Manual

jueves, 29 de febrero de 2024 22:53

Cultura

Bienvenidos a una jornada de descubrimiento y aprendizaje continuo en el mundo del desarrollo ágil de software. Este manual es una contribución personal de David Averos, Scrum Master apasionado y comprometido con la excelencia en la práctica de Scrum. A través de las páginas que siguen, David comparte su experiencia acumulada y conocimiento con el fin último de enriquecer y perpetuar la cultura de Scrum.

El propósito de este compendio no es otro que ofrecer una guía sustancial que sirva de faro en la implementación, consolidación y maduración de las prácticas de Scrum en cualquier entorno de desarrollo. Desde la presentación de los principios fundacionales hasta la descompresión de las ceremonias y artefactos que componen este marco de trabajo, cada elemento ha sido cuidadosamente seleccionado y explicado para fomentar una comprensión integral de Scrum.

Este manual aspira a ser más que un recurso educativo; pretende ser un aliado en el día a día, una herramienta práctica que se puede consultar en el ajetreo de los sprints, en la reflexión de las retrospectivas y en la visión estratégica de la planificación de productos. Es un testimonio de las posibilidades que Scrum ofrece para transformar equipos, proyectos y, en última instancia, empresas, impulsándolos hacia la excelencia operativa y la creación de valor constante para los clientes.

En un mundo donde la adaptabilidad y la respuesta rápida a los cambios no son solo deseables, sino esenciales, Scrum se erige como la metodología de elección para aquellos que buscan no solo sobrevivir, sino prosperar en el dinámico paisaje del desarrollo de software. Que este manual sirva como su mapa y brújula en ese esfuerzo, y que cada página les acerque más a la maestría en el arte de Scrum.

Con cada capítulo, te invitamos a sumergirte en el espíritu de colaboración, aprendizaje y mejora continua que Scrum promueve. Bien sea que estés dando tus primeros pasos en este marco ágil o que busques perfeccionar tus habilidades existentes, hay algo aquí para cada miembro del equipo que comparte el deseo de crear software de manera más efectiva, eficiente y en armonía con los valores de Scrum.

Embarquémonos juntos en este viaje de transformación y crecimiento.

— David Averos, Scrum Master

Índice

tradicionales a los ágiles, evidenciando el cambio de paradigma en las empresas públicas en Ecuador hacia prácticas más flexibles y dinámicas que se alinean con los valores de Scrum.

- **Lean Management**
  Exploración de los principios de Lean Management y cómo su enfoque en la eficiencia y la eliminación de desperdicios complementa y enriquece la implementación de Scrum en proyectos de desarrollo de software.

- **Introducción**
  Introducción a los fundamentos de Scrum, su historia y su evolución desde una solución de nicho hasta convertirse en una metodología dominante en la industria del software, destacando su agilidad y adaptabilidad.

- **Scrum Descifrado**
  Un análisis profundo de Scrum, desmitificando sus principios, prácticas y marco de trabajo, y cómo se diferencia de otros métodos ágiles para ofrecer una guía práctica y clara a los equipos de desarrollo.

- **Roles**
  Descripción detallada de los roles dentro de un equipo Scrum: Product Owner, Scrum Master y el Equipo de Desarrollo, delineando sus responsabilidades y la importancia de la colaboración para el éxito del proyecto.

- **Artefactos**
  Exploración de los artefactos de Scrum, incluyendo el Product Backlog, Sprint Backlog y los incrementos, y cómo cada uno contribuye a la transparencia y organización del proceso de desarrollo.

- **Sprint**
  Una visión completa del ciclo de vida del Sprint, desde la planificación hasta la retrospectiva, y cómo cada fase impulsa el progreso hacia los objetivos del proyecto.

- **Fortalezas de Scrum**
  Reflexiones finales sobre las fortalezas de Scrum, incluyendo la capacidad de la metodología para adaptarse a cambios, su enfoque centrado en el cliente y la promoción de un trabajo colaborativo y eficiente.

Primer Vistazo

viernes, 12 de enero de 2024 16:16

Metodología de desarrollo enfocado desde el desarrollo del producto software

- Tradicional
  Las empresas públicas en ecuador suelen usar metodologías tradicionales por temas de contraloría

- Metodología Ágil / SCRUM
  Este escenario vemos que los roles son pocos, el reto es ver quien toma dicho rol, los requerimientos son muy cambiantes y el equipo es dinámico ante este nuevo enfoque que toma la industria.

BA: gestiona la base de datos
Tester Team: son todo un equipo
Developer: son como 30 personas

Retomando el problema nacional es que tener procedimientos y acciones no debemos de dejar delado aspectos normativos

- Ya si SCRUM fue hace 20 años
  Actualmente vamos con una nueva metodología ágil que es Devops que es poner en producción rápida

Índice de SCRUM

Indice

Temario	
(1) Introducción. ¿Cómo surge?. Metodologías ágiles. Lean SDLC. ¿Cuando un método es ágil?. Diferencias entre metodologías.	(4) Prácticas para la gestión del proyecto y del producto.
(2) Introducción a SCRUM. Principios. Características.	(5) Conclusiones. Plataforma requerida. Fortalezas de SCRUM. Reflexiones.
(3) Practica SCRUM Propietario del Producto. El Scrum Master. El Equipo. Interdependencias de los roles. Historias de usuario. Product Backlog. Sprint. Planificación del Sprint. Estimación de póquer. Ejecución del Sprint. Fin del sprint: Sprint review. Sprint retrospective. Daily [illegible]nzar / Parar / Continuar). Incremento	(6) Caso practico

Grabación de audio iniciada: 16:19 lunes, 15 de enero de 2024

En scrum importa la complejidad ya no el tiempo, este es lo desafío

Grabación de audio iniciada: 16:58 lunes, 15 de enero de 2024

Lean Management

lunes, 22 de enero de 2024 8:31

"Es importante que lideres el cambio y que no lo gestiones"

Contexto Histórico

-2003 by: Mary y Tom Poppendieck publicaron su libro "Lean software development: an Agile Toolkit"
El libro describe cómo puedes aplicar los principios Lean al desarrollo de software. Al final, el desarrollo de software Lean se reduce a 7 principios.

Introducción

Es una filosofía de gestión que se originó en el ámbito manufacturero de Toyota y que **busca eliminar el desperdicio**, mejorar la eficiencia y maximizar el valor para el cliente.

- Consiste en mejorar continuamente los procesos de trabajo, los propósitos y las personas, todo esto a través de una cultura mas no con un manual de reglas y normas.

Relación con SCRUM:

- SCRUM comparte principios Lean, como la entrega iterativa y continua, la adaptabilidad a cambios y la mejora continua.
- Ambos enfoques buscan maximizar el valor para el cliente y minimizar el desperdicio de recursos.
- SCRUM se beneficia de los principios Lean al adoptar prácticas como la planificación iterativa, la revisión continua y la colaboración estrecha con el cliente.

Ideas

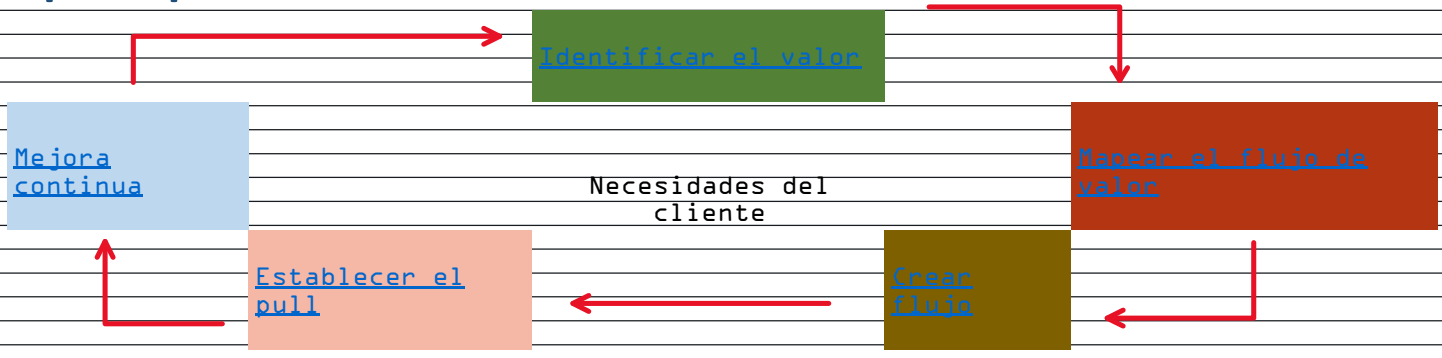
Basado en 3 ideas simples:

i. **Entregar valor desde la perspectiva de tu cliente**

ii. **Eliminar el desperdicio (cosas que no aportan valor al producto final)**

iii. **Mejora continua**

5 principios básicos



"Es importante que lideres el cambio y que no lo gestiones"

La frase destaca la importancia de un liderazgo activo y comprometido durante los procesos de cambio. No se trata simplemente de llevar a cabo tareas administrativas, sino de ser un catalizador que impulsa y guía a las personas a medida que se embarcan en la transformación.

1. Identificar el valor

Preparado el equipo para el cambio, como Lean manager, debes trabajar con tu equipo para reconocer y definir el valor del trabajo que realizan.

Es esencial comprender qué actividades y características son verdaderamente valiosas desde la perspectiva del cliente, diferenciar entre actividades valiosas y desperdicios, y fomentar la colaboración del equipo en este proceso.

- Actividad colectiva
  - Todos los miembros del equipo estén alineados. La colaboración es clave para garantizar que todos compartan la misma comprensión de lo que constituye valor en el contexto del proyecto.

7 Desperdicios en el Desarrollo de Software

Los 7 desperdicios comunes en el desarrollo de software, que incluyen tareas que no añaden valor al cliente.

i. **Transporte**
Cambiar de tarea con mucha frecuencia, innumerables interrupciones de colegas

ii. **Inventario**
código o funcionalidades no entregadas

iii. **Movimiento**
reuniones innecesarias o esfuerzo adicional para encontrar información

iv. **Espera**
esperar a que se completen las pruebas, la revisión del código, etc

v. **Sobreproducción**
Producción de funcionalidades que nadie usará

vi. **Sobre procesamiento**
complejos e innecesarios algoritmos que resuelven problemas simples

vii. **Defectos**

El Aseguramiento de la calidad es el ejemplo más claro de un desperdicio necesario.
Este no genera un valor directo para el cliente final, pero garantiza que el valor del proceso de desarrollo no se pierda.

Trazabilidad con SCRUM:

Este principio se alinea con la idea de tener una clara lista priorizada de elementos de trabajo en el Product Backlog.

En SCRUM, el Product Owner es responsable de definir y priorizar los elementos del backlog en función del valor que aportan al cliente.

Relevancia en el Desarrollo de Software:

Al entender y especificar el valor desde la perspectiva del cliente, los equipos de desarrollo pueden centrarse en la entrega de funcionalidades que realmente importan, evitando el desperdicio de recursos en características que no añaden valor percibido por el cliente.

Trazabilidad entre Principios

Es crucial haber identificado claramente qué actividades y características son percibidas como valiosas por el cliente.

Estas son aquellas que satisfacen directamente sus necesidades y expectativas. Pueden variar según el contexto del proyecto y la naturaleza del producto o servicio, pero algunas ejemplos generales podrían incluir:

Actividades Valiosas:

- **Desarrollo de Funcionalidades Clave:**
  La implementación de funciones esenciales que cumplen con los requisitos fundamentales del cliente.
- **Entrega Rápida de Incrementos:**
  La capacidad de proporcionar entregas frecuentes y rápidas de nuevas funcionalidades o mejoras.
- **Adaptabilidad a Cambios:**
  La flexibilidad para responder a cambios en los requisitos del cliente y ajustar el producto en consecuencia.
- **Buena Experiencia de Usuario:**
  La mejora continua de la interfaz de usuario y la experiencia general del cliente al utilizar el producto.
- **Alta Calidad del Software:**
  La entrega de un producto libre de errores y con un rendimiento óptimo.
- **Soporte Eficiente:**
  La disposición y capacidad para brindar un soporte eficiente en caso de problemas o consultas.

Características Valiosas:

- La buena implementación de un Modelo de Calidad del Producto
- Personalización: La posibilidad de personalizar el producto según las necesidades individuales del cliente.
- Innovación: Introducción de nuevas funcionalidades innovadoras que anticipan las necesidades futuras del cliente.

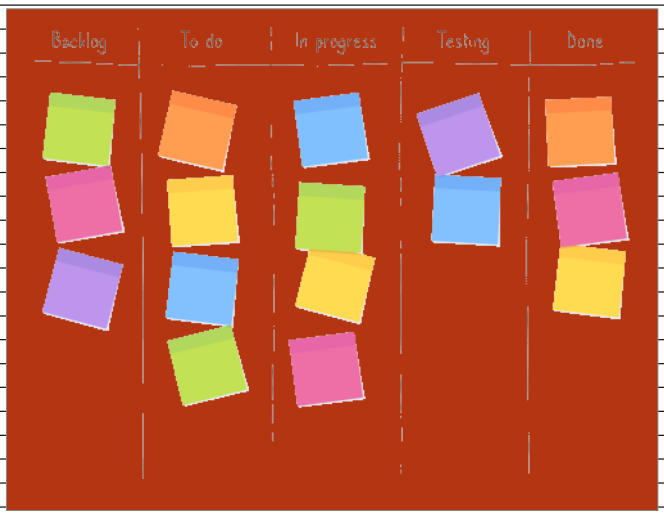
2. Mapea el Flujo de Valor

Implica visualizar y comprender todo el proceso de desarrollo, desde la concepción de la idea hasta la entrega al cliente.

- Busca identificar y eliminar actividades que no añaden valor al producto final.

Visualización con Kanban

La visualización del flujo de valor se realiza utilizando tableros Kanban.
Estos tableros proporcionan una representación visual del proceso, permitiendo a los equipos comprender de manera rápida y clara cómo fluyen las tareas y dónde pueden surgir mejoras.



Al comprender visualmente el flujo de valor, los equipos pueden optimizar el proceso para minimizar demoras y entregar más rápidamente un producto de calidad al cliente.

3. Crear Un Flujo de Trabajo Continuo

Establecer un flujo de trabajo constante y eficiente, minimizando los tiempos de espera y eliminando obstáculos para lograr una entrega continua de valor al cliente.

- Entrega Continua:
  Fomenta la entrega continua de productos o incrementos de software, evitando acumular grandes cantidades de trabajo antes de entregar al cliente.

- Visualización del Flujo con Kanban:
  El uso de tableros Kanban puede ayudar a visualizar y gestionar el flujo de trabajo, identificando cuellos de botella y optimizando el proceso.

Dividir el trabajo en grupos más pequeños y visualizar el flujo de trabajo, podrás detectar y eliminar fácilmente los obstáculos del proceso

4. Crear un Sistema Pull

Se enfoca en garantizar que el trabajo se realice a la velocidad adecuada, evitando la sobreproducción y asegurando que cada tarea se inicie cuando haya capacidad para llevarla a cabo.

- **Trabajo Iniciado por la Demanda:**
  Las tareas se inician según la demanda del sistema, evitando comenzar nuevas actividades antes de que sea necesario.

- **Evitar la Sobreproducción:**
  Busca evitar la generación de trabajo innecesario o exceso de tareas que podrían acumularse sin ser finalizadas.

- **Colaboración y Comunicación:**
  La implementación de un sistema de retiro fomenta la colaboración y comunicación efectiva entre los miembros del equipo, ya que las tareas se asignan según la capacidad y la necesidad.

- **Optimización de Recursos:**
  Al trabajar según la demanda, se optimizan los recursos y se evitan cuellos de botella que podrían surgir al abordar más trabajo del que el equipo puede manejar.

5. Mejora Continua

Recuerda, tu sistema no está aislado ni estático. Pueden ocurrir problemas en cualquiera de los pasos anteriores. Es por ello que debes asegurarte de que los empleados de todos los niveles participen en la mejora continua del proceso.

- Establecer una cultura dentro del equipo en la que la reflexión, la adaptación y la optimización constante sean fundamentales.

  a. **Feedback y Reflexión:**
     - Se fomenta la retroalimentación constante sobre el desempeño del equipo y el proceso.
     - La reflexión permite identificar oportunidades de mejora.

  b. **Adaptación Continua:**
     - Los equipos deben estar dispuestos a ajustar y adaptar sus procesos según las lecciones aprendidas y los cambios en el entorno del proyecto.

  c. **Experimentación y Aprendizaje:**
     - Se alienta la experimentación y la disposición a probar nuevas ideas. El aprendizaje constante contribuye a la evolución y perfeccionamiento del proceso.

  d. **Conexión con SCRUM:**
     - En SCRUM, la retrospectiva al final de cada sprint es una oportunidad para la mejora continua.
     - Los equipos revisan lo que salió bien, lo que podría mejorarse y aplican ajustes en sprints futuros.

  e. **Cambio Cultural:**

- Establece un cambio cultural en el equipo, donde la mejora continua no es vista como una tarea única, sino como una parte inherente del modo de trabajar.

f. Aplicación a Todo Nivel:

- La mejora continua no se limita al trabajo operativo; también se aplica a la gestión, la comunicación y cualquier otro aspecto del proceso.

Lean en el SDLC

- Mas info en: https://www.lean.org/explore-lean/what-is-lean/
- Lean StarUp

  Eric Ries, un ingeniero y emprendedor en serie, desarrolló una metodología basada en los principios Lean para ayudar al éxito de las startups. En el 2011, plasmó sus ideas en un libro llamado "The Lean Startup". El concepto consiste en 5 principios básicos que tienen como objetivo ayudar a las startups a ser más flexibles y responsivas a los cambios
- Los https://businessmap.io/es/gestion-lean/valor-desperdicios/7-desperdicios-de-lean

Introducción

lunes, 22 de enero de 2024 7:08

Como surge?

Contexto Histórico

Es un marco de trabajo ágil para el desarrollo de software que surgió en la década de 1980.

Propuesto por Jeff Sutherland y Ken Schwaber
Quienes se inspiraron en el mundo del rugby para crear un enfoque colaborativo y adaptable para el desarrollo de software.

- "Scrum" proviene de una formación de rugby donde los jugadores se agrupan para trabajar juntos de manera eficiente.

- 1995
Sutherland y Schwaber presentaron formalmente SCRUM como un proceso en la conferencia OOPSLA (Object-Oriented Programming, Systems, Languages & Applications).

Problemas con otras metodologías

Scrum trata de centrarse en los siguientes problemas que presentan otras metodologías agiles y tradicionales.

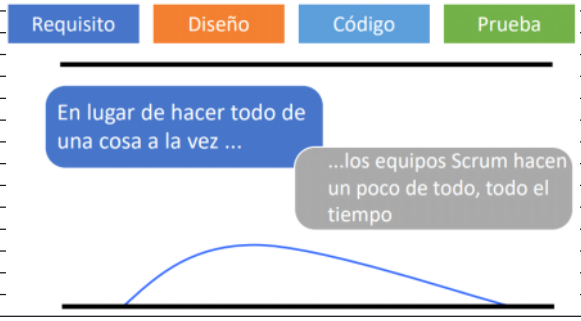
• Requerimientos fuera de control	• No cumple con los tiempos	• Estimaciones deficientes
• Retrabajo excesivo	• Baja Calidad de software	• Insatisfacción de los profesionales participantes

Ágil & Scrum

Scrum es un framework que busca ofrecer y crear el **más alto valor de negocio** en el menor tiempo.

- El negocio fija las prioridades. Los equipos se auto-organizan a fin de determinar la mejor manera de entregar las funcionalidades de más alta prioridad.
- Cada dos semanas o un mes, cualquiera puede ver el software real funcionando y decidir si liberarlo o seguir mejorándolo en otro sprint.

SCRUM REQUIERE
- Disciplina
- Espíritu de equipo
- Objetivo claro
- Mismas reglas para todos
- Diversión
- Entrega de resultados



Respetando los Fundamentos

Sigue siendo una metodología ágil así que TIENE QUE cumplir con el Agile Manifesto, esto siempre lo vamos a valorar más.

Además este también tiene sus 12 principios:

1. Satisfacción del Cliente

MAYOR prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor

2. Requisitos no son INMUTABLES

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo.

Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

3. Entregas Fast

Entregamos software funcional frecuentemente, **entre dos semanas y dos meses**, con preferencia al periodo de tiempo más corto posible.

4. El de Sistemas ya no está SOLO

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana **durante todo el proyecto**

5. El humano importa

Los proyectos se desarrollan en torno a individuos motivados.

Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

6. Me lo decís en la Jeta

El método más eficiente de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara

7. Medida del progreso

El software funcionando

8. Constancia

Los procesos Ágiles promueven el desarrollo sostenible.

Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

9. Técnico y Diseño

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

10. Al final siempre gana lo simple

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Explicación:
Se centra en la idea de que, en el desarrollo de software, se debe buscar la simplicidad y minimizar la complejidad innecesaria.

Maximizar la cantidad de trabajo no realizado
Este concepto destaca la importancia de enfocarse en lo que realmente importa y proporciona valor al usuario. Maximizar la cantidad de trabajo no realizado significa evitar realizar tareas, funciones o características que no contribuyen significativamente a los objetivos del proyecto o que podrían introducir complejidad innecesaria.

11. Equipo Autónomos

Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

Autoorganizado:
- Aquel que tiene la capacidad de tomar decisiones relacionadas con su trabajo sin depender de una supervisión externa constante.
- Los miembros del equipo se organizan y colaboran entre sí para planificar, ejecutar y mejorar su trabajo. Tienen la autonomía para tomar decisiones operativas y resolver problemas sin esperar instrucciones detalladas de arriba.

12. Feedback pero a las personas

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Anglicismos

- **Burndown chart:** Cuadro de incendio o avance
- **Framework:** Marco de trabajo
- **Product Backlog:** Lista de pendientes del producto
- **Product Owner:** Propietario del producto
- **Scrum Board:** Tablero Scrum
- **Scrum Master:** Maestro Scrum
- **Sprint:** Iteración
- **Sprint Backlog:** Lista de pendientes del Sprint

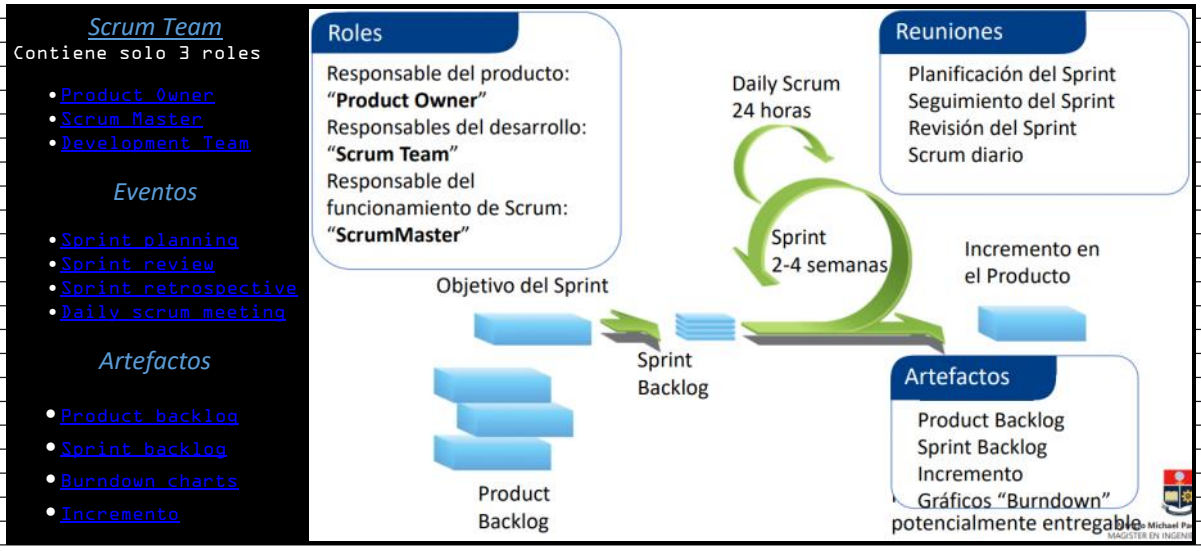
Usado para

- Software comercial
- Desarrollos internos
- Desarrollos bajo Contrato
- Proyectos Fixed-price
- Aplicaciones Financieras
- Aplicaciones certificadas ISO 9001
- Sistemas Embebidos
- Sistemas con requisitos 7x24 y 99.99% de disponibilidad
- Joint Strik Fighter
- Desarrollo de video juegos
- Sistemas críticos de soporte vital, aprobados por laFDA
- Software de control satelital
- Sitios Web
- Software para Handheld
- Teléfonos portátiles
- Aplicaciones de Network switching
- Aplicaciones de ISV
- Algunas de las más grandes aplicaciones en uso

SCRUM Descifrado

sábado, 27 de enero de 2024 14:38

Framework



Es una a practica de desarrollo muy simple, no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto

Principios

Es un Modelo de Desarrollo Incremental puede ser utilizado para desarrollar cualquier producto o administrar cualquier trabajo.

Donde sus principios son:

1. **Entrega Temprana y Continua con Flexibilidad a Cambios:**
   - Objetivo principal es satisfacer al cliente mediante la entrega rápida y continua de software útil y valioso.
   - Receptivo a los requisitos cambiantes, incluso en etapas avanzadas del desarrollo, para adaptarse mejor a las necesidades emergentes.

2. **Colaboración Continua:**
   - Colaboración estrecha y permanente entre los responsables de negocio y los desarrolladores es crucial.
   - El trabajo en equipo y la comunicación directa y cara a cara son fundamentales para el éxito del proyecto.

3. **Enfoque en Profesionales Motivados:**
   - Los proyectos se construyen con individuos motivados. Proporcionar el entorno y el apoyo necesario y confiar en ellos para hacer el trabajo es esencial.
   - La motivación y el compromiso del equipo contribuyen significativamente a la productividad y calidad del trabajo.

4. **Simplicidad y Eficiencia:**
   - La simplicidad y la eliminación de trabajo que no agrega valor son esenciales.
   - Se enfatiza en métodos de trabajo eficientes, incluyendo la comunicación cara a cara como la forma más eficiente y efectiva de transmitir información.

5. **Medición de Progreso y Desarrollo Sostenible:**
   - El progreso se mide principalmente a través de entregas de software funcional.
   - Los procesos ágiles apoyan un ritmo de desarrollo sostenible, asegurando que el equipo pueda mantener un ritmo constante y efectivo de trabajo indefinidamente.

6. **Enfoque en la Calidad y Mejora Continua:**
   - La atención constante a la excelencia técnica y al buen diseño mejora la agilidad.
   - La reflexión regular sobre cómo ser más efectivo y ajustar el comportamiento en consecuencia es una parte integral del proceso.

7. **Autoorganización y Autonomía del Equipo:**
   - Se promueve que los equipos sean auto-organizados para fomentar la responsabilidad, creatividad y eficiencia.
   - Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan y tienen autonomía en la toma de decisiones.

Características

- El producto progresa en una serie de "Sprint" que duran **entre dos semanas y dos meses**
- Los requerimientos se encuentran en el "Product backlog" reunidos en una lista
- No contiene prácticas de ingeniería pre-descriptas
- Usado para proyectos complejos con requerimientos cambiantes
- Basado en un control de proceso empírico y no hay prácticas de ingeniería prescritas
- No dice Qué hacer sino Cómo hay que hacer las cosas

Trazabilidad

1. Seguimiento Preciso del Producto/Servicio:
   - La trazabilidad en SCRUM implica un seguimiento detallado del producto o servicio a lo largo de todo su ciclo de vida.

2. Registro y Seguimiento de Requerimientos:
   - Cada requerimiento se identifica y registra para rastrear su evolución desde el origen hasta la entrega final.

3. Documentación Integral:
   - Toda la documentación, códigos y guiones de prueba se vinculan con su fuente de origen, facilitando la comprensión de su desarrollo e historia.

4. Bidireccionalidad en la Trazabilidad:
   - Permite conectar un requisito con el código que lo implementa y viceversa, asegurando una relación clara entre requisitos y su implementación.

5. Trazabilidad Vertical:
   - Asegura que cada requisito se refleja en el diseño, y que este a su vez se traduce en código y pruebas correspondientes.

6. Trazabilidad Horizontal:
   - Identifica posibles conflictos o inconsistencias entre diferentes etapas como requerimientos, diseño, codificación y pruebas.

Roles

lunes, 22 de enero de 2024 16:09

"Para tu rol que tomes requerimos que estes COMPROMETIDO mas no involucrado"

Grabación de audio iniciada: 16:09 lunes, 22 de enero de 2024

Scrum Team: son todos

Se refiere a todos los roles que abarca Scrum siendo solo 3 y otros aquí vemos una traza entre cada rol:

- PO: PRODUCT OWNER: El que tiene dominio del negocio, se comunica directamente con el cliente

  Le pasa la posta a

- SM: SCRUM MASTER: él es el que hacer todo el escenario del sprint trabajando con
- TEAM: DEVELOPER TEAM: son todos los que codifican, diseñan, analizan y eso

Aquí una traza entre cada uno:

Interesados	PO	PM	SM	TEAM
	>> Asegura que el proyecto cumple objetivos			
	>> Administra al equipo			Auto-organizado
	>> Administra el alcance y el presupuesto			
	>> Administra la comunicación entre involucrados			
	>> Gestiona riesgos y elimina obstáculos			
	>> Administra el proceso			
	>> Gestiona la visión		Mejora continua	
>> Investigación de mercado. Visión, Voz del Cliente				
	<< Asegura que el proyecto logra los objetivos			
	<< Negocia el trabajo con el equipo			
	<< Gestiona el alcance, cronograma, y presupuesto			
	<< Gestiona la comunicación con los interesados			
		>> Visualiza y distribuye información		
		>> Remueve impedimentos		

ESCALABILIDAD

Normalmente los equipos son de 7 ± 2 personas

- La escalabilidad proviene de equipos de equipos

Factores a tener cuenta

- Tipo de aplicación
- Tamaño del equipo
- Dispersión del equipo
- Duración del proyecto

Scrum se ha utilizado en múltiples proyectos de más de 500 personas

PRODUCT OWNER

"La voz del cliente"

Es solo una persona

★ Conocedor con Dominio del problema

- Por ende define las funcionalidades del producto
- No es un ingeniero de software necesariamente
- Representa a la gestión del proyecto
  - Responsable preguntar el costo, ver el triángulo
  - Determina si, si es rentable el proyecto

- El representa al usuario y todos los interesados en el producto final.
  - Da prioridades en cada iteración
  - aprueba o nel los resultados

- Hace el Product Backlog
  - De la manera más clara y menos ambigua posible.

- Da fechas y contenido de los Release

SCRUM MASTER

"Para gobernar hay que saber servir"

- Responsable de promover y enseñar los valores y prácticas de SCRUM para generar una cultura
- Vela por que cooperen los DEVELOPER TEAMS siendo así funcionales
- Si pasa algo externo es el primero en velar que el sprint se cumpla
- Garantizar la Simplicidad
- Vela por que cumplan los roles
  - Que los mismos se encuentre Motivados
- Es el facilitador, como el coach y tracker, un revolvedor en pocas palabras.

DEVELOPER TEAMS

Es una vista de Web Development Team pero es básicamente lo mismo que Developer Teams

Son técnicamente todo los roles que venían en los tradicional en un solo rol:

Características:

- Son auto organizados ni siquiera el scrum master ni nadie indica al equipo cómo convertir elementos del producto backlog en incrementos.

Son:

- De 3 - 9 personas con autogestión y autoorganización
- Multidisciplinarios que cubren todas las habilidades como ser:
  - DEVS
  - ANALLISTA
  - DISEÑADORES
  - QA : Velan por la calidad
- Puede haber cambios de miembros solo entre los sprints

Responsables de:

- Transforman la pila del sprint en un in+2Bcremento de la funcionalidad del software
- Cada uno debe ser un experto en casi todo
- Full time: debe estar al 100%

Y los otros?

En developer team se encuentran pero siendo más específicos son:

- Arquitecto:
  Encargado de realizar las decisiones de la arquitectura para el equipo.
- Especialista:
  en la mayoría de equipos agiles no existe especialistas, sin embargo en proyectos a gran escala es necesario de conocimientos específicos.
- Expertos técnicos
  Son parte del equipo de manera temporal, los cuales ayudan a superar problemas con altas dificultades, así mismo transfieren parte de sus habilidades a los desarrolladores.
- Tester independiente:
  En algunas ocasiones se utilizan equipos de pruebas independientes, para ayudar a validar el trabajo realizado a lo largo del ciclo de vida.
- Integrador/Operador:
  - Encargado de adaptar y construir todo el sistema desde los diversos subsistemas.

Todos estos roles se agregar en al developer-team

Artefactos

lunes, 5 de febrero de 2024 16:22

Introducción

Estos artefactos juegan un papel crucial en SCRUM al proporcionar transparencia, orientación y estructura al proceso de desarrollo. El Product Backlog y el Sprint Backlog ayudan a gestionar el trabajo, mientras que los Burndown Charts y el Incremento permiten al equipo monitorizar y evaluar su progreso hacia los objetivos del proyecto.

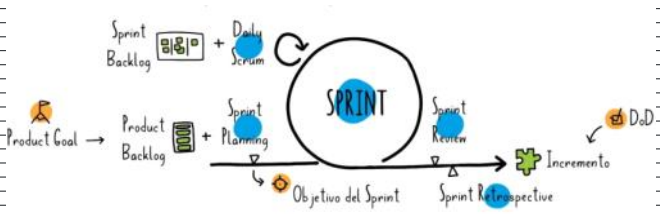
- Product backlog
- Sprint backlog
- Burndown charts
- Incremento

Grabación de audio iniciada: 16:25 lunes, 5 de febrero de 2024

0- 40 Sprints
40- 1: Artefactos
54- 1:00:00 Explicacion de lo que se hara el viernes app poker

Leer hasta diapositiva 115

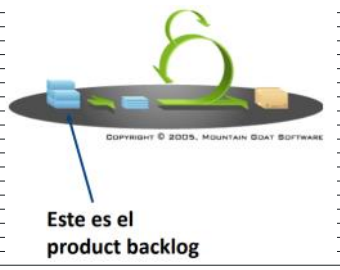
Por equipo:
- Goma
- Cartulinas A4 cualquier color
- Regla
- tijeras



Product backlog

Es una **lista dinámica** y priorizada de todas las funcionalidades, mejoras, correcciones y requisitos pendientes para el producto.

- Se mantiene y gestiona por el Product Owner y se utiliza para planificar y priorizar el trabajo en los Sprints.
- Re priorizada al comienzo de cada Sprint



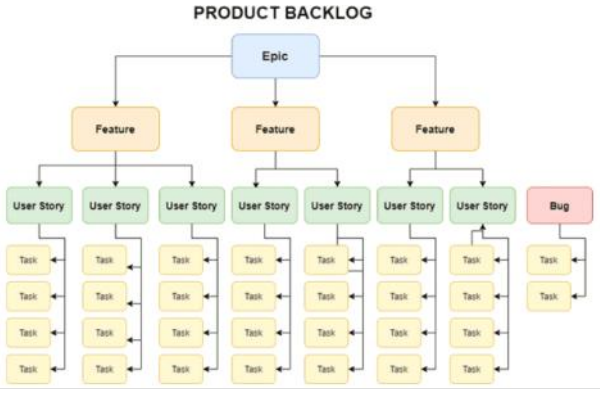
Pasos generales para la creación:

1. **Identificación de Requisitos y Funcionalidades:**
   - El Product Owner trabaja en estrecha colaboración con las partes interesadas para identificar y recopilar todos los requisitos y funcionalidades potenciales del producto.
2. **Priorización:**
   - Una vez recopilados, el Product Owner prioriza los elementos del Product Backlog en función de su valor para el cliente y para el éxito del producto. Esto se hace teniendo en cuenta factores como los objetivos del negocio, las necesidades de los usuarios y la viabilidad técnica.
3. **Desglose en Historias de Usuario:**
   - Los elementos del Product Backlog suelen ser desglosados en historias de usuario, que son descripciones cortas y centradas en el usuario de una funcionalidad o requisito específico. Estas historias de usuario representan las necesidades y expectativas del usuario final.
4. **Estimación:**
   - Una vez desglosados, el equipo puede estimar el esfuerzo necesario para completar cada historia de usuario. Esta estimación puede basarse en puntos de historia, horas de trabajo o cualquier otro método de estimación utilizado por el equipo.
5. **Refinamiento Continuo:**
   - El Product Backlog es un artefacto dinámico que se actualiza y refina continuamente a medida que se obtiene más información y se clarifican los requisitos. El Product Owner trabaja con el equipo y las partes interesadas para revisar y ajustar la priorización y los detalles de los elementos del Product Backlog según sea necesario.

Jerarquía en los Product Backlog

Épica
son grandes unidades de trabajo que representan funcionalidades o áreas del producto que son demasiado grandes para implementarse en un solo Sprint. Son objetivos de alto nivel que requieren desglose adicional en funcionalidades más pequeñas.

Feature (Funcionalidad)
son unidades de trabajo más pequeñas que las Épicas, pero aún representan una parte significativa del producto. Son funcionalidades coherentes que pueden implementarse en uno o varios Sprints.



Historias de Usuario
son descripciones cortas y centradas en el usuario de una funcionalidad o requisito específico del producto. Se redactan desde la perspectiva del usuario final y describen cómo el usuario interactuará con el sistema para lograr un objetivo.

Task (Tarea)
Son las unidades de trabajo más pequeñas y concretas del Product Backlog. Representan las actividades específicas que deben completarse para implementar una Historia de Usuario o una parte de una Funcionalidad.

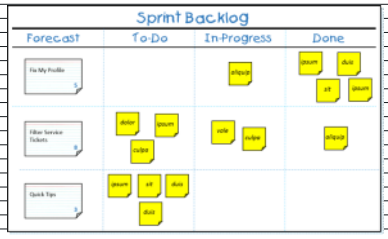
Trazabilidad General:

- Las Épicas representan objetivos de alto nivel y se desglosan en Funcionalidades.
- Las Funcionalidades son desgloses más detallados de las Épicas y se desglosan en Historias de Usuario.
- Las Historias de Usuario representan requisitos específicos del producto y se desglosan en Tareas, que son las acciones concretas para implementar esas Historias de Usuario.

Sprint Backlog

Es una lista de tareas específicas que el equipo selecciona del Product Backlog para completar durante un Sprint.

Herramientas
- Tableros KANBAN



- Se crea durante el Sprint Planning Meeting
- Se puede actualizar, añadir, borrar el Sprint Backlog durante el Sprint según sea necesario, por cualquier miembro.
- Las estimaciones las hacen los propios DEVELOPER TEAMS ya que ellos son los que chambean
  Reforzar el compromiso de
  - Todo el equipo respecto a las fechas que dan al cliente.
  - Cada miembro del equipo respecto al resto.
  - Hacer que todo el mundo se sienta oído.
-

¿Cómo Estimar?

La estimación del trabajo restante es actualizada diariamente

Hay diferentes métodos:

1. **Días Ideales:**
   Cada tarea se estima en la cantidad de días ideales o jornadas de trabajo necesarias para completarla. Se basa en la capacidad del equipo y en la suposición de que el trabajo se realiza de manera ininterrumpida.
   - Ejemplo: Si una tarea se estima en 2 días ideales y el equipo tiene una capacidad de trabajo de 6 horas al día, la tarea se consideraría que llevará 2 días en completarse.
   -

Tareas	L	M	M	J	V
Codificar UI	8	4	8		
Codificar negocio	16	12	10	4	
Testear negocio	8	16	16	11	8
Escribir ayuda online	12				
Escribir la clase foo	8	8	8	8	8
Agregar error logging			8	4	

2. **Planning Poker (Póker de Planificación):**
   Cada miembro del equipo estime el esfuerzo necesario para completar una tarea utilizando una baraja de cartas de póker con valores que representan puntos de historia. Se discuten las diferencias en las estimaciones y se llega a un consenso.
   - Ejemplo: Si se utiliza una baraja de cartas de póker, un miembro del equipo podría estimar una tarea como un "3", mientras que otro podría estimarla como un "5". Se discuten las razones de las estimaciones y se llega a un consenso sobre la estimación final.
3. **Talla de Camisas (Shirt Sizes):**
   Las tareas se estiman utilizando tallas de camisa que representan diferentes niveles de esfuerzo o complejidad. Las tallas de camisa pueden ser pequeñas, medianas, grandes, etc.
   - Ejemplo: Una tarea que se estima como una "M" (mediana) indica que tiene un nivel de esfuerzo o complejidad moderado, mientras que una tarea estimada como "XL" (extra grande) indica que es más compleja y requerirá más esfuerzo.

Otros métodos de estimación que también se utilizan en SCRUM incluyen Fibonacci, donde los valores de las estimaciones siguen la secuencia de Fibonacci (1, 2, 3, 5, 8, 13, etc.)

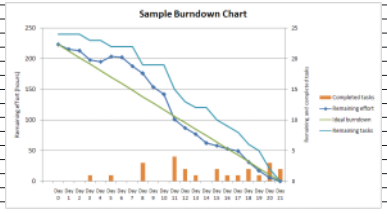
Reuniones

1. Reunión
   Las primeras cuatro horas se dedican al Product Owner
   - Establecer la meta del Sprint
   - Identificar la funcionalidad que se va a construir en el Sprint
2. Reunión
   Las segundas cuatro horas el equipo planea su propio Sprint
   - Se identifican y estiman las tareas para satisfacer
   - Se crea un Sprint Backlog
   - Las tareas son distribuidas por decisión de los miembros del equipo
   - Los miembros del equipo se comprometen a cumplir con la meta del Sprint

Entradas		Salida
• Backlog del producto actualizado • Retorno del último Sprint • Rendimiento del equipo en los Sprints anteriores		Pila del sprint (Sprint Backlog)

Burndown Charts

Son gráficos visuales que muestran la cantidad de trabajo restante en el Sprint Backlog a lo largo del tiempo. Ayudan al equipo a rastrear su progreso y a identificar cualquier desviación en el ritmo de trabajo planificado.



Herramientas

- Gráfico Burn-Up.
  Utilizado por el Product Owner.
  Muestra: las versiones previstas de un producto, funcionalidades de cada una de ellas, velocidad estimada, fechas probables para cada versión, margen de error previsto en las estimaciones, y avance real
- Gráfico Burn-Down.
  Utilizado por el Scrum Team para seguimiento del trabajo de cada Sprint

Incremento

Es la suma de todas las historias de usuario completadas y aceptadas durante un Sprint. Representa una versión potencialmente entregable del producto que se ha mejorado y ampliado con cada Sprint.

- Demostración de los objetivos alcanzados en cada sprint

Release

- No es necesario liberar el incremento al final del Sprint, pero debe estar en estado liberable.
  - Dependiendo de la situación, el propietario del producto decide si se lanzará o no. La decisión de liberar el Incremento está en manos del Product Owner, pero debe estar en estado utilizable/liberable.

Sprint

viernes, 2 de febrero de 2024 16:33

Destripando a Sprint

Es un breve espacio de tiempo para hacer SCRUM
Cada sprint debe cumplir el mismo tiempo, a diferencia de XP que en este puede variar las fechas del sprint.

El proceso de los Sprints en SCRUM sigue un ciclo iterativo que se repite en cada Sprint.
La imagen muestra solo un Sprint de muchos.

General

- Cada iteración o vueltita, se llama Sprint
- Realizar una revisión de los requisitos con los stakeholders
- El producto es diseñado, codificado y probado en el Sprint

Dentro de un sprint

- SCRUM gestiona la evolución del proyecto mediante reuniones breves de seguimiento en las que se revisa el trabajo realizado desde el hito anterior y los planes para el hito siguiente

Tiempos y Consistencia a base de:

- Ser eventos de **LONGITUD FIJA**, limitada a un mes, siendo lo normal de trabajar de 2 a 3 semanas.
- Diarias
  Se realiza una reunión diaria llamada Daily Scrum, y al final del Sprint, se realiza una reunión de revisión de Sprint, llamada Sprint Review
- Suelen ser muchas reuniones
  - La alternativa (juntarse a demanda, sin saber de antemano cuándo y con agendas cambiantes) es infinitamente peor.
- No hay cambios en un sprint. Planee la duración del sprint entorno a cuánto tiempo usted puede comprometerse a mantener los cambios fuera del sprint.

Existiendo las siguientes reuniones:

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Es importa el tamaño

Objetivo

SER una guía clara que orienta las actividades y decisiones del equipo durante el Sprint, permitiendo un enfoque cohesivo y efectivo en la consecución de metas específicas dentro de un plazo definido.

Recordemos que el fin de SCRUM es: ofrecer y crear el más alto valor de negocio, por ende cada Sprint tiene como objetivo dar ese valor.

Roles y Sprints

Solo el SCRUM MASTER puede cancelar el Sprint cuando:

- La tecnología no funciona

- Las circunstancias del negocio cambiaron
- El equipo tuvo interferencias

DEVELOPER TEAMS
- Un incremento de funcionalidad

Sprint Planning

Meeting
Como todo es un proceso en el cual

Paso a paso:
1. Es una reunión al inicio de cada Sprint donde el equipo y el Product Owner planifican el trabajo a realizar durante el Sprint:
   El equipo selecciona los temas a partir del Product backlog que pueden comprometerse a completar
2. Creamos el Sprint Backlog
   - Se identifican tareas y cada una es estimada (1-16 horas)
   - Realizado colaborativamente
3. El diseño de Alto Nivel es considerado

Trazabilidad : Planning a Daily
Se establecen los objetivos del Sprint y se seleccionan las historias de usuario del *Product backlog* que se abordarán durante el Sprint. Se definen las tareas necesarias y se asignan a los miembros del equipo.

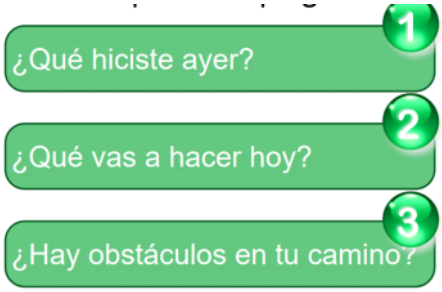
Daily Scrum Meeting

El SCRUM TEAM comparte el progreso, las dificultades y los planes para el día siguiente.

Parámetros:
- Diaria: solo hay una reunión de Daily Scrum programada por día.
- Dura 15 minutos
- Parados

Considerar:
- No es para resolver problemas
- No es reportarse al SCRUM MASTER
- Todos invitados, todo pueden hablar

TODOS responden:



Ahora

Sprint Review

Al finalizar el Sprint, se lleva a cabo una revisión para demostrar el trabajo realizado al Product Owner y otras partes interesadas.

Roles:

- Scrum Team participa y presenta lo realizado durante el sprint
  Informal
  - Duración de 4 horas.
    - Regla de 2 hs preparación
  - No usar diapositivas
  - Van los stakeholders

- Product Owner
  - Se presenta al product owner y a los implicados todas las funcionalidades implementadas
  - Los miembros del equipo presentan las funcionalidades
  - trata con los asistentes y con el team las posibles modificaciones en la pila de producto

- SCRUM MASTER
  - Anuncia el lugar y la fecha de la próxima revisión del Sprint, esto Al finalizar

Reglas

- Las funcionalidades no finalizadas completamente no se presentan
- Al final de la reunión se interroga individualmente a todos los asistentes para recabar impresiones, sugerencias de cambio y mejora, y su relevancia.

Al finalizar:

- El Product Owner decide si la funcionalidad presentada cumple con los objetivos del Sprint
- Se actualiza y vuelve a priorizar el Product Backlog

Trazabilidad : Review y Retrospective

Después del Sprint review y antes de la próxima Sprint planning meeting, el ScrumMaster convoca a una Sprint retrospective del Sprint con el Team.

Sprint Retrospective

Identificar que cosas se pueden cambiar para hacer el trabajo más agradable y productivo en las próximas iteraciones, por lo que se realiza al finalizar el Sprint

Reglas

- Normalmente 15 a 30 a 90 minutos
- Se realiza luego de cada sprint
- Participa el SCRUM TEAM

Roles:

- ScrumMaster
  - Hace que el Team revise, su proceso de desarrollo Scrum, para hacerlo más eficaz y eficiente para el próximo Sprint.
  - No proporciona respuestas, sino que ayuda al equipo a encontrar la mejor forma de trabajar con Scrum.

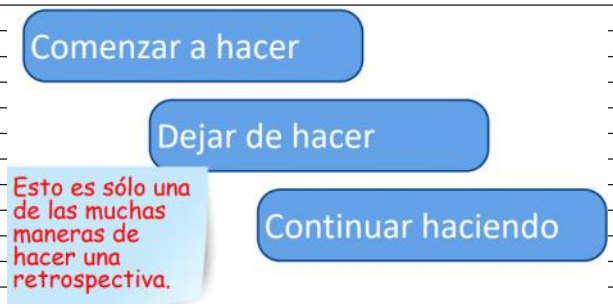
Responder:

- Que cosas hicimos bien?
- En qué cosas podemos mejorar?

Maneras de hacer retrospectiva:

Start - Stop - Continue

Todo el equipo se reúne y discute lo que les gustaría:



Ciclo de los Sprints:

- Después de la Sprint Retrospective, el equipo inicia un nuevo Sprint Planning Meeting para el próximo Sprint, seleccionando nuevas historias de usuario y estableciendo nuevos objetivos.
- El equipo trabaja en las tareas planificadas durante el Sprint, realizando Daily Scrums para mantenerse actualizado y resolver problemas.
- Al final del Sprint, se lleva a cabo una Sprint Review para mostrar el trabajo realizado y recibir retroalimentación.
- Posteriormente, se realiza una Sprint Retrospective para reflexionar sobre el Sprint y buscar mejoras.
- El ciclo continúa con un nuevo Sprint Planning Meeting para iniciar el siguiente Sprint.

Resumen

1. Sprint Planning

Definir qué se hará en el Sprint y cómo se logrará.

Duración Sugerida: 1-2 horas por semana de duración del Sprint.

Temas Clave para Cubrir:

- Selección de Tareas: Explicar cómo el equipo selecciona elementos del Product Backlog para incluir en el Sprint Backlog.
- Definición de Objetivos: Enfatizar la importancia de establecer metas claras y alcanzables para el Sprint.
- Estimaciones: Discutir cómo el equipo estima el esfuerzo necesario para cada tarea.
- Planificación Detallada: Ilustrar cómo se desglosan las tareas en actividades más pequeñas y se asignan a los miembros del equipo.

2. Daily Scrum Meeting

Objetivo: Sincronizar las actividades del equipo y planificar el trabajo del día siguiente.
Duración Sugerida: 15 minutos.
Temas Clave para Cubrir:

- Formato: Explicar la estructura de la reunión: qué hizo ayer, qué hará hoy, y si hay algún impedimento.
- Propósito: Aclarar que el objetivo es la transparencia y la identificación temprana de problemas.
- Participación: Resaltar la importancia de la participación de todos los miembros del equipo de desarrollo.

3. Sprint Review

Objetivo: Presentar y revisar el trabajo completado durante el Sprint.

Duración Sugerida: 1 hora por cada semana de duración del Sprint.
Temas Clave para Cubrir:

- Demostración: Mostrar lo que se ha completado durante el Sprint.
- Feedback del Product Owner y Stakeholders: Discutir cómo se recoge y se integra el feedback.
- Revisión del Backlog: Explicar cómo el Product Backlog se actualiza basado en el trabajo completado y el feedback recibido.

4. Sprint Retrospective

Reflexionar sobre el Sprint pasado para mejorar en el próximo.

Duración Sugerida: 45 minutos por cada semana de duración del Sprint.
Temas Clave para Cubrir:

- Evaluación del Proceso: Discutir qué fue bien y qué se puede mejorar en términos de proceso.
- Dinámica del Equipo: Hablar sobre cómo el equipo trabajó juntos y cómo se puede mejorar la colaboración.
- Acciones de Mejora: Identificar acciones concretas para mejorar en el próximo Sprint.

Fortalezas de SCRUM

martes, 13 de febrero de 2024 22:46

Seguimiento tradicional vs SCRUM

El seguimiento tradicional en los proyectos se da enfocado al tiempo que ya se ha trabajado. En Scrum se ve el avance mediante el tiempo que queda para terminar.

De esta manera tenemos la oportunidad de recalcular el plan de trabajo y la rapidez con que hacemos el trabajo. Así,el equipo puede decidir si trabajar más rápido o más lento para lograr el objetivo de la fecha.

CALIDAD EN SCRUM

El Testing nos ayuda a validar que lo que se solicito como un requerimiento se este cubriendo en el sistema.

Pruebas de Software común:

- ✓ Pruebas Unitarias
- ✓ Pruebas Funcionales
- ✓ Pruebas de Regresión
- ✓ Pruebas de Humo

**Las pruebas nos ayudan a conocer la salud del proyecto y a decidir si liberar o no el producto.

Valores del equipo

- ***Respeto***
  Cada uno es igual sin importar el titulo o experiencia
- ***Proactividad***
  No esperes a que te digan que hacer
- ***Receptividad***
  Buscar aprender nuevas habilidades, conocimientos para ser funcional.
- ***Foco***
  Céntrate en esto, cuando estamos en SCRUM solo existe el SCRUM

CE-ARC
Compromiso
Enfoque
Adaptabilidad
Respeto
Coraje