5.3 - Técnicas y herramientas útiles para los Developers

En el dinámico mundo de Scrum, las responsabilidades de los Developers trascienden las fronteras de las tareas convencionales. Se espera que estén equipados para abordar una amplia variedad de desafíos, adaptándose con agilidad a las demandas cambiantes del proyecto. En esta sección, exploraremos algunas técnicas y herramientas cruciales que pueden facilitar a los Developers esta navegación a través del vibrante entorno de Scrum. Dentro de ellas veremos:

- Técnicas para el desarrollo ágil
- Técnicas para estimar el trabajo
- Técnicas y herramientas de software para trabajar con Scrum

> **Consejo** Al seleccionar y emplear diversas técnicas y herramientas, es vital hacerlo con un enfoque centrado en el valor y la eficiencia. Considera la compatibilidad con las necesidades del organización, la facilidad de integración y cómo pueden potenciar una colaboración efectiva y una entrega ágil de incrementos de alta calidad. A menudo, una herramienta simple pero bien elegida o una técnica aplicada con pericia puede marcar una significativa diferencia, optimizando procesos y fomentando un ambiente de trabajo más productivo y armonioso.

5.3.1 - Para el desarrollo ágil

En esta sección, exploraremos técnicas y herramientas que pueden facilitar un proceso de desarrollo más fluido y colaborativo, promoviendo la calidad y eficiencia en cada incremento.

> **Nota:** Para aquellos interesados específicamente en el ámbito del desarrollo de software, les invitamos a consultar nuestra otra publicación enfocada especialmente en este sector, disponible a través de CertMind, donde profundizamos en técnicas y herramientas adaptadas a las particularidades del

> desarrollo de software.

Programación por parejas – Pair programming

La programación por parejas es una técnica de desarrollo ágil en la que dos desarrolladores colaboran en la misma tarea, trabajando en un único equipo o computadora. Este enfoque fomenta la colaboración, la comunicación directa y un intercambio fluido de ideas, lo que a menudo resulta en soluciones más innovadoras y eficientes.

Aunque puede parecer que esta técnica reduce la cantidad de funcionalidad producida, en realidad, tiende a mejorar notablemente la calidad del resultado final. Al tener dos pares de ojos revisando el trabajo en tiempo real, es más probable que se identifiquen y corrijan errores en una etapa temprana, lo que puede prevenir costos adicionales a largo plazo.

- En la programación por parejas, 2 Developers trabajan en un solo PC.
- La programación por parejas mejora notablemente la calidad del incremento, sin impactar el tiempo de entrega.
- Una sola persona sola agrega más funcionalidad, pero con menos calidad.
- La calidad alta puede ahorrar costes futuros en un proyecto.

Tip para el Pair Programming

La programación por parejas no es una sesión de enseñanza; de hecho, funciona mejor cuando ambos desarrolladores tienen un nivel similar de conocimiento y habilidades. Esto permite una colaboración más fluida y equitativa, donde ambos pueden contribuir significativamente al progreso de la tarea. Asegúrate de emparejar a desarrolladores con niveles de habilidad compatibles para aprovechar al máximo esta técnica.

El pair programming no solo se usa en programación de software

- **Diseño Gráfico:** Dos diseñadores podrían trabajar juntos en un diseño, con uno manejando las herramientas de diseño y el otro proporcionando retroalimentación en tiempo real.
- **Redacción y Edición:** Un escritor y un editor podrían colaborar en tiempo real en un documento. Uno escribe mientras el otro revisa la gramática, el estilo, etc.
- **Análisis de Datos:** En un proyecto de análisis de datos, un estadístico podría implementar modelos mientras otro experto en el dominio proporciona contexto y verifica la validez del análisis.
- **Educación:** En un entorno educativo, dos instructores podrían colaborar para enseñar una clase en línea o preparar material didáctico.
- **Investigación Científica:** Dos investigadores podrían colaborar en un experimento o en la redacción de un artículo científico.
- **Planeación Estratégica en Negocios:** En la gestión y planeación estratégica, un par podría trabajar en el análisis SWOT de un proyecto o en la elaboración de un plan de negocios.
- **Gestión de Proyectos:** Un administrador de proyecto y un stakeholder podrían revisar juntos el progreso de un proyecto, identificando riesgos y planificando mitigaciones en

tiempo real.

> **Caso práctico: El pair programming puede ser agotador** En una empresa de tecnología, se instauró la técnica de programación por parejas como una actividad regular pero no constante. Los equipos encontraron que trabajar en parejas durante periodos específicos (como por la mañana o en sesiones de no más de dos horas) era beneficioso sin ser agotador. Esta moderación permitió mantener altos niveles de energía y concentración, favoreciendo la calidad y eficiencia del trabajo realizado.

> **Caso práctico: Cambiando de pareja** En una startup, se estableció una rotación de parejas de programación con cada nuevo Sprint. Esta rotación no solo permitió que los Developers fortalecieran sus relaciones interpersonales, sino que también promovió una comprensión más profunda y diversificada del proyecto en su conjunto, ya que cada miembro del equipo tuvo la oportunidad de trabajar y aprender de diferentes perspectivas y enfoques a lo largo del tiempo.

Tip: el Pair Programming no se aplica a todas las tareas

El "Pair Programming" no es una regla fija que deba aplicarse a todas las tareas. Es una herramienta especialmente efectiva para abordar desafíos complejos, explorar nuevas tecnologías o desarrollar funcionalidades críticas. Considera implementarlo estratégicamente, especialmente en situaciones donde la colaboración directa puede potenciar la creatividad, la calidad y la eficiencia. Recuerda, la clave es usarlo donde agregue el mayor valor al proceso de desarrollo del producto.

La refactorización

La refactorización, en un contexto más amplio que el desarrollo de software, puede ser entendida como el proceso de revisar, ajustar y mejorar las estructuras existentes de un proyecto o sistema para hacerlo más eficiente, manejable o adaptable a nuevas necesidades, sin alterar su función fundamental. Esto puede aplicarse a una variedad de ámbitos, como procesos empresariales, estrategias de marketing, estructuras organizativas, etc.

> **Ejemplo: Antes de la Refactorización** Un equipo utiliza una plantilla de informe densa y complicada, que contiene una cantidad significativa de campos irrelevantes. Esto no solo consume tiempo sino que también dificulta la comprensión clara de los puntos clave. **Después de la Refactorización** - El equipo decide revisar y simplificar la plantilla, eliminando los campos innecesarios y dando prioridad a la información más crítica. Como resultado, los

> informes ahora son más concisos, centrados y fáciles de leer, facilitando la toma de decisiones y la comunicación efectiva dentro del equipo.

Tip: La refactorización debe ser vista como una actividad continua

La refactorización es un compromiso con la mejora constante. No es un "arreglo único", sino un proceso en el que se está siempre atento a las oportunidades de optimizar y ajustar para alcanzar una mayor eficiencia y efectividad. Además, es vital evitar la "refactorización por refactorización". Cada cambio debe tener un propósito claro y estar orientado a lograr mejoras tangibles, evitando alteraciones que no agregan valor significativo.

5.3.2 - Para estimar el trabajo

La estimación consiste en definir entre todos una suposición lo más precisa posible, de lo que se puede lograr y en cuanto tiempo.

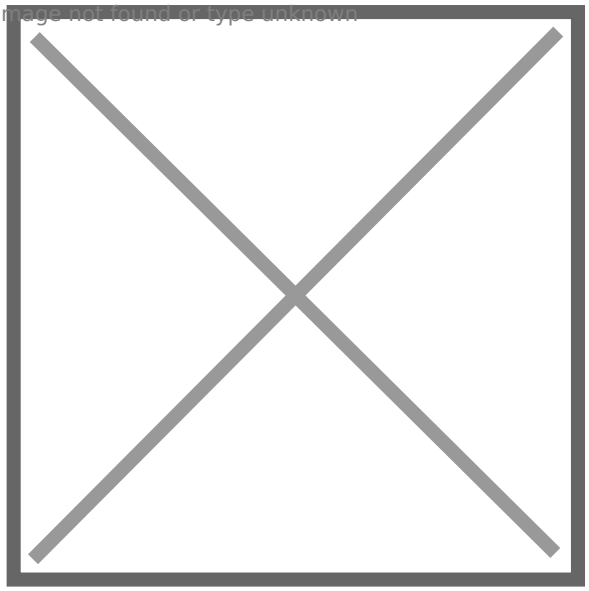
La estimación permite:

- Analizar dependencias
- Organizar las actividades prioritarias
- Definir las mejores técnicas de trabajo para ese Sprint
- Pronosticar diferentes escenarios (predecir tiempos y fechas de entrega)

Nota: El momento en que se realiza la estimación es durante la Planificación del Sprint.

Estimar con Planning Poker

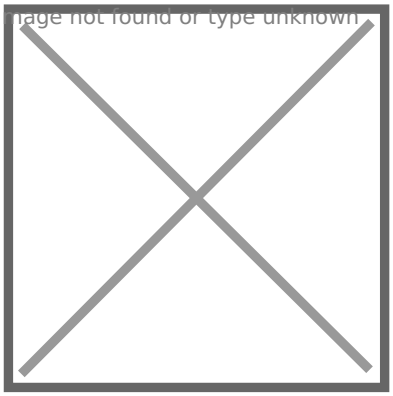
Image not found or type unknown



El póker de planificación consiste en un conjunto de tarjetas numéricas que sirven para realizar la estimación de tareas o historias de usuario.

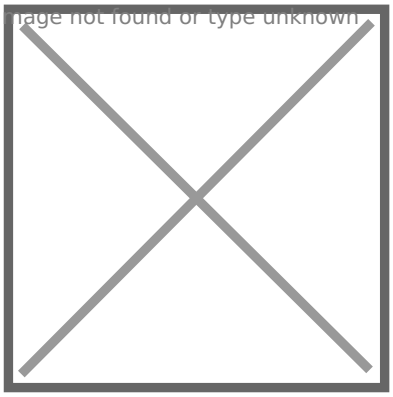
El póker puede ir numerado de 1 a 10 o utilizando la sucesión de Fibonacci

Estimar por afinidad



Consiste en encontrar las similitudes de las Historias de Usuario y tareas que se van a estimar. Lo que hace el equipo es agrupar los elementos que son similares. La mejor manera de “encontrar similitudes” es visualizar el panorama completo y luego agrupar elementos.

Estimar con Tallas de camiseta (t-shirt)



Se asigna una talla de camiseta a cada Historia de Usuario o tarea (desde XS hasta XXL) que representará el esfuerzo relativo de ese trabajo.

Con esta técnica se elimina la necesidad de la precisión de un modelo numérico, por lo que el equipo puede pensar de manera más abstracta en las estimaciones del trabajo a estimar.

¿Qué pasa cuando se Subestima/Sobreestima un Sprint?

Cuando los equipos Scrum son nuevos, o tienen poca experiencia es típico que sucedan las siguientes 2 situaciones:

Sobreestimación: Cuando los Developers proponen demasiado tiempo para el trabajo del Sprint y acaban mucho antes de la Revisión del Sprint. En este caso el Product Owner agregará más trabajo

al Sprint en curso (para que la Revisión no se cambie de fecha).

Subestimación: Cuando los Developers proponen muy poco tiempo para el trabajo del Sprint, lo que puede causar que el incremento no esté completamente terminado. En este caso la Revisión se hace sobre los elementos que se hayan completado y los demás quedan pendientes para ser terminados en el próximo Sprint (la revisión NO se aplaza).

Otras posibles técnicas que podrían ser útiles para mejorar el flujo de trabajo durante un Sprint

- Mesa de ayuda cruzada (Cross-Functional Help Desk): Un espacio donde los miembros del equipo pueden ofrecer o solicitar ayuda en diferentes áreas de especialización, fomentando la colaboración y el aprendizaje mutuo.
- Time-boxing: Establecer un tiempo límite para ciertas actividades puede ayudar a mantener el enfoque y evitar la procrastinación, así como fomentar la toma de decisiones eficiente.
- Revisión intermedia (Mid-Sprint Review): Una revisión a mitad de camino del Sprint puede ayudar a identificar cualquier obstáculo o cambio necesario a tiempo, permitiendo ajustes antes de la Revisión de Sprint.
- Kanban visual: Utilizar un tablero visual Kanban para monitorear el progreso de las tareas, lo que puede facilitar la identificación de cuellos de botella y ayudar a mantener un flujo de trabajo estable.
- Daily Stand-ups Mejoradas: Incorporar segmentos breves donde se compartan conocimientos o trucos útiles que hayan ayudado a los miembros del equipo a ser más eficientes.
- Refactorización de Procesos: Similar a la refactorización de código, esta técnica implica revisar y ajustar los procesos actuales para hacerlos más eficientes, eliminando pasos innecesarios o redundantes.
- Reuniones de Alineación de Equipos (Team Alignment Meetings): Organizar reuniones breves y focalizadas donde los equipos puedan alinear expectativas, definir objetivos claros y discutir estrategias para alcanzarlos con éxito.
- Workshops de Resolución de Problemas: Organizar talleres donde los equipos puedan colaborar para encontrar soluciones a problemas específicos, utilizando técnicas como el brainstorming o el pensamiento de diseño (Design Thinking).

Revision #4
Created 6 September 2022 22:40:16 by CertMind
Updated 6 September 2023 21:43:21 by CertMind