

Checklist para Scrum Masters

Fuente original : Michael James (mj4scrum@gmail.com).

<http://www.colabpro.com>

14 September 2007 (Revised 24 July 2012)

Traducción : José Vázquez Sánchez. (a113779@gmail.com)

<http://www.gestiondeproyectosit.es>

[@PepeVazquezS](#)

(09 Marzo de 2013)

INTRODUCCIÓN

Sobre la dedicación del Scrum Master.

Un ScrumMaster adecuado puede manejar dos o tres equipos a la vez. Si te sientes satisfecho con limitar tu rol a la organización de reuniones, la aplicación de timeboxes en las dinámicas Scrum, y con responder y ayudar a la gente con los impedimentos que explícitamente se declaren, podrías llegar a funcionar con la dedicación a tiempo parcial en esta función.

El equipo del cual seas Scrum Master muy probablemente seguirá siendo superior a la media, y podrá satisfacer las expectativas en su organización aplicando Scrum, y probablemente no ocurrirá nada catastrófico en el proyecto con tu dedicación parcial. Sin embargo, si eres capaz de imaginar un equipo que tiene una gran oportunidad para lograr cosas que nadie creía posible, dentro de una organización transformada, deberías considerar que no puedes ser solamente un Scrum Master adecuado, debes ser un Gran ScrumMaster.

Y un gran ScrumMaster puede manejar un solo equipo a la vez.

Se recomienda un ScrumMaster dedicado por cada equipo de unos siete miembros, sobre todo cuando el equipo está empezando con Scrum.

Si aún no has descubierto todo el trabajo que hay que hacer para el desarrollo del proyecto y/o producto, sintoniza con el propietario del producto, con tu equipo, con las prácticas de ingeniería, y con la organización que está fuera del equipo.

Si bien no hay una receta única para todos, esta checklist describe las cosas típicas que los ScrumMasters suelen pasar por alto.

Checklist para Scrum Masters

Parte I - ¿Cómo lo está haciendo el Product Owner?

Los Scrum Masters ayudan a mejorar el trabajo del Product Owner, ayudándoles a encontrar maneras de mantener el Product Backlog y el Release Plan. (Ten en cuenta que sólo el dueño del producto puede dar prioridad a la los PBI (Product Backlog Items))

- ¿Está el Product Backlog priorizado de acuerdo con lo que transmite el Product Owner?.
- ¿Estan los requisitos y deseos de todos los interesados capturados en el Product Backlog ? Recuerda: el backlog es emergente.
- ¿Es el Product Backlog de un tamaño manejable? Para mantener un número manejable de elementos, mantener las cosas con mayor detalle en la parte superior, con las epicas en la parte inferior. Es contraproducente sobreanalizar demasiado los PBI que estén en la parte baja de la Pila de Producto. Los requisitos cambiarán en las conversaciones en curso entre desarrollo de productos y los actores / clientes.
- ¿Cumplen los requisitos (especialmente los que están cerca de la parte superior de la Pila de Producto) la definición INVEST de las historias de usuario?

Independiente, Negociable, de Valor, eStimable, pequeño (Smart) , y Testeable?
- Has educado al dueño del producto sobre la deuda técnica y cómo evitarla?. Una manera de afrontarla puede ser escribir pruebas automatizadas y refactorizaciones para cada PBI que haya tenido dificultades y se haya "parcheado" para evitar retrasos en la entrega.
- Es el Product Backlog un radiador información, claramente visible y accesible para todos los interesados?
- Si estás utilizando una herramienta automatizada para la gestión del Backlog, sabe el PO cómo usarla fácilmente?.Las herramientas de gestión automatizadas introducen el peligro de convertirse en refrigeradores información sin la irradiación activa del ScrumMaster.
- ¿Puedes ayudar a irradiar información, mostrando las impresiones y opiniones sobre el Backlog de todos los interesados?

Checklist para Scrum Masters

- ¿Puedes ayudar a irradiar la información mediante la creación esquemas o diagramas gráficos para para transmitir la visión
- ¿Has ayudado al Product Owner a organizar los elementos del backlog en releases o categorizaciones en base a temas?.
- ¿ Ayudas al Product Owner en el Grooming del Product Backlog y facilitas la sesión con todo el equipo para sanear el Backlog y prepararlo de cara al siguiente Sprint?.
- ¿Te aseguras que todas las historias están correctamente preparadas para el Siguiete Sprint y no hay ambigüedades en la definición que puedan retrasar la descomposición en tareas durante el Sprint Planning?
- ¿Conoce todo el mundo el plan de liberación y este sigue coincidiendo con la realidad?. Debes tratar de mostrar a todos los Gráficos Burndown de la release y mantenerlo actualizaados después de que los PBI han sido reconocidos como "Done" en cada reunión de revisión de Sprint. Los gráficos que muestran tanto la tasa de PBIs realmente obtenidos como los nuevos añadidos, permiten el descubrimiento temprano de cambios de alcance durante el Sprint.
- ¿El Product Owner ajusta el plan de Releases después de la última reunión de revisión Sprint?. La mayoría de Product Owners que entregan sus productos a tiempo y suficientemente probados tienen la buena costumbre de volver a planear cada Sprint. Esto probablemente requiere diferir un poco de trabajo para futuras versiones, pero permite que el trabajo más importante sea descubierto antes de comenzar el siguiente Sprint.

Checklist para Scrum Masters

Parte II - ¿Cómo lo está haciendo el equipo?

Aunque al Scrum Master se le anima a trabajar fomentando la colaboración con los miembros del equipo, existe el riesgo de que éstos se pierdan en las tareas técnicas. Ten en cuenta tus principales responsabilidades para con el equipo:

- ¿Esta tu equipo en el estado de flujo? Algunas características de este estado son :
 - Objetivos claros (las expectativas y las reglas son discernibles y las metas son alcanzables, alineadas apropiadamente con el conjunto de habilidades y capacidades de los miembros del equipo).
 - Concentración y enfoque: un alto grado de concentración por parte del equipo limitado a un campo de atención concreto.
 - Pérdida del sentimiento de auto-conciencia, la fusión de la acción y la conciencia.
 - Feedback directo e inmediato (los éxitos y los fracasos en el curso de la actividad son evidentes, por lo que el comportamiento se puede ajustar según sea necesario).
 - Equilibrio entre el nivel de habilidad necesaria y el desafío que supone conseguir los logros. (la actividad a realizar no es ni demasiado fácil ni demasiado difícil).
 - Un sentido de control personal sobre la situación o actividad a realizar.
 - La actividad es intrínsecamente gratificante, así que tampoco es excesivamente necesario motivar al equipo.

- ¿Los miembros del equipo parecen estar alineados y celebrar el éxito de los otros como suyo propio?

- ¿Los miembros del equipo se sienten responsables, y tienen espíritu de mejora?

- ¿Existen problemas / oportunidades que el equipo no está discutiendo porque están demasiado incómodos unos con otros?

- ¿Has probado una variedad de formatos y lugares para las reuniones de Retrospectiva del Sprint?

Checklist para Scrum Masters

- ¿El equipo mantuvo el enfoque en las metas del Sprint?. Tal vez deberías llevar a cabo una revisión a mediados de Sprint para volver a revisar los criterios de aceptación de los items del product Backlog comprometidos para este Sprint.
- ¿El SprintBacklog refleja lo que el equipo está haciendo?. Cuidado con la "materia oscura" de las tareas no reveladas y las tareas más grandes que un día de trabajo. Las tareas no relacionadas con los compromisos de Sprint son impedimentos para esos compromisos.
- ¿Tu equipo tiene entre 3-9 personas con una mezcla de las habilidades suficientes para construir un incremento del producto potencialmente entregable?
- ¿El tablero Scrum del equipo está actualizado y el equipo se siente cómodo con él?.
- Son los artefactos (Tablero de tareas, Sprint Burndown, lista de impedimentos, etc) visibles para el equipo, y adecuados para el equipo a producto que vas a desarrollar?
- ¿Están estos artefactos adecuadamente protegidos de entrometidos? El exceso de control de la actividad diaria de la gente fuera del equipo puede impedir la transparencia interna del equipo y la autogestión.
- ¿Los miembros del equipo se ofrecen voluntarios para las tareas?
- ¿Se ha hecho explícita la deuda técnica en los elementos del backlog, para poco a poco hacer que el código sea un lugar más agradable para trabajar?.
- ¿Los miembros del equipo dejan sus cargos en la puerta de la sala del equipo, son colectivamente responsables de todos los aspectos del trabajo acordado (pruebas, documentación del usuario, etc)?

Checklist para Scrum Masters

Parte III - ¿Cómo son nuestras prácticas de ingeniería?

- ¿Tiene tu ecosistema de desarrollo algún mecanismo que permite a cualquier persona (mismo equipo o un equipo diferente) detectar convenientemente cuando se ha causado un fallo de regresión (se ha roto previamente alguna funcionalidad que ya se había dado por concluida)?

Normalmente, esto se logra a través de la implantación de test xUnit, JUnit, NUnit, etc.

- ¿Hay un equilibrio adecuado de pruebas del sistema automatizadas de extremo a extremo (también conocidas como "pruebas funcionales") y pruebas automatizadas de código?.

- ¿El equipo elabora las pruebas del sistema y las pruebas unitarias en el mismo lenguaje que el sistema que estamos desarrollando? . La Colaboración no mejora con lenguajes de scripting propietarios o herramientas de reproducción de pruebas que sólo un subconjunto del equipo sabe cómo mantener.

- ¿El equipo descubrió la zona gris existente entre las pruebas del sistema y las pruebas unitarias?.

- ¿Tienes un servidor de integración continua que hace saltar una alarma cuando alguien provoca un fallo de regresión? Este bucle de retroalimentación puede ser reducido a horas o minutos? ("Builds diarios son para los débiles." - Kent Beck).

- ¿Hay alguna tarea del servidor de integración continua encargada de la ejecución de los Test?.

- ¿Tienes un sistema de BugTracking totalmente sincronizado con el IDE de desarrollo en el que los issues estén integrados en el propio IDE y el equipo pueda trabajar en ellos sin tener que cambiar entre distintas herramientas?.

- ¿Dispones de un sistema de Planificación Ágil en tu ecosistema de desarrollo que permita a cada miembro del equipo conocer todos los PBI que están comprometidos para el Sprint y visualizar de manera rápida el grado de avance de la release?.

Checklist para Scrum Masters

- ¿Los miembros del equipo han descubierto las ventajas del diseño continuo y la refactorización constante , como una alternativa de diseño emergente?. El Refactoring tiene una definición estricta: cambios en la estructura interna sin cambiar la conducta externa. El Refactoring debe ocurrir varias veces por hora, siempre que haya código duplicado, lógica condicional compleja (visible por sangrado excesivo o métodos largos), los identificadores mal nombrados, acoplamiento excesivo entre los objetos, etc . El Refactoring con confianza sólo es posible con una cobertura de pruebas automatizada adecuada. Descuidar la refactorización hace que sea difícil cambiar el producto en el futuro, sobre todo porque es difícil encontrar buenos desarrolladores dispuestos a trabajar en código incorrecto.

- ¿La definición de "hecho" para cada elemento del Product Backlog incluye cobertura completa de pruebas automatizadas y refactorización?. Las Técnicas de Test Driven Development (TDD) aumentan la probabilidad de lograr esto.

- ¿Hacen los miembros del equipo pair programming la mayor parte del tiempo?. La programación en parejas puede mejorar drásticamente el mantenimiento del código y reducir la tasa de errores. Desafía los límites de las personas y, a veces parece tomar más tiempo (si se mide por líneas de código en lugar de funcionalidad entregable). Predicar con el ejemplo iniciando la programación por parejas cada día de trabajo con los miembros del equipo. Algunos de ellos comienzan a preferir trabajar de esta manera.

Checklist para Scrum Masters

Parte IV - ¿Cómo lo está haciendo la organización?

- ¿Existe una comunicación adecuada entre distintos equipos que colaboran en la construcción de un mismo producto?. "Scrum de Scrums" es sólo una manera de lograr esto, y no necesariamente la mejor.
- ¿Los equipos son capaces de forma independiente de producir trabajo que funciona, abarcando incluso los límites de la arquitectura?.
- ¿Se reúnen entre sí los ScrumMasters, trabajando para resolver la lista de impedimentos de la organización?.
- En su caso, ¿están los impedimentos organizacionales visibles en la pared de la oficina del director de desarrollo? . ¿Puede ser cuantificado el costo de retraso, la pérdida que supone el retraso en la salida a tiempo del producto en el mercado, la pérdida de calidad, o la pérdida de oportunidades con los clientes? (Aprender de los errores : Ken Schwaber - "Un Scrum Master muerto es un ScrumMaster inútil.").
- ¿Está tu organización comprometida con planes de carrera que sean compatibles con los objetivos colectivos de sus equipos?. La respuesta es "no" si no hay un incentivo de carrera para hacer el trabajo de programación o arquitectura a expensas de las pruebas, automatización de pruebas o documentación del usuario.
- ¿Tu organización ha sido reconocida por la prensa u otras fuentes independientes como uno de los mejores lugares para trabajar, o un líder en su industria?.
- ¿Es tu organización una organización de aprendizaje?.