

LISTA DE COMPROBACIÓN DE RIESGOS EN PROYECTOS SOFTWARE

Esta lista agrupa los riesgos de proyectos software en las siguientes categorías:

- A. Elaboración de la Planificación
- B. Organización y Gestión
- C. Ambiente/Infraestructura de Desarrollo
- D. Usuarios finales
- E. Cliente
- F. Personal Contratado
- G. Requisitos
- H. Producto
- I. Fuerzas Mayores
- J. Personal
- K. Diseño e Implementación
- L. Proceso

A. Elaboración de la Planificación

- A.1. Las definiciones de la planificación, de los recursos y del producto han sido impuestas por el cliente o un directivo superior, y no están equilibradas.
- A.2. Planificación optimista, «mejor caso» (en lugar de realista, «caso esperado»).
- A.3. La planificación no incluye tareas necesarias.
- A.4. La planificación se ha basado en la utilización de personas específicas de un equipo, pero estas personas no están disponibles.
- A.5. No se puede construir un producto de tal envergadura en el tiempo asignado.
- A.6. El producto es más grande que el estimado (en líneas de código, en el número de puntos función, o en relación con el tamaño del proyecto anterior).
- A.7. El esfuerzo es mayor que el estimado (por líneas de código, número de puntos función, módulos, etc.).
- A.8. La reestimación debida a un retraso en la planificación es demasiado optimista o ignora la historia del proyecto.
- A.9. La presión excesiva en la planificación reduce la productividad.
- A.10. La fecha final ha cambiado sin ajustarse al ámbito del producto o a los recursos disponibles.
- A.11. Un retraso en una tarea produce retrasos en cascada en las tareas dependientes.
- A.12. Las áreas desconocidas del producto llevan más tiempo del esperado en el diseño y en la implementación.

B. Organización y Gestión

- B.1. El proyecto carece de un promotor efectivo en los superiores.
- B.2. El proyecto languidece demasiado en el inicio difuso.
- B.3. Los despidos y las reducciones de la plantilla reducen la capacidad del equipo.
- B.4. Dirección o marketing insisten en tomar decisiones técnicas que alargan la planificación.
- B.5. La estructura inadecuada de un equipo reduce la productividad.
- B.6. El ciclo de revisión/decisión de la directiva es más lento de lo esperado.
- B.7. El presupuesto varía el plan del proyecto.
- B.8. La dirección toma decisiones que reducen la motivación del equipo de desarrollo.

- B.9. Las tareas no técnicas encargadas a terceros necesitan más tiempo del esperado (aprobación del presupuesto, aprobación de la adquisición de material, revisiones legales, seguridad, etc.).
- B.10. La planificación es demasiado mala para ajustarse a la velocidad de desarrollo deseada.
- B.11. Los planes del proyecto se abandonan por la presión, llevando al caos y a un desarrollo ineficiente.
- B.12. La dirección pone más énfasis en las heroicidades que en informarse exactamente del estado, lo que reduce su habilidad para detectar y corregir problemas.

C. Ambiente/Infraestructura de Desarrollo

- C.1. Los espacios no están disponibles en el momento necesario.
- C.2. Los espacios están disponibles pero no son adecuados (por ejemplo, falta de teléfonos, cableado de la red, mobiliario, material de oficina, etc.).
- C.3. Los espacios están sobreutilizados, son ruidosos o distraen.
- C.4. Las herramientas de desarrollo no están disponibles en el momento deseado.
- C.5. Las herramientas de desarrollo no funcionan como se esperaba; el personal de desarrollo necesita tiempo para resolverlo o adaptarse a las nuevas herramientas.
- C.6. Las herramientas de desarrollo no se han elegido en función de sus características técnicas, y no proporcionan las prestaciones previstas.
- C.7. La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado.

D. Usuarios Finales

- D.1. Los usuarios finales insisten en nuevos requisitos.
- D.2. En el último momento, a los usuarios finales no les gusta el producto, por lo que hay que volver a diseñarlo y a construirlo.
- D.3. Los usuarios no han realizado la compra del material necesario para el proyecto y, por tanto, no tienen la infraestructura necesaria.
- D.4. No se ha solicitado información al usuario, por lo que el producto al final no se ajusta a las necesidades del usuario, y hay que volver a crear el producto.

E. Cliente

- E.1. El cliente insiste en nuevos requisitos.
- E.2. Los ciclos de revisión/decisión del cliente para los planes, prototipos y especificaciones son más lentos de lo esperado.
- E.3. El cliente no participa en los ciclos de revisión de los planes, prototipos y especificaciones, o es incapaz de hacerlo, resultando unos requisitos inestables y la necesidad de realizar unos cambios que consumen tiempo.
- E.4. El tiempo de comunicación del cliente (por ejemplo, tiempo para responder a las preguntas para aclarar los requisitos) es más lento del esperado.
- E.5. El cliente insiste en las decisiones técnicas que alargan la planificación.
- E.6. El cliente intenta controlar el proceso de desarrollo, con lo que el progreso es más lento de lo esperado.
- E.7. Los componentes suministrados por el cliente no son adecuados para el producto que se está desarrollando, por lo que se tiene que hacer un trabajo extra de diseño e integración.
- E.8. Los componentes suministrados por el cliente tienen poca calidad, por lo que tienen que hacerse trabajos extra de comprobación, diseño e integración.
- E.9. Las herramientas de soporte y entornos impuestos por el cliente son incompatibles, tienen un bajo rendimiento o no funcionan de forma adecuada, con lo que se reduce la productividad.

- E.10. El cliente no acepta el software entregado, incluso aunque cumpla todas sus especificaciones.
- E.11. El cliente piensa en una velocidad de desarrollo que el personal de desarrollo no puede alcanzar.

F. Personal Contratado

- F.1. El personal contratado no suministra los componentes en el período establecido.
- F.2. El personal contratado proporciona material de una calidad inaceptable, por lo que hay que añadir un tiempo extra para mejorar la calidad.
- F.3. Los proveedores no se integran en el proyecto, con lo que no se alcanza el nivel de rendimiento que se necesita.

G. Requisitos

- G.1. Los requisitos se han adaptado, pero continúan cambiando.
- G.2. Los requisitos no se han definido correctamente. y su redefinición aumenta el ámbito del proyecto.
- G.3. Se añaden requisitos extra.
- G.4. Las partes del proyecto que se no se han especificado claramente consumen más tiempo del esperado.

H. Producto

- H.1. Los módulos propensos a tener errores necesitan más trabajo de comprobación, diseño e implementación.
- H.2. Una calidad no aceptable requiere de un trabajo de comprobación, diseño e implementación superior al esperado.
- H.3. Utilizar lo último en informática alarga la planificación de forma impredecible.
- H.4. El desarrollo de funciones software erróneas requiere volver a diseñarlas y a implementarlas.
- H.5. El desarrollo de una interfaz de usuario inadecuada requiere volver a diseñarla y a implementarla.
- H.6. El desarrollo de funciones software innecesarias alarga la planificación.
- H.7. Alcanzar el ámbito del producto o las restricciones de velocidad requiere más tiempo del esperado, incluyendo el tiempo para volver a diseñar e implementar.
- H.8. Unos requisitos rígidos de compatibilidad con el sistema existente necesitan un trabajo extra de comprobación, diseño e implementación.
- H.9. Los requisitos para crear interfaces con otros sistemas, otros sistemas complejos, u otros sistemas que no están bajo el control del equipo de desarrollo suponen un diseño, implementación y prueba no previstos.
- H.10. El requisito de trabajar con varios sistemas operativos necesita más tiempo del esperado.
- H.11. El trabajo con un entorno software desconocido causa problemas no previstos.
- H.12. El trabajo con un entorno hardware desconocido causa problemas imprevistos.
- H.13. El desarrollo de un tipo de componente nuevo para la organización consume más tiempo del esperado.
- H.14. Depender de una tecnología que aún está en fase de desarrollo alarga la planificación.

I. Fuerzas mayores

- I.1. El producto depende de las normativas del gobierno, que pueden cambiar de forma inesperada.
- I.2. El producto depende de estándares técnicos provisionales, que pueden cambiar de forma inesperada.

J. Personal

- J.1. La contratación tarda más de lo esperado.
- J.2. Las tareas preliminares (por ejemplo, formación, finalización de otros proyectos, adquisición de licencias) no se han completado a tiempo.
- J.3. La falta de relaciones entre la dirección y el equipo de desarrollo ralentiza la toma de decisiones.
- J.4. Los miembros del equipo no se implican en el proyecto, y por lo tanto no alcanzan el nivel de rendimiento deseado.
- J.5. La falta de motivación y de moral reduce la productividad.
- J.6. La falta de la especialización necesaria aumenta los defectos y la necesidad de repetir el trabajo.
- J.7. El personal necesita un tiempo extra para acostumbrarse a trabajar con herramientas o entornos nuevos.
- J.8. El personal necesita un tiempo extra para acostumbrarse a trabajar con hardware nuevo.
- J.9. El personal necesita un tiempo extra para aprender un lenguaje de programación nuevo.
- J.10. El personal contratado abandona el proyecto antes de su finalización.
- J.11. Alguien de la plantilla abandona el proyecto antes de su finalización.
- J.12. La incorporación de nuevo personal de desarrollo al proyecto ya avanzado, y el aprendizaje y comunicaciones extra imprevistas reducen la eficiencia de los miembros del equipo existentes.
- J.13. Los miembros del equipo no trabajan bien juntos.
- J.14. Los conflictos entre los miembros del equipo conducen a problemas en la comunicación y en el diseño, errores en la interfaz y tener que repetir algunos trabajos.
- J.15. Los miembros problemáticos de un equipo no son apartados, influyendo negativamente en la motivación del resto del equipo.
- J.16. Las personas más apropiadas para trabajar en el proyecto no están disponibles.
- J.17. Las personas más apropiadas para trabajar en el proyecto están disponibles, pero no se pueden incorporar por razones políticas o de otro tipo.
- J.18. Se necesitan personas para el proyecto con habilidades muy específicas y no se encuentran.
- J.19. Las personas clave sólo están disponibles una parte del tiempo.
- J.20. No hay suficiente personal disponible para el proyecto.
- J.21. Las tareas asignadas al personal no se ajustan a sus posibilidades.
- J.22. El personal trabaja más lento de lo esperado.
- J.23. El sabotaje por parte de la dirección del proyecto deriva en una planificación ineficiente e inefectiva.
- J.24. El sabotaje por parte del personal técnico deriva en una pérdida de trabajo o en un trabajo de poca calidad, por lo que hay que repetir algunos trabajos.

K. Diseño e Implementación

- K.1. Un diseño demasiado sencillo no cubre las cuestiones principales, con lo que hay que volver a diseñar e implementar.
- K.2. Un diseño demasiado complejo exige tener en cuenta complicaciones innecesarias e improductivas en la implementación.
- K.3. Un mal diseño implica volver a diseñar e implementar.
- K.4. La utilización de metodologías desconocidas deriva en un periodo extra de formación y tener que volver atrás para corregir los errores iniciales cometidos en la metodología.
- K.5. El producto está implementado en un lenguaje de bajo nivel (por ejemplo, ensamblador) y la productividad es menor de la esperada.
- K.6. No se puede implementar la funcionalidad deseada con el lenguaje o bibliotecas utilizados: el personal de desarrollo tiene que utilizar otras bibliotecas, o crearlas él mismo para conseguir la funcionalidad deseada.
- K.7. Las bibliotecas de código o clases tienen poca calidad, y generan una comprobación extra, corrección de errores y la repetición de algunos trabajos.
- K.8. Se ha sobreestimado el ahorro en la planificación derivado del uso de herramientas para mejorar la productividad.
- K.9. Los componentes desarrollados por separado no se pueden integrar de forma sencilla, teniendo que volver a diseñar y repetir algunos trabajos.

L. Proceso

- L.1. La burocracia produce un progreso más lento del esperado.
- L.2. La falta de un seguimiento exacto del progreso hace que se desconozca que el proyecto esté retrasado hasta que está muy avanzado.
- L.3. Las actividades iniciales de control de calidad son recortadas, haciendo que se tenga que repetir el trabajo.
- L.4. Un control de calidad inadecuado hace que los problemas de calidad que afectan a la planificación se conozcan tarde.
- L.5. La falta de rigor (ignorar los fundamentos y estándares del desarrollo de software) conduce a fallos de comunicación, problemas de calidad y repetición del trabajo. Un consumo de tiempo innecesario.
- L.6. El exceso de rigor (aferramiento burocrático a las políticas y estándares de software) lleva a gastar más tiempo en gestión del necesario.
- L.7. La creación de informes de estado a nivel de directiva lleva más tiempo al desarrollador de lo esperado.
- L.8. La falta de entusiasmo en la gestión de riesgos impide detectar los riesgos más importantes del proyecto.
- L.9. La gestión de riesgos del proyecto software consume más tiempo del esperado.

Fuentes:

- Principles of Software Engineering Management (Gilb, 1998).
- Software Risk Management (Boehm, 1989).
- A Manager's Guide to Software Engineering (Pressman, 1993).
- Third Wave Project Management (Thomsett, 1993).
- Assessment and Control of Software Risks (Jones, 1994).
- Connell, S. Desarrollo y Gestión de Proyectos Informáticos. (McGraw-Hill Iberoamericana, 1997).