

Actividad 2: Análisis de Riesgo

Kalio Fernando O'Farril Villalpando

Dirección y Gestión de Proyectos de Software

Grupo 1001

Análisis

R1 (Riesgo de adecuación del código desarrollado a los requisitos por excesiva complejidad algorítmica):

En diversas ocasiones he visto de primera mano como un algoritmo relativamente sencillo puede presentar dificultad a el equipo de desarrollo. Sumando a lo recién mencionado, también hay diversos casos de uso y situaciones marginales pero posibles que deben ser meticulosamente probadas para asegurar el correcto funcionamiento. Por el tiempo de entender los algoritmos, desarrollar la lógica de negocio, crear casos de prueba meticulosos y ejecutarlos, el riesgo R1 tiene una alta probabilidad de generar retrasos, lo que conlleva a un muy alto impacto ya que se nos ha dicho que si la aplicación no es entregada en tiempo la posible falta de interés del cliente es alta. Cabe añadir a lo ya dicho la poca disponibilidad de los expertos en finanzas para dudas con respecto a la lógica de negocio, lo que hace aún más el nivel de riesgo para este caso.

Este riesgo puede verse como un riesgo de calidad debido a que el software entregado no cumplirá con los requisitos impuestos inicialmente. Esto conlleva al riesgo de tiempo que conlleva un riesgo de coste y por la situación planteada puede llevar al fracaso total del proyecto.

Podemos identificar dos umbrales de riesgo:

- Una historia de usuario no es completada en un Sprint.
- Se repite la situación anterior.

Como acción preventiva, al ver que un recurso de desarrollo está presentando dificultades y expone sus preocupaciones en el daily scrum, podemos redirigir el esfuerzo de otros miembros del equipo (desarrolladores, stakeholders, etc...) a apoyar el recurso para evitar retrasos.

La acción correctiva en este caso es similar, pero incrementando el énfasis. Quizás dejando de lado aspectos no funcionales de la aplicación como podrían ser manuales, estéticas, etc... para dirigir la atención de todo el equipo en apoyar a terminar el desarrollo de la crucial lógica de negocio.

R2: Riesgo de calendario y temporización

Por el alto énfasis que ha hecho el cliente en la fecha de entrega, el riesgo de calendario y temporización es un riesgo de muy alto impacto. Considerando que ninguno de los miembros del equipo de desarrollo estaban presentes durante el desarrollo de la aplicación original de legado, es seguro que las dudas van a surgir. El hecho de tener poca disponibilidad del equipo del negocio como expertos del negocio también es algo que puede llevar al desarrollo a un alto durante días en lo que se tiene la oportunidad de dialogar con éstos para aclarar las dudas.

Este es un riesgo de cronograma, que al igual que el riesgo anterior es un riesgo de tiempo que conlleva un riesgo de costes y a el fracaso del proyecto.

Los umbrales para este riesgo son:

- La falta de disponibilidad de los expertos.
- El retraso continuo de entregables.
- El incumplimiento con la fecha de entrega.

Para una acción preventiva se puede considerar aprovechar las primeras semanas del proyecto durante las cuales se tendrá la mayor disponibilidad de los expertos de finanzas para repasar y aclarar todas las dudas en cuanto a lógica de negocio se refiere. Se deberá hacer énfasis con el equipo de desarrollo para que comenten en el daily Scrum sus avances y si ven riesgos o futuros retrasos en cualquier entregable del proyecto.

Como acción correctiva se puede, al igual que en el riesgo R1, dejar de lado las funcionalidades menos cruciales para enfocarse en entregar un producto que satisface los requerimientos importantes para el negocio. De igual forma, no se deberá tocar el dinero de holguera durante todo el proyecto para hacer uso de algún recurso externo en caso de que la fecha de término sea próxima y aún sean necesarios varios requerimientos. Por último, si el retraso se ve inminente, será apropiado tener una discusión con el cliente y negociar para aceptar un ligero de retraso en la fecha de entrega.

R3: No disponibilidad de un experto en finanzas en caso de que surja un problema de envergadura.

Las dudas en cuanto a lógica de negocio son importantes temas a tratar y pueden llegar a convertirse en riesgos de alto impacto si no se solucionan rápidamente ya que ésto puede ocasionar un alto total en el desarrollo de la aplicación hasta no tener las respuestas concretas. De igual manera puede llevar a retrabajos que no solo consumen tiempo sino que bajan la moral del equipo y pueden dar pauta a discusiones y futuras disfunciones entre los miembros del mismo. Es por eso que la no disponibilidad de un experto de finanzas es un riesgo importante a ser considerado y es necesario tener planes preventivos y de contingencia.

Este es un riesgo de recursos humanos que puede desencadenar riesgos de calidad, tiempo y costes.

Para este riesgo se han identificado los siguientes umbrales de riesgo:

- Quedarán dudas en la lógica de negocio después de las primeras semanas en las que teníamos disponibilidad de más de un experto en finanzas.
- Los entregables se han detenido o retrabajado por falta de experiencia en el área del negocio.
- La disponibilidad del experto que estará para todo el proyecto no cubre las necesidades del equipo.

Las acciones preventivas para este riesgo involucran tener sesiones iniciales extensas durante el inicio del proyecto con los expertos en finanzas para tener claras las normas del negocio así como aclarar las posibles dudas que se creen se pueden llegar a tener. Si el tiempo apremia, tener diagramas de flujo de negocio puede ser ilustrativo.

Como acciones correctivas se puede hablar con los clientes y hacerles ver que la poca disponibilidad atrasará indudablemente el proyecto y es necesario que se negocien nuevos horarios de disponibilidad de los expertos.

R4: Problemas en la adecuación del diseño de la API debido a que el software que se va a desarrollar tiene que ser reutilizable en contextos muy diferentes.

La versatilidad y reutilización del código es un importante requerimiento en todos los proyectos modernos. Afortunadamente existen diversas librerías y tecnologías que hacen de ésta una tarea más sencilla, haciendo de esto un riesgo de baja probabilidad. Para evitar retrabajos y conflictos es importante que los múltiples requerimientos y los diversos casos de uso para una funcionalidad queden aclarados desde el principio y se comience el desarrollo teniendo en cuenta ésto.

Este es un riesgo tanto técnico como de integración, ya que además de tener una arquitectura que soporte la reutilización con ligeras modificaciones necesitamos tecnologías que nos faciliten este labor.

Los umbrales de riesgo son:

- Se encuentra que un endpoint del API no es flexible para los distintos contextos.
- Se ha tenido que hacer más de un retrabajo sobre API's existentes para garantizar la flexibilidad deseada.

Como acción preventiva se tiene que dar énfasis en las etapas tempranas donde se hará una arquitectura preparada para soportar distintos contextos en el desarrollo. Se necesita dejar en claro desde un principio cuales son los diferentes caminos y aplicaciones que tendrá un API.

La acción correctiva consta en que el equipo de desarrollo se junte e identifique que es lo que esta ocasionando retrabajos o dudas y se aclare una nueva arquitectura que disminuirá las dudas o fallas en el diseño del API.

Conclusiones

Haciendo uso del análisis de los riesgos expuestos previamente, se pueden clasificar los riesgos mediante su impacto y probabilidad. Con ello podemos fabricar la siguiente Matriz de Impacto:

Matriz de Riesgos			
Riesgos	Probabilidad	Impacto	Nivel de Riesgo
R1	0.9	0.9	0.81
R2	0.7	0.9	0.63
R3	0.7	0.5	0.35
R4	0.3	0.3	0.09

En donde el color de cada riesgo representa en donde se encuentra con respecto a la Matriz de Riesgos mostrada a continuación:

			Impacto				
			Muy Bajo	Bajo	Medio	Alto	Muy Alto
			0.1	0.2	0.3	0.4 - 0.7	0.8 - 1.0
Ocurrencia	Frecuente	0.9 - 1.0					R1
	Probable	0.7 - 0.8				R3	R2
	Ocasional	0.5 - 0.6					
	Posible	0.3 - 0.4			R4		
	Improbable	0.1 - 0.2					