

5.1 - Desarrollo ágil de producto

En el dinámico mundo actual, la figura del Developer se transforma constantemente, abrazando roles más estratégicos y creativos. En este escenario, el desarrollo ágil de productos emerge como un enfoque que no sólo refina tus habilidades técnicas, sino que potencia tu capacidad para contribuir activamente en la creación de soluciones innovadoras.

En este módulo, te invitamos a explorar cómo, como developer, puedes ser un actor clave en este proceso. Desde participar activamente en la definición de la arquitectura del producto, hasta adoptar prácticas de documentación eficientes y ser un protagonista en la implementación de pruebas continuas, tu responsabilidad es esencial para el éxito del producto.

Prepárate para sumergirte en un camino donde tu conocimiento técnico se une con una visión estratégica, permitiéndote participar de manera significativa en la entrega continua de valor. ¡Adelante!

5.1.1 - Autogestión y Gestión del Tiempo: Claves en un buen Scrum Developer

Dentro de un equipo Scrum, se espera que los Developers sean proactivos y asuman un papel central en la autogestión de su tiempo y esfuerzos. Esto no solo fomenta la responsabilidad individual, sino que también promueve una cultura de colaboración y respeto mutuo.

Caso práctico: Imagina un escenario donde un Developer, Carlos, se da cuenta de que una tarea específica está consumiendo más tiempo del esperado, lo cual puede afectar la entrega oportuna del incremento planeado para el Sprint. En lugar de esperar a ser dirigido, toma la iniciativa de comunicar la situación al equipo, proponiendo una sesión de brain-storming para encontrar una solución eficiente. Gracias a esta proactividad, el equipo pudo reajustar y repriorizar las tareas, garantizando la entrega exitosa del incremento.

Tips para una Efectiva Autogestión y Gestión del Tiempo

- **Establece Metas Claras:** Asegúrate de entender bien el objetivo de cada Sprint y cómo contribuyen tus tareas a alcanzarlo.
- **Comunicación Continua:** Fomenta la comunicación constante con tu equipo para identificar y abordar problemas a tiempo.
- **Priorización Efectiva:** Aprende a priorizar tareas basándote en su valor y urgencia, esto ayudará a evitar el desperdicio de tiempo en elementos menos críticos.
- **Feedback Continuo:** Promueve una cultura de feedback constante para facilitar la mejora continua y la adaptación ágil.

La autogestión, combinada con una efectiva gestión del tiempo y priorización, no solo facilita la entrega oportuna de productos de alta calidad, sino que también contribuye a un ambiente de trabajo más armonioso y productivo.

5.1.2 - Consideraciones clave para el desarrollo ágil de productos

En el trayecto de construcción de productos ágiles, es imperativo tener en cuenta varios factores esenciales que influenciarán el éxito del producto final. Aquí te presentamos tres preguntas centrales que todo Developer debería tener en mente:

- ¿Por qué debería definirse la arquitectura del producto?
- ¿Qué debo considerar respecto a la documentación del producto?
- ¿Qué implican exactamente las pruebas del producto?

¿Por qué debería definirse la arquitectura de producto

La "Arquitectura del Producto" puede definirse como el esquema estructural que ilustra cómo se organizarán y interactuarán las distintas partes o componentes de un producto. Esta estructura no solo comprende la organización de elementos técnicos, sino también la definición de las relaciones y dependencias que existen entre ellos. El establecimiento de una arquitectura sólida desde el principio, ayuda a prever desafíos y facilita el proceso de incorporación de cambios o ajustes en etapas posteriores del desarrollo. En pocas palabras, funciona como el mapa guía que orienta el desarrollo del producto de una manera organizada y estratégica, asegurando que todas las piezas encajen de manera óptima para cumplir con los objetivos propuestos.

Definir la arquitectura del producto no es solo una cuestión de organización, sino que establece una hoja de ruta clara que facilita la identificación y solución de posibles problemas antes de que surjan. Permite una colaboración fluida entre los miembros del equipo y garantiza que todos trabajen hacia una visión unificada, optimizando los recursos y el tiempo disponible.

Antes de iniciar la construcción del producto es importante identificar la relación entre todos los componentes que harán parte del mismo. Definir una correcta arquitectura permitirá que los equipos de trabajo sepan como organizar sus entregables, se facilitará la integración y la mejora continua.

Es indispensable que, antes de iniciar con el desarrollo se defina el diseño técnico del producto o del incremento de producto a desarrollar. Esta actividad podría hacerse de forma iterativa antes de desarrollar los incrementos en los distintos Sprint, o de forma global al inicio del desarrollo.

Para garantizar que la arquitectura de producto cumple exactamente con los requerimientos y necesidades del cliente se debe hacer en conjunto entre los Developers y el Product Owner.

Tip para el Developer

Involúcrate activamente en las discusiones sobre la arquitectura del producto. Tu conocimiento técnico es vital para diseñar una arquitectura que no solo cumpla con los requisitos del cliente, sino que también sea sostenible y escalable a largo plazo. No dudes en compartir tus ideas y sugerencias.

La arquitectura está estrechamente relacionada con el objetivo del producto (Product Goal). El diseño arquitectónico del producto o sistema sienta las bases para lograr los objetivos y metas del producto, asegurando que se puedan implementar las características y funcionalidades deseadas de manera efectiva.

Ejemplo de arquitectura 1: Construyendo un vehiculo electrico

Al embarcarse en la construcción de un vehículo eléctrico, es vital analizar cuidadosamente los siguientes aspectos cruciales y entender cómo interactuarán entre sí y quién será responsable de su desarrollo, algunos ejemplos que debes considerar al definir la arquitectura son:

- **Sistema de Propulsión:** ¿Motor eléctrico de eje central o trasero? ¿batería de iones de litio o plomo? (en el piso del auto).
- **Sistema de Frenado:** ¿Frenos regenerativos para recuperar energía o solo ABS tradicional?.
- **Sistema de Navegación:** ¿GPS integrado con asistentes de conducción autónoma?.
- **Interfaz de Usuario:** ¿Tablero digital o analógico? ¿Controles táctiles o de voz?.
- **Conectividad:** ¿Bluetooth, Wi-Fi? ¿Conectividad celular para actualizaciones y telemetría?.

Ejemplo de arquitectura 2: Construyendo un software LMS

Una empresa de desarrollo de software planea construir una aplicación para realizar cursos virtuales. Al ser un proyecto de software, en la definición de arquitectura se deberán tener en cuenta los siguientes elementos:

- **Front-end:** ¿Será HTML, CSS, JavaScript para la interfaz de usuario?.
- **Back-end:** ¿Se usará un framework de base Ej. Laravel PHP para la lógica del servidor??
- **Base de datos:** ¿Qué motor de base de datos? ¿Relacional o no relacional?
- **APIs:** ¿JSON Web Token para autenticación? RESTful APIs para interactuar con servicios externos, como plataformas de redes sociales o bases de datos de terceros.
- **Infraestructura:** ¿Qué proveedor? ¿Qué servicios? ¿Cómo será la red? Servidores en la nube con escalabilidad automática.

Ejemplo de arquitectura 3: Construcción: Diseño de una Casa Moderna

- **Cimientos:** ¿Será hormigón armado o pilotes? ¿Madera?
 - **Estructura Principal:** ¿Marco de acero o madera para las paredes y techos?
 - **Sistema de Techo:** ¿Tejas de asfalto o metal?
 - **Sistema Eléctrico:** ¿Cableado empotrado o visible? (puntos de luz, enchufes y switches).
 - **Plomería:** ¿Tubos de PVC o cobre? (suministro de agua y desagüe).
 - **Espacio Habitacional:** Distribución de habitaciones, baños, cocina, etc.
 - **Sistema de Climatización:** ¿Aire acondicionado centralizado o radiadores?
 - **Energía eléctrica:** ¿Se usará energía alternativa en el futuro? ¿Se van a instalar múltiples medidores?
-

¿Qué debo considerar respecto a la documentación del producto?

La documentación es una herramienta vital en el desarrollo de productos, ya que facilita la trazabilidad y comprensión de las funcionalidades y características del producto. Es importante considerar no solo qué documentar, sino cómo hacerlo de una manera que sea accesible y útil para todos los involucrados. Además, incentivar la documentación proactiva puede ser un método efectivo para garantizar que se mantenga actualizada y relevante.

La documentación es una herramienta indispensable que, aunque a veces despreciada, puede ser un salvavidas en el ciclo de vida del desarrollo de productos. Vamos a desglosar cómo puede realizarse de manera eficiente y sin dolor:

- Muchos trabajadores odian documentar.
- Se pueden definir incentivos por mantener una buena documentación (no necesariamente económicos).
- Definir las reglas detalladas de qué se debe documentar y cómo.
- Se puede crear una Wiki y allí almacenar todo el contenido de forma organizada.

- Se puede incluir la documentación como parte de los criterios de la Definición de Terminado (DoD).

Tip para el Developer

Considera la documentación como una herramienta que te ayudará, no como una tarea tediosa. Al mantener una documentación precisa y bien organizada, facilitas la gestión de la calidad y la colaboración efectiva en tu equipo.

Caso práctico Pedro, uno de los developers, toma la iniciativa de crear una Wiki para almacenar toda la documentación del proyecto. A medida que el proyecto avanza, el equipo la mantiene actualizada, facilitando la incorporación de nuevos miembros al equipo y sirviendo como una valiosa referencia para revisar decisiones pasadas.

Caso Práctico: Motivando la Documentación Laura, una developer experimentada, notó que el equipo evitaba la documentación, lo que causaba confusión y retrasos. Para remediarlo, instauró junto al Scrum Master sesiones dedicadas a la documentación en cada sprint, complementándolo con un sistema de recompensas para el miembro que más contribuyera. Este enfoque transformó la documentación de una tarea despreciada a una actividad valorada que mejoraba la eficiencia del equipo.

¿Qué implican exactamente las pruebas del producto?

Las pruebas no son una fase aislada en el proceso ágil, sino una integración continua que asegura la calidad y el valor del producto en cada etapa del desarrollo. Los Developers deben adoptar una postura proactiva hacia la realización de pruebas, asegurando que cada incremento cumpla con los estándares de calidad establecidos y contribuyendo al éxito a largo plazo del producto.

Una prueba es la forma de comprobar el correcto funcionamiento del producto o uno de sus incrementos.

Tip para el Developer

Adopta un enfoque proactivo hacia las pruebas, viéndolo como una oportunidad para mejorar continuamente el producto, en lugar de una tarea de última hora. Incorpora pruebas regulares en tu flujo de trabajo para garantizar que cada incremento del producto sea de alta calidad.

Tip para el Developer

Recuerda que las pruebas no son una tarea aislada o exclusiva de un equipo de pruebas; eres co-responsable de la calidad de los entregables. Aprovecha tu conocimiento y habilidades para integrar pruebas continuas. Considera que las pruebas son exclusivas para proyectos de software, sino en todas las facetas del desarrollo de productos. Esto asegurará que el producto no solo funcione como se espera, sino que también cumpla con los estándares de calidad desde el inicio.

- En agilidad no se consideran a las pruebas como una fase separada, sino como parte integral del desarrollo de producto.
- En metodologías tradicionales, es una fase más dentro del desarrollo cascada o waterfall, lo cual no siempre resulta eficiente.
- Desde la agilidad se prueba continuamente, porque es una de las claves para mantener la calidad del producto.

Caso práctico: Cuando no se hacen pruebas o no son suficientes En un equipo, inicialmente se pasaba por alto la fase de pruebas o se realizaban de manera muy superficial, lo que llevaba a múltiples errores en las versiones finales del producto. Pero, al establecer definiciones de terminado (DoD) centradas en procesos específicos, se creó una pauta clara para realizar pruebas exhaustivas en cada incremento. Esto no solo redujo los errores sino que también facilitó que cada miembro del equipo pudiera llevar a cabo pruebas efectivas de sus propios entregables, mejorando así la calidad general del producto.

Revision #7

Created 6 September 2022 22:35:22 by CertMind

Updated 15 September 2023 22:49:23 by CertMind