

Caso de estudio: Scrum Mastery

Eres la nueva Scrum Master del equipo que trabaja en el desarrollo de la app intercambiagram. Tu primer reto será analizar las dinámicas de trabajo del equipo para identificar antipatrones.

Situación actual

El Product Owner está distante al equipo (Si el PO está distante, no hay transparencia; ni los Devs ni el SM pueden que tiene de cara el PO hacia el futuro), no participa de las reuniones (Está bien si no participa de la Daily Scrum, pero es un antipatrón que no se presente en la Sprint Planning) y solo se presenta en la Sprint Review para conocer qué hizo el equipo durante el Sprint (Es un antipatro conocer que hizo el equipo durante el Sprint, se tiene que obtener Feedback del Cliente Final, Es decir, no es para seguimiento, es para inspeccionar el crecimiento del Producto, para obtener un buen feedback de cliente, tanto directo como indirecto). Durante la Sprint Planning reparte trabajo a cada Developer según las capacidades de cada uno.

De vez en cuando el PO se presenta en las Daily Scrum y cada Developer le presenta sus avances del día anterior (Esto es un antipatrón, no debería pasar nunca, el Daily Scrum es principalmente para los Devs, o quienes estén muy interesados en cara al objetivo final, de lo contrario, se puede entorpecer el proceso, además, no deberían mostrar sus avances los Devs, sino que tienen que ver si están avanzando por el camino correcto, para saber si van a cumplir o no. Aca se cumplen los 3 valores de Scrum, Transparencia, Inspección y Adaptación, Transparencia -> para que se sepa que se está haciendo, siempre importante. Inspección -> que vamos a hacer con esta información, nos estamos acercando al objetivo o no. Adaptación -> Necesitamos cambiar el plan que definimos en la Sprint Backlog). El PO toma nota y detecta que uno de los Developers el día anterior había presentado los mismo avances que reporta en esa daily Scrum. El PO reclama inmediatamente ante todos, que ese Developer no hizo nada el último día, ya que no presenta avances nuevos (Si no está presentando cambios nuevos, está perdido, no es claro el camino a seguir.)

El PO gestiona su Product Backlog en un documento de Excel que está alojado en su computador personal ([Antipatrón de Transparencia](#)). Se sospecha que cada vez que su jefe pide alguna nueva característica de producto ([El PO es el representante del Cliente Final y no su Jefe](#)), este lo incluye en el Backlog por orden de llegada. Aunque los pedidos del vicepresidente siempre tienen máxima prioridad porque el vicepresidente no le gusta ser desafiado y todos tienen que hacerle caso.

Cuando llega la Sprint Review los Developers proyectan en un archivo de power point la cantidad de historias de usuario terminadas, el % de cumplimiento (historias terminadas / historias comprometidas), el total de puntos de historia completados, y un informe detallado de las historias terminadas ([Antipatrón. En el Sprint Review los Devs no tienen que proyectar nada, sino que el Producto tiene que ser utilizado por el Cliente Final y, estos, tienen que recibir el Feedback. Las cosas técnicas al Cliente Final no funcionan.](#))

En la Sprint Review los Developers presentan los resultados únicamente al PO. El PO prefiere no involucrar stakeholders a estas sesiones porque nunca entienden sobre lo que se está trabajando, además no tienen el producto completo terminado para mostrar. ([Antipatrón. Los Stakeholders como clientes tienen que estar involucrados en Review del Producto](#))

El Product Owner de vez en cuando pregunta durante la Sprint Review de forma aleatoria a algún Developer sobre su aporte durante el Sprint ([Antipatrón. El PO no debería preguntar a ningún Dev sobre su aporte durante el Sprint.](#))

Durante la Sprint Planning el PO muestra su backlog ([Antipatrón. Aquí no habría Transparencia](#)), pero reparte ([Antipatrón. No debería repartir, no habría transparencia entonces, además, el Scrum Team no puede planificar](#)) historias a cada Developer para que este la detalle mejor y la escriba en formato de historia de usuario. A continuación, el Developer más senior define los story points de cada historia.

El anterior Scrum Master, era responsable de enviar un correo al finalizar cada evento con copia al PO, al jefe del PO y algunos stakeholders con los avances y conclusiones de la reunión.

Observas que a lo que llama este equipo Sprint Backlog, se trata de la lista de historias asignadas para el Sprint por el PO (Antipatrón. No hay planificación.)

El equipo no tiene un Definition of Done (Antipatrón. Si no tienen un Definition of Done, no tienen una lista del trabajo terminado, porque cuando lo intentaron incorporar no encontraron valor en hacerlo, parecía una pérdida de tiempo.

Cuando facilitas tu primera Sprint Retrospective te enteras de que solo 2 developers asisten, los demás dicen estar muy ocupados para hacer "jueguitos" (Es la herramienta más poderosa en el entorno de trabajo de Scrum, para la Inspección y Adaptación sobre el trabajo. Si, en la Inspección, no hay un buen análisis, se estanca el rendimiento. Uno de los principios de la Agilidad, es buscar la mejora continua y esto mismo ocurre en este evento. No hay mejora del equipo, si no se la toman en serio todos los Devs.)

Las daily Scrum duran dos horas, porque cada Developer tarda entre 15 y 20 min presentando todos sus avances. Además siempre suelen llegar tarde la mayoría de los Developers. El Developer Senior pregunta a cada persona los avances y es quien dirige la conversación.

El PO te busca para pedirte que los Sprints que hasta ahora son de dos semanas, pasen a tener una duración de un mes porque hacer cada dos semanas los eventos de Scrum es ineficiente. Por otra parte, dos semanas no es suficiente para entregar valor. Su argumento es que la agilidad se trata de ser flexibles y no debemos ser tan fundamentalistas con la teoría porque el caso de ellos es diferente al de los demás. (Antipatron. Si no se están teniendo resultados es porque hay que volver al Sprint Planning; si no se está alcanzando el objetivo, se tiene que reevaluar la situación. Es muy importante que el ST cumpla con los objetivos del Sprint.)

Por último, el Developer senior te informa que el equipo decidió pasar a hacer las daily scrum cada 2 días, esto debido a que como las daily scrum duran tanto tiempo, les queda poco tiempo en el día para desarrollar. (Antipatron. Es

mejor la constancia, para que los Devs no pierdan tiempo en algo que no avanza hacia adelante.)

Proyecto del curso

De acuerdo con el caso anterior, identifica todos los antipatrones presentes y diseña tu Backlog de transformación ágil. Cada elemento de tu backlog debe ser una estrategia específica que solucione un antipatrón. A continuación, ordénalo por valor teniendo presente que el elemento más arriba en la lista corresponde a la intervención de más valor que podrías implementar.

Los antipatrones identificado son los siguientes:

1. Acercar al PO al equipo (que no esté distante), haciéndolo participar de la Sprint Planning.
2. Que en el Sprint Review, el PO, no pida conocer que hizo el equipo, sino que se centre en el Cliente Final y los Stakeholders, para poder obtener el importante Feedback.
3. El PO no tiene porque estar en la Daily Scrum a menos que sea muy importante, por lo tanto, que no participe a menos que sea muy importante. Además, los Devs no deberían rendirle cuenta a nadie, sino que se dediquen a ver si se está avanzando hacia los objetivos.
4. Tienen que volver hacia el Sprint Planning, los Devs, si no se puede avanzar con la Sprint.
5. Que el Jefe del PO haga cosas de su cargo, y, que el PO sea el representante del Cliente Final.
6. Que no se proyecte, ni técnico ni nada, sino que se entregue el Producto al Cliente Final y a los Stakeholders, para obtener el Feedback necesario.

7. Integrar a los Stakeholders en la Sprint Review, para que puedan dar Feedback al ST de cómo se están moviendo con el proyecto.
8. Promover más la planificación.
9. Crear una Definition of Done para poder saber qué trabajo está hecho, qué trabajo está terminado.
10. Hacer que el ST se tome en serio el Sprint Retrospective, es la fuente más importante de Inspección y Adaptación sobre el proyecto, para poder saber cómo nos estamos moviendo y cambiar para poder tener éxito.

Y, Ordenándolos del más importante al menos, serían los siguientes:

1. Prioridad 6
2. Prioridad 4
3. Prioridad 2
4. Prioridad 8
5. Prioridad 3
6. Prioridad 9
7. Prioridad 5
8. Prioridad 1
9. Prioridad 10
10. Prioridad 7