



*Universidad Nacional de Tucumán*



Universidad Nacional de Tucumán  
Facultad de Ciencias Exactas y Tecnología

## Ingeniería de Software II

# **Gestión del proyecto Estimación y riesgo**

Mg. Héctor A. Valdecantos

# Planificación y estimación

- La planificación del proyecto es la primera actividad en la administración de un proyecto de software
  - Se necesita una planificación efectiva
- Disponer de un tiempo para planificación y adaptación del plan
- Métricas de proceso y proyectos sirven de input para una estimación.
- Factores:
  - Complejidad del proyecto
  - Tamaño del proyecto
  - Grado de incertidumbre estructural

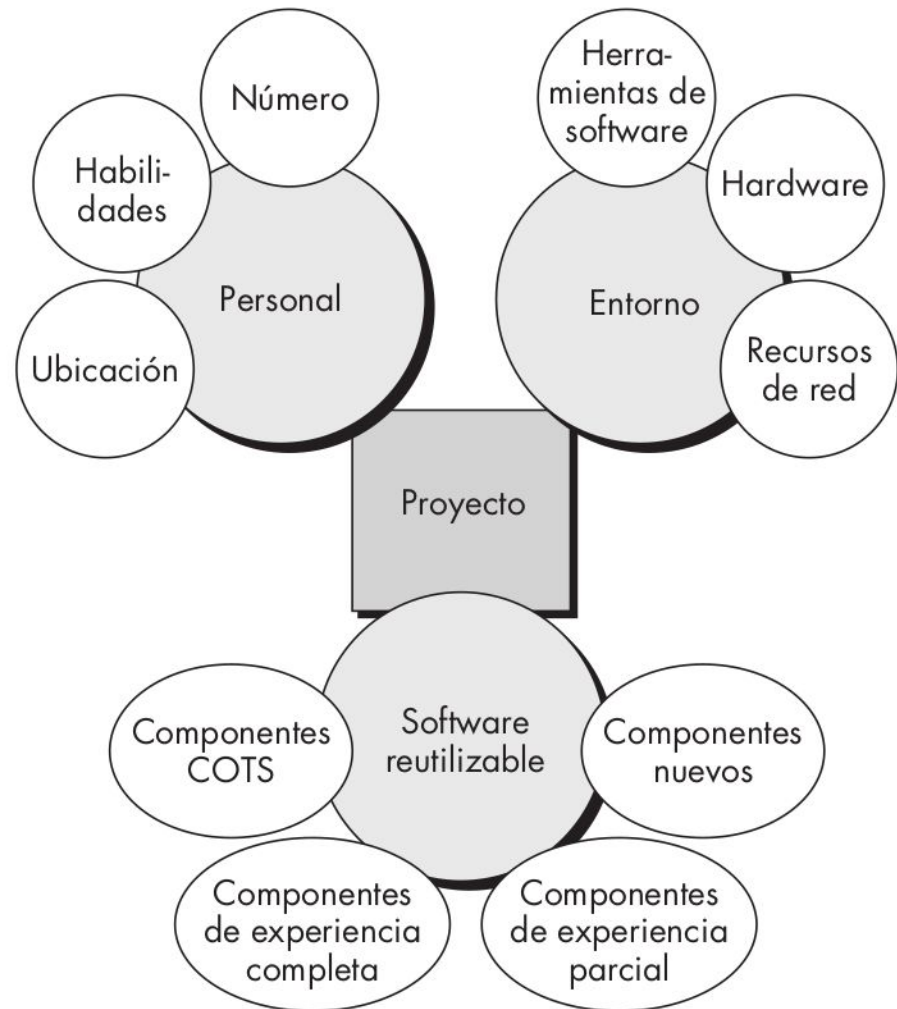
# Alcance del software y factibilidad

- La funcionalidad, los datos, las interfaces para los usuario, el rendimiento o performance, las restricciones, interfaces y fiabilidad determinan el alcance.
- Se define a través de
  - Narrativas desarrolladas después de comunicarse con todos los stakeholders
  - El conjunto de casos de usos
- Podemos construir un software de este alcance?
  - Tecnología: es factible tecnológicamente?
  - Finanza: la disponibilidad financiera es la adecuada?
  - Tiempo: la velocidad de desarrollo es suficiente?
  - Recursos: contamos con los recursos?

# Estimación de recursos

Especificaciones principales de recursos:

- descripción
- disponibilidad
- momento en el que se requerirá el recurso
- duración del tiempo que se aplicará el recurso.



# Estimación del proyecto de software

- **Muchas variables** (es más un arte que una ciencia)
  - Humanas, técnicas, ambientales, políticas, etc
- **Opciones para estimaciones con un riesgo aceptable**
  - Retrasar las estimaciones las hacen menos riesgosas
  - Basar estimaciones en proyectos ya completados
  - Usar **técnicas de descomposición** simples para estimar costo y esfuerzo
  - Usar **modelos empíricos** para estimar costo y esfuerzo

$$d = f(v_i)$$

$$d = \{esfuerzo, costo, duracion, \dots\}$$

$$v = \{LOC, FP, \dots\}$$

# Técnicas de descomposición

- Estimar un proyecto de software es complejo
  - Dividirlo en problemas más pequeños (Divide & conquer)
- Las estimaciones de proyecto son tan buenas como la estimación de tamaño que se hizo.
  - Se usa principalmente LOC, FP
- Datos históricos sirven para definir los valores optimista, más probable y pesimista (indicar el grado de incertidumbre)

$$estimacion = \frac{h_{optimista} + 4h_{probable} + h_{pesimista}}{6}$$

- La probabilidad de que el resultado final LOC o FP quede fuera es muy baja.
- Con la variable de tamaño estimada se estiman esfuerzo, costo, duración, etc

# Modelos empíricos

- Fórmulas derivadas empíricamente para estimar
  - Obtenidas de un número limitado de proyectos
  - No están contemplados todas las clases de proyectos
- Implementados en software de estimación que usan LOC o FP como argumentos
- El modelo de estimación debe ser calibrado:
  - Se usa data histórica  $Estimacion = A + B(e_v)^C$

$A, B, C$  : constantes empíricas

$e_v$  : variable de estimacion (LOC o FP)

- Ejemplos:

$$E = 5.2 \times (\text{KLOC})^{0.91}$$

Modelo Walston-Felix

$$E = 5.5 + 0.73 \times (\text{KLOC})^{1.16}$$

Modelo Bailey-Basili

$$E = 3.2 \times (\text{KLOC})^{1.05}$$

Modelo Boehm simple

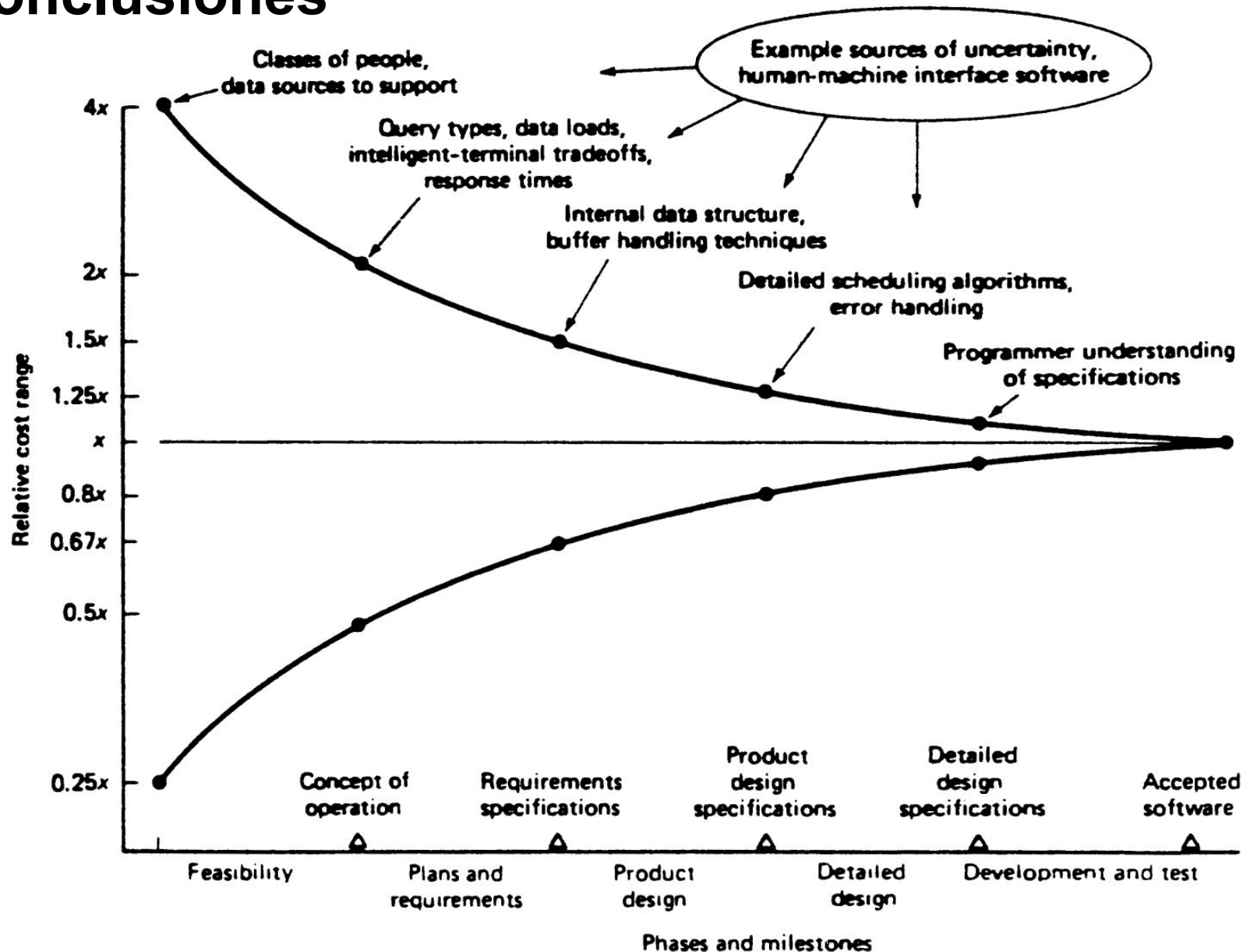
$$E = 5.288 \times (\text{KLOC})^{1.047}$$

Modelo Doty para KLOC > 9

# Modelos empíricos: COCOMO

- Una jerarquía de modelos de estimación de software que llevan el nombre COCOMO, de Barry Boehm
- COnstructive COst MOdel: modelo constructivo de costos.
- Evolucionó a COCOMO II
  - Modelo de composición de aplicación
  - Modelo de etapa temprana de diseño
  - Modelo de etapa post-arquitectónica

# Conclusions



# Conclusiones sobre estimaciones

- Estimar antes de comenzar el proyecto es difícil
  - Cuánto durará?
  - Cuánto esfuerzo requerirá?
  - Cuánto gente estará implicada?
  - Qué recursos se necesitan?
- Definir una agenda de desarrollo es difícil
- Se estima en base a requerimientos de alto nivel y datos históricos
- Si el esfuerzo inicial es  $x$  meses de esfuerzo:
  - el esfuerzo real va de  $0.25x$  a  $4x$  (Basado en datos recabados por Boehm, 1995)
- Es difícil de estimar el tamaño al comienzo del proyecto, las estimaciones terminan siendo subjetivas (se necesita experiencia).

# Técnicas de estimación

- Definir una agenda de desarrollo es difícil
- Se estima en base a requerimientos de alto nivel
- Muchas incertidumbres:
  - El ambiente de producción puede ser desconocido
  - se puede desconocer el skill del equipo
- La estimación es usada para determinar el budget
  - Muchas veces, el presupuesto se mantiene y se negocian las funcionalidades
- Dos tipos de tecnicas de estimacion
  - Basada en la experiencia
  - Modelado algorítmico de costos

# Técnicas de estimación

- Si el esfuerzo inicial es x meses de esfuerzo:
  - el esfuerzo real va de 0.25x a 4x (Basado en datos recabados por Boehm, 1995)
- Modelado algorítmico
  - $\text{Effort} = A * \text{Size}^B * M$ 
    - A es una constante que depende las prácticas que la organización maneja y el tipo de software
    - Size puede ser las SLOC, FP, otra
    - B usualmente entre 1 and 1.5, relacionado con el tamaño y la complejidad.
    - M un factor construido combinando atributos del proceso, producto y desarrollo.
- Es difícil de estimar el tamaño al comienzo y B y M son estimaciones subjetivas (se necesita experiencia).



*Universidad Nacional de Tucumán*



Universidad Nacional de Tucumán  
Facultad de Ciencias Exactas y Tecnología

Ingeniería de Software II

# Gestión de Riesgos del Software

Mg. Héctor A. Valdecantos

# La administración de proyectos de software es desafiante (Sommerville)

- El software es intangible
  - A diferencia de otras ingenierías.
- Grandes proyectos de software son únicos (one-off)
  - Proyectos grandes son diferentes de los proyectos previos. Los cambios de tecnología también los hacen únicos.
- Los procesos de software son variables y específicos
  - Los modelos se adaptan a la organización.

# Definición de Riesgo

Se trata de identificar o anticipar las potenciales causas que pueden afectar negativamente:

- Agenda (schedule)
- El presupuesto (budget)

Y luego tomar acción para anular, disminuir, o actuar sobre los riesgos.

**Riesgo:** Probabilidad de que una circunstancia adversa ocurra. Los riesgos son una amenaza para el proyecto, para el producto o servicio de software que se está desarrollando y para la organización a cargo.

# Categorías de riesgos (Sommerville)

- Riesgos del proyecto

- Afectan la agenda o los recursos. La agenda se dilata y los costos se incrementan. Ejemplo: Un ingeniero experimentado abandona el proyecto.

- Riesgos del producto

- Afectan la calidad del software en desarrollo. Ejemplo: una librería de software adquirida no se desempeña como lo esperado.

- Riesgos del negocio

- Afectan a la organización que desarrolla el software. Ejemplo: un nuevo producto de software presentado por la competencia

# Riesgos de producto, proyecto y negocio comunes

Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organizational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool underperformance	Product	CASE tools, which support the project, do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

# Proceso de la gestión de riesgos (Sommerville)

## 1. Identificación de riesgos

- Generar un listado de riesgos potenciales para el proyecto, producto y negocio.

## 2. Análisis de riesgos

- Evaluar la probabilidad y consecuencias de estos riesgo para una priorización.

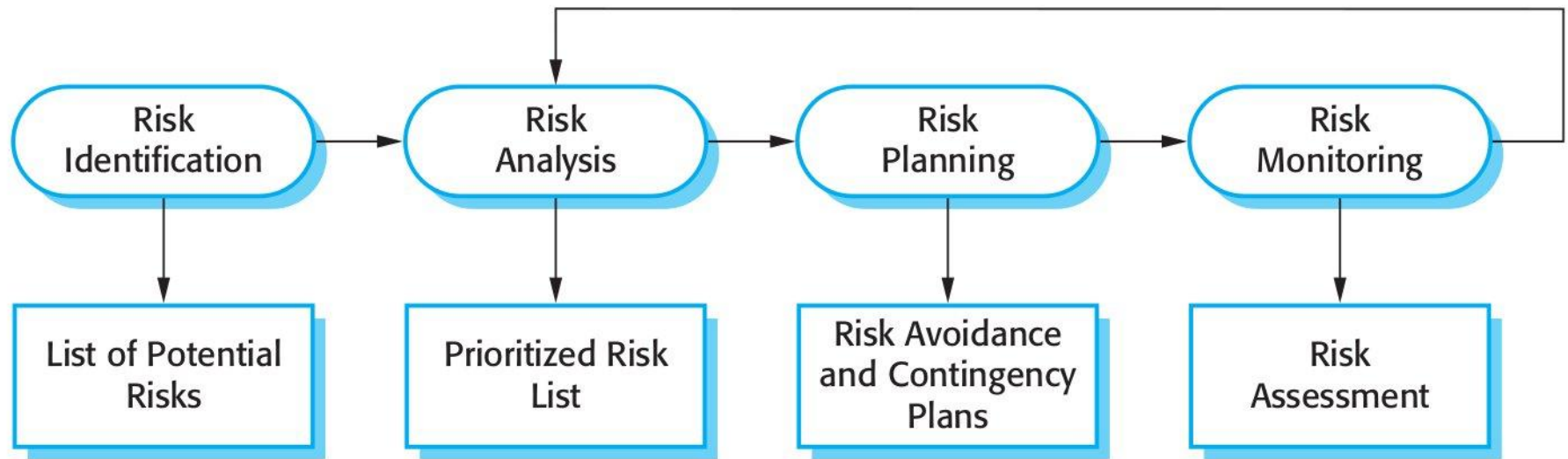
## 3. Planificación de riesgos

- Planificar cómo abordar el riesgo, ya sea para mitigar o anular el riesgo.

## 4. Monitorización de riesgos

- Regularmente evaluar los riesgos y el plan para abordarlos cuando se disponga de información más detallada.

# Proceso de la gestión de riesgos



- Proceso iterativo que se aplica a lo largo del proyecto
- La prevención de riesgos y planes de contingencia deben actualizarse cuando surja nueva información

# Identificación de riesgos (Sommerville)

1. Riesgos tecnológicos: aquellos que derivan de la tecnología de software o hardware usados en el desarrollo.
2. Riesgos de personas: aquellos asociados con la gente en el equipo de desarrollo.
3. Riesgos organizacionales: aquellos derivados del ambiente organizacional donde el software se está desarrollando.
4. Riesgos de herramientas: derivados de herramientas CASE y/o software de apoyo usados para el desarrollo.
5. Riesgos de requisitos: derivados de los cambios que se efectúan a los requerimientos y el proceso de la administración de cambios.
6. Riesgos de estimación: relacionados a las estimaciones de los recursos necesarios para el desarrollo.

# Identificación de riesgos (Sommerville)

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. (1) Reusable software components contain defects that mean they cannot be reused as planned. (2)
People	It is impossible to recruit staff with the skills required. (3) Key staff are ill and unavailable at critical times. (4) Required training for staff is not available. (5)
Organizational	The organization is restructured so that different management are responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7)
Tools	The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9)
Requirements	Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11)
Estimation	The time required to develop the software is underestimated. (12) The rate of defect repair is underestimated. (13) The size of the software is underestimated. (14)

[Ian Sommerville. Software Engineering (9th ed.), pag. 599, 2010]

# Análisis de riesgos (Sommerville)

No es posible hacer una evaluación numérica precisa de la probabilidad y gravedad del riesgo, sólo se debe confiar en el juicio y experiencia previa. Se hace una valoración por intervalos:

## 1. Probabilidad del riesgo

- muy baja (<10%)
- baja (10-25%)
- moderada (25-50%)
- alta (50-75%)
- muy alta (>75%)

## 2. La gravedad del riesgo se la puede evaluar como:

- catastrófica (amenaza a la supervivencia del proyecto),
- seria (puede causar atrasos grandes)
- tolerable (se pueden contener los atrasos)
- insignificante

# Análisis de riesgos (Sommerville)

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget (7).	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project (3).	High	Catastrophic
Key staff are ill at critical times in the project (4).	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused. (2).	Moderate	Serious
Changes to requirements that require major design rework are proposed (10).	Moderate	Serious
The organization is restructured so that different management are responsible for the project (6).	High	Serious
The database used in the system cannot process as many transactions per second as expected (1).	Moderate	Serious
The time required to develop the software is underestimated (12).	High	Serious
Software tools cannot be integrated (9).	High	Tolerable
Customers fail to understand the impact of requirements changes (11).	Moderate	Tolerable
Required training for staff is not available (5).	Moderate	Tolerable
The rate of defect repair is underestimated (13).	Moderate	Tolerable
The size of the software is underestimated (14).	High	Tolerable
Code generated by code generation tools is inefficient (8).	Moderate	Insignificant

[Ian Sommerville. Software Engineering (9th ed.), pag. 600, 2010]

# Planificación de riesgos

Qué hacer cuando un riesgo ocurre? Definir estrategias para los riesgos identificados:

## 1. Estrategias de anulación o prevención

- Para reducir las probabilidades de que un riesgo surja. El ejemplo de tratar con componentes defectuosos (pag. siguiente).

## 2. Estrategias de minimización

- Para reducir el impacto de un riesgo. Ej: estrategia para el personal enfermo (pag. siguiente).

## 3. Planes de contingencia

- Si sucede lo peor, se está preparado y se cuenta con una estrategia para abordar la situación. Ej: estrategia para problemas financieros (pag. siguiente).

# Planificación de riesgos

Risk		Strategy
Organizational financial problems	3	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.
Recruitment problems		Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.
Staff illness	2	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	1	Replace potentially defective components with bought-in components of known reliability.
Requirements changes		Derive traceability information to assess requirements change impact; maximize information hiding in the design.
Organizational restructuring		Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance		Investigate the possibility of buying a higher-performance database.
Underestimated development time		Investigate buying-in components; investigate use of a program generator.

# Monitorización de riesgos

La supervisión o monitorización de riesgos es un proceso continuo acerca de si las suposiciones sobre los riesgos del producto, el proceso, y el negocio han cambiado o no.

- Evaluar regularmente los riesgos identificados
  - Verificar la probabilidad de que ocurra el riesgo en cada revisión administrativa del proyecto
- Observar los factores involucrados
  - Determinar los factores que hacen los riesgos más probables de aparecer. Ej: cambios de requerimientos.

# Monitorización de riesgos

Risk type	Potential indicators
Technology	Late delivery of hardware or support software; many reported technology problems.
People	Poor staff morale; poor relationships amongst team members; high staff turnover.
Organizational	Organizational gossip; lack of action by senior management.
Tools	Reluctance by team members to use tools; complaints about CASE tools; demands for higher-powered workstations.
Requirements	Many requirements change requests; customer complaints.
Estimation	Failure to meet agreed schedule; failure to clear reported defects.

[Ian Sommerville. Software Engineering (9th ed.), pag. 602, 2010]

# Conclusión

- Se debe identificar y valorar los riesgos mayores
- Los riesgos se analizan para lograr una clasificación
  - Probabilidad y criticidad
- Se planifica qué hacer para evitar, minimizar o actuar en caso de que un riesgo se materialice
- Se los evalúa de manera iterativa en el proyecto

# Bibliografía

Pressman, Roger S. Software engineering: a practitioner's approach. 7th edition. Palgrave Macmillan, (2010).

Ian Sommerville. Software Engineering (9th ed.). Addison-Wesley Publishing Company, USA, (2010).

Boehm, Barry W. "Software engineering economics." IEEE transactions on Software Engineering 1 (1984): 4-21.