

Actividad | #3| Análisis de riesgo

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo Garcia

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 04 de febrero del 2024

Contenido

Introducción.	3
Descripción.	4
Justificación.	5
Desarrollo.....	6
Conclusión.	12

Introducción.

A partir de las 2 actividades anteriores donde realizamos el plan de trabajo, así como la metodología que elegimos para la realizar la calendarización del proyecto y los requerimientos de este mismo, así como asignar las tareas o sprints a los diferentes equipos que conformamos, así como tiempos estimados de entrega antes esperados, esto sin incluir los riesgos que revisaremos mas adelante ya que pueden alargar los tiempos de entrega así como parar la producción del software por completo, por lo que veremos un análisis de riesgo con base a los requerimientos solicitados en la actividad 2 por lo que veremos el riesgo que tienen cada una de las actividades o del mismo personal por lo que se le asignara una probabilidad de riesgo así como el impacto de este, si es que afecta de una manera critica, seria o tolerable del riesgo por lo que realizaremos una estrategia de reducción de riesgos así como su supervisión y gestión de estos, veremos si los riesgos son tecnológicos, así como riesgos humanos.

Descripcion.

Como vimos en la actividad anterior, realizamos diagramas de Gantt ya que mostramos a quien le asignamos las actividades correspondientes a diferentes equipos en nuestro entorno de trabajo de Jira por lo que mostramos fechas de inicio y fin para cada una de las actividades, por lo que en esta actividad revisaremos las actividades anteriores para poder detectar factores de riesgo tales como tecnológicos y humanos dado el caso de que alguna parte del sistema en desarrollo o módulos no se pueda integrar de manera correcta al sistema principal por lo que es un riesgo tecnológico dado a que se puede realizar en lenguajes de código diferentes al que esta realizado el sistema principal o que algun miembro del equipo encargado de tareas criticas e importantes como las bases de datos o el mismo funcionamiento del sistema principal haya tenido una enfermedad y no se presente y no realice su parte de la tarea ese mismo día por lo que debemos pensar bien que probabilidad hay de que pase dicho suceso así como proporcionarle un grado de riesgo, ya que hay contratiempos que son tolerables, serios que se pueden controlar así como los críticos en caso de que sucedan, por lo que tendremos que tener un plan para anularlos y contenerlos.

Justificación.

Realizaremos el análisis de riesgo ya que es necesario saber poder identificar los riesgos a los que nos sometemos al momento de realizare el software así como empezaremos desde los requerimientos identificando cual es la probabilidad de que ocurra un factor de riesgo en esta etapa tal que el cliente olvide decir una función del sistema que necesita y no lo dijo en el momento, cual es el riesgo de que el sistema falle, cual es el riesgo de que el personal pueda retrasar la entrega de las actividades así como los posibles factores que debemos tener en cuenta para este análisis de riesgo para poder garantizar el cumplimiento de todos los objetivos del proyecto así poder planear medidas de contención para cada uno de los problemas que identifiquemos en este análisis de riesgo, por lo que en base a este análisis podremos reducir costos del proyecto así como poder adaptarnos a los cambios que lleguen a suceder y presentar al cliente su producto de una manera adecuada revisando el código así como solucionar todos los problemas que tengamos durante el transcurso del proyecto

Desarrollo.

Realizaremos un análisis de riesgo teniendo en cuenta los siguientes elementos

- Riesgo
- Probabilidad
- Impacto
- Estrategia de Reducción
- Supervisión

por lo que es necesario tener claras las categorías que estaremos analizando tales como los riesgos tecnológicos, riesgos en el personal, riesgos en la organización, riesgos en las herramientas que utilizaremos, así como los requerimientos que realizamos y la estimación el tiempo requerido para la entrega de las actividades del proyecto, por lo que definiremos las categorías, riesgos, probabilidades e impactos que tienen los riesgos, así como la prioridad de estos en la siguiente tabla

Categoría	Riesgo	Probabilidad	Impacto	Prioridad
Tecnológico	Problemas de integración de los módulos al sistema principal	Media 25-50%	Alto	Alta
Solución	Realizar revisiones antes de la integración, así como verificar que el módulo funciona correctamente, así como verificar que el módulo y el sistema estén hechos en el mismo lenguaje de programación			
Tecnológico	Mala optimización, lentitud del sistema	Media 25%	Alto	Alta
Solución	Verificar que el código no tenga líneas innecesarias, así como verificar que el hardware es el correcto para la ejecución del programa, así como revisar que no			

	haya interferencias entre los módulos			
Tecnológica	Protección de datos del cliente y usuarios	Media 35%	Medio	Alto
Solución	Implementar la encriptación de datos críticos, así como datos biométricos para iniciar sesión			
Tecnológico	El sistema no reconoce lectores de huella o escáner de códigos de barras	Baja 10%	Medio	Medio
Solución	Verificar que el hardware esté conectado correctamente, reiniciar el módulo que opera estas funciones, así como verificar que algún modulo no interfiera en el software			
Tecnológico	Fallas en servidores, bases de datos o redes LAN	Baja 10%	Alto	Medio
Solución	Reiniciar los servidores, así como verificar que no se tenga un ataque DDOS, verificar que las redes estén conectadas correctamente y no tenga interferencia con alguna otra red			
Requerimientos	El cliente requiere nuevos módulos fuera del sprint actual	Alta >50%	Alto	Alta
Solución	Pedirle al cliente con anterioridad todos los detalles de los requerimientos antes de empezar con el desarrollo, así como indicarle que agregar más módulos fuera de lo indicado puede retrasar la entrega así como subir el presupuesto del proyecto			
Requerimientos	Funcionalidades mal definidas	Media 35%	Medio	Medio
Solución	Pedirle al cliente que nos indique para que necesite ese modulo, como lo utilizara, quienes lo utilizaran y como es que debe de funcionar antes del			

	desarrollo			
Requerimientos	El backlog no está ordenado correctamente conforme a prioridades	Medio <35%	Medio	Medio
Solución	Ordenar el backlog conforme a prioridades que nos indique el cliente así priorizando las más altas			
Personal	El equipo carece de habilidades técnicas	Media 35%	Alto	Alta
Solución	Buscar con anterioridad a desarrolladores con las capacidades necesarias, así como tener un equipo definido con anterioridad, así como pedir recomendaciones a estos mismos desarrolladores			
Personal	Rotación de personal (desarrolladores, testers, etc.)	Bajo 15%	Alto	media
Solución	Establecer un salario fijo que sea atractivo, así como establecer el tiempo que quiera trabajar para poder conseguir desarrolladores antes de su renuncia en caso de que lo haga			
Organizacional	Exceso de carga al equipo (demasiados sprints exigentes en poco tiempo)	Medio 35%	Medio	Media
Solución	Establecer tiempos de entrega más largos para los sprints más complejos así como aumentar los desarrolladores que trabajaran en este sprint, en un periodo de tiempo más corto (menos desarrolladores más tiempo)			
Organizacional	El equipo no se integra correctamente	Bajo 10%	Medio	Medio
Solución	Establecer vínculos con el equipo, separar en equipos distintos aquellos colaboradores que tienen diferencias entre ellos y verificar y tratar de solucionar sus diferencias a través del dialogo			

Herramientas	El equipo en el que se realiza el código no funciona correctamente	Bajo 10%	Alto	Medio
Solución	Verificar que no existan bugs en el código, así como revisar que el código este escrito en el mismo lenguaje de programación que establecimos, así como realizar revisiones y depuraciones en el Código			
Estimación	Errores no detectados en módulos críticos	Medio 35%	Alto	Alta
Solución	Realizar revisiones al terminar un módulo, así como pruebas intensivas para comprobar su funcionamiento			
Tecnológico	Bugs después del lanzamiento	Baja >10%	Alto	Alta
Solución	Realizar revisiones para solucionar los bugs detectados, así como establecer una conexión con el cliente indicando que se esta trabajando en la solución			
Organizacional	El cliente no participa en las revisiones de los sprints	Medio>25%	Alto	Alta
Solución	Pedirle al cliente que participe en las revisiones de los sprints ya que es de suma importancia para establecer prioridades, así como cambios que este quiera hacer dado que se le puede olvidar alguna instrucción que no dio en los requerimientos, así como posibles mejoras que pueda aportar			
Estimación	Demoras en la entrega de hardware y servicios de terceros (terminales bancarias)	Medio <25%	Medio	Media
Solución	Pedir con anterioridad las terminales y el hardware necesario para su instalación y guardarlas en bodega para su uso cuando se requiera realizar una prueba, así como tener disponible el material para su instalación			
Estimación	El tiempo requerido para construir el	Medio 35%	Medio	media

	software esta subestimada			
Solución	Establecer tiempos de realización de sprints un poco mas largos para tener un tiempo un poco mas largo para el desarrollo dado a que pueden suceder estos riesgos que pueden atrasar el proyecto, así poder tener el software en tiempo y forma			

Una vez que tengamos los posibles riesgos, procederemos a realizar nuestra estrategia para la reducción de los posibles riesgos que detectamos así como su supervisión y gestión de estos por lo que priorizaremos aquellos riesgos con prioridad alta empezando por los requerimientos ya que uno de ellos es que el cliente requiera módulos que no definió al principio y al utilizar la metodología de scrum, este tendrá un impacto medio-alto ya que agregamos módulos que no estaban planeados por lo que es necesario pedirle al cliente que puede aumentar el presupuesto y el tiempo de entrega se puede retrasar, es importante tener que explicarle al cliente que estas primeras etapas son muy importante por lo que se podría indicar que tiene un lapso de tiempo de algunos días para que tenga una lista con los requerimientos necesarios así como sus funciones bien definidas e indique prioridades para los módulos por lo que es necesario indicarlo correctamente en el backlog y así tener un mayor control sobre lo que pide el cliente, organizar de manera correcta las prioridades en el backlog es necesario para una buena gestión de las tareas en la metodología scrum, por lo que podremos proceder al personal, ya que puede haber algunas disconformidades entre el personal así como no encontrar el personal necesario con las habilidades requeridas para la construcción del sistema así como las disconformidades que pueden tener los colaboradores al momento de relacionarse con el personal, esto podremos solucionarlo de distintas formas ya que podremos hacer diferentes equipos de desarrolladores que previamente se hayan

conocido o trabajado juntos con anterioridad así como poder realizar los trabajos por home office así para evitar que caigan en conflictos al momento de verse uno al otro de manera personal por lo que es necesario tener en cuenta como idear los equipos para evitar esto conflictos internos, de la misma forma podremos resolver 2 problemas que pueden estar ligadas unas a otras que afectarían de manea media-alta a el desarrollo del proyecto ya que al momento de asignar sprints muy demandantes y que necesiten mucho trabajo en un corto periodo de tiempo puede hacer que los desarrolladores tengan tanta carga de estrés que haga que estos mismos bajen su productividad así como la posible rotación del personal haciendo que renuncien del equipo de desarrollo del proyecto por lo que establecer tareas más sencillas o prolongar el tiempo establecido es una buena forma en la que este problema pueda solucionarse, así como algunos problemas tecnológicos ya que puede se que un módulo no se pueda integrar de manera correcta por lo que una buena forma de solucionarlo es realizar pruebas del módulo por separado así como depurar su código en caso de que este tenga un error así como poder integrarlo con sus respectivas pruebas revisando que sea l mismo lenguaje de código utilizado, con este análisis de riesgo se pretende realizar correcciones a diferentes riesgos detectados así como poder tener abiertas las posibilidades de que algún riesgo ocurra y poder tener un plan para su contención.

Presentamos anteriormente una tabla de posibles riesgos que pueden ocurrir, así como su impacto, su probabilidad y una posible solución ya que hay diferentes formas de solucionar un solo problema, ya que para su supervisión se puede utilizar la comunicación con los colaboradores por privado así como en las reuniones diarias haciendo análisis del código escrito y preguntado si tuvieron algún problema al momento del desarrollo, ya que de esta manera podremos detectar posibles riesgos que puedan ocurrir.

Conclusión.

Vemos la importancia que tiene el análisis de riesgo dentro de las metodologías al momento de planear el proyecto no solo para poder ver los posibles riesgos que pueden suceder al momento de desarrollar, ya que el poder ver el entorno completo es de bastante ayuda al momento de definir los riesgos al momento de los requerimientos del cliente ya que es la primera etapa del proyecto por lo que identificar todas las etapas del proyecto así como todos los posibles riesgos que encontraremos durante el proceso del análisis es una ventaja la cual puede hacer que nos enfoquemos en las soluciones dentro de este campo, así poder reducir el impacto dentro del proyecto ya que tendremos una contramedida para poder mitigar estos posibles riesgos ya que algunos son mas comunes que otros, así que se pueden ver diferentes formas en las que un riesgo puede ocurrir y como es que existen diferentes maneras de poder corregirlo por lo que por lo que identificar y explicar de manera correcta un problema hace que nuestro trabajo para encontrarle solución sea más fácil, así como supervisar el equipo de desarrollo correctamente integrándonos a un trabajo en equipo y no de manera individual hacer que el entorno de trabajo sea más amigable y evitar así algunos de los riesgos que pueden demorar el proyecto así como reducir su eficiencia.

Referencias.

Hackemi. (s. f.). *GitHub - Hackemi/ingenieria-en-software-1: Actividades entregadas en PDF*. GitHub. <https://github.com/Hackemi/ingenieria-en-software-1>

Agile Board - jira. (s. f.). JIRA.

<https://edgarcuamea.atlassian.net/jira/software/projects/SCRUM/boards/1/timeline>