

Scrum Reference Card

de Michael James

Acerca de Scrum

Un Marco de Gestión

Scrum es un marco de gestión para el desarrollo incremental de productos, valiéndose de uno o más equipos multi-funcionales, auto-organizados, de aproximadamente siete personas cada uno.

Proporciona una estructura de roles, reuniones, reglas y artefactos. Los equipos son los responsables de crear y adaptar sus procesos dentro de este marco.

Scrum utiliza iteraciones de longitud fija que se denominan Sprints, que son típicamente de dos semanas o 30 días de duración. Los equipos Scrum intentan generar un incremento de producto potencialmente entregable (debidamente probado) en cada iteración.

Una Alternativa a la Cascada

El enfoque incremental e iterativo de Scrum cambia las fases tradicionales del desarrollo en "cascada" por la capacidad de desarrollar un subconjunto de características de alto valor en primer lugar, incorporando feedback temprano.

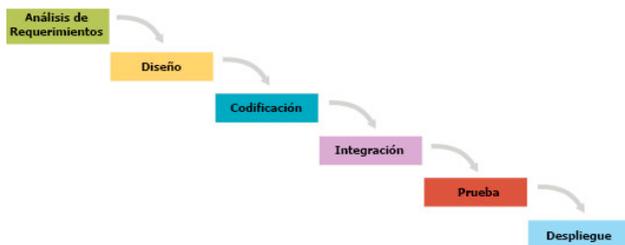


Figura 1: el desarrollo tradicional en "cascada" depende de una perfecta comprensión de los requisitos del producto desde el principio y un mínimo de errores en cada fase.

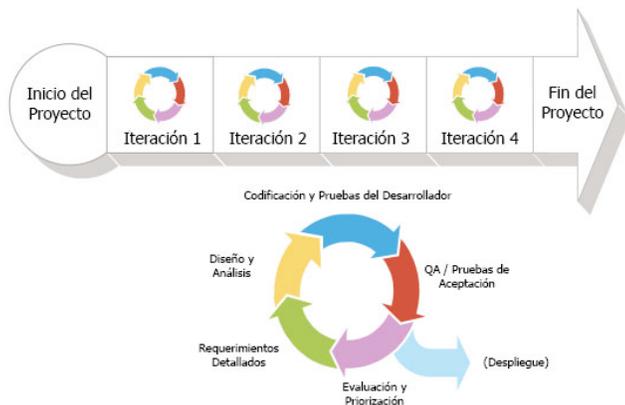


Figura 2: Scrum combina todas las actividades de desarrollo en cada iteración, adaptándose a las realidades emergentes, en intervalos fijos.

El mayor beneficio de Scrum se experimenta en el trabajo complejo que implica la creación de conocimiento y colaboración, tal como sucede en el desarrollo de nuevos productos. Scrum se asocia generalmente con el desarrollo de software orientado a objetos. Su uso también se ha extendido al desarrollo de productos como semiconductores, hipotecas y sillas de ruedas.

¿Haciendo Scrum o Pretendiendo Hacer Scrum?

Los chequeos de realidad incesantes de Scrum pueden exponer las limitaciones disfuncionales en los individuos, equipos y organizaciones. Muchos dicen hacer Scrum y modifican las partes que requieren traspasar los impedimentos organizacionales y terminan privándose a sí mismos de la mayoría de los beneficios.

Roles de Scrum

Product Owner

- Única persona responsable de maximizar el retorno de la inversión (ROI) del esfuerzo de desarrollo
- Responsable de la visión del producto
- Constantemente re-prioriza el Backlog del Producto, ajustando las expectativas a largo plazo, como los planes de liberaciones
- Es el árbitro final de las preguntas sobre requerimiento
- Acepta o rechaza cada incremento del producto
- Decide si se debe liberar
- Decide si se debe continuar con el desarrollo
- Considera los intereses de los stakeholders
- Puede contribuir como miembro del equipo
- Tiene un papel de liderazgo

Equipo de Desarrollo de Scrum

- Multifuncional (incluye miembros con habilidades de testing y a menudo otros no llamados tradicionalmente desarrolladores: analistas de negocio, expertos de dominio, etc.)
- Auto-organizado/auto-gestionado, sin roles asignados externamente
- Negocia los compromisos con el Product Owner, de un Sprint a la vez
- Tiene autonomía con respecto a la forma de lograr sus compromisos
- Intensamente colaborativo
- Tiene mayor probabilidad de éxito al encontrarse establecido en un mismo lugar, sobre todo para los primeros Sprints
- Tiene más éxito si el involucramiento con el equipo es a largo plazo y full-time. Scrum promueve evitar el traslado de personas o dividirlos entre otros equipos.
- 7 ± 2 miembros
- Tiene un papel de liderazgo

ScrumMaster

- Facilita el proceso de Scrum
- Ayuda a resolver los impedimentos
- Crea un ambiente propicio para la auto-organización del equipo
- Captura datos empíricos para ajustar las previsiones
- Protege al equipo de interferencias externas y distracciones para mantener el flujo del equipo (también conocido como *la zona*)
- Aplica los timeboxes
- Mantiene visibles los artefactos Scrum
- Promueve la mejora de las prácticas de ingeniería
- No tiene autoridad en la gestión del equipo (cualquier persona que tenga autoridad sobre el equipo no es, por definición, su ScrumMaster)
- Tiene un papel de liderazgo

Reuniones de Scrum

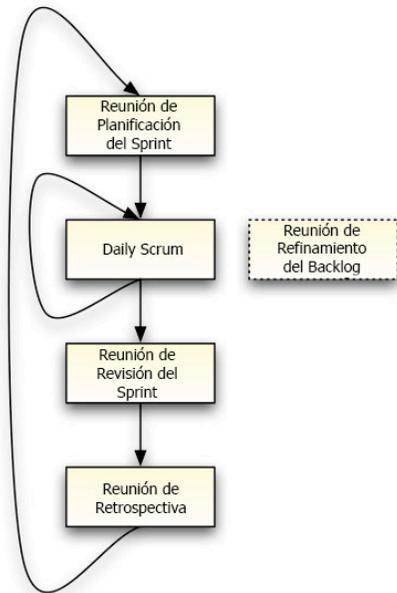


Figura 3: flujo de Scrum

Todas las reuniones de Scrum son facilitadas por el ScrumMaster, quien no tiene autoridad para tomar decisiones en ellas.

Reunión de Planificación de Sprint

Al comienzo de cada Sprint, el Product Owner y el equipo tienen una Reunión de Planificación del Sprint donde negocian qué ítems del Backlog del Producto intentarán convertir en producto funcionando durante el Sprint. El Product Owner es el responsable de declarar cuáles son los ítems más importantes para el negocio. El equipo es responsable de seleccionar la cantidad de trabajo que cree que podrán realizar sin acumular deuda técnica. El equipo "toma" el trabajo desde el Product Backlog hacia el Sprint Backlog.

Cuando los equipos se encuentran frente a un trabajo complejo con una incertidumbre inherente, deben trabajar juntos para intuitivamente estimar su capacidad para comprometerse con los ítems, mientras aprenden de los sprints anteriores. Planificar su capacidad por hora y comparar sus estimaciones con datos reales hace que el equipo simule precisión y reduzca su sentido de responsabilidad con de sus compromisos. A menos que el trabajo sea realmente predecible, se debe descartar esas prácticas dentro de los primeros Sprints o evitarlos por completo.

Hasta que un equipo haya aprendido a lograr un incremento del producto potencialmente entregable en cada Sprint, se debe reducir la cantidad de funcionalidad comprometida. El fracaso en cambiar viejos hábitos conduce a la deuda técnica y la muerte eventual del diseño, como se muestra en la Figura 15.

Si la parte superior del Backlog del Producto no ha sido refinada, una parte importante de la reunión de planificación se debe dedicar a esto, tal como se describe en la sección de la Reunión de Refinamiento del Backlog.

Hacia el final de la Reunión de Planificación del Sprint, el equipo desglosa los ítems seleccionados en una lista *inicial* de tareas del Sprint (*Sprint Tasks*) y hace un compromiso final para realizar el trabajo.

El tiempo máximo asignado (también conocido como *timebox*) para la planificación de un Sprint de 30 días, es de ocho horas, reducido proporcionalmente en caso de un Sprint más corto.

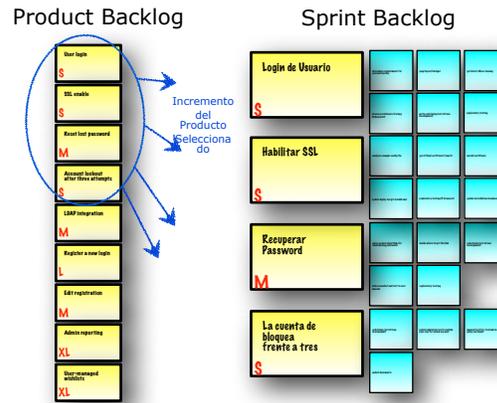


Figura 4: los resultados de la Reunión de Planificación del Sprint son ítems del Product Backlog (PBIs) comprometidos y Sprint Tasks subordinadas.

Daily Scrum y Ejecución del Sprint

Cada día, a la misma hora y en el mismo lugar, los miembros del equipo de desarrollo pasan 15 minutos reportándose entre sí. Cada miembro del equipo resume lo que hizo el día anterior, lo que hará hoy, y qué impedimentos está enfrentando.

Mantenerse de pie en el Daily Scrum ayudará a que sea breve. Los temas que requieren atención adicional pueden ser discutidos por los interesados después de que cada miembro del equipo haya reportado.

Al equipo puede resultarles útil mantener una lista de Tareas del Sprint (*Sprint Task List*), un Sprint Burndown Chart, y un Listado de Impedimentos. Durante la ejecución del Sprint es común descubrir tareas adicionales necesarias para alcanzar las metas del Sprint. Los impedimentos causados por los problemas que escapan al control del equipo se consideran *impedimentos organizacionales*.

Para el Product Owner, casi siempre es útil asistir a la Daily Scrum. Pero cuando alguno de los asistentes también es el jefe del equipo, el efecto del *arma invisible* impide la auto-organización y el liderazgo emergente. Las personas que carecen de la experiencia real de la auto-organización no verán este problema, al igual que los peces no son conscientes del agua. Por el contrario, un equipo que necesita experiencia adicional en los requisitos del producto se beneficiará de una mayor participación del Product Owner, incluyendo la asistencia a la Daily Scrum.

La Daily Scrum tiene la intención de alterar los viejos hábitos del trabajo por separado. Los miembros deben permanecer atentos para detectar signos de la antigua estrategia. Por ejemplo, mirar sólo en el ScrumMaster cuando se habla es un síntoma de que el equipo no ha aprendido a operar como una entidad auto-organizativa.

Reunión de Revisión del Sprint

Después de la ejecución del Sprint, el equipo mantiene una reunión de revisión del Sprint para mostrar un incremento del producto al Product Owner y a todos los demás interesados.

La reunión debe ofrecer una demostración en vivo, no un informe.

Después de la demostración, el Product Owner revisa los compromisos contraídos en la Reunión de Planificación del Sprint y declara qué ítems considera terminados. Por ejemplo, un ítem de software que es meramente "código completado" no se considera terminado, porque el software no probado no es entregable. Los elementos incompletos son devueltos al Backlog del Producto y clasificados de acuerdo a las prioridades del Product Owner como candidatos para futuros Sprints.

El ScrumMaster ayuda al Product Owner y a los stakeholders a convertir su feedback en los nuevos ítems del Product Backlog o Product Backlog Items (PBIs) para su priorización por el Product Owner. A menudo, descubrir nuevo alcance supera la tasa del equipo de desarrollo. Si el Product Owner determina que el alcance recién descubierto es más importante que las expectativas originales, el nuevo alcance desplaza el viejo alcance en el Product Backlog.

La reunión de revisión de Sprint es el encuentro apropiado para los stakeholders (incluso a los usuarios finales). Es la oportunidad de revisar y adaptar el producto a medida que emerge, y de forma iterativa refinar la comprensión de cada uno de los requisitos. Nuevos productos, especialmente productos de software, son difíciles de visualizar en el vacío. Muchos clientes necesitan ser capaces de reaccionar ante una pieza de software que funcione para descubrir lo que realmente quieren. El desarrollo iterativo, un enfoque dirigido por el valor, permite la creación de productos que no podrían haber sido especificados por adelantado en un enfoque dirigido por planes.

Reunión de Retrospectiva del Sprint

Cada Sprint finaliza con una retrospectiva. En esta reunión, el equipo reflexiona sobre su propio proceso. Inspeccionan su comportamiento y adoptan las medidas para adaptarlo en los futuros Sprints.

Los ScrumMasters dedicados encontrarán alternativas a las reuniones obsoletas, temidas, que todo el mundo espera. Una retrospectiva en profundidad requiere de un ambiente de seguridad psicológica que no se encuentra en la mayoría de las organizaciones. Sin seguridad, la discusión retrospectiva o bien evitará los problemas incómodos, o se deteriorará en la culpa y la hostilidad.

Un impedimento común para la plena transparencia en el equipo es la presencia de personas que realizan evaluaciones de desempeño.

Otro impedimento para una retrospectiva profunda es la tendencia humana a sacar conclusiones y proponer acciones con rapidez. *Agile Retrospectives*, el libro más popular sobre este tema, describe una serie de medidas para ralentizar este proceso: preparar el escenario, recabar datos, generar ideas, decidir qué hacer, cerrar la retrospectiva.¹ Otra guía recomendada para ScrumMasters, *The Art of Focused Conversations*, divide el proceso en pasos similares: objetivo, reflexivo, interpretativo y de toma de decisiones (ORID)².

Un tercer impedimento a la seguridad psicológica es la distribución geográfica. Equipos dispersos geográficamente por lo general no colaboran, así como aquellos que comparten una sala de equipo.

Las retrospectivas a menudo exponen impedimentos organizacionales. Una vez que un equipo ha resuelto los obstáculos dentro de su influencia inmediata, el ScrumMaster debe trabajar para expandir esa influencia.

El ScrumMaster debe utilizar una variedad de técnicas para facilitar retrospectivas, incluyendo la redacción en silencio, líneas de tiempo, e histogramas de satisfacción. En todos los casos, el objetivo es lograr una comprensión común de múltiples perspectivas y desarrollar acciones que llevarán al equipo al siguiente nivel.

Reunión de Refinamiento del Backlog

La mayoría de los PBIs inicialmente deben refinarse, ya que son grandes y poco comprendidos. Los equipos han encontrado útil tomar un tiempo de la ejecución de cada Sprint para preparar el Backlog del Producto para la próxima Reunión de Planificación del Sprint.

En la reunión de refinamiento del Backlog, el equipo estima la cantidad de esfuerzo que se debe invertir para completar los Ítems del Backlog del Producto y proporciona información técnica para ayudar al Product Owner a priorizarlos³. Los grandes ítems se dividen y se clarifican, teniendo en cuenta temas tanto de negocio como técnicos. A veces, un subconjunto del equipo, junto con el Product Owner y los stakeholders, descomponen y dividen los ítems del Backlog del Producto antes de involucrar a todo el equipo en la estimación.

Un ScrumMaster experimentado puede ayudar al equipo a identificar finas rebanadas verticales de trabajo que tienen valor comercial, mientras promueve una definición rigurosa de "terminado" que incluye las pruebas adecuadas y refactorización.

Es común escribir los PBIs en forma de *Historias de Usuario*⁴. En este enfoque, los PBIs de gran tamaño son llamados *Epics*. El desarrollo tradicional rompe las características en tareas horizontales (que se asemeja a fases en cascada) que no pueden ser priorizadas de manera independiente y que carecen de valor del negocio desde la perspectiva del cliente. Este hábito es difícil de cambiar.

La agilidad requiere aprender a dividir las Historias de Usuario que representan características más pequeñas del producto. Por ejemplo, en una aplicación para registros médicos la Epic "mostrar todo el contenido de los registros de alergia de un paciente a un médico" produjo la Historia "informar si existen o no registros de alergia existe". Mientras que los ingenieros prevén importantes desafíos técnicos en analizar los aspectos internos de los registros de alergias, la presencia o ausencia de algún tipo de alergia era la cosa más importante que los médicos necesitaban saber. La colaboración entre los empresarios y técnicos para dividir esta Epic produjo una Historia que representa el 80% del valor de negocio con el 20% del esfuerzo de la Epic original.

Como la mayoría de los clientes no utilizan la mayoría de las características de los productos, es conveniente dividir las Epics para entregar las historias más importantes primero. Aunque la entrega de características de menor valor más adelante es probable que incluya algún retrabajo, la repetición es mejor que la ausencia de trabajo.

La reunión de refinamiento carece de nombre oficial y también es llamada "Backlog Grooming", "Mantenimiento del Backlog" o "Story Time".

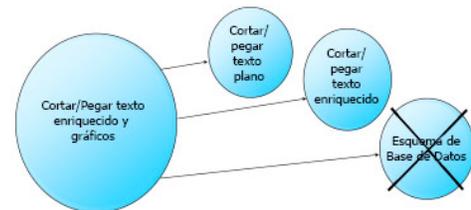


Figura 5: durante el refinamiento del Backlog, los PBIs grandes (a menudo llamados "Epics") cerca de la parte superior del Backlog del Producto se dividen en finas porciones de funciones verticales ("Historias"), y no fases de ejecución horizontales.

Artefactos de Scrum

Backlog del Producto

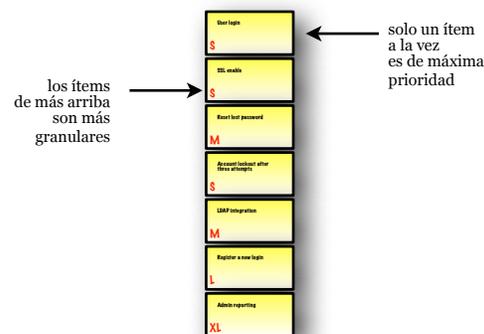


Figura 6: Backlog del Producto

- Lista ordenada de funcionalidad deseada
- Visible para todos los stakeholders
- Cualquier stakeholder (incluido el equipo) puede agregar ítems
- Constantemente re-priorizado por el Product Owner
- Los Ítems superiores son más granulares que los inferiores
- Mantenido durante la reunión de Refinamiento del Backlog

¹ "Agile Retrospectives", Pragmatic Bookshelf, Derby/Larson (2006)

² "The Art of Focused Conversations", New Society Publishers (2000)

³ El equipo debe colaborar para producir una estimación conjunta para cada ítem. Ver <http://blogs.danube.com/estimation-game>

⁴ "User Stories Applied: For Agile Software Development", Addison Wesley, Cohn (2004)

Ítems del Backlog del Producto (PBIs)

- Especifica el *qué* más que el *cómo* de una característica centrada en el cliente
- A menudo escrita en forma de *Historia de Usuario*
- Tiene una definición de "terminado" abarcadora de todo el producto para evitar la deuda técnica
- Puede tener criterios de aceptación específicos del ítem
- El esfuerzo es calculado por el equipo, de preferencia en unidades relativas (por ejemplo, puntos de la historia)
- El esfuerzo es de aproximadamente 2-3 personas 2-3 días, o menos para equipos más avanzados

La cuenta se bloquea ante tres intentos

Criterios de Aceptación:

Pequeña

Figura 7: un PBI representa una función centrada en el cliente, que normalmente requiere varias tareas para lograr la definición de hecho.

Backlog del Sprint

- Consiste en PBIs comprometidos negociados entre el equipo y el Product Owner durante la Reunión de Planificación del Sprint
- El alcance comprometido es fijo durante la ejecución del Sprint
- Las tareas iniciales son identificadas por el equipo durante la reunión de planificación del Sprint
- El equipo descubrirá las tareas adicionales necesarias para cumplir con el compromiso de alcance fijo durante la ejecución de Sprint
- Visible para el equipo
- Referenciado durante la Daily Scrum Meeting

Ítems del Backlog Comprometidos	Tareas No Iniciadas	Tareas En Progreso	Tareas Completadas
📄	📄📄📄	📄	📄📄
📄	📄📄📄	📄	
📄	📄📄📄		
📄	📄		

Figura 8: el Backlog del Sprint se representa a menudo con un "radiador de información", tal como un taskboard físico.

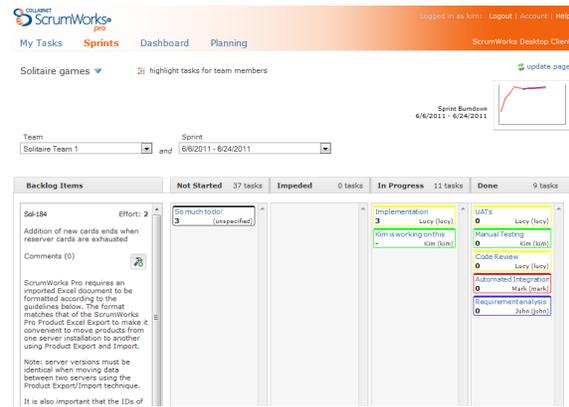


Figura 9: a veces se representa el Backlog del Sprint electrónicamente en una herramienta de colaboración.

Tareas del Sprint

- Especifica *cómo* alcanzar el *qué* del PBI
- Requiere aproximadamente un día de trabajo
- El esfuerzo restante se re-estima a diario, por lo general en horas
- Durante la ejecución de Sprint, una *persona de contacto* puede ofrecerse para ser el principal responsable de una tarea
- Propiedad de todo el equipo, se espera colaboración



Figura 10: las tareas del Sprint necesarias para completar un Ítem del Backlog se componen de un mix de actividades que ya no se realizan en fases separadas (por ejemplo: captura de requisitos, análisis, diseño, implementación, desarrollo, pruebas).

Sprint Burndown Chart

- Indica el total de horas restantes de las tareas del equipo dentro de un Sprint
- Re-estimado a diario, por lo tanto puede subir antes de bajar
- Diseñado para facilitar la auto-organización del equipo
- Algunas variaciones, como agrupar por persona de contacto o agregando tendencias, tiende a reducir su efectividad para fomentar la colaboración
- Parecía una buena idea en los primeros días de Scrum, pero en la práctica ha sido a menudo mal utilizada como un informe de gestión, invitando a la intervención. El Scrum Master debe descontinuar el uso de esta tabla si se convierte en un impedimento para la auto-organización del equipo.

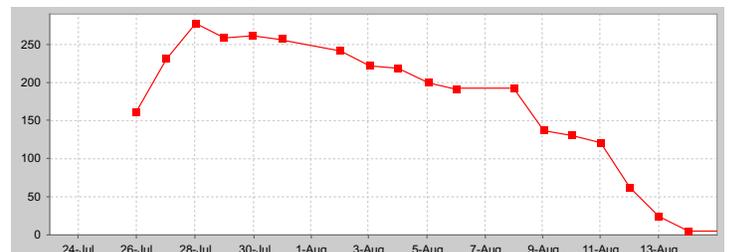


Figura 11: Sprint Burndown Chart

Product / Release Burndown Chart

- Realiza un seguimiento del esfuerzo restante del Product Backlog de un Sprint al próximo
- Puede utilizar unidades relativas como *Puntos de Historia* en el eje Y
- Utiliza tendencias históricas para ajustar las previsiones

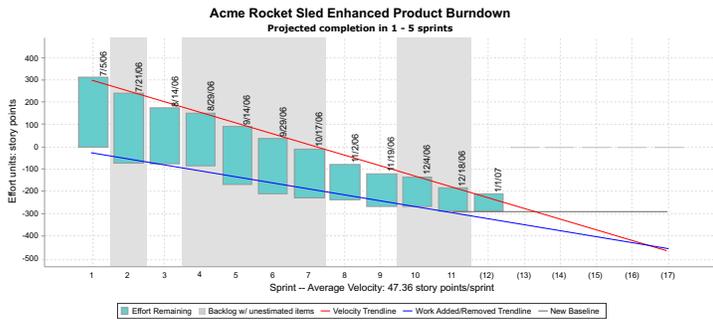


Figura 12: una variante del Release Burndown Chart popularizado por Mike Cohn. La línea roja sigue la trayectoria de los PBIs completados en el tiempo (velocidad), mientras que la línea azul sigue los nuevos PBIs añadidos (descubrimiento del nuevo enfoque). La intersección proyecta la fecha en que estará completa la entrega, basada en tendencias históricas.

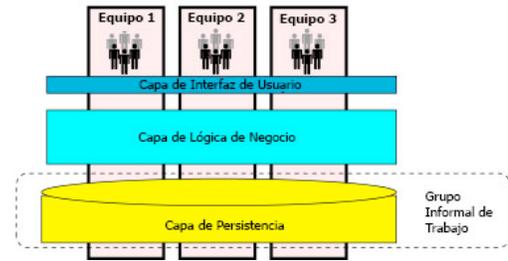


Figura 14: los equipos de características aprenden para abarcar los componentes arquitectónicos.

Más malas noticias: aún es difícil

La Agilidad representa un desafío particularmente para las grandes organizaciones. Creyendo hacer Scrum, muchas no han logrado superarlo⁵. Los ScrumMasters en las grandes organizaciones debían reunirse con regularidad, promoviendo la transformación a través de una lista visible de los impedimentos de organización, y leer libros como *Scaling Lean & Agile Development*⁶.

Escalamiento

Malas noticias: es duro

Scrum aborda los requisitos inciertos y los riesgos técnicos mediante la agrupación de personas de múltiples disciplinas en un solo equipo (idealmente en una única sala) para maximizar el ancho de banda de la comunicación, la visibilidad y la confianza.

Cuando los requisitos son inciertos y los riesgos técnicos son altos, agregar mucha gente a la situación empeora las cosas. Agrupar personas por especialidad también empeora las cosas. Agrupar a las personas por los componentes arquitectónicos (por ejemplo, equipos de componentes) empeora las cosas...con el tiempo.

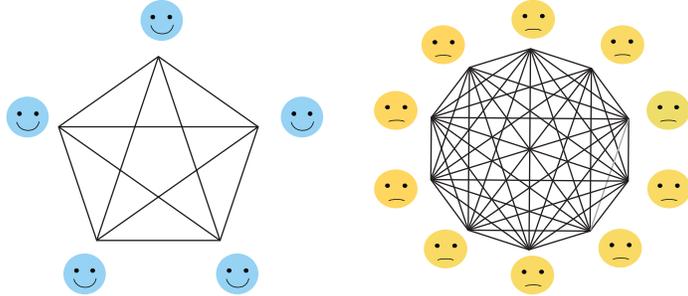


Figura 13: las vías de comunicación aumentarán como el cuadrado del tamaño del equipo.

Buenas noticias: equipos de características

El enfoque más exitoso para abordar este problema ha sido la creación de "equipos de características" totalmente multi-funcionales, capaces de operar en todas las capas de la arquitectura con el fin de ofrecer las características centradas en el cliente. En un gran sistema, esto requiere aprender nuevas habilidades.

A medida que los equipos se centran en el aprendizaje, en lugar de micro-eficiencias de corto plazo, pueden ayudar a crear una *organización de aprendizaje*.

Prácticas relacionadas

Lean

Scrum es un marco de gestión general, coincidente con el movimiento ágil del desarrollo de software, que está inspirado en parte por métodos de fabricación Lean como el Sistema de Producción Toyota⁷.

eXtreme Programming (XP)

Al no establecer Scrum las prácticas específicas de ingeniería, los ScrumMasters son responsables de promover un mayor rigor en la definición de "terminado". Los ítems que se denominan "terminados" deben permanecer así. Las pruebas de regresión automatizadas evitan las sorpresas. Diseño, arquitectura e infraestructura deben surgir con el tiempo, sujetos a reconsideración continua y perfeccionamiento, en lugar de ser "finalizados" desde el principio, cuando no sabemos nada.

El ScrumMaster puede inspirar al equipo para aprender las prácticas de ingeniería asociadas con XP: integración continua, Test-Driven Development (TDD), refactorización constante, programación en parejas, check-ins frecuentes, etc.; la aplicación informada de estas prácticas impide la deuda técnica.

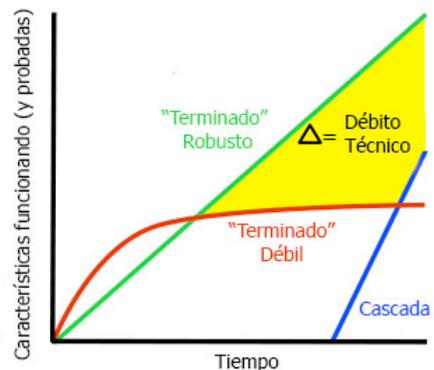


Figura 15: la línea verde recta representa el objetivo general de los métodos ágiles: entrega rápida y sostenible de las funciones valiosas. Implementar Scrum correctamente implica aprender a satisfacer una definición rigurosa de "terminado" para evitar la deuda técnica⁸.

⁵ "Seven Obstacles to Enterprise Agility", Gantthead, James (2010) <http://www.gantthead.com/content/articles/255033.cfm>

⁶ "Scaling Lean & Agile Development", Larman/Vodde, Addison Wesley (2008)

⁷ El movimiento Ágil definido en <http://agilemanifesto.org>

⁸ Gráfico inspirado en una conversación con Ronald E. Jeffries

Equipos Auto-Organizados

Los equipos comprometidos superan a los equipos manipulados

Durante la ejecución del Sprint, los miembros del equipo desarrollan un interés intrínseco en objetivos compartidos y aprenden a gestionarse para alcanzarlos. La natural tendencia humana a rendir cuentas a un grupo de compañeros contradice años de hábito de los trabajadores. Permitir a un equipo convertirse en auto-organizado, en lugar de manipularlo por medio de castigos y recompensas extrínsecas, contradice años de hábito para los directivos⁹. La observación del ScrumMaster y las habilidades de persuasión aumentan la probabilidad de éxito, a pesar de la incomodidad inicial.

Desafíos y oportunidades

En términos de desempeño, los equipos auto-organizados pueden superar radicalmente a los equipos tradicionales. Los grupos de tamaño familiar se auto-organizan naturalmente cuando se reúnen ciertas condiciones:

- Los miembros se comprometen a metas claras y de corto plazo
- Los miembros pueden evaluar el progreso del equipo
- Los miembros pueden observar la contribución de los otros miembros
- Los miembros se sienten seguros de dar su opinión sin tapujos

El psicólogo Bruce Tuckman describe las etapas de desarrollo del equipo como "forming, storming, norming, performing"¹⁰. La auto-organización óptima lleva su tiempo. El equipo puede desempeñarse peor durante las iteraciones tempranas de lo que han actuado como un grupo de trabajo manejado tradicionalmente¹¹.

En el trabajo complejo, los equipos heterogéneos superan a los equipos homogéneos. También experimentan mayores conflictos¹². Los desacuerdos son habituales y saludables en un equipo comprometido, el rendimiento del equipo será determinado por lo bien que el equipo se encargue de estos conflictos.

La *teoría de la manzana podrida* sugiere que una sola persona negativa ("que requiere esfuerzo de retención del grupo, que expresa afecto negativo, o la violación de importantes normas interpersonales"¹³) puede reducir el rendimiento de un grupo entero de manera desproporcionada. La aparición de tales individuos es rara, pero su impacto se magnifica por la renuencia de un equipo para eliminarlos. Esto puede ser parcialmente mitigado dándole a los equipos mayor influencia de decisión en quién se une a ellos.

Otras personas que no se desempeñan en un nivel óptimo en situación de jefe/empleado (por no ser desafiados o por el *micromanagement*) van a brillar en un equipo Scrum.

La auto-organización se ve obstaculizada por condiciones tales como la distribución geográfica, dinámicas de jefe/trabajador, miembros del equipo part-time, y las interrupciones no relacionadas con los objetivos de Sprint. La mayoría de los equipos se beneficiarán de un ScrumMaster a tiempo completo que trabaje duro para mitigar este tipo de impedimentos¹⁴.

Cuándo es apropiado Scrum?

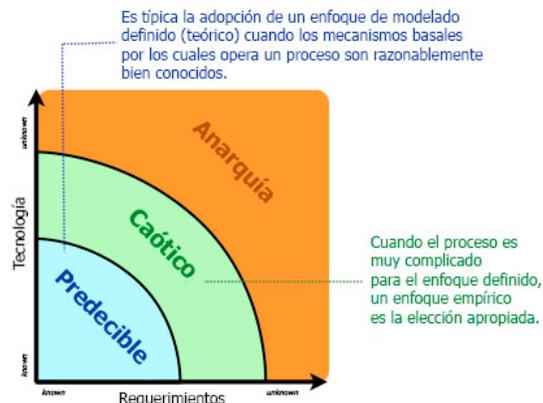


Figura 16: en un marco empírico, Scrum es apropiado para trabajos que incluyan requisitos con un alto grado de incertidumbre y/o incertidumbre técnica.^{15 16}

Scrum se aplica a los tipos de trabajo que las personas han encontrado difícil de manejar con procesos definidos, que incluyen requisitos inciertos combinados con riesgos impredecibles en la tecnología de aplicación. Al decidir si se debe aplicar Scrum, a diferencia de los enfoques impulsados por la planificación, tales como los descritos en la Guía del PMBOK®, se recomienda tener en cuenta los mecanismos subyacentes bien entendidos o si el trabajo depende de la creación de conocimiento y colaboración. Por ejemplo, Scrum no fue pensado originalmente para los tipos repetitivos de producción y servicios.

También es aconsejable considerar si existe un compromiso suficiente para fomentar un equipo auto-organizado.

Sobre el Autor



Michael James aprendió a programar varios años atrás. Ha trabajado directamente con Ken Schwaber para convertirse en un instructor de Scrum. Realiza Coaching en equipos técnicos, gerentes y ejecutivos con el objetivo de optimizar la entrega de valor en las organizaciones. Por favor envíe feedback a mj4scrum@gmail.com o bien <http://twitter.com/michaeldotjames>

Traducción al español:

Para aportes a la mejora de la versión en español, por favor envíe feedback a Martín Alaimo: malaimo@gmail.com o <http://twitter.com/martinalaimo>

Aprender más

- Ejemplo de ScrumMaster Checklist: <http://ScrumMasterChecklist.org>
- Online Scrum training: <http://ScrumTrainingSeries.com>
- Versión actualizada disponible en inglés: <http://ScrumReferenceCard.com>

⁹ La motivación intrínseca está relacionada con el dominio, la autonomía y el propósito. Las "Recompensas" lo dañan <http://www.youtube.com/watch?v=u6XAPnuFjJc>

¹⁰ Developmental Sequence in Small Groups, Psychological Bulletin, 63 (6): 384-99 Tuckman, referenciado numerosas veces por Schwaber

¹¹ "The Wisdom of Teams: Creating the High-Performance Organization", Katzenbach, Harper Business (1994)

¹² "Group Genius: The Creative Power of Collaboration", Sawyer, Basic Books (2007), (libro #2 de la lista de lecturas recomendadas para ScrumMasters, de Michael James)

¹³ How, when, and why bad apples spoil the barrel: Negative group members and dysfunctional groups, Research in Organizational Behavior, Volume 27, 181-230, Felps/Mitchell/Byington, (2006)

¹⁴ Ejemplo de una lista detallada de las responsabilidades de un ScrumMaster full-time: <http://ScrumMasterChecklist.org>

¹⁵ Versión extensamente modificada de un gráfico de "Strategic Management and Organizational Dynamics", Stacey (1993), referenciado en *Agile Software Development with Scrum*, Schwaber/Beedle (2001)

¹⁶ "Process Dynamics, Modeling, and Control", Ogunnaike, Oxford University Press, 1992