

Gestión de riesgos Software

Ingeniería del Software

Jesús Aguirre Pemán
Enrique Ballesteros Horcajo
Jaime Dan Porras Rhee
Ignacio Iker Prado Rujas
Alejandro Villarín Prieto

28 de Febrero de 2013

Índice general

I	Identificación de los riesgos	1
II	Análisis del riesgo	5
1.	Deficiencias del personal	6
2.	Planificaciones y presupuestos poco realistas	7
3.	Desarrollo de las funciones y propiedades erróneas	8
4.	Desarrollo erróneo del interfaz de usuario	10
5.	Chapado	10
6.	Continua corriente de cambios en los requisitos	11
7.	Deficiencias en componentes proporcionados externamente	11
8.	Deficiencias en tareas desarrolladas externamente	12
9.	Deficiencias en rendimiento en tiempo real	12
10.	Exprimir las capacidades informáticas	13
III	Priorización del riesgo	15
1.	Deficiencias del personal	16
2.	Planificaciones y presupuestos poco realistas	16
3.	Desarrollo de las funciones y propiedades erróneas	17
4.	Desarrollo erróneo del interfaz de usuario	18
5.	Chapado	18
6.	Continua corriente de cambios en los requisitos	18
7.	Deficiencias en componentes proporcionados externamente	19
8.	Deficiencias en tareas desarrolladas externamente	19
9.	Deficiencias en rendimiento en tiempo real	19
10.	Exprimir las capacidades informáticas	20
11.	Prioridad de los riesgos	20
IV	Gestión del riesgo	23
1.	Baja temporal de algún miembro del equipo por enfermedad	24

2.	El cliente rechaza las funciones que hemos desarrollado	25
3.	Los programas desarrollados son muy difíciles de usar y poco efectivos	25
4.	Poco tiempo para asimilar los pasos a seguir	26
5.	Falta de conocimiento por parte de los componentes del equipo . . .	27
6.	Retraso en las entregas por mala planificación	28
7.	El producto no se ajusta a lo que el cliente necesita por falta de comunicación	28

Parte I

Identificación de los riesgos

En esta parte se identificarán todos los riesgos que pueden afectar a nuestro proyecto.

- **Deficiencias del personal**

- Baja temporal de algún miembro del equipo por enfermedad
- Baja definitiva de algún miembro del equipo por enfermedad
- Abandono de la asignatura por parte de algún miembro del equipo
- Abandono del proyecto por algún miembro del equipo
- Abandono de la carrera por algún miembro del equipo
- Baja del supervisor del proyecto

- **Planificaciones poco realistas**

- Retraso en las entregas por mala planificación
- No entregar todo lo acordado en la planificación por falta de tiempo
- Pérdidas insubsanables
- Mayores gastos de lo esperado
- Cierre del proyecto por ser inviable

- **Desarrollo de las funciones y propiedades erróneas**

- Las funciones son ineficientes
- Poca calidad de las funciones y propiedades realizadas
- Las distintas partes del proyecto no cumplen con su cometido
- Dificultad para hacer que las distintas funciones del proyecto se coordinen entre ellas
- El producto no se ajusta a lo que el cliente necesita por falta de comunicación
- El cliente rechaza las funciones que hemos desarrollado
- El cliente no sabe que funciones debe desarrollar el producto
- El producto no funciona debidamente en la plataforma en que se quiere usar
- El lenguaje no permite realizar todas las funciones
- Los programas desarrollados son muy difíciles de usar y poco efectivos

- **Desarrollo erróneo del interfaz de usuario**

- La interfaz de usuario es demasiado difícil de construir
- Falta de recursos para el desarrollo de la interfaz
- El cliente considera que la interfaz es difícil de usar
- Al cliente no le resulta atractiva la interfaz de usuario
- El cliente decide cambiar por completo la interfaz de usuario
- **Chapado**
 - Abandono del proyecto
- **Continúa corriente de cambios en los requisitos**
 - El cliente cambia de opinión acerca de lo que debe hacer el proyecto
 - El cliente no sabe que espera que haga el producto
 - Los distintos clientes aportan visiones muy distintas del producto
- **Deficiencias en componentes proporcionados externamente**
 - Las librerías de java no son eficientes para nuestro proyecto
 - Los programas proporcionados son muy difíciles de usar y poco efectivos
 - Los recursos son proporcionados demasiado tarde
- **Deficiencias en tareas desarrolladas externamente**
 - Poco tiempo para realizar correcciones
 - Poco tiempo para asimilar los pasos a seguir
 - Poco tiempo para realizar el proyecto
- **Deficiencias en rendimiento en tiempo real**
 - Falta de recursos para realizar el proyecto
 - Nuestro producto no cumple con los requisitos de rendimiento
 - Nuestro producto no garantiza la calidad de uso
- **Exprimir las capacidades informáticas**
 - Falta de conocimiento por parte de los componentes del equipo

Parte II

Análisis del riesgo

En esta parte se analizan todos los riesgos identificados, exponiendo cuál es la probabilidad de que se produzcan y explicando la consecuencia que tendrían si llegaran a producirse.

1. Deficiencias del personal

Baja temporal de algún miembro del equipo por enfermedad

- **Probabilidad:** Frecuente.
- **Consecuencia:** Crítica. Aumento de la carga de trabajo entre los restantes miembros, disminución de la calidad del producto, retraso en las entregas.

Baja definitiva de algún miembro del equipo por enfermedad

- **Probabilidad:** Improbable
- **Consecuencia:** Catastrófica. Aumento de la carga de trabajo entre los restantes miembros, disminución de la calidad del producto, retraso en las entregas.

Abandono de la asignatura por parte de algún miembro del equipo

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Aumento de la carga de trabajo entre los restantes miembros, disminución de la calidad del producto, retraso en las entregas.

Abandono del proyecto por algún miembro del equipo

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Aumento de la carga de trabajo entre los restantes miembros, disminución de la calidad del producto, retraso en las entregas.

Abandono de la carrera por algún miembro del equipo

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Aumento de la carga de trabajo entre los restantes miembros, disminución de la calidad del producto, retraso en las entregas.

Baja del supervisor del proyecto

- **Probabilidad:** Remota.
- **Consecuencia:** Seria. Cambio en la organización del proyecto y en su desarrollo.

2. Planificaciones y presupuestos poco realistas

Retraso en las entregas por mala planificación

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Cambio en la planificación del proyecto y empeoramiento de los resultados.

No entregar todo lo acordado en la planificación por falta de tiempo

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Cambio en la planificación del proyecto y empeoramiento de los resultados.

Pérdidas insubsanables

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Cambio en la planificación del proyecto y empeoramiento de los resultados.

Mayores gastos de lo esperado

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Cambio en la planificación del proyecto y empeoramiento de los resultados.

Cierre del proyecto por ser inviable

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Suspenso en IS.

3. Desarrollo de las funciones y propiedades erróneas

Las funciones son ineficientes

- **Probabilidad:** Probable.
- **Consecuencia:** Menor. Empeoramiento de la calidad del software.

Poca calidad de las funciones y propiedades realizadas

- **Probabilidad:** Probable.
- **Consecuencia:** Menor. Empeoramiento de la calidad del software.

Las distintas partes del proyecto no cumplen con su cometido

- **Probabilidad:** Remota
- **Consecuencia:** Crítica. Necesidad de volver a desarrollar las partes del proyecto que no funcionan correctamente.

Dificultad para hacer que las distintas funciones del proyecto se coordinen entre ellas

- **Probabilidad:** Probable.
- **Consecuencia:** Menor. Aumento del esfuerzo.

El producto no se ajusta a lo que el cliente necesita por falta de comunicación

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Habría que rehacer gran parte del proyecto, con todo el coste que esto supone.

El cliente rechaza las funciones que hemos desarrollado

- **Probabilidad:** Probable.
- **Consecuencia:** Crítica. Necesidad de cambiar enormemente el desarrollo del proyecto.

El cliente no sabe qué funciones debe desarrollar el producto

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Aumento del tiempo necesario para desarrollar el producto.

El producto no funciona debidamente en la plataforma en que se quiere usar

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Es necesario desarrollar de nuevo el producto.

El lenguaje no permite realizar todas las funciones

- **Probabilidad:** Remota.
- **Consecuencia:** Seria. Será necesario buscar soluciones alternativas.

Los programas desarrollados son muy difíciles de usar y poco efectivos

- **Probabilidad:** Probable.
- **Consecuencia:** Crítica. El cliente rechaza el producto y habría que rehacerlo de nuevo, con todo el esfuerzo y costes que esto supone.

4. Desarrollo erróneo del interfaz de usuario

La interfaz de usuario es demasiado difícil de construir

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Aumento del esfuerzo, el coste y gran disminución de la calidad.

Falta de recursos para el desarrollo de la interfaz

- **Probabilidad:** Remota.
- **Consecuencia:** Crítica. Gran aumento del esfuerzo y el coste, y gran disminución de la calidad.

El cliente considera que la interfaz es difícil de usar

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Volver a realizar la interfaz de usuario.

Al cliente no le resulta atractiva la interfaz de usuario

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Volver a realizar la interfaz de usuario o modificarla en su mayor parte.

El cliente decide cambiar por completo la interfaz de usuario

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Desarrollo de una nueva interfaz de usuario aumentando coste y esfuerzo.

5. Chapado

Abandono del proyecto

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. Suspenso en IS

6. Continua corriente de cambios en los requisitos

El cliente cambia de opinión acerca de lo que debe hacer el proyecto

- **Probabilidad:** Improbable.
- **Consecuencia:** Catastrófica. cambio total en el desarrollo del proyecto. Sería necesario empezar de nuevo.

El cliente no sabe que espera que haga el producto

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Supondría tener que estar rediseñando requisitos, re-escribiendo código y rehaciendo el producto según le pareciera al cliente.

Los distintos clientes aportan visiones muy distintas del producto

- **Probabilidad:** Probable.
- **Consecuencia:** Seria. Es muy difícil avanzar en el proyecto.

7. Deficiencias en componentes proporcionados externamente

Las librerías de java no son eficientes para nuestro proyecto

- **Probabilidad:** Improbable.
- **Consecuencia:** Crítica. Sería necesario implementar funciones que pueden llegar a ser muy complejas.

Los programas proporcionados son muy difíciles de usar y poco efectivos

- **Probabilidad:** Probable.

- **Consecuencia:** Crítica. Falta de información, aumento del esfuerzo y necesidad de buscar otras alternativas.

Los recursos son proporcionados demasiado tarde

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Aumento del esfuerzo, necesidad de buscar otras alternativas, disminución de la calidad y posibles retrasos.

8. Deficiencias en tareas desarrolladas externamente

Poco tiempo para realizar correcciones

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Los errores son corregidos muy tarde y se propagan mucho.

Poco tiempo para asimilar los pasos a seguir

- **Probabilidad:** Probable.
- **Consecuencia:** Crítica. Aumento desmesurado de la dificultad del proyecto.

9. Deficiencias en rendimiento en tiempo real

Falta de recursos para realizar el proyecto

- **Probabilidad:** Remota.
- **Consecuencia:** Crítica. Empeoramiento de la calidad software.

Nuestro producto no cumple con los requisitos de rendimiento

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Necesidad de realizar de nuevo el trabajo para que cumpla con los requisitos mínimos.

Nuestro producto no garantiza la calidad de uso

- **Probabilidad:** Ocasional.
- **Consecuencia:** Crítica. Es necesario realizar de nuevo el producto.

10. Exprimir las capacidades informáticas

Falta de conocimiento por parte de los componentes del equipo

- **Probabilidad:** Frecuente.
- **Consecuencia:** Crítica. Empeoramiento enorme de la calidad, aumento de la dificultad de desarrollo, retraso en las entregas.

Parte III

Priorización del riesgo

A continuación se exponen las prioridades de cada riesgo, calculadas según su probabilidad y consecuencia. En la siguiente sección serán ordenados según su prioridad.

1. Deficiencias del personal

- Baja temporal de algún miembro del equipo por enfermedad
 - Prioridad: Intolerable.
- Baja definitiva de algún miembro del equipo por enfermedad
 - Prioridad: Media.
- Abandono de la asignatura por parte de algún miembro del equipo
 - Prioridad: Media.
- Abandono del proyecto por algún miembro del equipo
 - Prioridad: Media.
- Abandono de la carrera por algún miembro del equipo
 - Prioridad: Media.
- Baja del supervisor del proyecto
 - Prioridad: Baja.

2. Planificaciones y presupuestos poco realistas

- Retraso en las entregas por mala planificación
 - Prioridad: Alta.
- No entregar todo lo acordado en la planificación por falta de tiempo
 - Prioridad: Alta.
- Cierre del proyecto por ser inviable
 - Prioridad: Media.

3. Desarrollo de las funciones y propiedades erróneas

- Las funciones son ineficientes
 - Prioridad: Media.
- Poca calidad de las funciones y propiedades realizadas
 - Prioridad: Media.
- Las distintas partes del proyecto no cumplen con su cometido
 - Prioridad: Media.
- Dificultad para hacer que las distintas funciones del proyecto se coordinen entre ellas
 - Prioridad: Media.
- El producto no se ajusta a lo que el cliente necesita por falta de comunicación
 - Prioridad: Alta.
- El cliente rechaza las funciones que hemos desarrollado
 - Prioridad: Intolerable.
- El cliente no sabe qué funciones debe desarrollar el producto
 - Prioridad: Alta.
- El producto no funciona debidamente en la plataforma en que se quiere usar
 - Prioridad: Media.
- El lenguaje no permite realizar todas las funciones
 - Prioridad: Baja.

Los programas desarrollados son muy difíciles de usar y poco efectivos

- Prioridad: Intolerable.

4. Desarrollo erróneo del interfaz de usuario

- La interfaz de usuario es demasiado difícil de construir
 - Prioridad: Alta
- Falta de recursos para el desarrollo de la interfaz
 - Prioridad: Media.
- El cliente considera que la interfaz es difícil de usar
 - Prioridad: Alta.
- Al cliente no le resulta atractiva la interfaz de usuario
 - Prioridad: Alta.
- El cliente decide cambiar por completo la interfaz de usuario
 - Prioridad: Alta.

5. Chapado

- Abandono del proyecto
 - Prioridad: Media.

6. Continua corriente de cambios en los requisitos

- El cliente cambia de opinión acerca de lo que debe hacer el proyecto
 - Prioridad: Media.
- El cliente no sabe qué espera que haga el producto
 - Prioridad: Alta.
- Los distintos clientes aportan visiones muy distintas del producto
 - Prioridad: Alta.

7. Deficiencias en componentes proporcionados externamente

- Las librerías de java no son eficientes para nuestro proyecto
 - Prioridad: Baja.
- Los programas proporcionados son muy difíciles de usar y poco efectivos
 - Prioridad: Alta.
- Los recursos son proporcionados demasiado tarde
 - Prioridad: Alta.

8. Deficiencias en tareas desarrolladas externamente

- Poco tiempo para realizar correcciones
 - Prioridad: Alta.
- Poco tiempo para asimilar los pasos a seguir
 - Prioridad: Intolerable.

9. Deficiencias en rendimiento en tiempo real

- Falta de recursos para realizar el proyecto
 - Prioridad: Media.
- Nuestro producto no cumple con los requisitos de rendimiento
 - Prioridad: Alta.
- Nuestro producto no garantiza la calidad de uso
 - Prioridad: Alta.

10. Exprimir las capacidades informáticas

- Falta de conocimiento por parte de los componentes del equipo
 - **Prioridad:** Intolerable.

11. Prioridad de los riesgos

En esta sección se ordenan los riesgos según su prioridad y se eligen aquellos riesgos que van a ser tratados.

Riesgos con prioridad Intolerable
1. Baja temporal de algún miembro del equipo por enfermedad.
2. El cliente rechaza las funciones que hemos desarrollado.
3. Los programas desarrollados son muy difíciles de usar y poco efectivos.
4. Poco tiempo para asimilar los pasos a seguir.
5. Falta de conocimiento por parte de los componentes del equipo.

Riesgos con prioridad Alta
6. Retraso en las entregas por mala planificación.
7. El producto no se ajusta a lo que el cliente necesita por falta de comunicación.
8. No entregar todo lo acordado en la planificación por falta de tiempo.
9. El cliente no sabe qué funciones debe desarrollar el producto.
10. Los programas proporcionados son muy difíciles de usar y poco efectivos.
11. La interfaz de usuario es demasiado difícil de construir.
12. El cliente considera que la interfaz es difícil de usar.
13. Al cliente no le resulta atractiva la interfaz de usuario.
14. El cliente decide cambiar por completo la interfaz de usuario.
15. El cliente no sabe qué espera que haga el producto.
16. Los recursos son proporcionados demasiado tarde.
17. Poco tiempo para realizar correcciones.
18. Nuestro producto no cumple con los requisitos de rendimiento.
19. Nuestro producto no garantiza la calidad de uso.

Riesgos con prioridad Media
20. Baja definitiva de algún miembro del equipo por enfermedad.
21. Abandono de la asignatura por parte de algún miembro del equipo.
22. Abandono del proyecto por algún miembro del equipo.
23. Abandono de la carrera por algún miembro del equipo.
24. Las funciones son ineficientes.
25. Poca calidad de las funciones y propiedades realizadas.
26. Las distintas partes del proyecto no cumplen con su cometido
27. Dificultad para hacer que las distintas funciones del proyecto se coordinen entre ellas.
28. El producto no funciona debidamente en la plataforma en que se quiere usar.
29. Falta de recursos para el desarrollo de la interfaz.
30. El cliente cambia de opinión acerca de lo que debe hacer el proyecto.
31. Falta de recursos para realizar el proyecto

Riesgos con prioridad Baja
32. Baja del supervisor del proyecto.
33. El lenguaje no permite realizar todas las funciones.
34. Las librerías de java no son eficientes para nuestro proyecto.

Por lo tanto, los riesgos que serán tratados son los siguientes:

- Baja temporal de algún miembro del equipo por enfermedad.
- El cliente rechaza las funciones que hemos desarrollado.
- Los programas desarrollados son muy difíciles de usar y poco efectivos.
- Poca tiempo para asimilar los pasos a seguir.
- Falta de conocimiento por parte de los componentes del equipo.
- Retraso en las entregas por mala planificación.
- El producto no se ajusta a lo que el cliente necesita por falta de comunicación.

Parte IV

Gestión del riesgo

En esta parte se tratarán los riesgos que han sido elegidos por su prioridad en la sección anterior.

Para ello, se analizarán diversos aspectos, que se muestran a continuación.

- **Indicios de que se produzca:** Hechos que nos pueden llevar a pensar que es posible que ocurra el riesgo.
- **Mitigación:** En este momento el riesgo ya se ha producido. Aquí se describen las acciones que se deben llevar a cabo para minimizar las consecuencias, desde el momento en el que se produce el riesgo hasta el momento en el que se pone en marcha la acción de contingencia.
- **Acción de contingencia:** Acciones que se deben llevar a cabo una vez que el riesgo haya ocurrido, para que pueda ser resuelto.
- **Prevención:** Acciones que se deben realizar para evitar que el riesgo se produzca.

A continuación se exponen todos los riesgos:

1. Baja temporal de algún miembro del equipo por enfermedad

- **Indicios de que se produzca:**
 - Periodos de gripe.
 - Los miembros del equipo se encuentran cansados por sobrecarga de trabajo.
 - Desmotivación en el trabajo, cansancio frecuente.
- **Mitigación:**
 - Repartir el trabajo de forma equitativa entre los restantes miembros del grupo.
 - En caso de que esté de baja por una enfermedad contagiosa, comprobar que el resto de el equipo no ha sido contagiado.
- **Acción de contingencia:**

- Suplir la baja contratando a otro trabajador.

■ **Prevención:**

- La enfermedad puede ser imprevisible.
- No sobrecargar a los trabajadores. Tener cuidado en temporadas de gripe.

2. El cliente rechaza las funciones que hemos desarrollado

■ **Indicios de que se produzca:**

- El cliente discute todas las propuestas que le ofrecemos y ninguna de nuestras posibles soluciones le parece adecuada.

■ **Acción de contingencia:**

- Reunirnos de nuevo con el cliente, analizar lo que nos pide y llegar a un acuerdo para hacer una síntesis entre las funciones que ya hemos desarrollado y lo que el cliente pide.

■ **Prevencion:**

- Hacer más reuniones con el cliente para conocer mejor su idea, es decir, qué quiere y cómo lo quiere, para conocer todos los detalles que debe cumplir el producto.
- Una vez conocidos los detalles a fondo comenzar algunas funciones y a medida que se van desarrollando mostrárselas al cliente para que de su opinión y así realizar un trabajo que sea de su agrado.

■ **Mitigacion:**

- Modificar las funciones hasta que sean del gusto del cliente.

3. Los programas desarrollados son muy difíciles de usar y poco efectivos

■ **Indicios de que se produzca:**

- Hay poca visibilidad del producto.

4. *POCO TIEMPO PARA ASIMILAR LOS PASOS A SEGUIR* **Gestión de riesgos**

- Los requisitos son poco claros.
- El equipo no tiene claro lo que hay que hacer.
- El equipo no sabe qué es exactamente lo que el cliente desea.
- No se sabe qué pasos hay que seguir debido a una mala planificación y un mal diseño.
- **Mitigación:**
 - Hablar con el cliente para saber dónde está el problema.
 - Probar, monitorizar y depurar el programa, para ver qué partes son problemáticas y qué partes son poco efectivas.
 -
- **Acción de contingencia:**
 - Rehacer las partes problemáticas.
 - Comprobar que los problemas se han resuelto.
 - Entregar al cliente la nueva versión del programa.
- **Prevención:**
 - Elaborar prototipos para que el programa pueda ser probado y corregido.
 - Mejorar la planificación, aumentar la claridad de los requisitos.
 - Hablar más con el cliente, para tener claro qué es lo que él quiere.

4. **Poco tiempo para asimilar los pasos a seguir**

- **Indicios de que se produzca:** Vamos retrasados en las clases y el profesor tiene que explicar más rápido, o se aproxima una época de mucho trabajo y exámenes por lo que no podemos dedicar tanto tiempo a la asignatura y al proyecto.
- **Acción de contingencia:**
 - Repartir bien el trabajo.
 - Marcarnos fechas para realizar cada parte muy cortas y ser muy estrictos.

- Para ello debemos dedicar más tiempo al proyecto. De este modo llevaremos una información muy actualizada de que debemos llevar hecho, cómo hacerlo y todo el grupo conocerá el estado del proyecto.
- También podríamos dedicar un tiempo a ayudarnos a comprender mejor lo que debemos saber de la asignatura.

■ **Prevención:**

- Llevar las clases de Ingeniería del Software al día, incluso estudiar por adelantado.
- Comprender todo antes de comenzar con el proyecto y dedicarnos a éste desde el principio, cuidando no dejarlo todo para cuando se aproxime la fecha de entrega.

- **Mitigación:** Dedicar nuestro tiempo únicamente a estudiar la asignatura y a realizar el proyecto.

5. Falta de conocimiento por parte de los componentes del equipo

■ **Indicios de que se produzca:**

- Los miembros del equipo no saben qué hacer.
- Se hacen avances torpes e inseguros.
- Poca organización.
- En determinados momentos parece que el proyecto se va a detener.

- **Mitigación:** Recordemos que si el riesgo ocurre, esto nos llevaría a un empeoramiento de la calidad del producto, aumento de la dificultad del desarrollo y retrasos en las entregas.

- Contratar o conseguir la colaboración de alguien con experiencia.
- Analizar qué aspectos son los desconocidos.
- Corrección de errores por fallos de concepto.
- Reorganizar las partes incorrectas del proyecto, esta vez teniendo conocimiento de cómo se deben de hacer las cosas.
- Hablar con el cliente para acordar nuevas fechas.

■ **Acción de contingencia:**

- Rehacer las partes incorrectas del proyecto.
- Mejorar la calidad del producto.

■ **Prevención:**

- Tener claros los conceptos de la IS.
- En caso de duda, consultar con gente con experiencia, para saber cómo actuar.
- En caso de que el equipo se encuentre desorientado o algún miembro del equipo no sepa qué hacer, pedir ayuda y consultar a alguien con experiencia en la IS.

6. Retraso en las entregas por mala planificación

■ **Indicios de que se produzca:**

- Conforme avanza el proyecto la relación entre esfuerzo real y planificación no se cumple, por tanto se necesita más tiempo para realizar la misma tarea. Esto es lo que lleva a los retrasos.

■ **Mitigación:**

- Dedicar más tiempo a la entrega hasta que se acabe, aunque eso implique menor calidad.

■ **Acción de contingencia:**

- Volver a realizar una nueva planificación basándonos en la experiencia de los últimos hechos y con más cuidado que la última vez.
- Para cumplir con las entregas también será necesario aumentar la cantidad de horas dedicadas.

- **Prevención:** Realizar una planificación más extensa que indique en qué momento debemos tener cada parte, sin necesidad de que exista una entrega.

7. El producto no se ajusta a lo que el cliente necesita por falta de comunicación

■ **Indicios de que se produzca:**

- En determinadas ocasiones, el equipo se encuentra desorientado acerca de lo que tiene que hacer.
- Existen diferentes posturas dentro del equipo acerca de lo que se tiene que hacer.
- Hay poca comunicación con el cliente.
- En las reuniones con el cliente no se saca nada en claro. Hay ambigüedades.

■ **Mitigación:**

- Detectar qué partes del programa son las que fallan.
- Hablar con el cliente para saber por qué el programa no se ajusta a sus necesidades.
- Estudiar a fondo aquellas características que no son necesarias en el programa, y aquellas que lo son y que no están presentes.

■ **Acción de contingencia:**

- Mostrar al cliente qué características se cambiarán, para obtener el visto bueno.
- Rehacer el programa, teniendo en cuenta la opinión del cliente, y haciendo los cambios necesarios.

■ **Prevención:**

- Evitar ambigüedades en la especificación.
- Al hablar con el cliente, dejar claras todas las características que ha de tener el programa.

