

Casos de uso vs Historias de usuario

El primer de cualquier proyecto es definir los requerimientos, la cantidad de requerimientos define también la complejidad del proyecto

El diagrama de casos de uso en el cual se muestra un actor en este caso dice administrador que puede ser una persona un rol Hardware o Software que interactúa con un sistema informático cada elipse significa una funcionalidad computacional y el actor saca provecho o la utiliza, y de alguna manera hay una cierta relación por qué con el enfoque tradicional porque los casos de uso son o corresponden a una especificación de requerimiento muy precisa muy detallada muy paso a paso por lo tanto yo voy a ocupar casos de uso y tener una primera pista o idea voy a ocupar casos de uso cuando los requerimientos estoy seguro por razonablemente seguro que se van a implementar, hacer un buen caso de uso requiere esfuerzo requiere tiempo no es algo excesivamente difícil pero no es algo trivial. Entonces un caso de uso es una forma de escribir funcionalidad que implica que un actor persona Hardware o software interactúa con un sistema informático y ejecuta alguna funcionalidad específica, en cambio una historia de usuario es una funcionalidad computacional resumida en una tarjeta porque una tarjeta porque es una descripción relativamente genérica no tan precisa mucho menos detallada que un caso de uso por lo tanto requiere menos tiempo Y cuando utilizamos historias de usuario Cuando no tenemos la certeza absoluta que esa funcionalidad se vaya a programar, está se adapta mejor a una gestión de proyecto en la cual esté ocurriendo una situación no tan definida no tan clara se está haciendo una exploración suponemos que tal o cual funcionalidad es interesante pero como la vamos a priorizar a partir de las necesidades del negocio a partir de lo que el cliente requiere podría ocurrir perfectamente que no sólo una ni dos sino una buena cantidad de historias de usuarios finalmente no se implementan debido a que nos damos cuenta que hay otras historias que son más valiosas que aportan más valor al programa, ahí tenemos una diferencia si yo invertí varias horas en construir un caso de uso no lo implemento podríamos decir que perdí esas varias horas en cambio con la historia de usuario se invierte menos tiempo Y por cierto no tiene ningún drama ningún problema. Si bien la parte más importante es lo que no está escrito que es la conversación cuando nosotros definimos una historia de usuario, nos juntamos con el Product Owner y la analizamos la discutimos y ahí vemos si aporta y cómo se podría implementar.

Si bien ya está definido que se usa uno u otro pero también se pueden combinar, podríamos partir ocupando historias de usuario hacer esa conversación inicial y después cuando me convenció que la historia de usuario describe una funcionalidad que sí vale la pena implementar entonces la podría describir con casos de uso suponiendo que sea una exigencia definida para la gestión del proyecto a veces ocurre así existen métodos metodologías que explícitamente trabajan casos de uso y nunca está de más y por otro lado no es cierto la historia de usuarios son una herramienta muy flexible que nos permite capturar rápidamente la funcionalidad básica al menos para empezar a avanzar en nuestro proyecto acuérdense que la plantilla que se ocupa con historias de usuarios dice así dice yo como rol necesito funcionalidad para beneficio miren interesante ejemplo yo como vendedor necesito registrar las ventas para cobrar mi comisión a fin de mes rol funcionalidad software eso es lo que estamos definiendo y beneficio que gano yo que gana el que ocupa esta funcionalidad si nos damos cuenta que el aporte que el valor que lo que entrega Es realmente útil entonces Obviamente que lo vamos a implementar

Políglotas

Una pregunta que nos podríamos hacer es porqué existen tantos lenguajes de programación, bueno la respuesta es tan sencillo como que habiendo tantos problemas cada uno requiere diferentes herramientas para resolverlo, y cada lenguaje de programación tiene ciertas funciones y características que lo hacen adecuado para tareas específicas, partiendo de esa premisa, cuál sería una forma efectiva de volverse políglota por así decirlo de los diferentes lenguajes de programación.

Hay que entender que cada lenguaje de programación viene programado con palabras clave para realizar un acción y no siempre es la misma, entonces habría que por así decirlo crear una estructura general para programar. En este caso ya existe una forma para programar que es tan sencilla de comprender y a la vez fácil de pasar a cualquier lenguaje, el pseudocódigo es código en su forma más primitiva y no cuesta mucho pasarlo a un lenguaje de programación entonces:

- Cómo primer paso conocer la estructura del pseudocódigo desde cómo definir variables hasta como crear ciclos.
- Luego conocer las bases para crear objetos y métodos.
- Conocer los tipos de datos, aunque son los mismos prácticamente solo cambian los dígitos que pueden almacenar, así como también los operadores y como es su sintaxis
- Aprenderse las palabras clave equivalentes a las de pseudocódigo cómo imprimir o leer, etc.
- Por último aprenderse las librerías en caso de ser necesarias y aprender a cómo inicia el código del lenguaje, es decir inicio del código y fin del código

Con esto ya puedes cambiarte fácilmente de un lenguaje a otro

Seguridad para IS

Al adentrarse al mundo laboral un ingeniero en software debe contar con ciertos conocimientos y habilidades que aprendió durante su etapa estudiantil. Hablando específicamente de seguridad esta ha venido cobrando mayor importancia para las organizaciones dado el marcado crecimiento de las nuevas tecnologías de la información, servicios Web, comercio electrónico, etc. Es por ello que existe la necesidad de contar con nuevos profesionales en este entorno. Estos profesionales deben contar con conocimientos como:

Fundamentos de seguridad, Seguridad organizativa, Requisitos de seguridad, Seguridad en desarrollo de software, Seguridad de sistemas de información, Riesgos de seguridad, Servicios de seguridad, Gestión de seguridad, Certificación, normas y estándares para la seguridad.

Según la UES un ingeniero en software debe saber supervisar la información confidencial de una empresa y controlar su tráfico, así como también crear sistemas de seguridad digitales, por ejemplo algún programa que cifre datos o los proteja contra virus, robo de datos, etc.

Con conocimientos cómo seguridad informática optimizar y dar mantenimiento a los programas, servidores, etc., de manera que los datos fluyan sin ningún riesgo o problema

Según la UAEH:

- Diseñar e implementar sistemas de seguridad en las organizaciones bajo políticas internas y lineamientos generales.
- Desarrollar soluciones orientadas al uso y desarrollo de aplicaciones móviles avanzadas, para la conectividad de productos y equipos

Por Juan Emmanuel Poot Escamilla