hace falta memorizar los valores de la tabla, pero si entender cada tipo de sistema.

2. Determinar un conjunto de **15 factores de multiplicación** representando distintos atributos:

   Atributos del hardware:
   TIME: limitaciones en el uso de CPU
   STOR: limitaciones en el uso de memoria
   VIRT: volatilidad de la máquina virtual
   TURN: frecuencia de cambio en el modelo de explotación

   Atributos del software:
   RELY: confiabilidad
   DATA: tamaño de la BD
   CPLX: complejidad de las funciones, datos, interfaces

   Atributos del persona:
   ACAP: codificación de los analistas
   AEXP: experiencia del personal
   PCAP: calificación de programadores
   VEXP: experiencia del personal en la máquina virtual
   LEXT: experiencia en el lenguaje de programación a usar

   Atributos del proyecto:
   MODP: uso de prácticas modernas de programación
   TOOL: uso de herramientas de desarrollo de SW
   SCED: limitaciones en el cumplimiento de la planificación

   Valores f_k para factor k

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	

No hay que aprenderse todos los factores, solo 4 o 5.

3. Ajustar el **estimador de esfuerzo** escalándolo según el **factor de multiplicación final**.

$$esfuerzo = E_i * \Pi_{k=1}^{15} f_k$$

4. Calcular el **estimador de esfuerzo** de cada fase principal.

   La siguiente tabla sólo es para sistemas orgánicos. Hay tablas diferentes para los otros tipos de sistemas.

   Se divide en 4 fases, e incluye el estimador de esfuerzo para distintos tamaños.

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. Estimación de tiempos y recursos

Hay dos niveles de estimación/planificación de tiempos y recursos:

- **Global**: abarca las metas parciales (milestones) y la fecha final.

  En este tipo de planificación, se establecen los objetivos generales y las milestones del proyecto, y se determina cómo se llevará a cabo el mismo a lo largo del tiempo, analizando tiempos y costos.

  Dependiendo de la estimación del esfuerzo necesario para el proyecto, se puede tener cierta flexibilidad en la planificación global, que está relacionada con la asignación de recursos.

  Ej.: un proyecto de 56 PM puede programarse en 8 meses (7 personas), 7 meses (8 personas), o 9 ⅓ meses (6 personas).

  Igual la profe dice q esto no es tan lineal. La relación entre el esfuerzo y el tiempo varía según proyecto y recursos disponibles.

  Un enfoque común para hacer la planificación global es estimar el tiempo programado del proyecto M (en meses) como una función del esfuerzo en persona/mes.

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

  SE es tamaño estimado del SW.

Además, se usa la regla “Rule of Thumb” como método rápido para verificar modificaciones en la planificación. Sugiere que el tiempo programado **M** = √esfuerzo aproximadamente.

Ejemplo: suponer esfuerzo = 56 personas/meses.

IBM: M = 4.1 * 560.36 = 17.46 meses

=> 56/17.46 < 4 personas

COCOMO: M = 2.5 * 560.38 = 11.54 meses

=> 56/11.54 < 5 personas

“Rule of thumb”: M = √56 = 7.48 meses

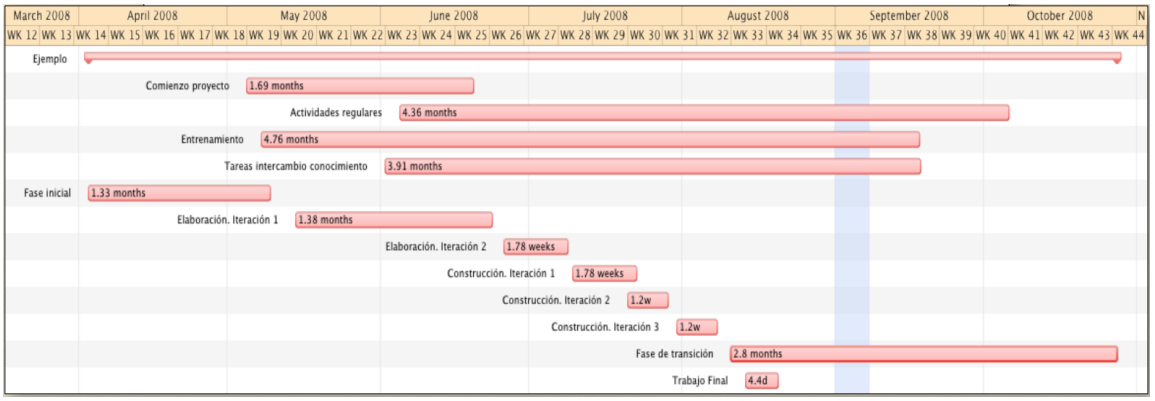
=> 56/7.48 < 8 personas

Luego de estimar el tiempo programado del proyecto **M** lo que se hace es determinar la **duración de cada milestone parcial** del proyecto, que se hace conociendo cómo será la **distribución de los RRHH y la distribución del esfuezo**.

Ejemplo de planificación global

Planificación global

#	Info	Title	Expected Duration	Given Planned Work	Expected Start	Expected End
0		**Ejemplo**	7.4 months		03/04/2008	27/10/2008
1		Comienzo proyecto	1.69 months	24.2 days	05/05/2008	19/06/2008
2		Actividades regulares	4.36 months	35.13 days	05/06/2008	06/10/2008
3		Entrenamiento	4.76 months	49.37 days	08/05/2008	18/09/2008
4		*Tareas intercambio conocimi...*	3.91 months	19.56 days	02/06/2008	18/09/2008
5		Fase inicial	1.33 months	22.67 days	03/04/2008	09/05/2008
6		Elaboración. Iteración 1	1.38 months	55.16 days	15/05/2008	23/06/2008
7		Elaboración. Iteración 2	1.78 weeks	35.88 days	26/06/2008	08/07/2008
8		**Construcción. Iteración 1**	1.78 weeks	**24.63 days**	10/07/2008	22/07/2008
22		Construcción. Iteración 2	1.24 weeks	28.22 days	21/07/2008	29/07/2008
23		Construcción. Iteración 3	1.24 weeks	27.03 days	31/07/2008	08/08/2008
24		Fase de transición	2.8 months	179.62 d...	11/08/2008	27/10/2008
25		Trabajo Final	4.44 days	6.44 days	14/08/2008	20/08/2008



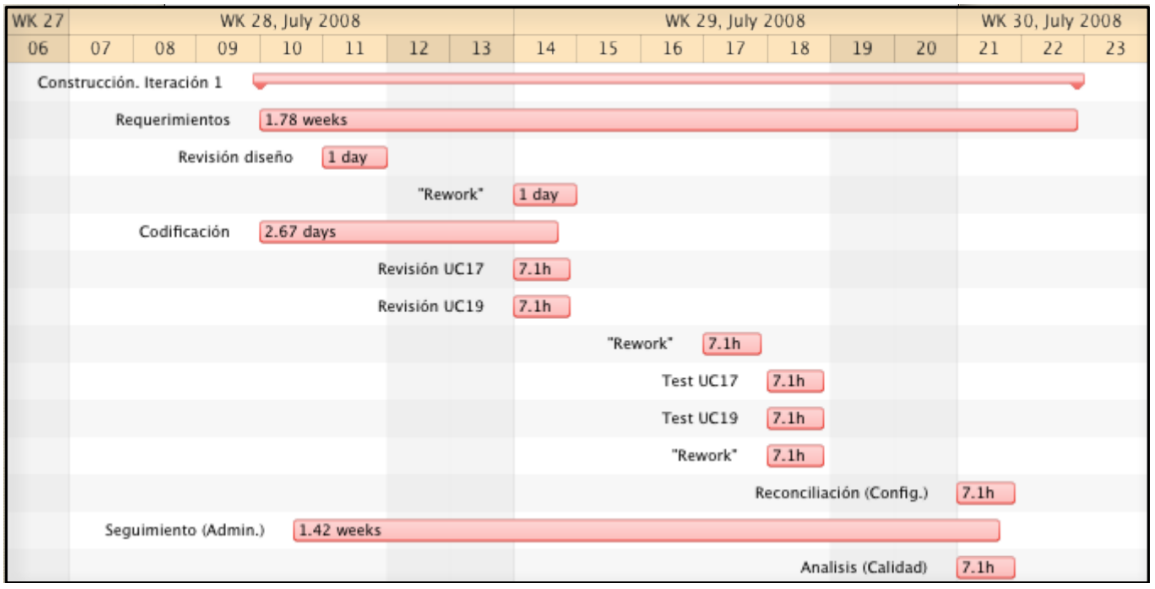
▼ **Detallada**: asignación de las tareas de más bajo nivel a los recursos.

Implica la descomposición de las metas parciales en tareas específicas y su asignación detallada de recursos, incluidas las personas y los tiempos para llevar a cabo esas tareas. Lo que se hace es:

1. Descomponer las metas en tareas especificas.
2. Asignar a cada tarea los recursos necesarios para llevarla a cabo (personas, definición de fechas de inicio y finalización, estimación de duración de la tarea, etc).
3. Repetimos hasta acomodar todas las tareas dentro del marco de tiempo establecido o los recursos disponibles. Si esta acomodación no puede lograrse, se debe ajustar la planificación global.

Se debe tener en cuenta que la planificación detallada evoluciona con el tiempo a medida que se obtiene más información y se realizan ajustes. Se podrían agregar nuevas tareas o modificaciones.

Ejemplo de planificación detallada:



Luego, debemos tener en cuenta algunos puntos para la estructura del trabajo en equipo:

- Organización jerárquica: hay un administrador con la responsabilidad global. Tiene programadores, testers y un administrador de configuración.
- Equipos democráticos: grupos chicos con liderazgo rotativo.
- Alternativa: se usa más para sistemas grandes, se tienen equipos para desarrollo, testing, etc, cada uno con su líder general.

4. Plan para la administración de la configuración

Esto se realiza cuando el proyecto fue iniciado y ya se conoce la especificación de los requerimientos y el entorno de la operación. Aquí, lo que se hace es identificar items de configuración y especificar procedimientos a usar para controlar e implementar los cambios de estos items.

⬇️ Esto me parece q la profe no toma un pingo

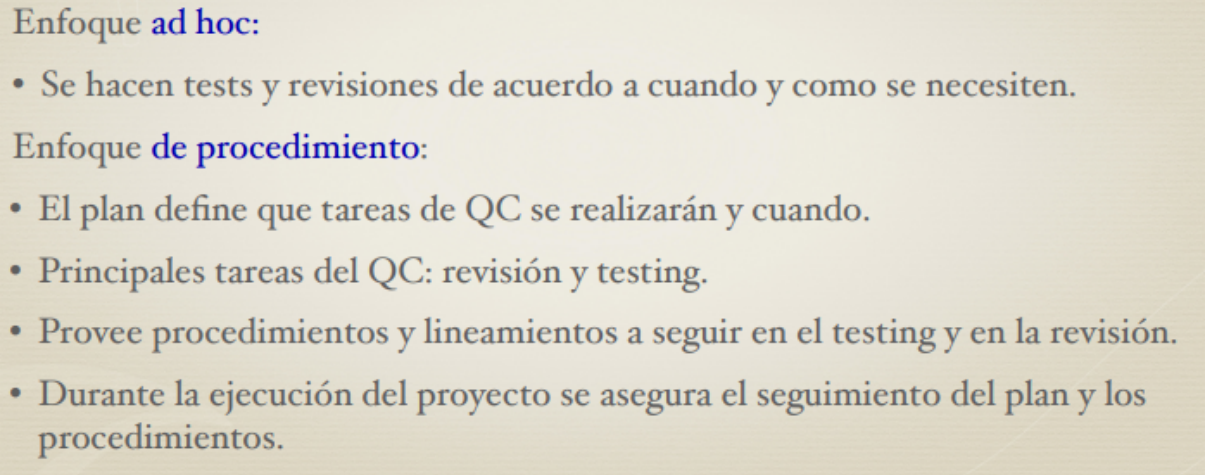
Algunas actividades a seguir para hacer esto son:

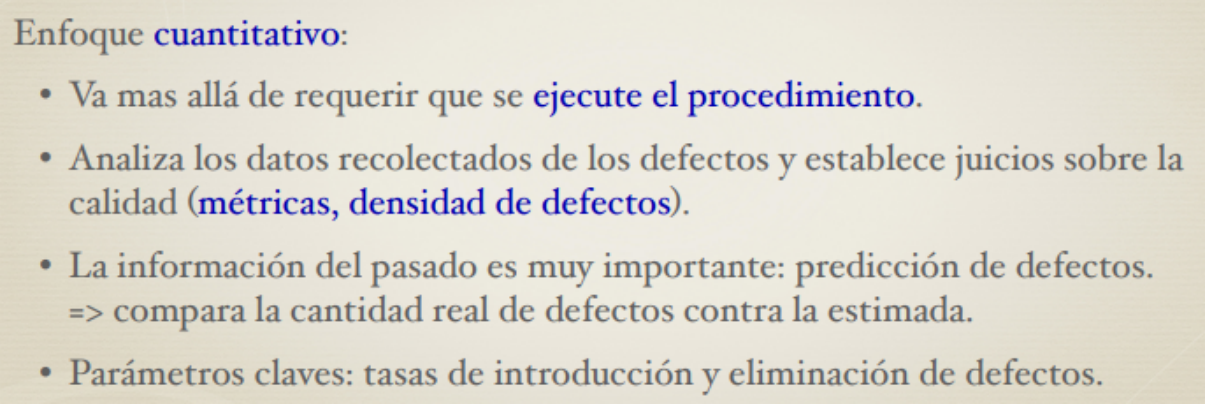
- Identificar los items de configuración (IC).
- Definir un esquema de nomenclatura para cada IC.
- Definir la estructura de directorios necesaria.
- Definir el procedimiento para el versionado y los métodos para rastrear los cambios en los IC.
- Definir las restricciones de acceso.
- Definir los procedimientos para el control de cambios.
- Identificar y definir las responsabilidades de la administración de configuración.
- Identificar los puntos en los que se crearán las “baselines”.
- Definir el procedimiento de backup.
- Definir el procedimiento de “release”.

5. Planeamiento de la calidad

El objetivo de esto es entregar un software de alta calidad, que es proporcional a la cantidad de defectos con respecto a la cantidad de líneas de código.

Para esto, llevamos a cabo controles de calidad (QC) + revisiones + testing. Existen tres enfoques para la administración de los controles de calidad (QC):





Según sea el enfoque, se elabora el documento **plan de calidad:**

- Se establecen actividades a realizar
- Se establece nivel de plan dependiendo del modelo de predicción disponible
- Se definen las QC a realizar
- Debe especificar los niveles esperados de defectos que cada tarea de QC debe encontrar.

6. Administración del riesgo

La administración de riesgo es un intento de minimizar las fallas generadas por **riesgos** en un proyecto.

▶️ **Riesgo**: condición o evento de ocurrencia incierta que puede causar la falla del proyecto.

Dentro de la administración de riesgos tenemos dos áreas importantes:

1. Evaluación de riesgos → se hace durante el planeamiento del proyecto

    ▼ a. Se identifican los riesgos

Ésto se hace mediante listas de control, experiencias pasadas, brainstorming, etc.

Algunos de los riesgos más importantes son:

- Personal insuficiente o mal entrenado
- Tiempos y costos irreales
- Componentes incompatibles o de baja calidad
- Software legado, entre otros …

▼ b. Se hace un análisis de los riesgos identificados y se definen prioridades

Para enfocar la atención en aquellos riesgos más críticos, se establece la probabilidad de que esos riesgos se materialicen en algo grave dentro de nuestro sistema.

Para ello, ordenaremos los riesgos identificados según su **valor de exposición al riesgo (RE)**, en otras palabras, es el valor esperado de pérdida debido a un riesgo.

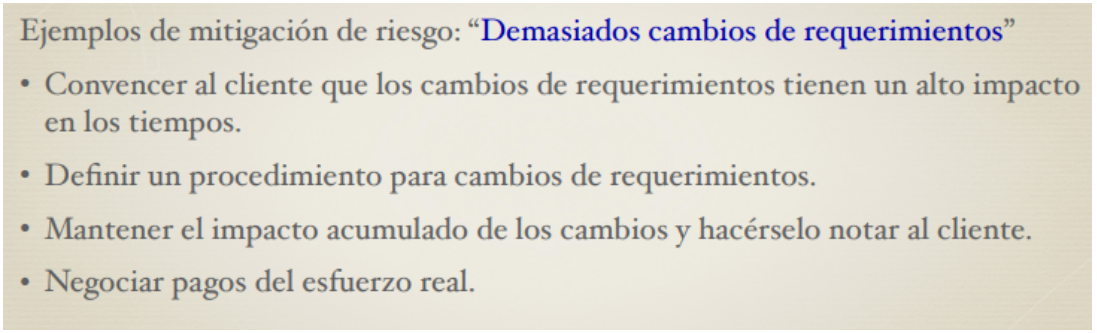
Luego, se clasifican las RE y sus impactos como Bajas, Medias o Altas.

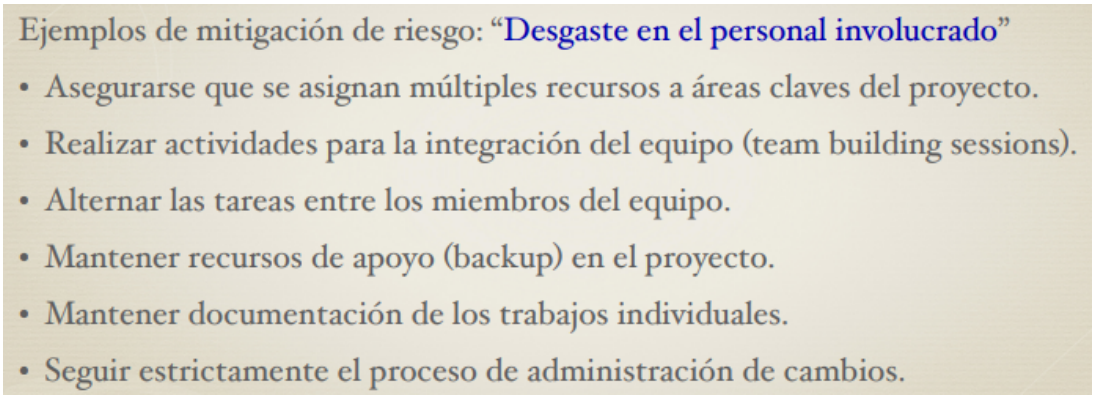
2. Control de riesgos → se hace durante la realización del proyecto

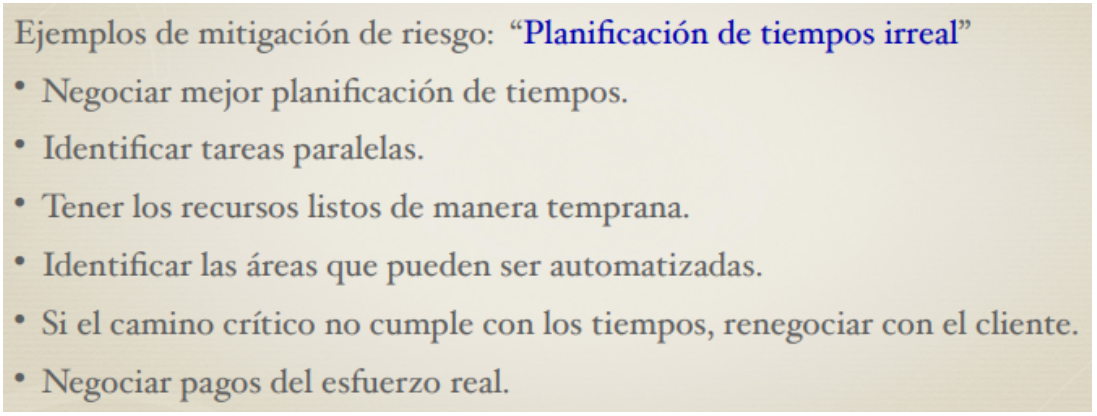
▼ a. Se planea la administración de los riesgos

Se definen las acciones a seguir en el proyecto por si el riesgo se materializa, así se disminuye el impacto.

Algunos ejemplos:







b. Se realiza la resolución de los riesgos

c. Se hace un seguimiento sobre los riesgos.

7. Plan para el seguimiento del proyecto

Es meramente un documento que sirve de guía. Establece cómo se llevará a cabo el proyecto de desarrollo de software, incluyendo los recursos, actividades y plazos.

Para garantizar que la ejecución de un proyecto sea acorde al plan detallado en el documento, se realiza un **monitoring:** se supervisa el progreso del proyecto en comparación con el plan. Ésto incluye utilizar medidas específicas, entre ellas:

- Tiempo: ve el cronograma y chequea que se estén cumpliendo los plazos
- Esfuerzo: es el principal recurso, se fija que se esté dentro del presupuesto
- Defectos: determinan la calidad
- Tamaño: es un dato importante a medir porque mucha info se normaliza respecto al tamaño.

El monitoring en sí tendrá como objetivo observar la ejecución del proyecto de modo que se puedan hacer acciones correctivas cuando sea necesario. Hay tres niveles de seguimiento:

1. Nivel de actividad

   Se asegura que cada actividad se realiza apropiadamente y a tiempo.

2. Reportes de estado

   Se hacen semanalmente y contienen resumen de actividades completadas y pendientes y cuestiones que necesitan atención o deben ser resueltas

3. Análisis de metas parciales

   Se hace una revisión en cada meta parcial para analizar los esfuerzos y tiempos reales vs estimados. Se aplican medidas correctivas.

> 💡 Las mediciones vistas proveen los datos para éstos seguimientos.

fin