JEFF SUTHERLAND

SCRUM

EL ARTE DE HACER EL DOBLE DE TRABAJO EN LA MITAD DE TIEMPO





Lectulandia

Este es el libro definitivo sobre la metodología que está revolucionando el mundo. El *scrum* (término procedente del *rugby*, que hace referencia a la forma en la que el equipo trabaja conjuntamente para hacer avanzar la pelota por el campo) es un sistema de trabajo creado por el autor del libro, Jeff Sutherland, que logra que hagamos «el doble de trabajo en la mitad de tiempo». El *scrum* acaba con el papeleo, la burocracia y la jerarquización en las empresas y en los proyectos personales, y apuesta por el trabajo en equipo, por que todos nos sintamos implicados de verdad en aquello que hacemos, por acabar con todos los impedimentos que se interponen en nuestro camino y por que todos nos sintamos satisfechos y realizados con los objetivos conseguidos.

Lectulandia

Jeff Sutherland

Scrum

El arte de hacer el doble de trabajo en la mitad de tiempo

ePub r1.0
XcUiDi 07.08.2019

Título original: *SCRUM. The Art of Doing Twice the Work in Half the Time*
Jeff Sutherland, 2014
Traducción: Enrique Mercado

Editor digital: XcUiDi
ePub base r2.1

más libros en lectulandia.com

Índice de contenido

Cubierta

Scrum

Prefacio. ¿Por qué scrum?

Capítulo uno. El mundo no marcha bien
- Una nueva manera de pensar
- Remedio en el FBI
- Resumen

Capítulo dos. Los orígenes de scrum
- Aprender a pensar como robot
- No persigas cascadas
- Inspección y ajuste
- Renovarse o morir
- Shu Ha Ri
- Resumen

Capítulo tres. Equipos
- La larga fila gris
- Scrum en la hora de la revuelta
- Un equipo que lleve a cabo el trabajo
- Scrum en la guerra
- El tamaño sí importa, aunque no como suele pensarse
- El Scrum Master
- No rechaces al jugador sino al juego
- Alcanzar la grandeza
- Resumen

Capítulo cuatro. Tiempo
- El sprint
- Parada diaria
- Una y otra vez
- Resumen

Capítulo cinco. El desperdicio es un crimen
- Haz una cosa a la vez

Hacer a medias es no hacer en absoluto
Hazlo bien a la primera
Trabajar de más sólo crea problemas
Sé razonable
Flujo
Resumen

Capítulo seis. Planea realidades, no fantasías
Planeación de una boda
El tamaño importa, pero sólo relativamente
Serie de Fibonacci: Por todas partes
El oráculo de Delfos
Póker de planeación
No hay tareas, sólo historias
Escribe historias cortas
Lista y terminada
Planeación del sprint
Conoce tu velocidad
Resumen

Capítulo siete. Felicidad
Felicidad es éxito
Cuantificar la felicidad
Vuelve visible todo
Dar felicidad
Rompe la burbuja de la felicidad
Feliz ahora, feliz mañana
Resumen

Capítulo ocho. Prioridades
Pendientes: qué hacer y cuándo
El responsable del producto
Observa, oriéntate, decide, actúa
Lo primero es lo primero
Lanzamiento
Dinero por nada y modificaciones gratis
Riesgo
He aquí lo que harás mañana
Resumen

Capítulo nueve. Cambia el mundo
Educación

Pobreza
Gobierno
Cómo trabajaremos todos algún día
¿Qué no podemos hacer?
Resumen

Agradecimientos

Apéndice. Implementación de scrum: cómo empezar

Autor

Notas

Prefacio

¿Por qué Scrum?

Creé Scrum, con Ken Schwaber, hace veinte años, como una manera más rápida, confiable y eficaz de crear *software* en la industria de la tecnología. Hasta ese momento —y aun en 2005—, la mayoría de los proyectos de desarrollo de *software* se ejecutaban usando el método en cascada, donde un proyecto se completaba en etapas separadas y avanzaba paso a paso hacia el lanzamiento último a los consumidores o usuarios de *software*. El proceso era lento e impredecible, y a menudo no resultaba en un producto que la gente necesitara o pagara. Retrasos de meses o hasta años eran endémicos al proceso. Los primeros planes paso a paso, expuestos en cómodo detalle en gráficas de Gantt, daban seguridad a la dirección de que teníamos el control del proceso de desarrollo, pero, casi sin falta, pronto nos atrasábamos respecto al programa y rebasábamos desastrosamente el presupuesto.

Para remediar estas fallas, en 1993 inventé una nueva forma de hacer las cosas: Scrum, que representa un cambio radical respecto a las metodologías prescriptivas y verticales de gestión de proyectos del pasado. Scrum se asemeja en cambio a los sistemas evolutivos, adaptativos y capaces de autocorregirse. Desde su concepción, su marco se ha convertido en el modo en que la industria de la tecnología crea *software* y productos nuevos. Pero aunque Scrum se ha vuelto célebremente exitoso en la gestión de proyectos de *software* y *hardware* en Silicon Valley, sigue siendo relativamente desconocido en la práctica general de negocios. Por eso escribí *Scrum*: para revelar y explicar el sistema de gestión de Scrum a empresas fuera del mundo de la tecnología. En este libro hablaré de los orígenes de Scrum en el Toyota Production System y en el ciclo OODA de la aviación de combate. Describiré cómo organizar proyectos alrededor de equipos pequeños y por qué ésta es una forma muy eficaz de trabajar. Explicaré cómo priorizar proyectos, cómo establecer «*sprints*» de una semana a un mes para cobrar impulso y

responsabilizar a todos los miembros del equipo, cómo realizar breves paradas diarias para estar al tanto de lo que se ha hecho y de los retos que inevitablemente aflorarán. Asimismo, expondré cómo Scrum incorpora los conceptos de mejora continua y productos mínimos viables para obtener realimentación inmediata de los consumidores, en lugar de esperar hasta que un proyecto concluya. Como verás en las páginas que siguen, Scrum se ha usado ya para hacer todo, desde automóviles accesibles de 100 millas por galón (160 km por 3785 L) hasta la incorporación de los sistemas de base de datos del FBI al siglo XXI.

Sigue leyendo. Descubrirás cómo puede ayudar Scrum a transformar la manera de trabajar, crear, planear y pensar de tu compañía. Creo firmemente que Scrum puede contribuir a revolucionar la forma en que trabajan las empresas de prácticamente todas las industrias, así como revolucionó la innovación y rapidez de arribo al mercado en una deslumbrante serie de nuevas compañías y una variedad pasmosa de nuevos productos salidos de Silicon Valley y el mundo de la tecnología.

—Dr. Jeff Sutherland

Capítulo uno

El mundo no marcha bien

Jeff Johnson estaba seguro de que aquél no iba a ser un buen día. El 3 de marzo de 2010, el Federal Bureau of Investigation (Buró Federal de Investigación, FBI) de Estados Unidos canceló su más ambicioso proyecto de modernización, que supuestamente impediría otro 11 de septiembre pero que al final se había convertido en una de las mayores debacles de *software* de todos los tiempos. Durante más de una década, el FBI había intentado poner al día su sistema de cómputo y, al parecer, no lo había conseguido. *Una vez más*. Sin embargo, en esta ocasión se trataba de un proyecto de Johnson.

Éste había llegado a esa agencia siete meses antes, invitado por el nuevo director de información, Chad Fulgham, con quien había trabajado en Lehman Brothers. Nombrado subdirector de la división de informática, su oficina se hallaba en el piso más alto del J. Edgar Hoover Building, situado en el centro de Washington. Era una oficina grande, con vista al monumento a Washington. Johnson jamás imaginó que pasaría los dos años siguientes en el sótano de ese edificio, en una oficina de hormigón sin ventanas, intentando componer algo que todos juzgaban irreparable.

«No fue una decisión fácil», dice. Su jefe y él habían resuelto darse por vencidos y acabar con un programa que ya había durado casi una década y costado cientos de millones de dólares. Para ese momento, lo más lógico era que el propio FBI se hiciera cargo del proyecto. «Pero tenía que terminarse y terminarse bien».

Se trataba del muy esperado sistema de cómputo que introduciría al FBI a la época moderna. En 2010 —la era de Facebook, Twitter, Amazon y Google —, este organismo seguía archivando sus informes en formato impreso. Usaba el Automated Case Support (Soporte Automatizado de Casos), sistema basado en mainframes que había sido de vanguardia en los años ochenta. Pero muchos de sus agentes especiales no lo empleaban. Era demasiado lento y

engorroso en una época de ataques terroristas y criminales sumamente dinámicos.

Cuando un agente quería hacer algo —*cualquier cosa*, desde pagar a un informante hasta perseguir a un terrorista y presentar un informe sobre un asaltabancos— debía seguir casi el mismo procedimiento de treinta años antes. Johnson lo describe de esta manera: «Se hacía un documento en un procesador de palabras y se imprimían tres copias, una para el visto bueno, otra por si la primera se perdía y una más para encerrar en rojo —¡en rojo!— las palabras clave que había que trascribir en la base de datos. Uno mismo indizaba sus informes».

Una vez aprobada una solicitud, la copia se devolvía a su autor, marcada con un número. Un número en una hoja de papel era la forma en que el FBI seguía la pista de todos sus expedientes. Este método era tan permeable y anticuado que se le creía en parte la causa de que el FBI no hubiera podido «unir los puntos» que indicaban que activistas de Al Qaeda habían entrado al país en las semanas y meses previos al 11 de septiembre. Una de sus oficinas había sospechado de alguien. Otra se había preguntado por qué se enseñaba a volar a tantos extranjeros misteriosos. Otra más había incluido a alguien en una lista de individuos por vigilar, pero sin decírselo a las demás. Nadie en la agencia había asociado entre sí todos esos datos.

La comisión del 11 de septiembre hizo indagaciones tras el ataque para averiguar la razón básica de que eso hubiera pasado. Los analistas, concluyó, no tenían acceso a la información que debían examinar. «El mal estado de los sistemas de información del FBI», se lee en el reporte correspondiente, «implicaba que ese acceso dependía sobre todo de las relaciones personales de los analistas con los agentes de las brigadas o unidades operativas en poder de la información».

Antes del 11 de septiembre, el FBI nunca había evaluado la amenaza terrorista general para Estados Unidos. Había muchas razones de ello, desde interés exclusivo en el ascenso hasta no compartir información. Pero ese reporte señalaba el anacronismo tecnológico como posible razón clave de la extrema ineptitud del FBI en los días anteriores al 11 de septiembre. «Los sistemas de información de la agencia eran totalmente insuficientes», concluía el reporte de dicha comisión. «El FBI no podía saber qué sabía: no contaba con ningún mecanismo eficiente para capturar o compartir sus conocimientos institucionales».

Cuando el Senado comenzó a hacer preguntas incómodas, el FBI respondió en esencia: «No se preocupen; estamos por lanzar un plan de

modernización». Este plan, llamado Virtual Case File (Expediente virtual, VCF), lo cambiaría todo, supuestamente. Para aprovechar al máximo la crisis, dijeron funcionarios de la agencia, sólo necesitaban setenta millones de dólares más, aparte de los cien ya presupuestados para aquel plan. Si se leen artículos de entonces sobre el VCF, se verá que en ellos aparecen muchas veces las palabras *revolucionario* y *transformación*.

Sin embargo, este programa se canceló tres años después. No había dado ningún resultado. El FBI gastó ciento setenta millones de dólares de los contribuyentes en la compra de un sistema de cómputo que nunca se usó: ni una sola línea de código ni aplicación, ningún clic con el ratón. Un fracaso absoluto. Y no porque IBM o Microsoft hubieran cometido un error, sino porque, literalmente, la *vida* de muchas personas estaba en peligro. Como dijo al *Washington Post* el senador Patrick Leahy, de Vermont, entonces el demócrata de más alto rango en el comité judicial de la cámara alta:

> Se contaba con información que habría podido impedir los acontecimientos del 11 de septiembre. Estaba ahí, en algún lado, pero no se utilizó. […] No he visto que se resuelva ningún problema. […] Bien podría llegar el siglo XXII sin que dispongamos aún de la tecnología del XXI[1].

Curiosamente, muchos empleados que estaban en el FBI en la época del desastre del VCF ya no trabajan ahí.

La agencia anunció en 2005 un nuevo programa, Sentinel. Esta vez sí daría resultado. Esta vez sí se tomarían todas las precauciones y se aplicarían los controles y procedimientos presupuestales indicados. El FBI había aprendido la lección. ¿El precio? Únicamente cuatrocientos cincuenta y un millones de dólares. El sistema estaría en operación en 2009.

¿Qué pasó esta vez? En marzo de 2010, la respuesta apareció en el escritorio de Jeff Johnson. Lockheed Martin, la compañía contratista elegida para generar el sistema Sentinel, había gastado ya cuatrocientos cinco millones de dólares, llevaba un año de retraso y apenas había desarrollado la mitad del proyecto. En un análisis independiente se estimó que, para terminarlo, se necesitarían de seis a ocho años más y que los contribuyentes tendrían que aportar al menos otros trescientos cincuenta millones de dólares.

Johnson debía encontrar la manera de evitarlo.

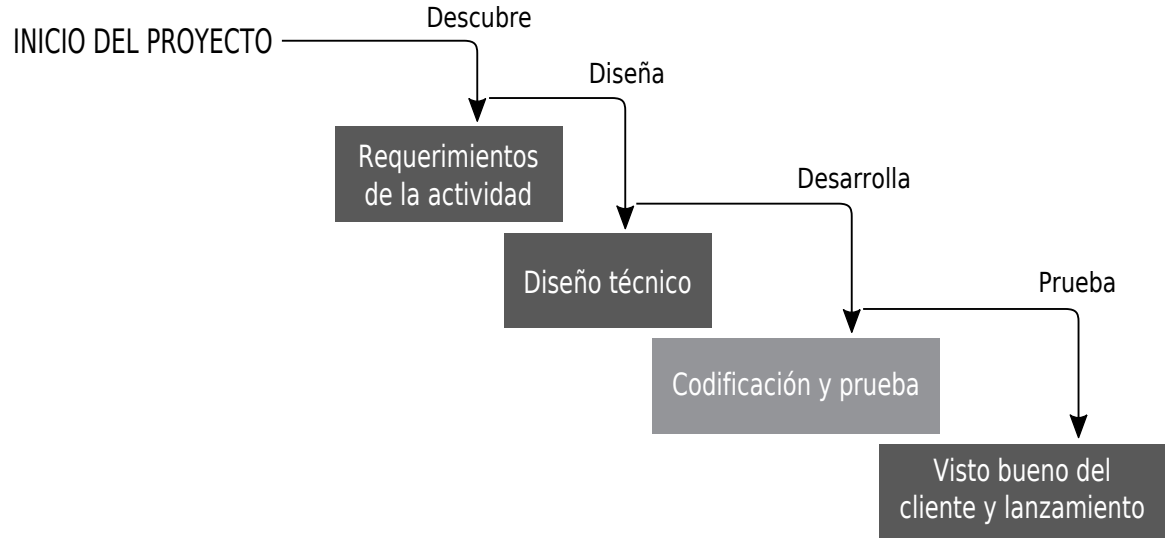
Qué marchó mal y cómo se resolvió ese problema es el motivo de que yo haya escrito este libro. El error no fue la falta de individuos inteligentes. No

fue que el FBI no haya tenido el personal indicado ni la tecnología correcta. No se debió a la ética de trabajo ni a la ausencia de empuje competitivo.

Se debió a la *manera* en que la gente trabaja. A la manera en que la *mayoría* trabajamos. A la forma en que todos creemos que *debemos* hacerlo, porque así es como nos lo enseñaron.

Una vez en conocimiento de lo ocurrido, todo *resulta* aparentemente lógico: la gente de Lockheed se reunió antes de pujar por el contrato, estudió los requerimientos y se puso a planear cómo crear un sistema que satisfaciera esas necesidades. Puso a trabajar meses enteros a muchas personas inteligentes para que determinaran lo que se debía hacer. Luego dedicó varios meses más a planear cómo hacerlo. Elaboró vistosos diagramas con todas las actividades por efectuar y en los que se especificaba el tiempo que consumirían todas y cada una de ellas. Después, con colores cuidadosamente elegidos, mostró cada parte del proyecto vertiéndose sobre la siguiente, como en una cascada.

MÉTODO EN CASCADA



Estos diagramas se llaman *gráficas de Gantt*, en honor a Henry Gantt, su inventor. La aparición de las computadoras personales en la década de 1980, las cuales facilitaron la elaboración de estas gráficas —y las *complicaron*—, habría de convertirlas en obras de arte. En ellas se muestra en detalle cada paso de un proyecto, cada indicador importante, cada fecha de entrega. Son impresionantes. El único problema es que nunca *aciertan*.

Henry Gantt inventó sus famosas gráficas en 1910. Su usuario primigenio, en la Primera Guerra Mundial, fue el general William Crozier, jefe de artillería del ejército estadunidense. Quien haya estudiado esa conflagración

sabe que no se distinguió precisamente por su capacidad organizativa. Yo nunca he podido entender cómo un recurso de la Primera Guerra Mundial ha terminado por convertirse en la herramienta por antonomasia de la gestión de proyectos del siglo XXI. Pese a que fracasó en la guerra de trincheras, por alguna razón sigue siendo popular.

Exponer a la vista de los involucrados todo lo que se debe hacer en un gran proyecto resulta demasiado tentador. He visitado numerosas compañías con empleados cuya *única labor* es poner al día las gráficas de Gantt. El problema es que una vez que ese plan maravilloso se enfrenta con la realidad se viene abajo. Pero en lugar de abandonarlo, o de abandonar la forma en que se le concibió, los directivos contratan gente para que haga parecer que funciona. En esencia, pagan para que se les mienta.

Este lamentable patrón recuerda los informes que el politburó soviético recibía en la década de 1980, antes del desplome de la URSS: un espejismo total. Ahora como entonces, los informes son más importantes que la realidad que supuestamente deben describir; y si surge una discrepancia, el problema es la realidad, no las gráficas.

Cuando yo era cadete en West Point, dormía en la antigua habitación del expresidente Dwight Eisenhower. De noche, la luz de los faroles se reflejaba en una placa de oro en la repisa de la chimenea y a veces me despertaba. DWIGHT D. EISENHOWER DURMIÓ AQUÍ, se leía en esa placa. Yo recordaba entonces que él dijo una vez que, aunque planear el combate es importante, los planes se evaporan en cuanto suena el primer disparo. Al menos tuvo cordura suficiente para no recomendar el uso de las gráficas de Gantt.

Lockheed presentó al FBI esas preciosas gráficas y éste la contrató. En teoría, la tarea estaba ahora tan bien planeada que nada podía salir mal. «Miren, eso también está en nuestro plan, con codificación de colores, fechas y gráficas de barras».

Pero cuando Johnson y su jefe, el director de información, Chad Fulgham, examinaron dicho plan en la primavera de 2010, descubrieron lo que realmente era y lo que son *todos* los diagramas de ese tipo: pura mentira. Cuando se pusieron a analizar el desarrollo y las fechas de entrega reales, se dieron cuenta de que el problema era irresoluble. Antes siquiera de que acabaran de corregirse errores ya detectados en el *software*, aparecían nuevos.

Fulgham notificó entonces al inspector general del Departamento de Justicia que el proyecto Sentinel podía llevarse hasta su fin si el FBI asumía su desarrollo y reducía el número de desarrolladores, con lo que la mitad más

difícil del proyecto se entregaría en menos de un quinto del tiempo y con menos de un décimo del monto presupuestado. En los informes del inspector general al Congreso, usualmente parcos, es palpable el escepticismo. En el de octubre de 2010, tras exponer sus nueve reservas frente a esa propuesta, los vigías de la oficina del inspector general remataron: «En suma, tenemos muchas inquietudes y dudas de que con ese nuevo enfoque el proyecto Sentinel pueda terminarse dentro del presupuesto, a tiempo y con una funcionalidad similar a la prevista»[2].

Una nueva manera de pensar

Este nuevo enfoque se llama *Scrum*. Yo fui su inventor, hace veinte años. Hoy en día es el *único* recurso de eficacia comprobada para llevar a cabo proyectos como ése. Hay dos formas de hacer las cosas: el antiguo método «en cascada», que gasta cientos de millones de dólares sin lograr nada, y el nuevo modo que, con menos personas y en menos tiempo, permite hacer más con mayor calidad y a menor costo. Sé que parece demasiado bello para ser verdad, pero los resultados lo confirman. Funciona.

Hace veinte años, yo desesperaba por hallar una nueva forma de concebir el trabajo. Y gracias a incontables investigaciones y experimentos, así como a la revisión de datos pasados, me di cuenta de que todo lo que necesitaba era otra forma de organizar el esfuerzo humano. Nada de esto es nuevo; ya se ha dicho antes. Estudios de tiempos de la Segunda Guerra Mundial exponen mejores maneras de trabajar. Pero, por algún motivo, nunca se pusieron en práctica. Yo he intentado hacer precisamente eso en las dos últimas décadas y hoy este método es ubicuo en el primer campo en que lo apliqué, el desarrollo de *software*. En gigantes como Google, Amazon y Salesforce.com, lo mismo que en nuevas empresas de las que no se oye hablar aún, este marco ha cambiado radicalmente cómo hace las cosas la gente.

La razón de que este marco funcione es simple: estudié cómo trabaja la gente *en realidad*, no cómo *dice* trabajar. Analicé investigaciones de décadas atrás y buenas prácticas de compañías del mundo entero, y examiné atentamente a los mejores equipos de esas compañías. ¿Qué los volvía superiores? ¿Qué los hacía diferentes? ¿Por qué algunos equipos alcanzan la grandeza mientras otros se estancan en la mediocridad?

Por razones que presentaré en capítulos posteriores, llamé *Scrum* a ese marco para trabajar en equipo. Este término procede del *rugby* y se refiere al

modo en que un equipo se desempeña en común para mover el balón por la cancha. Acoplamiento, unidad de propósito y claridad de metas van de la mano. Ésa es la metáfora perfecta para mi manera de entender el trabajo en común.

Tradicionalmente, la gerencia quiere dos cosas de un proyecto: control y predictibilidad. Esto se traduce en gran número de documentos, gráficas y tablas, como en el caso de Lockheed. Meses de esfuerzo se invierten en planear cada detalle, a fin de que no haya errores ni excesos de costos y de que el producto esté listo a tiempo.

El problema es que este escenario optimista jamás se vuelve realidad. Todo ese esfuerzo invertido en planear, con intención de restringir cualquier cambio y conocer lo incognoscible, es inútil. Todo proyecto implica problemas y golpes de inspiración. Tratar de reducir un empeño humano de cualquier alcance a tablas y gráficas codificadas con color resulta absurdo y está condenado al fracaso. La gente no trabaja así, los proyectos no avanzan de ese modo. No es así como fructifican las ideas ni como se hacen las grandes cosas.

Eso provoca, en cambio, que la gente no consiga lo que quiere y se frustre. Los proyectos se atrasan, rebasan su presupuesto y, demasiado a menudo, acaban en fracasos lamentables. Esto se aplica en particular a los equipos creativos, que deben hacer algo nuevo. En la mayoría de los casos, la gerencia no advierte la cercanía al fracaso hasta que ha invertido millones de dólares y miles de horas para nada.

Scrum pregunta por qué hacer algo implica tanto tiempo y esfuerzo y por qué somos tan malos para calcular cuánto tiempo y esfuerzo absorberá. La catedral de Chartres tardó cincuenta y siete años en construirse. Es casi indudable que, al principio del proyecto, los canteros dijeron al obispo: «Veinte años máximo. Tal vez quince».

Scrum abraza la incertidumbre y la creatividad. Dota de estructura al proceso de aprendizaje, lo que permite a los equipos evaluar qué crearon e, igualmente importante, cómo lo hicieron. Scrum se vale del trabajo real en equipo y brinda a este último las herramientas necesarias para organizarse y aumentar en poco tiempo tanto la rapidez como la calidad de su trabajo.

Scrum se basa en una idea simple: cada vez que se ejecuta un proyecto, ¿por qué no revisar con regularidad para ver si lo que se está haciendo sigue la dirección correcta y es lo que la gente quiere? ¿Por qué no revisar si se puede hacer mejor y más rápido y qué lo impide?

Esto se llama ciclo de «inspección y ajuste». Cada tanto, haz una pausa, revisa lo que hiciste y ve si debes seguir haciéndolo y cómo podrías hacerlo mejor. La idea es sencilla, pero llevarla a la práctica requiere reflexión, introspección, honestidad y disciplina. Escribí este libro para enseñarte a hacerlo, no sólo en compañías de *software*. He visto usar satisfactoriamente Scrum para fabricar automóviles, administrar una lavandería, enseñar en el aula, hacer naves espaciales, planear una boda y hasta, como hace mi esposa, confirmar cada fin de semana la ejecución de la lista de pendientes del hogar.

El resultado final de Scrum —la meta de su diseño, si se quiere— son equipos que aumentan considerablemente su productividad. En los últimos veinte años, he formado muchos equipos así. He sido director general, de tecnología o de ingeniería de una docena de organizaciones, desde nuevas empresas cuyo personal cabe en una sala hasta grandes corporaciones con oficinas en todo el planeta. Y he sido consultor y asesor de varios cientos más.

Los resultados pueden ser tan espectaculares que grandes empresas de investigación y análisis como Gartner y Standish ya sostienen que el antiguo estilo de trabajar es obsoleto. Las compañías que siguen adhiriéndose a ideas probadas pero falsas de mando y control y que intentan imponer una predictibilidad rígida están condenadas al fracaso si sus competidores usan Scrum; la diferencia es enorme. Sociedades de capital de riesgo como OpenView Venture Partners, en Boston, de la que soy asesor, afirman que Scrum ofrece una ventaja competitiva demasiado grande para no aprovecharla. Y no dicen esto personas impulsivas y dispersas, sino hombres con la mira puesta en el dinero, quienes aseguran: «Los resultados son indiscutibles. Las compañías tienen sólo dos opciones: renovarse o morir».

Remedio en el FBI

El primer problema del equipo de Sentinel fueron los contratos. Cada cambio derivaba en una negociación contractual con Lockheed Martin. Así, Jeff Johnson y Chad Fulgham pasaron meses cancelando todos los contratos, trasladando las labores de desarrollo al FBI y reduciendo el personal de cientos a menos de cincuenta individuos. El equipo básico era aún más pequeño.

La primera semana, Johnson y Fulgham hicieron lo que suele hacerse en estos casos: imprimir toda la documentación sobre requerimientos. A quien

nunca ha visto qué significa esto en un gran proyecto le sorprenderá saber que bien puede representar cientos y cientos de páginas. He visto pilas de más de medio metro de alto. En un proyecto tras otro, he visto a gente cortar y pegar texto sin que nadie lea las miles de páginas resultantes. No puede hacerlo. Y ésta es precisamente la cuestión: la gente ha implantado un sistema que la obliga a respaldar una fantasía.

«El número de requerimientos ascendía a mil cien. La pila era de más de diez centímetros de alto», dice Johnson. Sólo pensar en eso me hace sentir lástima por quienes dedicaron algunas semanas de su vida a producir esos documentos inútiles. Pero el FBI y Lockheed Martin no están solos en esto; lo he visto repetirse en casi cada compañía con la que he trabajado. Esa pila inservible es una de las razones de que Scrum pueda constituir un cambio tan impresionante. Nadie debería dedicar un solo minuto de su vida a hacer algo sin sentido. No sólo es mal negocio, sino que también embota el entendimiento.

Una vez en poder de aquella pila, los equipos de Sentinel procedieron a ordenar por prioridad cada requerimiento, algo mucho más relevante y complicado de lo que parece. La gente suele decir que todo es importante, pero lo que se debe preguntar —lo que esos equipos se preguntaron— es qué es lo más *valioso* para al proyecto. Haz eso primero. En el desarrollo de *software* impera la regla, confirmada por décadas de investigación, de que ochenta por ciento del valor de un componente de *software* reside en veinte por ciento de sus funciones. Piensa en esto: ¿cuándo fue la última vez que usaste la función Editor de Visual Basic de Word? Quizá ni siquiera sepas qué es Visual Basic, menos aún para qué sirve. Pero está ahí y alguien dedicó tiempo a implementarla, aunque puedo asegurarte que dicha función no vuelve a Word mucho más valioso de lo que es.

Hacer que la gente priorice por valor la obliga a producir primero ese veinte por ciento. Al terminar suele darse cuenta de que no necesita el ochenta por ciento restante o de que lo que al principio le parecía importante en realidad no lo es.

Para el equipo de Sentinel, la cuestión era: «Aún está pendiente este enorme e importante proyecto, en el que ya hemos gastado cientos de millones de dólares. ¿Cuándo lo terminaremos?». Tras pensarlo bien, prometieron tenerlo listo para el otoño de 2011. El informe del inspector general del otoño de 2010 es un modelo de incredulidad:

> El FBI explicó que aplicará una «metodología ágil» para concluir el desarrollo de Sentinel usando menos empleados

> propios, de Lockheed Martin y de las compañías que han provisto los principales componentes del sistema. Planea reducir los empleados contratados de doscientos veinte a cuarenta y los propios de treinta a doce. […] Añadió que cree poder finalizar el programa con los veinte millones de dólares restantes en el presupuesto y doce meses después de que comience a aplicarse el nuevo método[3].

Que se haya entrecomillado la expresión «metodología ágil» indica lo poco que el inspector general sabía de Scrum. El término «ágil» (*agile* en inglés) data de una reunión de 2001 en la que diecisiete líderes de desarrollo de *software* escribimos lo que hoy se conoce como el *Agile Manifesto*. En ese documento proclamamos estos valores: personas antes que procesos, productos que funcionen antes que documentar lo que se supone que deben hacer, colaborar con los clientes antes que negociar con ellos y responder al cambio antes que seguir un plan. Scrum es el marco que yo creé para poner en práctica esos valores. No es ninguna metodología.

Claro que la promesa de doce meses de Johnson era algo engañosa, porque en realidad él no sabía —no podía saber— cuánto tardarían en terminar el proyecto. El FBI ignoraba cuán rápido podían trabajar sus equipos. Siempre les digo lo mismo a los ejecutivos: «Sabré la fecha de entrega cuando vea cuánto mejoran los equipos, cuán rápido avanzan, cuanto se *aceleran*».

También resultó crucial, desde luego, que los miembros de los equipos hayan previsto qué les *impediría* acelerarse. Como dijo Johnson: «Me encargué de quitar impedimentos». La noción de «impedimento» procede de la compañía que forjó muchas de las ideas que se hallan en la base de Scrum: Toyota, específicamente del sistema de producción de Toyota, descrito por Taiichi Ohno.

No entraré en detalles aquí, pero uno de los conceptos clave que Ohno formuló es la idea de «flujo». Es decir, la producción debía fluir veloz e ininterrumpidamente a lo largo del proceso, de modo que una de las tareas clave de la gerencia era identificar y quitar los impedimentos a ese flujo. Todo lo que obstruye es desperdicio. Ohno concede al desperdicio un valor moral tanto como económico en su libro clásico *The Toyota Production System*:

> No es exagerado decir que, en un periodo de bajo crecimiento, el desperdicio es un crimen contra la sociedad más que una

> pérdida financiera. Eliminar el desperdicio debe ser el primer objetivo de una compañía[4].

Ohno abunda en los diferentes tipos de desperdicio e impedimentos que pueden obstaculizar la producción. Para que Scrum pueda despegar en verdad, alguien en la alta dirección debe estar convencido de que los impedimentos son casi criminales. Más adelante explicaré *cómo* eliminar el desperdicio. Por ahora baste decir que el efecto de hacerlo es drástico, pese a lo cual la gente no lo hace porque para eso se precisa ser honesto con uno mismo y con los demás.

Johnson entendió que ésa debía ser su función.

El equipo de Sentinel tuvo que invertir tres meses para saber cuánto tardaría *realmente* en terminar el proyecto. ¿Por qué? Por el ya mencionado ciclo de inspección y ajuste. Scrum opera estableciendo metas secuenciales por cumplir en un periodo fijo. En el FBI se optó por ciclos de dos semanas, en la inteligencia de que, al final de cada uno, se dispondría de una adición completa al producto, es decir, de algo ya en operación, por enseñar a cualquiera, en particular a los interesados y, óptimamente, a los usuarios del producto.

Esta metodología permite a los equipos obtener realimentación sobre su trabajo casi en tiempo real. ¿Están siguiendo la dirección correcta? ¿Lo que, según sus planes, deben hacer después es *realmente* lo que deberían hacer, considerando lo que ya descubrieron en el ciclo que acaban de concluir?

En Scrum, llamamos *sprints* a estos ciclos. Al inicio de cada *sprint* hay una junta de planeación. El equipo decide entonces cuánto cree poder hacer en las dos semanas siguientes. Toma las tareas de la lista de prioridades y las escribe en papeletas adhesivas que pega en la pared. Luego decide cuántas de esas tareas puede llevar a cabo en las dos semanas siguientes.

Al final del *sprint*, el equipo vuelve a reunirse y muestra lo que logró en su periodo de trabajo en común. Determina asimismo cuántas de esas papeletas adhesivas cumplió. ¿Incluyó demasiadas en el *sprint* que acaba de terminar y no culminó todas? ¿No incluyó las suficientes? Lo importante en este caso es que el equipo empiece a tener un punto de referencia sobre el ritmo al que puede avanzar: su rapidez.

Después de mostrar lo que hizo —y aquí es donde entran en juego las ideas de Ohno—, el equipo no habla de lo que hizo, sino de *cómo* lo hizo. Pregunta: «¿Cómo podemos trabajar mejor en el siguiente *sprint*? ¿Qué nos estorbó en el anterior? ¿Qué nos impide avanzar tan rápido como sabemos que podemos hacerlo?».

Por eso Johnson precisó de unos meses para saber cuánto tardarían en terminar el proyecto. Quería evaluar la rapidez de cada equipo en varios *sprints* y ver después cuánto podían mejorar: cuán rápido podían avanzar en verdad. Una vez que examinó cuántas tareas llevó a cabo cada equipo en cada *sprint* y que revisó cuántas les faltaban para concluir el proyecto pudo pronosticar una fecha de entrega.

Además de saber cuán rápido marchaban los equipos, también quería saber qué les impedía avanzar, a fin de *acelerarlos* para que progresaran más pronto, aunque no trabajando más tiempo (luego explicaré por qué ésta es una ratonera que te retrasa más), sino *mejor* y con *más inteligencia*. Johnson asegura que sus equipos aumentaron tres veces su productividad: avanzaron tres veces más rápido en cuanto cobraron vuelo, en comparación con el momento inicial. ¿Por qué? Porque trabajaban mejor en común, sí, pero también, y sobre todo, porque identificaban las cosas que los retrasaban y se deshacían de ellas en cada *sprint*.

Finalmente, luego de dieciocho meses de codificación, el proyecto Sentinel puso en operación su base de datos, que acabó de desplegar en todo el FBI dos meses más tarde. «Fue una presión enorme», dijo Johnson en una entrevista. «Tómese en cuenta que ese sistema se usa para *todo*, desde pagar a informantes hasta guardar evidencias, expedientes y calendarios. Aún esta reunión está en Sentinel».

¿Lo más eficiente de Scrum desde su punto de vista? «Las demostraciones. Apuntar a un producto periódicamente demostrable». El equipo de Sentinel mostraba sus logros cada dos semanas. Y la demostración no estaba dirigida únicamente a sus miembros. Éstos ponían a probar el sistema a la misma gente que lo usaría. Todos los interesados en el proyecto enviaban a alguien a esas reuniones, así que a veces asistían a ellas muchas personas: de expedientes, inteligencia, agentes especiales, la oficina del inspector general, representantes de otras dependencias gubernamentales… A menudo también asistían el director y el subdirector del FBI, lo mismo que el inspector general en turno. Aquél no era un público fácil.

Eso fue lo que hizo que el proyecto funcionara, dice Johnson. «Scrum no se orienta a los desarrolladores, sino a los clientes y otros interesados. Esto fue un cambio organizacional. Mostrar el producto fue la parte más eficaz del procedimiento».

Mostrar el producto resultó eficaz porque, para decirlo suavemente, no había nadie que creyera en los presuntos avances del equipo de Sentinel. Nadie podía creer que éste progresara cada vez más rápido. «Yo había dicho

al Congreso que en veinte meses y con cinco por ciento del presupuesto haríamos lo que Lockheed no había podido hacer en diez años con noventa por ciento», dice Johnson. «Había escepticismo en el ambiente. Teníamos que rendir informes al subsecretario de Justicia. Planteábamos claramente nuestra situación, pero nuestra audiencia suponía que escondíamos algo. Cada vez que, en el pasado, ella había visto ese tipo de indicadores, los informes perdían detalle y la verdad *era* distinta».

Tal escepticismo se contagió al resto del FBI. «Los del sótano fracasarán de nuevo», era la idea. «El sistema va a fallar y tendremos que seguir usando documentos impresos».

Johnson contó a su equipo que, cuando era cadete de la Marina en Annapolis, había tenido que memorizar un pasaje del discurso «Citizenship in a Republic» (Ciudadano de una república) que el expresidente Theodore Roosevelt pronunciara en la Sorbona en 1910:

> No es el crítico el que importa; el que señala cómo tropieza el fuerte o qué habría podido hacer mejor el resuelto. El mérito es de quien está en el ruedo, cubierta la cara de polvo, sudor y sangre; de quien lucha con valentía; de quien falla y se equivoca una y otra vez, porque no hay esfuerzo sin tacha ni error; de quien asume la responsabilidad de actuar; de quien conoce grandes entusiasmos y devociones; de quien se entrega a una buena causa; de quien al final conoce el triunfo de su empeño y que si, en el peor de los casos, fracasa, lo hace al menos habiéndose atrevido a lo grande, así que no habrá de contarse nunca entre los espíritus tímidos e indiferentes que no conocen la victoria ni la derrota[5].

El equipo sufrió algunos retrasos mientras determinaba con exactitud cuán rápido podía hacer las cosas y cuán difíciles de hacer eran. Por fin, en julio de 2012 echó a andar Sentinel, cabalmente y para todos. Era impensable activarlo por etapas.

«Tuvo que hacerse de un día para otro. En un caso penal o de antiterrorismo, algo en Los Ángeles podía estar relacionado con algo en Chicago», dice Johnson. «No nos podíamos permitir perder pistas. *Debíamos mantener en todo momento un buen estado, cierto y seguro*».

Ese estado debía ser tan cierto y seguro como para poder presentarse en los tribunales. Los datos de Sentinel servían para procesar a personas y su integridad debía estar fuera de toda duda.

Aquel primer día, Johnson estaba nervioso e impaciente. Al llegar a su oficina puso en marcha Sentinel. Éste se cargó, lo cual era buena señal. Luego intentó aprobar un documento con una firma electrónica, tarea básica que decenas de miles de empleados del FBI realizaban sin cesar. Apareció un mensaje de error; no se podía hacer eso. Johnson empezó a alarmarse y en su cabeza surgieron visiones de desastre. Tras leer atentamente el código de error, comprendió el propósito del mensaje: no había insertado su tarjeta de identificación para verificar su identidad. Metió la tarjeta, hizo clic con el ratón y Sentinel estuvo listo para operar.

El efecto de Sentinel en el FBI ha sido inmenso. La posibilidad de comunicar y compartir información ha aumentado fundamentalmente las capacidades de la agencia. En enero de 2013, una de sus oficinas recibió el reporte de que la cuenta de una pequeña empresa había sido hackeada. Un millón de dólares se habían transferido a otro país antes de que los bancos estadunidenses pudieran evitarlo. Gracias a Sentinel, esa oficina pudo coordinarse con el agregado legal en la embajada del país de destino de la transferencia, quien alertó a las autoridades locales, las que a su vez atajaron la operación antes de que llegara al sistema bancario. Todo esto ocurrió en cuestión de horas, algo que habría sido imposible en los días de tres copias impresas y plumas rojas. Era la diferencia entre atrapar a un pillo y dejarlo escapar.

El equipo de Sentinel sigue en el sótano del FBI, aunque ya sin paneles entre los cubículos de sus miembros para que puedan verse entre sí. En una pared luce un póster con los principios de Agile, que yo contribuí a elaborar y he dedicado mi vida a aplicarlos en el mundo entero. Asombrosamente, tratándose de un lugar sin ventanas, una planta de lavanda crece bajo focos fluorescentes en la entrada. «Lavanda» fue el nombre en clave del prototipo de Sentinel. Los integrantes de este equipo continúan en sus puestos, haciendo mejoras y añadiendo funciones a su sistema.

De acuerdo con una vieja broma de la comunidad de Scrum, un día un pollo le dijo a un puerco:

—Deberíamos abrir un restaurante.

—¿Qué nombre le pondríamos? —preguntó el puerco.

—¿Qué te parece «Huevos con jamón»?

—¡No, gracias! —respondió el puerco—. ¡Yo me comprometería completo; tú sólo en parte!

La idea en Scrum es que los «puercos» son quienes se comprometen totalmente con el proyecto y los responsables de sus resultados; los «pollos»,

en cambio, son aquellos a quienes sólo se informa del progreso del proyecto, los interesados. En la sala de Sentinel hay una campana en forma de puerco. Cuando suena, quienes hicieron lo que nadie creía posible saben que se les llama a ellos. Hay otra campana, el timbre de la puerta, pero es para los pollos.

El mundo es cada vez más complicado y nuestro trabajo adquiere complejidad creciente. Piénsese en los automóviles. Antes, yo casi siempre reparaba mi auto. Hace treinta años podía reconstruir un radiador. Ahora, cuando abro el cofre bien podría estar viendo el interior de una computadora. Hoy un coche es básicamente eso; un Ford nuevo contiene más líneas de código que Facebook y Twitter juntos. Crear algo tan complejo representa un gran esfuerzo. Cada vez que la gente participa en un complicado esfuerzo creativo, sea lanzar un cohete al espacio, fabricar un interruptor mejor o capturar a un delincuente, los métodos administrativos tradicionales se desmoronan.

Nosotros lo sabemos, como individuos y como sociedad. Percibimos ecos de nuestra vida real en distopías laborales de ficción como las de la tira cómica *Dilbert* o la película *Office Space* (Enredos de oficina). Todos hemos dicho en casa, a nuestra pareja o amigos, que la «organización» corporativa moderna es una locura. A todos nos han dicho que llenar correctamente un formulario es más importante que hacer la labor respectiva o que debemos celebrar una reunión para preparar la reunión previa a la reunión. Es aberrante. Pero no dejamos de hacerlo, pese a que sea un completo y absoluto fracaso.

El lanzamiento de Healthcare.gov, la página en internet en la que los estadunidenses pueden contratar un seguro médico, es un buen ejemplo de ello. La interfaz era atractiva, clara, ingeniosa, un gran diseño. Se hizo en tres meses, utilizando Scrum. El motor, en cambio, era un desastre. No funcionaba. Se suponía que esa página debía enlazar bases de datos de la oficina federal de impuestos con bases de datos estatales, de compañías de seguros y el Departamento de Salud y Servicios Humanos. Una tarea compleja, que involucró a más de veinte contratistas a cargo de cosas diferentes y que lo planearon todo empleando técnicas de cascada. Pero no probaron la página hasta el final, en vez de hacer pruebas periódicas sobre la marcha.

La tragedia es que todos sabían que no debía ser así. Quienes trabajaban para esos contratistas no eran tontos. El problema fue que todos dijeron: «No es asunto mío». Hicieron su parte y ya. Nunca examinaron la página desde el

punto de vista del usuario, sino sólo desde el propio. La razón de que hayan podido actuar de esa manera es que no estaban acoplados: no los unía un propósito común. Scrum une a equipos para crear grandes cosas, para lo cual todos deben ver la meta última, pero también contribuir cada vez más a su cumplimiento. Nadie a cargo del proyecto Healthcare.gov insistió en probarlo todo al momento mismo en que se le hacía y, por lo que se refiere a sus fallas, esa página no es en absoluto atípica, lamentablemente. ¿Y la gente que arregló Healthcare.gov? Usó Scrum.

¿Cuántas veces no has oído hablar de un gran proyecto, con un costo de millones y millones, que se cancela no sólo a causa de sus gastos excesivos, sino también porque sencillamente *no funciona*? ¿Cuántos miles de millones de dólares se invierten cada año en producir *nada*? ¿Cuánto tiempo de vida pierdes haciendo algo que tu jefe y tú saben que carece de valor? Igualmente podrías abrir agujeros y volverlos a llenar, como si eso tuviera algún efecto.

Las cosas no tienen por qué ser así. Y en realidad no lo son. Que todos te hayan dicho siempre que así es como el mundo opera no significa que estén en lo cierto. Hay una manera diferente de hacer las cosas, una forma distinta de trabajar.

Y si *no* la adoptas, tus funciones serán objeto de subcontratación. O tu compañía desaparecerá. En el hipercompetitivo mundo del trabajo del siglo XXI no hay cabida para el desperdicio y la insensatez.

Otra cuestión importante: la productividad óptima —a la manera de Scrum— no tiene por qué limitarse a los negocios. ¿Qué pasaría si la gente usara este método para atacar los grandes problemas con que nuestra especie tiene que vérselas, como la dependencia del petróleo, la educación de mala calidad, la falta de agua potable en regiones pobres del mundo o el crimen endémico? ¿Y si realmente hubiera una mejor manera de vivir, trabajar y resolver problemas? ¿Una manera de cambiar el mundo? La *hay*. Hay quienes utilizan Scrum para tratar de resolver cada una de las cuestiones mencionadas y están teniendo gran efecto en su empeño.

En este libro conocerás modalidades fundamentales del buen trabajo en equipo y sabrás por qué somos fatales para hacer cálculos y por qué trabajar tiempo extra retrasa tus proyectos. Te conduciré por todas las investigaciones y aplicaciones que personas, científicos y organizaciones han realizado con diligencia a través de los años y te diré cómo Scrum conjunta todo eso en una forma que puedes poner en marcha mañana mismo.

Te enseñaré *cómo* hacerlo. Pero antes quisiera contarte cómo llegué aquí.

RESUMEN

Hacer planes es útil; seguirlos ciegamente, no. Resulta *tentador* trazar un sinfín de gráficas, exponer a la vista de todos lo que debe hacerse en un gran proyecto. Pero cuando los planes detallados se confrontan con la realidad se vienen abajo. Integra en tu método de trabajo el supuesto del cambio, el descubrimiento y nuevas ideas.

Inspecciona y ajusta. Cada tanto haz una pausa, revisa lo que hiciste y ve si lo debes seguir haciendo o si puedes hacerlo mejor.

Renovarse o morir. Aferrarse a la antigua manera de hacer las cosas, de mando y control y rígida predictibilidad, conduce al fracaso. Entre tanto, tus competidores dispuestos a cambiar te harán morder el polvo.

Equivócate pronto para que puedas corregir con toda anticipación. La cultura corporativa suele dar más relevancia a los formularios, procedimientos y reuniones que a la creación visible de valor que pueda ser inspeccionada por los usuarios a intervalos cortos. El trabajo que no produce valor verdadero es irracional. Avanzar en ciclos reducidos en la elaboración del producto permite obtener pronta realimentación de los usuarios y eliminar *de inmediato* esfuerzos obviamente inútiles.

Capítulo dos

Los orígenes de Scrum

Para los pilotos de caza estadunidenses en Vietnam, un periodo de servicio significaba cumplir cien misiones en territorio enemigo. Cincuenta por ciento de ellos eran derribados. A algunos se les rescataba, pero la mayoría no conseguía regresar. En 1967, como piloto de caza joven e inexperto, fui enviado de la base de la fuerza aérea en Mountain Home, Idaho, a la base Udorn Royal Thai, en el norte de Tailandia, para desempeñar la función más peligrosa en la fuerza aérea de Estados Unidos: la de reconocimiento.

Esto sucedió mucho antes de las misiones no tripuladas de Predator y de las imágenes satelitales confiables. Despojado de todas sus armas, a mi RF-4C Phantom se le equipó con cámaras y un tanque extra de combustible. Mi labor consistiría en incursionar en territorio enemigo para que mi navegante tomara fotos de antes y después de las misiones de bombardeo. La mayoría de estas misiones se hacían de noche, así que yo atravesaba a toda prisa la oscuridad tropical a apenas treinta metros del suelo, rozando casi las copas de los árboles. En cuanto cruzaba la frontera y entraba en Vietnam del Norte, mi pantalla de visualización frontal se encendía como una máquina de juegos y el ruidoso sistema de alerta de misiles comenzaba a disparar una ráfaga de silbatazos y pitidos. El cielo se iluminaba con el fuego trazador de las armas antiaéreas y yo sabía que, en cuestión de minutos, el radar antimisiles ubicaría mi avión a una altura de menos de ciento cincuenta metros, demasiado baja para permanecer en el área de reflejo terrestre.

Mi adrenalina fluía con fuerza en esos momentos, pero nunca perdí la calma. Por el contrario, el peligro casi me tranquilizaba. Atribuyo esto a la instrucción que recibí en la fuerza aérea para controlar el riesgo. Ese adiestramiento me enseñó a hacer cuatro cosas: «**O**bserva, **o**riéntate, **d**ecide y **a**ctúa». Específicamente, tenía que observar el área objetivo, determinar la

mejor vía de acceso y salida de la zona de ataque, orientarme ante la eventualidad de hechos inesperados y actuar decididamente con base en la intuición y la costumbre adquirida. La vacilación podía costar la vida a un piloto, pero la imprudencia también. En cuanto mi navegante había tomado sus fotografías, yo tiraba nuevamente de la palanca y salía de la zona de peligro, al tiempo que la fuerza de gravedad reducía mi visión al mínimo. Mi navegante solía perder el conocimiento por esa misma causa, y a veces el control de sus esfínteres, pero jamás se quejó; conmigo siempre regresaba a salvo.

Yo no pasaba de ser entonces un joven piloto de *jets* con la esperanza de sobrevivir a las misiones que se le encomendaban. Pero nunca imaginé que mi experiencia de vuelo y el entrenamiento que recibí para pensar y actuar en una situación de vida o muerte definirían mi manera de operar el resto de mis días. Llegué a Vietnam en 1967 junto con dos escuadrones de cazas F-4 y dos de aviones de reconocimiento RF-4C, cien naves en total. Estas naves de reconocimiento remplazaron a dos escuadrones de RF-101, de los que sólo cuatro de un total de cincuenta no habían sido derribados en un año. Los cuatro sobrevivientes tenían tantos agujeros de bala que ya resultaban inservibles. No sé cómo pudieron aterrizar luego de su última misión. El RF-4C era un avión de combate más resistente, pese a lo cual la mitad de ellos fueron derribados en un año. Aunque el índice de sobrevivencia mejoró, cincuenta por ciento de quienes llegaron conmigo no regresaron a la base, si bien a algunos se les rescató en la selva antes de que fueran hechos prisioneros.

Al volver de la guerra hice una maestría en estadística en Stanford y pasaba todo el tiempo posible en el Stanford Artificial Intelligence Laboratory. Luego me fui a dar clases de matemáticas a la Academia de la Fuerza Aérea, periodo durante el cual también hice un doctorado en biométrica en la Medical School de la University of Colorado. Un día pregunté a mi asesor, el doctor John Bailar, distinguido investigador en medicina y estadística, cómo podía hacer una tesis que no fuera a dar a un polvoriento estante de la biblioteca. Él me tendió trescientos artículos médicos sobre el cáncer, cada uno de ellos con gráficas de esa enfermedad que variaban mucho entre humanos y animales y entre tipos de tumores. Me dijo que si podía explicar esas diferencias, me daría mi doctorado. Así lo hice y obtuve mi título.

Para ello tuve que pasar varios años tratando de entender por qué una célula se vuelve cancerosa. Aprendí mucho sobre la teoría de sistemas y que

un sistema sólo tiene ciertos estados estables. Conforme una célula evoluciona, pasa de un estado estable a otro. Comprender las reglas para que un sistema adaptativo complejo pase de un estado a otro y para que éste sea positivo y no negativo fue algo a lo que dediqué casi una década.

Años más tarde se me ocurrió que las organizaciones, los equipos y las personas son sistemas adaptativos complejos. Lo que hace pasar a las células de un estado a otro es lo mismo que hace pasar a las personas de un estado a otro. Para que una célula cambie, primero inyectas energía en el sistema. Al principio hay caos, parece no haber reglas, todo está en estado de flujo. Cuando haces esto en organizaciones que intentan cambiar, la gente suele asustarse, no entiende qué pasa, no sabe qué hacer. Pero más pronto de lo que se cree, e igual que una célula, una organización se adapta a un nuevo estado estable. La pregunta es si este nuevo estado es mejor que el anterior; ¿la célula es sana o cancerosa? Me pregunté entonces *cómo deducir algunas reglas simples para que los equipos pudieran adoptar un estado más productivo, satisfactorio, sustentable, grato y efusivo.* Pasé los quince años siguientes tratando de resolver esto.

Durante el periodo presidencial de Ronald Reagan, el gobierno redujo drásticamente las subvenciones a la investigación científica, entre ellas la de mi investigación en los National Cancer Centers, donde me desempeñaba como investigador titular de recolección y análisis de datos para pruebas clínicas y estudios epidemiológicos en el Colorado Regional Cancer Center. Mientras pensaba qué hacer, recibí una llamada de MidContinent Computer Services, compañía que se había enterado de que yo era el principal experto en el campo de su más reciente tecnología. MidContinent ofrecía servicios a ciento cincuenta bancos de América del Norte. Su nuevo producto más demandado eran las redes de «cajeros automáticos». Estábamos en 1983, cuando disponer de efectivo solía significar hacer fila en un banco o formarse en la ventanilla de servicio en el auto. Se emitía un cheque para cambiarlo por la suma deseada y se le tendía al cajero.

Los cajeros automáticos remediaron este embrollo, pero MidContinent tenía problemas para que sus redes se entendieran entre sí. Necesitaba un especialista en sistemas para resolverlos y me hizo una lucrativa oferta como vicepresidente de sistemas avanzados. Sus computadoras en red eran iguales a las que yo había usado durante años en mi doctorado, así que yo resultaba idóneo para el trabajo.

O al menos eso creí. Nada es fácil nunca, ¿verdad? Cuando llegué a esa compañía, me topé con una división que ejecutaba sus proyectos utilizando el

método en cascada. Cientos de programadores de computadoras pasaban el día entero en su escritorio trabajando en apariencia, pero nunca hacían nada a tiempo ni dentro del presupuesto. En cuanto a los cajeros automáticos, sus costos eran treinta por ciento superiores a sus ingresos. Las ineficiencias eran inconcebibles.

Lo primero que hice fue dedicar un tiempo a averiguar cómo funcionaban las cosas. Ya te imaginarás cómo trataba la alta dirección al personal a mi cargo. Había muchos gritos, microgestión, conducta agresivapasiva y exigencias de trabajo más intenso y tiempo extra. Pero por más que la dirección presionaba, los proyectos seguían atrasándose crónicamente, rebasando el presupuesto y sin rendir lo que debían.

Decidí que lo mejor era cambiar todo. La operación estaba demasiado dañada para repararla poco a poco, así que opté por formar una compañía dentro de la compañía. Pedí al director general, Ron Harris, que me permitiera crear una organización aparte, con todos los involucrados en las redes de cajeros automáticos. Tendríamos nuestro propio equipo de ventas, de mercadotecnia y de finanzas. Harris era un director brillante y creativo que confiaba en mi trabajo. De haber habido otro director, quizá no lo habría permitido. Tras escuchar mi idea, Harris me dijo: «Si quieres echarte a cuestas ese dolor de cabeza, adelante».

Y así lo hice. Fui con los desarrolladores y gerentes y les dije: «Lo primero que debemos hacer es dejar de hacer cosas que están acabando con nosotros». Es como ese viejo chiste en el que alguien se da de topes contra la pared sólo por el alivio que siente al dejar de hacerlo. «Debemos descubrir una mejor manera de trabajar», les dije, «y tenemos que empezar de inmediato».

Nuestra pequeña compañía pasó a ser entonces un equipo dividido en subequipos. Los bonos no se basaban en el desempeño individual, sino en el de toda la compañía. Ideamos herramientas que diez años después se abrirían paso hasta Scrum, como los conceptos de «responsable del producto», «retraso del producto» y *sprints* semanales, que detallaré más adelante. Seis meses más tarde, éramos la división más rentable de la compañía. Los ingresos eran ya treinta por ciento superiores a los gastos. Nuestros sistemas Nonstop Tandem fueron las primeras computadoras de transacciones en línea en inspirar confianza a los bancos y las desplegamos en toda América del Norte. Hoy en día casi no hay ciudad estadunidense que no tenga un cajero automático. Y todas estas máquinas saben con exactitud cuánto dinero tienes. Mi equipo tuvo mucho que ver en eso y ahora te doy la bienvenida a él.

Aprender a pensar como robot

Tras mi primera carrera en el ejército y mi segunda en la academia, casi me sentía un extraño en los negocios. Pero esta perspectiva «desde fuera» fue una de mis ventajas principales. Desde el primer día, para mí fue un misterio por qué la gente insistía en trabajar en formas que ella misma *sabía* que eran ineficientes y antieconómicas, tanto como deshumanizadoras y deprimentes. Supongo que creía que ésa era la mejor manera de trabajar simplemente porque así es como lo hacían todos.

Aunque la pasé muy bien en MidContinent, ansiaba poner a prueba mis habilidades en nuevos retos. En las dos décadas siguientes trabajé en compañías grandes y pequeñas como vicepresidente de ingeniería o director tecnológico. En cada una de ellas intenté hacer que los equipos trabajaran en común de modo más eficiente. Uno de estos empleos me llevó a un edificio en Cambridge, Massachusetts, muy cerca del Massachusetts Institute of Technology (MIT). Doctores y profesores de esta institución acababan de poner una compañía fabricante de robots y se habían quedado sin espacio en su laboratorio en el MIT, así que subarrendaron algunas oficinas de la compañía en la que yo trabajaba.

Semanas después de haberse mudado con nosotros, sucedió algo totalmente inesperado: un robot de seis patas del tamaño de un gato entró corriendo a mi oficina y se puso a perseguirme en torno a mi escritorio. Luego llegaron las personas que lo controlaban, quienes se disculparon nerviosamente por lo ocurrido, pese a lo cual esto volvía a suceder cada tanto. Un robot escapaba del laboratorio corriendo por el edificio. Yo oía el tableteo mecanizado de sus piernas ir de aquí para allá en los pasillos.

Yo siempre ofrecía vino y cerveza los viernes en la tarde para que los empleados se relajaran y convivieran tras una dura semana de trabajo. Invitaba a la gente del MIT del otro lado del pasillo y una tarde apareció Rodney Brooks. Profesor de inteligencia artificial del MIT, Brooks era uno de los fundadores de la compañía de robots, así que le pregunté cómo funcionaban aquellas máquinas errantes.

«Hemos pasado décadas tratando de producir una máquina realmente inteligente», contestó. «Hemos invertido miles de millones de dólares y muchos años de trabajo fabricando las computadoras más grandes posibles con las más grandes bases de datos, pero lo único que logramos fue una computadora capaz de ganarnos en el ajedrez».

Sus robots, explicó, adoptan un enfoque totalmente distinto. En vez de fabricar algo con un solo cerebro central, produjeron un robot con seis patas, cada una de ellas con un cerebro propio. Un procesador en la espina dorsal contenía unas cuantas reglas: avanzar, retroceder, no chocar con otras patas. El chip de red neuronal en la cabeza del robot conocía esas reglas y fungía como árbitro entre todas las partes. Les decía a las patas qué veía por su cámara, cuándo chocaba con un obstáculo, este tipo de cosas.

Lo interesante, dijo Brooks, es que cada vez que se le encendía, un robot aprendía a caminar por vez primera. No disponía de una base de datos que le indicara dónde se hallaban las cosas en la habitación. El mundo era su base de datos. Conocía todo por primera vez siempre que se le encendía. Tropezaba con cosas y las deducía a partir de sus circunstancias reales, lo cual quiere decir que podía adaptarse a cualquier entorno.

«Déjame enseñarte», me dijo Brooks, y me llevó a su laboratorio. Ahí insertó un chip neuronal virgen en uno de sus robots insectoides, al que vi bambolearse mientras cobraba vida. Titubeante en un principio, recorrió la sala dando traspiés como un cervatillo que usara por primera vez sus patas. Pero a cada paso se volvía más seguro. Las patas aprendían rápidamente a colaborar y trabajar en común. Minutos más tarde, el robot ya corría por el cuarto. No tenía nada almacenado o programado sobre cómo caminar; unas cuantas reglas mantenían trabajando en conjunto a esos componentes. Aquellas patas no pensaban; actuaban. El ingenio y sencillez de ese sistema me dejó atónito. Ahí estaba algo que hacía justo lo que a mí me habían enseñado a hacer al volar en Vietnam: **o**bservar, **o**rientarse, **d**ecidir y **a**ctuar. Asimilaba su entorno y se conducía con decisión a partir de los datos que obtenía de él.

—¿Qué pasaría —pregunté entonces a Brooks— si pudiéramos idear un conjunto simple de instrucciones que permitiera que equipos de personas trabajaran en común, a la manera de esas patas? Tales equipos se autoorganizarían y autooptimizarían, justo como este robot.

—No sé —respondió él—. ¿Por qué no haces la prueba y me dices qué resulta?

No persigas cascadas

Comprendí entonces que si podía crear un sistema que, como ese robot, coordinara a seres pensantes independientes con realimentación constante

sobre su entorno se alcanzarían muy altos niveles de desempeño. Si agilizábamos el flujo de información entre las «patas» de un grupo, podríamos obtener eficiencias nunca antes vistas.

Mi conversación con Brooks tuvo lugar hace más de dos décadas. Durante muchos años, él fue director de robótica e inteligencia artificial del MIT, y el robot de extremidades delgadas e inseguras que yo conocí, el cual respondía al nombre de Gengis Kan, luce ahora en la Smithsonian Institution, como pieza de colección. Una de las compañías de Brooks, iRobot, fabrica en la actualidad la aspiradora Roomba, la que, para limpiar pisos, emplea la misma inteligencia adaptativa que Gengis Kan usó para perseguirme en mi oficina. La innovación más reciente de Brooks en Rethink Robotics, el robot Baxter, puede trabajar con humanos en un área común.

La labor de Brooks me sirvió de inspiración. En 1993 llevé conmigo esas ideas a Easel, compañía que me contrató como vicepresidente de tecnología de objetos. Los ejecutivos de esta corporación querían que mi equipo desarrollara en seis meses una línea de productos totalmente nueva, destinada a algunos de sus clientes principales, como Ford Motor Company, la cual usaba el *software* de Easel para diseñar y producir aplicaciones internas. Así, informé a mi equipo de desarrollo que sabía que no lo lograríamos siguiendo el viejo estilo de desarrollo de *software*.

Este estilo antiguo era el método en cascada que describí en el capítulo anterior: la diagramación esmerada en grandes gráficas de Gantt de todo lo relacionado con un proyecto; de cada tarea, calculada en horas exactas, en hermosos colores que fluían por la página como en un salto de agua. Esas gráficas eran imponentes en su precisión. Pero también pura mentira.

Yo sabía que la metodología en cascada nos haría exceder en meses, si no es que en años, nuestra fecha límite. Debíamos inventar una manera completamente distinta de hacer las cosas. Por lo tanto, fui con el director general y le dije que prescindiríamos de las gráficas de Gantt. Él se alarmó y me preguntó por qué.

—¿Cuántas gráficas de Gantt ha visto usted en su carrera? —le pregunté a mi vez.

—Cientos —contestó.

—¿Y cuántas de ellas resultaron atinadas?

Hizo una pausa.

—Ninguna.

Le dije entonces que al final de cada mes le entregaría *software* útil, en lugar de una defectuosa gráfica de Gantt. Podría probarlo y ver si íbamos por

buen camino. Tendríamos que hacerlo, pues de lo contrario no cumpliríamos la fecha límite.

Mi equipo y yo dedicamos varias semanas a leer cientos de ponencias, libros y artículos sobre la organización de equipos y el desarrollo de productos. Un día, uno de los desarrolladores llegó con un artículo de 1986 de la *Harvard Business Review* escrito por dos profesores de administración japoneses, Hirotaka Takeuchi e Ikujiro Nonaka. Llevaba por título «The New New Product Development Game» (El nuevo nuevo desarrollo de productos). Takeuchi y Nonaka habían estudiado equipos de algunas de las compañías más productivas e innovadoras del mundo: Honda, Fuji-Xerox, 3M, Hewlett-Packard y otras. Afirmaban que la vieja manera de desarrollar productos, representada por el sistema de planeación gradual de programas de la NASA —un sistema en cascada—, presentaba fallas de origen. En cambio, las mejores compañías seguían un proceso de desarrollo empalmado, más rápido y flexible. Sus equipos eran interfuncionales y tenían autonomía, autoridad para tomar decisiones y un propósito trascendente: perseguían algo más allá de ellos. La dirección no daba órdenes. Por el contrario, los ejecutivos eran líderes de servicio y facilitadores dedicados a quitar obstáculos a sus equipos, más que a decirles qué hacer y cómo desarrollar productos. Aquellos profesores japoneses comparaban el trabajo en común con un equipo de *rugby* y decían que los mejores equipos actúan como en un *scrum*: «La pelota circula entre los integrantes del equipo a medida que éste avanza por el campo como una unidad»[6].

El artículo de Takeuchi y Nonaka causó revuelo al momento de su publicación, siete años antes de que nosotros lo leyéramos en Easel. Asombró a todos, pero nadie hizo nada al respecto. El gerente estadunidense promedio era incapaz entonces de dotarlo de sentido, pese a que Toyota aumentaba ya rápidamente su participación de mercado aplicando ese enfoque. En Easel no teníamos nada que perder. Así, decidimos hacer la prueba, pese a que el artículo versaba sobre manufactura, no sobre desarrollo de *software*. A mí me pareció que las ideas de esos autores sacaban provecho de algo fundamental, un proceso descriptivo del trabajo en equipo óptimo en cualquier área. Esto se extendía a todos los experimentos que yo había hecho desde mi primer empleo en el sector privado, en MidContinent.

Fue así como nació Scrum. Entregamos el producto a tiempo, dentro del presupuesto y con menos errores que todos los productos previos.

Las posibilidades de esta nueva forma de gestión de proyectos me entusiasmaron tanto que toda mi labor futura se centró en afinar Scrum para

su uso en toda clase de compañías. En 1995 presenté, en una conferencia de investigación de la Association for Computing Machinery, un artículo en coautoría con Ken Schwaber titulado «SCRUM Development Process» (Proceso de desarrollo con SCRUM), que codificaba esas prácticas. Desde entonces hemos prescindido de tantas mayúsculas y modificado un poco la idea, pero los principios siguen en pie y las compañías que adoptan este proceso suelen ver beneficios inmediatos[7].

Inspección y ajuste

Los buenos equipos de trabajo que utilizan Scrum pueden alcanzar lo que nosotros llamamos «hiperproductividad». Aunque cueste creerlo, suelo ver aumentos de productividad de entre trescientos y cuatrocientos por ciento en los grupos que implementan correctamente Scrum. Los mejores equipos pueden obtener aumentos de hasta ochocientos por ciento y reproducir su éxito una y otra vez. Asimismo, pueden más que duplicar la *calidad* de su trabajo.

¿Cómo incorporar autonomía, trascendencia y fecundación mutua en un equipo que usa Scrum y obtener hiperproductividad de esa combinación? De eso tratará el resto del libro, pero aquí describiré la estructura básica.

Dado que Scrum proviene de técnicas que se emplean en la manufactura nipona, conviene saber un poco acerca de cómo las aprendieron los japoneses. Irónicamente, adquirieron muchas de ellas de un estadunidense, W. Edwards Deming, quien trabajó con el general Douglas Mac-Arthur durante la ocupación estadunidense de Japón tras la Segunda Guerra Mundial. Para reconstruir la economía, MacArthur despidió a casi toda la alta dirección japonesa, ascendió a gerentes de línea y llevó consigo a expertos estadunidenses en operaciones de negocios como Deming. La influencia de este último en la manufactura nipona fue profunda. Instruyó a cientos de ingenieros en lo que hoy se conoce como «control estadístico de procesos». La idea básica consiste en evaluar con exactitud qué se hace y cuán bien, y perseguir la «mejora continua». No una mejora de una sola vez, sino la mejora constante. Busca siempre algo en lo que puedas mejorar. Nunca te contentes con lo que eres. Para ello, experimenta sin cesar, a fin de ver si puedes conseguir una mejora. «¿Este método que acabo de probar es mejor? ¿Y este otro? ¿Y si sólo cambio esta cosa?».

En 1950, Deming pronunció ante líderes japoneses de negocios una charla que se volvería famosa. Entre ellos se encontraba Akio Morita, fundador de Sony. Dijo Deming:

> Por buenos que sean sus técnicos, como líderes ustedes deben buscar avances en la calidad y uniformidad de sus productos si quieren que sus técnicos hagan mejoras. En consecuencia, toca a la dirección dar el primer paso. Los técnicos de sus fábricas y compañías deben saber primero que a ustedes les apasiona la mayor calidad y uniformidad de los productos y que se sienten responsables de la calidad de estos últimos.
> Pero nada resultará de esto si sólo lo dicen. La acción es importante[8].

Y el método para la acción, al que quizá Deming debe su fama más que a nada, es el ciclo Planea, Haz, Revisa, Actúa (**P**lan, **D**o, **C**heck, **A**ct, PDCA). Este ciclo puede aplicarse a la producción de prácticamente cualquier cosa, sea un auto, un videojuego o hasta un avión de papel.

Cuando enseño a la gente a aplicar Scrum, me sirvo justo de eso: de aviones de papel. Divido al grupo en equipos y les digo que la meta es hacer el mayor número posible de aviones. Habrá tres roles en cada equipo: una persona confirmará que los aviones vuelen; otra participará en el proceso de montaje, pero buscará formas en que el equipo puede hacer mejores aviones y aumentar su producción, y todos los demás miembros se ocuparán de hacer el mayor número posible de aviones en el tiempo de montaje asignado.

Posteriormente indico que se realizarán tres ciclos de elaboración de aviones, de seis minutos cada uno. A planear cómo hacerlos se dedicará un minuto, tres a elaborarlos y probarlos y dos a revisar. En esta última fase, el equipo ve el modo de mejorar su procedimiento de elaboración de aviones. ¿Qué salió bien? ¿Qué salió mal? ¿El diseño debería cambiar? ¿En qué puede mejorar el procedimiento de elaboración? Finalmente, el equipo actúa. En el mundo de Deming, «actuar» significa cambiar la manera de trabajar con base en los resultados reales y las aportaciones reales del entorno. Esta estrategia es igual a la que siguió el robot de Brooks.

Recorre tres veces este ciclo y así hagas aviones de papel o naves espaciales, mejorarás, y mucho (trabajarás de dos a tres veces más rápido y con al menos el doble de calidad). El ciclo PDCA, una idea radical cuando Deming la transmitió a los japoneses, fue el medio que permitió a Toyota convertirse en la compañía automotriz número uno del mundo. Y es el medio

para hacer cualquier tipo de manufactura «saneada» (término de uso común en Estados Unidos para aludir a los conceptos del sistema de producción de Toyota), o de desarrollo de productos con Scrum.

Renovarse o morir

Una de las razones de que una nueva manera de hacer las cosas haya sido imperativa y de que tantas compañías la hayan adoptado es que el estado del desarrollo de *software* era pésimo. Casi todos los proyectos se atrasaban, excedían el presupuesto y no daban resultado. Y no porque la gente fuera tonta o codiciosa, sino por la forma en que concebía su trabajo. Insistía en el método en cascada, en que planear todo era posible e incluso en que en el curso de un proyecto de varios años nada cambiaría. Esto es sencillamente aberrante.

Lo supe de primera mano en BellSouth, al visitarla como consultor hace unos años. Esta compañía tenía ingenieros de primera, muchos de ellos procedentes de los famosos Bell Labs. Ejecutaban la cascada a la perfección. Competían por proyectos inmensos, de entre diez y veinte millones de dólares. Reunían todos los requisitos del cliente, se ausentaban dieciocho meses y entregaban a tiempo y dentro del presupuesto justo lo que el cliente había solicitado. BellSouth era una de las muy pocas compañías en todo el mundo capaces de hacer eso. El problema asomaba cuando el cliente ya no quería lo que había dicho que quería. Las circunstancias habían cambiado. Los ciclos de negocios se acortaban, y los clientes demandaban servicios con mayor capacidad de respuesta.

Fui contratado para ayudar a BellSouth a comprender qué hacía mal. Pronto me di cuenta de que el problema era la manera en que trabajaban sus empleados. Esto puede ser difícil de escuchar para quien tiene la impresión de hacer todo bien. Así, un día me planté frente a ciento cincuenta ingenieros de BellSouth y les dije que si no cambiaban de modelo y eran más sensibles al cliente, su empresa no duraría. Reaccionaron con hostilidad. Todos ellos eran hombres y mujeres inteligentes, pero creían que mis ideas no eran sino una moda gerencial más. No logré darme a entender, así que simplemente me alcé de hombros e hice una última advertencia: «Recuerden que hay sólo dos opciones: renovarse o morir». Como tal vez ya lo sepas, BellSouth desapareció tiempo después.

Shu Ha Ri

Scrum echa raíces en el pensamiento y la práctica japoneses. En un viaje reciente para reunirme con él, el profesor Ikujiro Nonaka me aclaró que en Japón no se ve a Scrum como la última moda en el ámbito del trabajo. Lo consideran una manera de hacer, una manera de ser, un modo de vida. Cuando enseño a la gente a aplicar Scrum suelo hablar de mi experiencia personal de años en el estudio del arte marcial japonés del aikido.

Scrum —como el aikido o el tango, incluso— es algo que sólo puedes aprender con la práctica. Tu cuerpo, mente y espíritu se sincronizan gracias a la práctica y mejora constantes. En las artes marciales se enseña el concepto de *Shu Ha Ri*, el cual apunta a diferentes niveles de maestría. En el estado *Shu* se conocen todas las reglas y formalidades. Se repiten como si fueran pasos de baile, para que el cuerpo las asimile. No son alteradas ni un ápice.

En el estado *Ha*, una vez dominadas las formalidades, pueden hacerse innovaciones. Se imprime brío extra al paso por la pista de baile.

En el estado *Ri* es posible desechar las formalidades, gracias a que ya se domina la práctica, y se puede ser creativo sin traba alguna, porque el significado del aikido o el tango ya ha arraigado tanto que cada paso expresa su esencia.

Scrum es muy parecido. Requiere atención y práctica, aunque también un esfuerzo continuo por llegar a un nuevo estado, en el que las cosas fluyan y sucedan. Si alguna vez has visto a grandes bailarines o gimnastas, sabes que sus movimientos pueden parecer casi naturales, como si ellos no hicieran otra cosa que ser. Parecería que no pudieran ser otra cosa que lo que son en ese momento. Lo experimenté un día en que un diminuto maestro de aikido me lanzó por los aires de tal forma que caí suavemente en la colchoneta, como un bebé al que se mete con delicadeza en la cuna.

Esto es lo que debes conseguir con Scrum. Ése es el estado que deseo que todas las personas alcancen en la vida. El trabajo no tiene por qué ser enojoso. Puede fluir; puede ser una expresión de dicha, un acoplamiento hacia un propósito más alto. Podemos mejorar. ¡Podemos ser excelentes! Sólo tenemos que practicar.

Dedicaré cada uno de los capítulos siguientes a un aspecto particular de Scrum. Estas inmersiones pretenden ofrecerte la lógica detrás de los conceptos y explicar por qué Scrum se estructura como lo hace. En el apéndice hallarás los aspectos básicos de Scrum (una magnífica descripción

del sistema), pero sólo te indicará *qué* hacer. Si continúas a mi lado, yo te diré *por qué*.

RESUMEN

Dudar es mortal. Observa, oriéntate, decide y actúa. Determina dónde estás, evalúa tus opciones, toma una decisión ¡y actúa!

Busca respuestas a tu alrededor. Los sistemas adaptativos complejos siguen unas cuantas reglas, que adquieren de su entorno.

Los grandes equipos. Son interfuncionales, autónomos, con autoridad y con un propósito trascendente.

No supongas; planea, haz, revisa, actúa. Planea qué harás. Hazlo. Revisa si lograste lo que querías. Actúa en consecuencia y cambia tu manera de hacer las cosas. Repite esto en ciclos regulares y alcanzarás la mejora continua.

Shu Ha Ri. Primero aprende las reglas y formalidades y, una vez que las domines, haz innovaciones. Por último, en un alto estado de maestría, desecha las formalidades y sólo *sé*, con el aprendizaje que ya has interiorizado y tomando decisiones de modo casi inconsciente.

Capítulo tres

Equipos

Los equipos son los que hacen las cosas en el mundo del trabajo. Los hay que fabrican automóviles, contestan teléfonos, efectúan cirugías, programan computadoras, transmiten noticias o irrumpen en departamentos ocupados por terroristas. Claro, también hay artesanos o artistas que trabajan por su cuenta, pero los equipos son lo que hace girar el mundo y en lo que se basa Scrum.

Aunque todos sabemos eso, en los negocios solemos centrarnos en los individuos, aun si la producción es un esfuerzo conjunto. Piensa en los bonos de desempeño, los ascensos o las contrataciones. Todo gira alrededor del actor individual, no del equipo. Éste es un grave error.

Los gerentes tienden a centrarse en el individuo porque esto tiene sentido intuitivo. Quieres a tu lado a los mejores y cada persona es distinta, así que si prestas toda tu atención en conseguir a quienes rinden más, obtendrás los mejores resultados, ¿no es así? Bueno, no es tan sencillo como parece.

Tomemos como ejemplo el proceso de calificar a los estudiantes en el aula. En Yale University, el curso de programación de computadoras del profesor Stanley Eisenstat, CS 323, sobresale por su dificultad. Cuando los alumnos se quejaron de que se les dejaban tareas muy prolongadas, el maestro no aligeró su carga, sino que empezó a fijarse en cuánto tiempo necesitaba cada uno de ellos para hacerlas. Joel Spolsky, alumno de Eisenstat en los años ochenta y dueño ahora de una empresa de *software*, comparó luego esos datos con las calificaciones obtenidas. Quería saber si había una correlación entre el tiempo dedicado a un proyecto y la calificación asignada. Curiosamente, no la hubo. Hay quienes trabajan rápido y obtienen la más alta calificación 10 y quienes trabajan meticulosamente y logran lo mismo. La única diferencia es la cantidad de tiempo invertida. ¿Cómo se aplica esto a los negocios?

Si eres gerente, es probable que no sólo quieras contratar a quienes obtienen la calificación más alta, sino también a quienes lo consiguen en el menor tiempo posible. En el estudio de Yale, los estudiantes rápidos rebasaron a sus compañeros lentos en una proporción de 10:1. Fueron diez veces más rápidos y obtuvieron una calificación igual de satisfactoria. Diez veces más rápido es mucho, ¿no? Así, parecería que las compañías deben contratar a los más rápidos y descartar a los lentos. Todo *indica* que éste es el mejor método para aumentar la productividad, pero lo cierto es que otros factores pueden ser más cruciales aún.

Si en vez de fijarte en los individuos te centras en los equipos verás algo muy interesante. En un estudio se examinaron tres mil ochocientos proyectos, desde tareas en despachos contables hasta desarrollo de *software* para acorazados y proyectos tecnológicos en IBM. Los analistas no tomaron en cuenta el desempeño individual, sino el de los equipos. Y al indagar cómo les iba emergió algo inesperado. Si el mejor de ellos ejecutaba una tarea en una semana, ¿cuánto crees que tardaba el peor? Podrías aplicar a este caso la misma proporción que se observó en Yale, de 10:1 (lo cual querría decir que el equipo lento demoró más de dos meses en hacer lo que el rápido liquidó en una semana). Pero la respuesta correcta es que la diferencia entre el desempeño grupal y el individual es mucho mayor. El equipo lento no tardó diez semanas, sino *dos mil*, en hacer lo que el rápido hizo en una. Así de grande es la diferencia entre el mejor y el peor equipo. ¿Dónde, entonces, deberías centrar tu atención? ¿En cada individuo, en el que podrías obtener una mejora de diez veces si, como por arte de magia, lograras volver genios a todos tus empleados, o en el equipo, donde puedes elevar la productividad en una magnitud enorme aun si sólo vuelves mediocres a tus peores equipos? Claro que apuntar a la mediocridad no te llevará más allá de ella. Pero ¿y si pudieras volver excelentes a todos tus equipos?

Cualquier cosa es posible en ciertos momentos y lugares, y con ciertos equipos de un número reducido de personas. Quizá nunca hayas estado en un equipo así, pero seguramente lo has visto en acción o escuchado leyendas de lo que es capaz de hacer. Crecí cerca de Boston y ahora vivo ahí, de manera que algunos de los grandes equipos que me vienen a la mente son los Celtics de los años ochenta y los Patriotas de Nueva Inglaterra de tiempos de Tom Brady. Cuando estos equipos jugaban, parecían hacerlo diferente a los demás. Pases y jugadas que antes se creían imposibles se volvían de repente parte del plan de juego. Era como si un estado de gracia se apoderara de los jugadores, quienes por un momento no podían hacer nada mal. Larry Bird cruzaba la

cancha y pasaba el balón sin ver siquiera lo que parecía ser madera pura. Pero justo cuando la pelota semejaba estar por salir de la cancha, Kevin McHale *aparecía* donde debía, lanzando a su vez el balón hacia un ala —como sin ver, de igual modo—, donde Robert Parish estaba idealmente ubicado para disparar. Un acoplamiento perfecto de confianza y propósito como éste genera grandeza.

Todos hemos visto a equipos de esta clase. Y algunos hemos tenido la suerte de estar en uno —o *más* de uno— de ellos en el curso de nuestra vida. Mientras concebía Scrum, indagué qué tenían los equipos de alto rendimiento que otros equipos no tuvieran. ¿A qué se debía, me pregunté, que algunos cambiaran el mundo y otros se estancaran en la mediocridad? ¿Cuáles eran los elementos comunes de los grandes equipos? Y, sobre todo, ¿era posible reproducir esos elementos?

La respuesta es sí.

En el artículo en el que detallaron lo que más tarde se convertiría en Scrum, «The New New Product Development Game» (El nuevo nuevo desarrollo de productos), los profesores Takeuchi y Nonaka describieron las características de los equipos que hallaron en las mejores compañías del mundo:

1. *Trascendentes*. Los buenos equipos tienen un propósito más allá de lo normal. Esta meta de realización grupal les permite pasar de lo ordinario a lo extraordinario. La sola decisión de no quedarse en el promedio sino aspirar a la grandeza cambia en efecto la manera en que los miembros de un equipo se ven a sí mismos y aquello de lo que son capaces.
2. *Autónomos*. Los buenos equipos se organizan y gestionan solos, tienen el poder para tomar sus propias decisiones sobre cómo trabajar y poseen la autoridad indispensable para apegarse a esas decisiones.
3. *Interfuncionales*. Los buenos equipos cuentan con todas las habilidades necesarias para llevar a cabo un proyecto: planeación, diseño, producción, ventas, distribución… Y estas habilidades se alimentan y refuerzan entre sí. Como lo describió uno de los miembros de un equipo que diseñó una cámara revolucionaria para Canon: «Cuando todos los integrantes están presentes en la misma sala, se apropian de la información de los demás sin siquiera proponérselo. Cada uno comienza a pensar entonces en términos de lo mejor para el grupo, no para sí mismo»[9].

¿Cómo crear un equipo que apunte a una meta más alta, se organice solo y se alimente constantemente de las habilidades de cada uno de sus miembros? Pasé mucho tiempo ponderándolo. Después de todo, no puedes nada más gritar a la gente que se organice mejor y aspire a la trascendencia; esto tiene que venir de dentro. Imponerlo dará al traste con lo que quieres hacer. ¿Existirá tal vez una serie de reglas sencillas que hagan magia?

La larga fila gris

Recordé entonces que yo había formado parte de uno de esos equipos mágicos. Esto sucedió a principios de los años sesenta, cuando era cadete de la Academia Militar Estadunidense, mejor conocida como West Point. Durante mi último año ahí se me nombró oficial instructor de mi compañía de cadetes, la L2, también llamada los Perezosos.

En 1963 había veinticuatro compañías en West Point, de la A1 a la M1 y de la A2 a la M2. Tres veces a la semana, estas compañías tomaban la plaza de armas y marchaban en uniforme de gala, con fusiles a un lado y espadas al otro, y con hombreras blancas y toda suerte de accesorios. Estas formaciones de desfile han sido una competencia en aquella academia desde hace casi doscientos años. En 1963, los Perezosos llevaban más de un siglo en el último lugar de esa contienda.

Un oficial instructor carece de poder directo. No pertenece a la estructura de mando de la compañía. Nadie responde a sus órdenes. Nadie tiene que hacer lo que él dice. Pero luego de cada desfile, los oficiales instructores de las diversas compañías se reúnen para calificar a cada una de ellas de acuerdo con varios criterios. Como instructor de los Perezosos, decidí hacer todo más transparente. Elaboraba gráficas de colores de lo que había salido bien y lo que no y las fijaba en el cuartel donde todos los miembros de la compañía pudieran verlas.

Al principio, las críticas eran sencillas: «Charlie arrastró su espada, Jim perdió el paso al dar vuelta, Dave saludó sin brío». No había culpas ni castigos; sólo se exponían los hechos que los demás instructores habían señalado en las evaluaciones. Pero ésas eran justo las razones de que la L2 ocupara el último lugar de la lista.

En unas cuantas semanas, los cadetes de mi grupo pulieron su conducta, de manera que las malas calificaciones comenzaron a apuntar al comandante de la compañía. Sus órdenes no eran tan claras como debían; el ritmo, no tan

vigoroso. No es de sorprender que se me amonestara por criticar al comandante, a lo que yo me limité a responder: «Las calificaciones son las calificaciones. Yo nada más muestro lo que es. Los soldados ya hicieron su parte, así que ahora el problema es él. ¿Quiere resolverlo o arrastrarlo para siempre?». Semanas después, la L2 era la compañía número uno de West Point.

El cadete más respetado en la historia de West Point es el general Douglas MacArthur. Alcanzó el rango más alto entre todos los graduados de esa institución y fue un oficial distinguido en las dos guerras mundiales. Como general de cinco estrellas galardonado con la Medalla al Mérito, tenía un lazo especial con el cuerpo de cadetes.

En mayo de 1962, un año antes de que se me designara instructor de mi compañía, el general MacArthur pronunció su último discurso en West Point. Basta imaginar la escena: tres mil hombres con su uniforme gris de cadetes en una sala inmensa de paredes de piedra con vastas columnas y candelabros enormes pendientes de altos techos. A unos diez metros se levantaba contra una pared un estrado que dominaba la sala. El general MacArthur, ya frágil, se puso de pie en el templete y pronunció el que hoy se conoce como el discurso de la «Larga fila gris» (en alusión al color del uniforme de los cadetes):

> Ustedes son el fermento que mantiene vivo nuestro sistema nacional de defensa. De sus filas emergen los grandes capitanes que tienen en sus manos el destino de la nación cuando resuenan los tambores de la guerra.
> La larga fila gris nunca nos ha fallado. Si así lo hiciera, un millón de espíritus con uniformes verde olivo, caqui, azul y gris se alzarían de sus tumbas bramando estas palabras mágicas: Deber, Honor, País[10].

Pareció en ese momento que una legión de aquellos espíritus se elevara detrás de MacArthur mientras daba su última orden al cuerpo de cadetes. Y entonces, tres mil hombres entrenados para la guerra, reacios a las lágrimas, rompieron a llorar.

> En mis sueños oigo otra vez el estrépito de las armas, las descargas de fusilería, el raro y lastimero murmullo del campo de batalla. Pero en el ocaso de mi memoria, vuelvo a West

> Point, donde siempre sonará y resonará el mismo coro: Deber, Honor, País.
> Hoy es el último día en que pasaré revista a ustedes. Pero quiero que sepan que, cuando cruce el río, mis últimos recuerdos serán para el cuerpo de cadetes de West Point[11].

Hasta nuestros días, todos los soldados de esa academia deben aprender de memoria este discurso, línea por línea, palabra por palabra, para poder graduarse. Se ha convertido en la guía espiritual del cuerpo de cadetes y del cuerpo de oficiales estadunidenses en general: Deber, Honor, País.

Casi un año después de que pronunciara esas palabras, el general MacArthur pasó a mejor vida. En su funeral marchó una compañía selecta. Al lento y rítmico redoblar de los tambores, los Perezosos, la peor compañía de cadetes durante más de cien años, desfiló detrás de la carroza con el féretro de uno de los mayores generales en la historia de Estados Unidos.

Meses más tarde, yo desfilé por última vez con los Perezosos, el día de nuestra graduación. Marchamos las veinticuatro compañías, la nuestra en el lugar veintitrés, dada nuestra posición en el alfabeto. Terminada la ceremonia, mi futuro suegro me dijo:

—La penúltima compañía se distinguió de las demás. Las otras marchaban; ella parecía flotar. ¿Qué compañía era ésa?

—La mía —respondí—. La que sepultó al general MacArthur.

Mi compañía había alcanzado la trascendencia.

Scrum en la hora de la revuelta

Cuando se habla de grandes equipos, suele aludirse a ese propósito trascendente. Pero aunque éste sea un elemento decisivo, no es la única pata del tripié. Igualmente crucial, aunque menos célebre, es la libertad de hacer tu trabajo como lo crees mejor, tener autonomía. En todos los grandes equipos, toca a los miembros decidir cómo cumplir las metas fijadas por quienes dirigen la organización.

La plaza de Tahrir es sinónimo ya de la revolución de Egipto y de la lucha aún en marcha en ese país, pero antes de enero de 2011 era una sucia glorieta más, muy transitada, en el centro de El Cairo. Al norte se levanta el Museo Egipcio, rojo y rosado, y al sur las altas paredes de la American University y el emblemático edificio gubernamental de Muqawama. Las oficinas del

Partido Democrático Nacional, del dictador Hosni Mubarak, estaban al oeste, igual que las de la Liga Árabe. Curiosamente, en el costado oriental de la plaza estaba nada menos que un Kentucky Fried Chicken, el cual se convirtió pronto en telón de fondo de manifestantes que arrojaban piedras a la policía.

A fines de enero de 2011, un pequeño grupo de manifestantes decidió marchar en esa transitada rotonda para protestar por el brutal asesinato de Khaled Said a manos de la policía. Lo que habría podido ser otra limitada protesta contra un régimen represivo prendió fuego, encendió la imaginación egipcia y acabó atrayendo a la plaza a millones de personas. El mes siguiente ocurrió lo inconcebible. Mediante el solo hecho de congregar a la gente y decir no, uno de los regímenes dictatoriales más antiguos y poderosos de Medio Oriente se vino abajo. La gente se reunía todos los días, noche a noche, llenando la plaza y erigiendo un país alterno en el que el dictador no mandaba y donde los individuos podían decir lo que pensaban, cambiando así su mundo.

Para los periodistas, esto constituyó todo un acontecimiento histórico. Entre los reporteros que invadieron El Cairo estaban los corresponsales de National Public Radio (NPR), una de las organizaciones periodísticas más importantes de Estados Unidos. Tomados por sorpresa, en un principio productores y reporteros de NPR incumplían fechas límite, perdían oportunidades noticiosas y satisfacían a duras penas las demandas de los ejecutivos en Washington.

J. J. Sutherland, mi hijo, fue enviado a Egipto a remediar esa situación. Experimentado productor y corresponsal de guerra, se le asignó a El Cairo para hacer funcionar debidamente la cobertura informativa. El caso era demasiado relevante como para no estar al aire todos los días, en cada programa, a toda hora. J. J. Sutherland fue a dar a un país donde los aeropuertos habían sido cerrados, del que los extranjeros intentaban huir desesperadamente y en el que la red de telefonía celular y la internet no operaban más. Sería el productor ejecutivo en el lugar de los hechos, pero, a la manera de los instructores de West Point, en NPR un productor es un facilitador y organizador, un animador o asistente más que un jefe o líder habitual. El trabajo de Sutherland consistiría en ayudar a su equipo a trabajar lo mejor posible. No en decirle qué hacer, sino en darle lo que necesitaba. La dirección había ordenado relatar los hechos y estar al aire varias veces al día y el equipo en el terreno tenía que idear cómo vencer ese reto, decidiendo qué hechos referir y cómo hacerlo usando la radio como medio.

Curiosamente, justo gracias a la dificultad de comunicarse con los ejecutivos en Washington, el equipo tuvo un éxito enorme. Sus miembros estaban prácticamente solos. Impedida la supervisión constante de Washington y dada la rapidez con que se sucedían los acontecimientos, tuvieron que organizar por sí mismos su trabajo. Uno de los conceptos clave de Scrum es que los equipos deben decidir cómo trabajar. Es responsabilidad de la dirección fijar las metas estratégicas, pero toca al equipo decidir cómo alcanzarlas. En cuanto a El Cairo, era imposible saber de lejos qué pasaba. Casi todos los días, algunos de los sucesos cubiertos por el equipo de NPR para el día siguiente se volvían obsoletos de inmediato, en vista del rápido desenvolvimiento de los hechos. Un choque en la plaza, un discurso, una renuncia o una batalla inesperados malograban el trabajo de todo el equipo. Éste tenía que hacer entonces hasta lo imposible para transmitir información oportuna.

Lo logró gracias a Scrum. Sus plazos de entrega lo obligaban a presentar información cada doce horas, en *Morning Edition* y *All Things Considered*. En cada ciclo, Sutherland hablaba con los miembros de su equipo y les hacía tres sencillas preguntas: «¿Qué han hecho desde la última vez que nos vimos? ¿Qué harán antes de nuestra próxima reunión? ¿Qué les estorba?». Planteando estas preguntas, uno de los rituales regulares de Scrum, forzaba a los corresponsales a comunicarse entre sí. Y su principal tarea como Scrum Master *de facto* era asegurarse de que lo que estorbaba a su equipo hubiera desaparecido para la siguiente reunión. El impedimento podía ser cualquier cosa, desde tratar con la burocracia egipcia hasta conseguir un cuarto de hotel seguro, encontrar choferes y traductores o sortear la custodia de la temida policía secreta de Egipto, la Mukhabarat.

¿Qué resultó de todo esto? Lo que había empezado en caos, conflictos personales e imposibilidad de transmitir rápidamente las noticias se convirtió en una máquina bien aceitada que la alta dirección ni siquiera tenía que conducir. Este equipo se dirigía solo. En las semanas siguientes, la brigada de NPR en El Cairo produjo más información de la concebible y de mayor calidad que la de la competencia, lo que le merecería después varios premios. Esta proeza habría sido imposible si el equipo no se hubiera imbuido de un propósito (documentar uno de los acontecimientos más importantes en su carrera) y si no hubiera gozado de autonomía (la oportunidad de decidir por sí mismo cómo producir las muchas hebras de la cobertura general).

Hoy se emplea Scrum en todas las tareas de NPR, desde diseño web y periodismo de investigación hasta creación de nuevos programas. También

usan Scrum equipos de *Chicago Tribune, New York Times, Washington Post* y *ProPublica*. En presencia de fechas límite muy estrictas, la rapidez es sencillamente esencial.

Un equipo que lleve a cabo el trabajo

La tercera pata del tripié de los grandes equipos es disponer de todas las habilidades necesarias para llevar a cabo el trabajo. En una estructura organizada al modo tradicional se tendría al equipo de planificadores, luego al de realizadores, después al de control de calidad, más tarde al de producción y por último al de embarque. Cada uno de estos equipos debe terminar su parte para que el proyecto pueda avanzar a la etapa siguiente. Ninguno puede hacerse cargo del producto por sí solo.

El ejemplo clásico de esto es el proceso de «fase-puerta» de la NASA, con el cual este organismo guió el programa del transbordador espacial y otros proyectos de los años sesenta, setenta y ochenta. Aunque ya ha cambiado mucho, he aquí cómo operaba. El primer paso era una «fase» de descubrimiento en la que se decidía qué hacer (fabricar un cohete que viajara a la luna, por ejemplo). Un grupo de estrategas se reunían en una sala para fantasear a ese respecto. Luego estaba una «puerta», en la que un gerente o grupo de gerentes autorizaba el proyecto en caso de juzgarlo digno de ejecutar. A continuación venía una fase de alcance, en la que toda la «gente de requerimientos» decidía qué haría el objeto proyectado. Después se alzaba otra puerta y otra serie de reuniones, tras de lo cual los ya para entonces colosales documentos pasaban a una nueva fase, de elaboración del plan de negocios y el del proyecto. Más tarde, todos estos planes pasaban por otra serie de reuniones y aprobaciones, luego de lo cual avanzaban a la fase de desarrollo, en la que se procedía a fabricar el objeto. Tras muchas otras reuniones y documentos, el producto iba a dar a manos de un grupo totalmente distinto, a cargo de la siguiente fase, de prueba. Aunque para entonces ningún miembro de este último grupo había visto jamás el producto, su labor era probarlo, certificarlo y hacerlo atravesar una puerta más, caracterizada por reuniones interminables y un nuevo altero de documentos que nadie leería nunca. Entonces, y sólo entonces, el producto llegaba a un *sexto* grupo de personas, responsable de su lanzamiento. La mera descripción de este proceso es de suyo agotadora. Así era como la NASA hacía las cosas.

A principios de los ochenta, ejecutivos de Fuji-Xerox viajaron a Estados Unidos para estudiar cómo operaba esa famosa agencia espacial. Cuando esos ejecutivos pusieron en marcha en Japón los mismos procedimientos, vieron bajar de inmediato la calidad, aumentar el índice de fallas y desplomarse su capacidad para cumplir, de modo que abandonaron al instante ese método a fin de evitar un error catastrófico. La Rogers Commission que examinó el desastre de 1986 del Challenger coincidiría más tarde con ellos. Como escribió célebremente el físico Richard Feynman en el apéndice F del informe de aquella comisión: «Parecería que para cualquier propósito de consumo, interno o externo, la dirección de la NASA exagera la confiabilidad de sus productos, hasta el punto mismo de la fantasía»[12].

El hecho es que al analizar los mejores equipos —como los de Toyota y 3M cuando Takeuchi y Nonaka escribieron su artículo, o como los de Google, Salesforce.com y Amazon hoy en día—, en ellos no está presente tal separación de roles. Cada equipo dispone del personal indispensable para hacer todo de cabo a rabo.

Nicola Dourambeis está a cargo de las prácticas Agile en Salesforce.com. Ella es responsable de doscientos equipos con Scrum en una empresa que suele aparecer en listas como las de «Los cien mejores centros de trabajo» de *Fortune* y «Las compañías más innovadoras del mundo» de *Forbes*. Dourambeis asegura que Scrum es su «fórmula secreta». «En un principio», cuenta, «hacíamos un lanzamiento importante tres o cuatro veces al año. Cuando crecimos, gestionando proyectos al habitual estilo en cascada, esa tasa se redujo a uno, en 2005-2006. Esto tenía que cambiar, así que adoptamos Scrum. Desde entonces hacemos tres lanzamientos al año. No muchas compañías emprendedoras de grandes dimensiones pueden hacer esto».

Lo que Dourambeis busca en un equipo es diversidad: de habilidades, manera de pensar y experiencia. Desea equipos desinteresados y autónomos, pero también interfuncionales. Equipos que puedan encargarse de un proyecto en su totalidad.

Una de sus pruebas de si un equipo va por buen camino consiste en preguntar, por ejemplo, a un ingeniero de redes: «¿En qué equipo estás?». Si este último responde mencionando el producto en el que trabaja (automatización o integración, por decir algo), no su especialidad (ingeniería de redes), Dourambeis mueve la cabeza aprobatoriamente. Cuando un experto se identifica con su especialidad antes que con el producto en el que colabora, ella sabe que aún tiene mucho por hacer.

Scrum en la guerra

Uno de los ejemplos más elocuentes de los equipos interfuncionales proviene del ejército. Las fuerzas especiales de Estados Unidos operan de ese modo. Un «equipo A» representativo de esas fuerzas consta de doce individuos: un líder (oficial de alto rango), un suboficial, un sargento (a cargo de la operación diaria), un sargento de inteligencia y dos sargentos de cada una de estas áreas: armas, demolición, médica y de comunicaciones. Cada equipo cuenta así con todas las capacidades necesarias para cumplir de principio a fin su misión, además de lo cual intercapacita a sus miembros en cada juego de habilidades. Quiere estar seguro, por ejemplo, de que si ambos médicos son abatidos, un especialista en comunicaciones podrá poner un parche a uno de los de armas. Otra cosa que distingue a las fuerzas especiales es que, a diferencia de gran parte del ejército «regular», no separan la recolección de inteligencia de la planeación de operaciones. Las cosas no se trasladan de un equipo a otro, lo cual puede inducir errores. Estas fuerzas no quieren desastres como el del Challenger. Así, existe una comunicación constante entre quienes recolectan inteligencia, planean qué hacer con ella y ejecutan las acciones.

Durante la guerra de Iraq, los equipos de fuerzas especiales demostraron ser muy aptos para liquidar enemigos. Podían localizar un blanco insurgente y eliminarlo esa misma noche. Entre 2003 y 2007 realizaron satisfactoriamente miles de misiones destinadas a desbaratar la insurgencia iraquí, en especial Al Qaeda en Iraq. Táctica y operativamente, tuvieron éxito en la mayoría de sus encomiendas. Dichos equipos interfuncionales, sumamente preparados, se contaron entre las fuerzas más letales que el mundo haya visto jamás. Pero pese a su habilidad y talento, su efecto estratégico era casi nulo. Durante los cuatro primeros años de esa guerra, los ataques contra fuerzas estadunidenses y civiles iraquíes aumentaron casi a diario. En los días más oscuros del conflicto hubo más de cien ataques contra tropas estadunidenses y ni siquiera la letalidad de las fuerzas especiales fue capaz de contener la escalada de violencia. A fines de 2006 y principios de 2007, la mayor parte de los expertos consideraban que Iraq era un caso perdido. Cada nueva muerte estadunidense se veía ya como un sacrificio en vano.

Fue así como, en 2007, el general David Petraeus encabezó la que se conocería como la *Oleada*, la cual implicó desplazar a Iraq decenas de miles de soldados más a fin de que vivieran entre la población. Esta nueva estrategia fue de gran relevancia, entre otras cosas porque logró convencer al pueblo iraquí de que los estadunidenses estaban de su parte, combatiendo a

los insurgentes que hacían explotar bombas en sus barrios y que ejercían la limpieza étnica. Otra razón de su trascendencia fue que el ejército estadunidense consiguió sobornar a decenas de miles de exinsurgentes para que se pasaran de su lado, programa que se denominó «Hijos de Iraq». Pero la estrategia tuvo un tercer elemento, consistente en algo que, según el periodista Bob Woodward, resultaría tan revolucionario como la invención del tanque o el avión.

Esa arma no era un nuevo artefacto o nave no tripulada, sino lo que el general Stanley McChrystal, entonces comandante del Joint Special Operations Command, llamó «guerra cooperativa». Implicaba equipos interfuncionales con miembros de todas las áreas del gobierno estadunidense a fin de identificar y desintegrar las redes de Al Qaeda. Como explicó el *Washington Post* el 6 de septiembre de 2008:

> La CIA [Central Intelligence Agency] aporta analistas de inteligencia y aviones espía con sensores y cámaras capaces de rastrear blancos, vehículos o equipo por hasta catorce horas. Expertos forenses del FBI [Federal Bureau of Investigation] diseccionan datos, desde información de teléfonos celulares hasta la «basura de bolsillo» que se extrae de extremistas. Funcionarios del Tesoro rastrean el flujo de fondos entre extremistas y desde gobiernos. Empleados de la National Security Agency interceptan conversaciones y datos de computadoras, y miembros de la National Geospatial-Intelligence Agency emplean equipo de alta tecnología para identificar los lugares en que presuntos extremistas usan teléfonos o computadoras[13].

Se crearon, entonces, equipos interfuncionales con todas las aptitudes necesarias para llevar la tarea hasta su fin. En vez de disponer equipos por separado cuyos expertos no acostumbran compartir información, todos los participantes trabajaban juntos, en la misma sala, compartiendo inteligencia y haciendo planes para ubicar y liquidar a agentes de Al Qaeda.

Antes, una organización de inteligencia designaba el blanco y luego cedía las operaciones a un equipo de fuerzas especiales, el que transfería a su vez la inteligencia reunida a otro equipo de análisis. Quienes practicaban este modelo de transferencia descubrieron lo que Fuji-Xerox había detectado décadas antes al tratar de implementar el sistema fase-puerta de la NASA (y una de las razones principales del desarrollo de Scrum). Cada vez que se

hacen transferencias entre equipos hay riesgo de desastres. Como se dijo en el artículo «Employing ISR: SOF Best Practices» (Empleo del ISR: Buenas prácticas de las Fuerzas Especiales), publicado en el *Joint Force Quarterly*:

> Los equipos interagencias permitieron eliminar las costuras organizacionales entre los diferentes actores de la coalición en Iraq, poniendo un «ojo avizor» en blancos de alta prioridad. […] Ceder responsabilidades entre unidades y organizaciones representaba un «parpadeo organizativo» en el que se perdía impulso y el blanco podía escapar[14].

Compartir información y recursos no es fácil. He visto a gerentes casi engarrotarse cuando sus recursos se asignan a un equipo fuera de su control directo. Renunciar a la microgestión y el control cotidiano es difícil, pero hacerlo en el hermético mundo de la inteligencia y las operaciones especiales es más difícil todavía; tanto que, pese a su eficacia, los equipos de Iraq se disolvieron en cuanto la Oleada fue juzgada un éxito. Christopher Lamb y Evan Munsing escribieron un artículo espléndido, «Secret Weapon: High-value Target Teams as an Organizational Innovation» (Arma secreta: los equipos con blancos de alto valor como innovación organizativa), en el que expusieron esto.

> En cuanto se conjuró el cuasifracaso en Iraq, el apoyo burocrático a los equipos interagencias empezó a declinar. En 2008, otros departamentos y organismos, en particular una agencia de inteligencia no identificada, empezaron a retirar personal y cooperación, en la creencia de que la compartición de información y la colaboración habían llegado demasiado lejos[15].

El arma más poderosa del arsenal estadunidense, aquello mismo que Bob Woodward estimó tan importante como la invención del tanque o el avión, fue dejado de lado por intereses estrictamente departamentales y los escrúpulos de mandos medios preocupados por su carrera. He visto suceder esto repetidamente en una gran institución financiera de Boston, cuyos directivos me llaman alarmados cuando tienen un proyecto de misión crítica en dificultades. Entonces me hacen impartir capacitación en Scrum a docenas de empleados e implantar equipos capaces de resolver la emergencia. Destinan individuos de toda la organización a equipos interfuncionales que

ataquen el problema. Y que lo resuelvan. Pero una vez pasada la crisis, devuelven a la gente a sus respectivos silos y feudos administrativos. El grado en que un equipo fabuloso comparte y ejerce transparencia amenaza estructuras fundadas en la reserva y la confusión. En general, a los gerentes no les gusta que sus colegas, equipos ni otras personas dentro de la estructura de poder sepan lo que ellos hacen o harán, ni cuán rápido. Creen que mantener esto en secreto es crucial para su poder. En vez de hacerlo con los intereses del bien mayor, armonizan con sus motivaciones propias, las que a menudo se reducen a codicia y ambición. Esta manera de pensar causó el gran fracaso administrativo que derivó en el colapso económico más reciente. Muchas compañías basaron exclusivamente sus actos en beneficios individuales de corto plazo; no pensaron en lo que beneficiaría a todos o en limitar el daño para la economía global.

El tamaño sí importa, aunque no como suele pensarse

Pero no porque la interfuncionalidad pueda dar excelentes resultados tú has de sentirte Noé e incluir en un equipo dos ejemplares de cada especie. La dinámica de grupos sólo opera correctamente en equipos pequeños. La cifra clásica son siete integrantes, dos de más o de menos, aunque he visto funcionar de maravilla a grupos de apenas tres individuos. Lo asombroso es que los datos demuestran que un equipo de más de nueve se vuelve torpe. En efecto: más recursos hacen más lento a un equipo.

En el desarrollo de *software* rige un principio llamado *ley de Brooks*, que Fred Brooks enunció en 1975 en su libro seminal *The Mythical Man-Month* (El mítico hombre-mes). En palabras sencillas, esta ley establece que «agregar personal a un proyecto atrasado lo atrasará aún más»[16], lo que se ha confirmado en un estudio tras otro. Lawrence Putnam, figura legendaria en este campo, dedicó lo mejor de su vida laboral a estudiar cuánto tardan las cosas en hacerse y por qué. Su labor corroboró que proyectos con veinte personas o más implicaban más esfuerzo que aquellos con cinco o menos. No sólo algo más de esfuerzo, sino mucho. Un equipo grande consumía cinco veces el número de horas de uno chico. Él vio reiterarse esto una y otra vez; a mediados de los años noventa decidió hacer un estudio amplio para determinar cuál era el tamaño de equipo correcto. Examinó cuatrocientos

noventa y un proyectos medianos de centenas de compañías. Todos ellos eran proyectos que requerían la creación de nuevos productos o funciones, no la reformulación de versiones previas. Putnam dividió esos proyectos por tamaño de equipo y advirtió algo de inmediato: en cuanto los equipos constaban de más de ocho miembros demoraban mucho más en realizar su tarea. Los grupos compuestos por entre tres y siete personas requerían de veinticinco por ciento del esfuerzo de aquellos de entre nueve y veinte para hacer el mismo trabajo. Este resultado se repetía en cientos de proyectos. Que grupos muy grandes hacen menos parece ser una regla fija de la naturaleza humana.

Pero ¿por qué? Para responder a esta pregunta hemos de considerar las limitaciones del cerebro humano. Tal vez ya hayas oído hablar del estudio clásico de George Miller, quien en 1956 demostró que el número máximo de unidades que la persona promedio puede retener en su memoria de corto plazo es de siete. Se supone que ésta es la causa de que los números telefónicos consten de siete dígitos. El problema del trabajo de Miller es que fue refutado por investigaciones posteriores.

En 2001, Nelson Cowan, de la University of Missouri, se preguntó si esa regla mágica de siete era cierta e hizo un extenso sondeo de las nuevas investigaciones sobre el tema. De ello resultó que el número de unidades que uno puede retener en su memoria de corto plazo no es de siete, sino de cuatro[17]. La gente cree ser capaz de memorizar un número mayor usando un recurso nemotécnico o mucha concentración. Pero las investigaciones insisten en que sólo podemos recordar cuatro fragmentos de datos. El ejemplo clásico es dar a alguien la siguiente serie de doce letras: *fbicbsibmirs*. La gente tiende a recordar cuatro de ellas, *a menos que* advierta que pueden dividirse en estas conocidas siglas: FBI, CBS, IBM, IRS. Si eres capaz de asociar recuerdos en tu memoria de corto plazo con otros en la de largo plazo podrás aumentar la cabida de la primera. Pero la parte de la mente que se concentra —la parte consciente— sólo puede contener cuatro unidades por vez.

Esto quiere decir que hay un límite preestablecido a lo que nuestro cerebro puede contener en un momento dado, lo cual nos regresa a Brooks. Cuando quiso saber por qué agregar personas a un proyecto lo demoraba más encontró dos razones. La primera es el tiempo que implica acelerar a una persona. Como es de suponer, acelerar a alguien retrasa a los demás. La segunda razón no tiene que ver con nuestra manera de pensar, sino, literalmente, con la capacidad de pensar de nuestro cerebro. El número de canales de comunicación crece drásticamente cuando aumenta el número de

personas con que tratamos, y nuestro cerebro sencillamente no puede manejar eso.

Para calcular el efecto del tamaño de un grupo, multiplica el número de sus miembros por ese mismo número menos uno y divide entre dos el resultado. Canales de comunicación = $n(n\text{-}1)/2$. Así, cinco personas equivalen a diez canales, seis a quince, siete a veintiuno, ocho a veintiocho, nueve a treinta y seis y diez a cuarenta y cinco. Nuestro cerebro no puede seguir a tantas personas a la vez. No sabemos qué hace cada una de ellas y cuando tratamos de averiguarlo nos atrasamos.

Igual que los de un equipo de fuerzas especiales, los miembros de un equipo con Scrum deben saber qué hacen los demás. Todo lo que se lleva a cabo, los retos que se enfrentan, el progreso alcanzado deben ser transparentes para cada uno. Y si el equipo crece demasiado, la posibilidad de que en todo momento cada quien se comunique claramente con los demás se ve mermada. Hay demasiadas contracorrientes. Con frecuencia, el equipo se divide social y funcionalmente en subequipos que empiezan a trabajar en propósitos cruzados. La interfuncionalidad se pierde. Reuniones que antes tardaban minutos ahora llevan horas.

No lo hagas. Mantén equipos reducidos.

El *Scrum Master*

A mi primer equipo con Scrum solía presentarle un video del equipo de *rugby* All Blacks preparándose para un partido. Los All Blacks, escuadra legendaria de la diminuta Nueva Zelanda, son un equipo trascendente. Antes de cada encuentro celebran la ceremonia de la *haka*, propia de los guerreros maoríes. La *haka* es una danza guerrera que llena de energía a quienes están por entrar en batalla. Al presenciarla, casi puedes ver la energía salir de cada jugador y fusionarse en un gran conjunto. Entre zapateos y aplausos sincronizados acompañados por cánticos —movimientos ritualizados del acto de cortar el cuello al enemigo—, se ve a hombres comunes y corrientes transformarse en algo grande, mayor. Invocan a un espíritu bélico que no admite derrotas ni abatimientos.

Aunque fue necesario proyectar varias veces ese video, mi equipo de programadores de computadoras ligeramente fuera de forma empezó a hablar por fin de qué podía hacer para asemejarse a aquella escuadra. Dio así con cuatro aspectos dignos de emulación. El primero fue intensa concentración en

la meta, producida y reforzada por el cántico maorí. El segundo, la colaboración radical: brazos y cuerpos unidos en pos del mismo objetivo. El tercero, ansia de aplastar: todo lo que se interpusiera en su camino debía ser eliminado. El cuarto, entusiasmo universal cuando un jugador se abría paso con la pelota. No importaba quién lo hiciera, que sucediera era motivo de celebración.

Establecimos entonces nuestro marco de *sprints*, reuniones diarias, revisiones y retrospectivas, y noté que necesitábamos a alguien cuya función fuera confirmar la eficacia del proceso. No un gerente, sino un líder de servicio, una combinación de capitán y entrenador. Viendo a diario a los All Blacks, preguntaba al equipo cómo debíamos llamar a esa persona. Optaron por *Scrum Master*. Este sujeto facilitaría las reuniones, comprobaría que hubiera transparencia y, sobre todo, ayudaría al equipo a saber qué le estorbaba. La parte clave de esto era percatarse de que, a menudo, los impedimentos no son que la máquina no funcione o que Jim el de contabilidad sea un idiota, sino el proceso mismo. Era deber del *Scrum Master* guiar al equipo a la mejora continua; preguntar con regularidad: «¿Cómo podemos hacer mejor lo que llevamos a cabo?».

Idealmente, al final de cada repetición, de cada *sprint*, el equipo se examinaba con detalle —sus interacciones, prácticas y procesos— y formulaba dos preguntas: «¿Qué cambios podemos hacer en la forma como trabajamos?» y «¿Cuál es nuestro principal escollo?». Si estas preguntas se responden con franqueza, un equipo puede avanzar más rápido de lo que cree.

No rechaces al jugador sino al juego

Es común que la baja moral, cohesión y productividad de un equipo se deban a una incomprensión fundamental de cómo trabajan los seres humanos. ¿Cuántas veces no te has juntado con un colega para quejarte de un tercero que «no pone de su parte», «siempre nos atrasa» o «toma malas decisiones», o has formado parte de un equipo que, al enfrentar un problema, lo primero que hace es ver a quién culpar?

Podría apostar que todos y cada uno de nosotros hemos tenido una conversación de ese tipo. También que, en un momento u otro, hemos sido la persona a la que se culpa del problema. Pero, asimismo, que cuando culpas a alguien buscas sus defectos personales, mientras que si se te culpa a ti aduces los factores situacionales que originaron el problema y por qué actuaste como

lo hiciste. ¿Y sabes qué? En cuanto a ti, tienes toda la razón. Pero con otros cometes uno de los errores humanos más comunes —y destructivos— al juzgar sus acciones. Incluso esto tiene un nombre: «Error fundamental de atribución».

Interesantes estudios sobre el tema se incluyen en el libro *Induction: Processes of Inference, Learning, and Discovery* (Inducción: procesos de inferencia, aprendizaje y descubrimiento), de John H. Holland y otros. Uno de los artículos citados en este libro se publicó a principios de la década de 1970, así que no es nada reciente. Este añejo asunto se ha reiterado una y otra vez. Todo se reduce a qué mueve a los seres humanos. Como fuere, ese grupo de investigadores reunió a gran número de estudiantes universitarios y les hizo un par de preguntas muy sencillas: «¿Por qué elegiste tu carrera?» y «¿Por qué elegiste a tu actual pareja?». Luego pidió a los alumnos responder a esos mismos cuestionamientos en relación con su mejor amigo, de lo que surgieron diferencias importantes. Cuando los estudiantes se referían a sí mismos, no lo hacían en términos personales, sino de lo que se les había preguntado. Decían cosas como «El campo de la química está muy bien remunerado», respecto a su carrera, o «Es muy cariñosa», sobre su novia. Pero al referirse a sus amigos, hablaban de las aptitudes y necesidades de éstos, como en «Siempre ha sido bueno en matemáticas» o «Es un poco dependiente, así que necesita una mujer enérgica»[18].

Esta manera de percibir el mundo resulta graciosa cuando la ves en otros; es demasiado obvio que emiten juicios equivocados. Pero antes de que te rías, debes reconocer que tú también haces eso todo el tiempo. Todos lo hacemos. Todos nos percibimos como reaccionando a una situación y a los demás como movidos por su carácter. Un curioso efecto secundario de esto es que cuando se nos pide señalar nuestros rasgos de personalidad y los de nuestros amigos siempre nos representamos como más aburridos. Decimos tener menos rasgos de carácter que nuestros amigos.

Los autores de *Induction* trazan un paralelo revelador entre nuestras equivocadas ideas de las motivaciones sociales y la manera en que personas no científicas o con una comprensión meramente intuitiva de la física ven el mundo físico.

Un físico intuitivo podría explicar por qué cae una piedra diciendo que posee la cualidad intrínseca de la gravedad, en vez de decir que la gravedad forma parte de un sistema de fuerzas que actúan sobre la piedra. De igual modo, cuando hablamos de otros, aludimos a sus propiedades inherentes en vez de verlas en relación con el entorno. Pero, de hecho, lo que motiva

nuestra conducta son nuestras interacciones con el entorno. Más que cualquier cualidad intrínseca, *el sistema que nos rodea* es lo que explica la casi totalidad de nuestro comportamiento. Scrum fue diseñado para cambiar este sistema. En lugar de buscar culpas y defectos, premia la conducta positiva, concentrando a la gente en el trabajo en común y en llevar a término sus proyectos.

Quizá la demostración más famosa de esta reacción humana a los sistemas sea el experimento de Milgram sobre la obediencia a figuras de autoridad, efectuado en Yale University a principios de los años sesenta. Este experimento era muy simple y, para ojos modernos, algo cruel. Pero también fue devastador e impactante y se enseña en todos los cursos de primer año de psicología. El doctor Stanley Milgram, profesor de Yale, tenía entonces una pregunta muy pertinente. Tres meses antes de que se pusiera en marcha el primer experimento, Adolf Eichmann, el arquitecto del Holocausto, fue sometido a juicio.

Una de las interrogantes más persistentes sobre el Holocausto era cómo había sido posible que tantos millones de personas fueran cómplices de tal horror. ¿Los alemanes eran fundamentalmente reprensibles en lo moral? ¿Había algo intrínsecamente malo en su composición cultural? ¿O de veras sólo habían seguido órdenes? Es fácil esgrimir crímenes contra la humanidad y culpar a individuos de sus actos. Esto es lo correcto, ¿no? Pero la pregunta que Milgram quería contestar era: ¿los estadunidenses comunes eran diferentes de los alemanes? ¿Habrían reaccionado de otro modo en una situación igual? Y la incómoda respuesta es no: los estadunidenses no se *habrían* comportado de otra manera. De hecho, dada la forma en que muchos países y culturas han reproducido este experimento, *nadie* lo habría hecho. En la situación indicada, todos podríamos ser nazis.

El experimento de Milgram consistía en lo siguiente: el sujeto, un individuo del común, era instruido por alguien con bata blanca (lo que le daba una apariencia de autoridad científica) para que administrara choques eléctricos cada vez más intensos a un tercero, un actor, situado en otro cuarto. El sujeto podía oír pero no ver al actor. Mientras los choques aumentaban, este último empezaba a gritar y suplicar. Finalmente el actor (quien, en algunas versiones del experimento, decía al sujeto que padecía una enfermedad del corazón) comenzaba a golpear la pared, pidiendo a gritos parar el experimento hasta que, por último, guardaba silencio.

Algunas personas se detenían en los 135 volts, cuando el actor gritaba, y preguntaban sobre el propósito del experimento. Casi todas continuaban una

vez que se les aseguraba que no se les haría responsables. Algunas reían nerviosamente al oír los gritos angustiosos salidos del cuarto contiguo. Cuando el sujeto quería hacer alto, el «científico» le decía: «*Siga*, por favor». Y si el sujeto no lo hacía, aquél agregaba: «El experimento requiere que usted *continúe*». En ausencia aún de una reacción, el científico añadía: «Es esencial que *continúe*». La mayoría de los sujetos parecían estar bajo enorme estrés y sudaban profusamente. Exhibían pulso y temperatura elevados mientras el instinto de «pelear o huir» se asentaba en ellos. Entonces, si seguían sin presionar el botón, el científico hacía un último intento: «No tiene otra opción. *Debe* proseguir».

Casi todos lo hacían, dando el último choque a alguien que, tras dejar de gritar, callaba. Milgram resumió así las implicaciones de esto en su artículo de 1974, «The Perils of Obedience»:

> Personas comunes que se limitan a cumplir su trabajo y carecen de toda hostilidad particular pueden volverse agentes de un terrible proceso destructivo. Además, aun si los efectos destructivos de su labor resultan patentemente claros y se les pide ejecutar acciones incompatibles con normas morales fundamentales, un número relativamente reducido de ellas tiene los recursos indispensables para resistirse a la autoridad[19].

Cuando este experimento se expone en el aula, suele decirse a los estudiantes que la culpa es del sistema en el que esas personas operaban, no de ellas mismas. Pero ésta es una lección difícil de interiorizar, porque si aceptas su verdad, ¿qué dice eso de ti?

Que *todos* somos criaturas del sistema en que estamos insertos. Lo que Scrum hace es aceptar esta realidad y, en vez de ver de quién es la culpa, examinar el sistema que produjo la falla y repararlo.

Otro experimento que ilumina un fenómeno similar se realizó en un seminario teológico a principios de los años setenta. Se pensaría que los seminaristas son las personas más compasivas del planeta, ¿no es así? A los sujetos de este experimento se les dijo que tenían que dar un sermón al otro lado del campus. A algunos se les indicó que debían apurarse, porque ya se les esperaba e iban retrasados, pero a los demás no. Al cruzar los jardines del plantel todos pasaron junto a alguien que pedía ayuda en una puerta. ¿Cuántos sujetos a los que se les dijo que tenían que apurarse se detuvieron a ayudar? Diez por ciento. De *seminaristas*.

A la gente le gusta culpar a individuos, no a sistemas. Se siente mejor. El error fundamental de atribución apela a nuestra noción de justicia. Si podemos culpar a otro, nos libramos del riesgo de actuar como él, de tener la misma probabilidad que cualquiera de apretar ese botón, dadas las circunstancias adecuadas.

¿Cómo se manifiesta en los negocios este error de culpar a individuos más que a sistemas? Tengo dos buenos ejemplos; el primero, la planta automotriz New United Motor Manufacturing, Inc. (NUMMI), en Fremont, California. Ésta fue una empresa conjunta de General Motors y Toyota. GM cerró la planta en 1982. Su dirección juzgó que la fuerza laboral era la peor de Estados Unidos: bebía en el trabajo, se ausentaba de sus deberes y saboteaba sutilmente los autos (metiendo, por ejemplo, un envase de Coca Cola en una puerta, donde vibraría e incomodaría al cliente). Toyota la reabrió en 1984. Y aunque GM la previno contra los obreros, le dijo que los gerentes eran de primera y debía recontratarlos. Toyota se negó, recontratando en cambio a la mayoría de los trabajadores originales, a algunos de los cuales envió incluso a Japón para que conocieran su sistema de producción. Muy pronto, la planta NUMMI producía autos con la misma precisión y tan pocos defectos como los elaborados en Japón. Mismas personas, diferente sistema. GM no alcanzó nunca tales niveles de calidad en ninguna de sus demás plantas en Estados Unidos. Abandonó la sociedad conjunta el mismo año en que quebró.

El segundo ejemplo que me viene a la mente es algo distinto. Me recuerda lo sistemático que es para muchas personas buscar culpables de un problema antes que la solución. Tiene que ver con la forma en que operan los capitalistas de riesgo con los que trabajo cuando deciden invertir en una empresa. La primera vez que me asocié con OpenView Venture Partners me sorprendió que, a diferencia de muchas otras sociedades de inversión de riesgo, a ésta no le importara cómo la empresa elegida había gastado el dinero *antes de* invertir en ella. El pasado no importa. OpenView decide si invertir o no con base en la situación presente de una compañía; todo lo demás es irrelevante. Quiere saber cómo va a gastar ésta el dinero que *ella* le dará. No importa cómo haya gastado el dinero de otros. Lo único relevante es el futuro, las *soluciones*.

Alcanzar la grandeza

Cuando un equipo comienza a acoplarse y sincronizarse puede parecer mágico. Lo sientes al entrar a la sala donde está. Lo ves cuando se despliega en el terreno. Parecería que flotara: se ha sobrepasado a sí mismo.

En fecha reciente visité a un amigo en su casa en Copenhague. Como cabe imaginar, siendo europeo, es fanático del futbol. No sé en qué torneo jugaba su equipo, pero verlo a él saltar y gritar frente al televisor fue todo un espectáculo. Ahí estaba un aficionado indignadísimo. Llegó entonces el momento del empate, de los últimos segundos, y el equipo de mi amigo se hizo del balón. Desde más allá de la media cancha, *y sin ver a sus compañeros*, un delantero disparó hacia una masa de jugadores frente a la portería. El problema es que no había nadie de su equipo ahí. Esto me decepcionó, pero de repente, como salido de la nada, apareció un jugador del equipo de mi amigo, justo en el momento y lugar indicados para anotar un gol de cabeza. Había corrido desde la media cancha hasta la masa de adversarios frente a la portería, donde aprovechó la oportunidad de cabecear. Fue una sorpresa absoluta. Pero el delantero que hizo el disparo había tenido fe en que su compañero estaría donde debía. Y éste la había tenido en que el balón sería puesto donde pudiera hacer algo con él. Es el tipo de sincronía que da gusto ver.

Y esto es lo que quiero ayudar a la gente a conseguir con Scrum. Es posible. No sólo las elites y los atletas y personas especiales pueden hacerlo. Todo se reduce a establecer el marco indicado con los incentivos correctos, y a dar a la gente la libertad, respeto y autoridad que necesita para hacer las cosas por sí sola. La grandeza no se puede imponer; tiene que venir de dentro. Pero vive en todos nosotros.

RESUMEN

Jala la palanca correcta. Cambia el desempeño de tu equipo. Este desempeño tiene *mucho* más influencia —en varios órdenes de magnitud— que el individual.

Trascendencia. Los grandes equipos tienen un propósito mayor que el individual; por ejemplo, sepultar al general MacArthur, ganar el campeonato de la NBA.

Autonomía. Concede a los equipos libertad para tomar decisiones sobre cómo actuar, para que se les respete como maestros en su oficio. La posibilidad de improvisar marcará toda la diferencia, sea

que la unidad cubra una revolución en Medio Oriente o haga una venta.

Interfuncional. El equipo debe tener todas las habilidades necesarias para terminar un proyecto, así su misión sea entregar *software* de Salesforce o capturar terroristas en Iraq.

Prefiere lo chico. Los equipos reducidos trabajan más rápido que los grandes. La regla son siete miembros, dos de más o de menos. Peca de cicatero.

Es absurdo culpar. No busques malas personas, sino malos sistemas: aquellos que incentivan la mala conducta y premian el bajo rendimiento.

Capítulo cuatro

Tiempo

El tiempo es la limitante suprema del esfuerzo humano y lo afecta todo, desde cuánto trabajamos hasta cuánto tardamos en hacer las cosas y cuán exitosos somos. El incesante flujo unidireccional del tiempo determina en lo fundamental nuestra manera de ver el mundo y a nosotros mismos. Como dijo célebremente el poeta británico del siglo XVII Andrew Marvell: «Si tuviéramos mundo suficiente, y tiempo», podríamos conseguirlo todo. Pero, claro, una sensación de mortalidad pende sobre cada uno de nuestros empeños. Sabemos que nuestro tiempo es limitado. ¿Desperdiciarlo no es entonces el mayor de los crímenes? Marvell otra vez:

> Aunque no podemos parar el sol
> es posible que corra a buen son[20].

¿Cómo lograrlo? Es fácil gritar desde un estrado «¡Aprovecha cada día!» para inspirar a una multitud, pero ¿cómo conseguirlo en verdad? Buena parte del trabajo consiste en decir a la gente que se siente, se apriete el cinturón e invierta largas horas. «No pienses en el mundo exterior», nos dicen implícitamente nuestros jefes. «No te preocupes por tus hijos, por ir a surfear y ni siquiera por tu cena; nada más trabaja, trabaja mucho y recibirás tu recompensa. Obtendrás ese ascenso. Harás esa venta. Terminarás ese proyecto».

Aunque no tengo nada contra los ascensos, ventas o proyectos, es un hecho que los seres humanos somos fatales para trabajar así. Somos pésimos para concentrarnos, pasamos en la oficina muchas más horas de las necesarias y somos muy malos para calcular cuánto tardaremos en hacer algo. Y hablo de todos por igual; así *somos* los seres humanos.

Cuando desarrollé Scrum no era mi intención crear un nuevo «proceso». Sólo quería reunir todas las investigaciones hechas durante décadas sobre cómo se trabaja mejor y emularlo. Quería incorporar buenas prácticas a mi plan y robar todas las buenas ideas que encontrara. Justo antes del primer Scrum en Easel, en 1993, trabajaba en una compañía muy cerca del Media Lab del MIT, laboratorio del que me robé una idea que se ha vuelto la esencia de Scrum: el *sprint*.

El *sprint*

A principios de los años noventa, el Media Lab producía toda suerte de cosas ingeniosas. La internet estaba entonces en ciernes y ese laboratorio hacía de todo, desde robots y la tinta electrónica que hace posibles los *e-readers* hasta nuevas formas de codificar el sonido. Fue una temporada muy intensa y yo solía contratar a estudiantes salidos de ese laboratorio porque estaban llenos de ideas, tenían una aptitud increíble para hacer cosas fabulosas y podían hacerlas rápido.

Esos jóvenes debían su celeridad a una política que el Media Lab aplicaba a todos sus proyectos. Cada tres semanas, todos los equipos debían mostrar a sus colegas lo que estaban haciendo. Era una demostración pública; cualquiera podía asistir. Y si el producto mostrado no era funcional ni fabuloso, los directores cancelaban el proyecto. Esto obligaba a los estudiantes a hacer rápido cosas ingeniosas y, sobre todo, les proporcionaba realimentación inmediata al respecto.

Piensa en tus propios proyectos. Apostaría que rara vez obtienes realimentación sobre ellos hasta su conclusión, lo que podría significar meses, incluso años. Podrías seguir meses enteros una dirección *totalmente* equivocada sin sospecharlo siquiera. Esto es desperdiciar grandes tramos de tu vida. En los negocios, tal cosa podría representar la diferencia entre éxito y fracaso. Veo ocurrir esto todo el tiempo: una compañía dedica años a un proyecto que en sus inicios parecía una buena idea, pero sin considerar el riesgo de que, una vez cruzada la línea de meta, el mercado haya cambiado por completo. Cuanto más pronto des cosas concretas a tus clientes, más rápido podrán decirte ellos si estás haciendo lo que realmente necesitan.

Así, cuando puse en marcha el primer Scrum en Easel y dije al director general que no le mostraría una larga y detallada gráfica de Gantt que ambos sabíamos que estaba equivocada, él replicó: «*¿Qué* me enseñarás entonces?».

Le respondí que cada mes le enseñaría una pieza funcional de *software*. No algo que operaría en segundo plano. No una pieza de arquitectura. Una pieza de *software* que un cliente pudiera usar en la práctica. Una función totalmente implementada.

«De acuerdo», me dijo. «Hazlo».

Mi equipo se embarcó entonces en lo que denominamos *sprints*. Los llamamos así porque este nombre evocaba intensidad. Trabajaríamos al máximo por un periodo corto, tras de lo cual haríamos una pausa para saber dónde estábamos.

Team WIKISPEED es un grupo fundado por alguien que responde al maravilloso nombre de Joe Justice. Este grupo hace autos. Autos que rinden más de ciento sesenta kilómetros por galón (3785 litros), tienen permiso para circular, reciben evaluaciones de cinco estrellas en riesgo de colisiones, alcanzan los doscientos veinticinco kilómetros por hora y pueden comprarse por menos de lo que costaría un Camry. WIKISPEED mejora constantemente sus vehículos; si quieres adquirir uno debes depositar veinticinco mil dólares en wikispeed.com para recibirlo en tres meses. Los miembros de ese equipo logran esto siguiendo Scrum. Como muchos de los mejores equipos actuales, ellos trabajan en *sprints* de una semana. Cada jueves examinan su gran cantidad de pendientes, desde hacer prototipos de un nuevo diseño de tablero de instrumentos hasta probar direccionales. Habiendo dispuesto esta lista por orden de prioridad, dicen: «Dada esa lista, ¿cuántas cosas podemos hacer esta semana?». Y por «hacer» entienden «terminar». Las nuevas características funcionan. El auto marcha. Cada semana. Cada *sprint*.

Al entrar un jueves cualquiera a la guarida de Team WIKISPEED, al norte de Seattle, puedes ver el glorioso caos organizado que es un taller de maquinaria. Hay cajas de herramientas, sierras, aparatos eléctricos, tornillos y llaves de tuerca. Una fresa de acanalar ocupa una esquina cerca de un chasís a medio hacer en la bahía tres. Un taladro y una dobladora de metal descansan a un lado, como muñecas ansiosas de que jueguen con ellas. El día en que nosotros visitamos el taller vimos arriba de ese chasís la foto del comprador del auto, Tim Myer. A Myer le gustan el alpinismo, las papas fritas y la sidra. No le agrada ignorar qué ocurre y no tener opciones. Puedes hallarlo en las montañas los fines de semana, y bailando en el Tractor cada tercer lunes por la noche.

Enfrente, en la bahía uno, está el primer auto producido por Team WIKISPEED, mismo que participó en un concurso de XPrize dotado con diez millones de dólares, de autos con eficiencia de combustible de ciento sesenta

kilómetros por galón. Team WIKISPEED obtuvo el décimo lugar, superando a más de cien competidores, desde grandes compañías automotrices hasta universidades. Invitado en consecuencia al 2011 Detroit Auto Show, ahí se le dio un lugar destacado, entre Chevy y Ford. Ese coche es ahora su banco de pruebas de nuevas ideas.

A un lado se levanta una pared de pizarrón blanco de tres y medio metros de altura que se extiende por todo el taller. En ella hay docenas de uno de los artilugios más comunes en Scrum: papeletas adhesivas de colores brillantes, cada una con una cosa por hacer: «taladrar tubo de dirección modular»… «preparar molde interior»… «forro de las salpicaderas», etcétera.

El pizarrón se divide en varias columnas: Pendientes… En proceso… Terminado. Al principio de cada *sprint*, los miembros del equipo de WIKISPEED ponen en la columna de Pendientes todas las papeletas con las actividades que creen poder hacer esa semana. En el transcurso de ella, un miembro asumirá una de esas tareas, pasando la papeleta a la columna En proceso y al acabar la pasará a Terminado. Todo el equipo puede ver qué está haciendo cada uno en todo momento.

Un detalle importante: nada pasa a Terminado si el cliente no lo puede usar. En otras palabras, tú debes poder *manejar* el auto. Y si lo haces y dices: «Las direccionales se traban», este problema se remediará en el siguiente *sprint.*

Los *sprints* son lo que se conoce como «cajas de tiempo». Su duración es fija. No deben ser de una semana uno y de tres otro. Tienes que ser congruente. Debes establecer un ritmo de trabajo que permita saber a la gente cuánto puede hacer en un periodo determinado. Esta cantidad de trabajo suele causar asombro.

Un elemento crucial de cada *sprint*, sin embargo, es que una vez que el equipo se compromete con lo que hará, el número de tareas se congela. Nadie fuera del equipo puede añadir una más. Más adelante explicaré por qué; por ahora baste saber que interferir y distraer al equipo lo retarda drásticamente.

Como ya dije, los *sprints* del primer Scrum eran de cuatro semanas. Casi al final del *sprint* inicial sentimos que no íbamos lo bastante rápido, que podíamos hacer más. Vimos entonces el video de los All Blacks ejecutando la *haka* y penetrando las líneas enemigas. «¿Por qué no somos como ellos?», nos preguntamos. «¿Por qué no tenemos ese mismo ánimo?». Nuestra meta no era ser sólo un buen equipo, sino el mejor. ¿Qué podíamos hacer para lograrlo? Una vez más, la respuesta resultó ser algo muy sencillo que robamos a otros: la reunión diaria.

Parada diaria

Afuera de una ciudad cuyo nombre no puedo decir, en una compañía que no puedo mencionar, un grupo se reúne todos los días para ponderar cómo llevar al espacio a determinadas personas. Como en realidad los cohetes son misiles balísticos intercontinentales con carga humana, hay cierta dosis de seguridad y reserva en este esfuerzo privado de viajes espaciales. Se trata además de un negocio, no de una quimera multimillonaria. Mientras escribo estas líneas, otro cohete privado acaba de acoplarse en la estación espacial internacional por segunda vez. Ni siquiera el gobierno estadunidense posee tal capacidad en este momento.

Pero en ese edificio particular, este día particular, esas personas particulares no saben cuán grande debe ser el compartimiento del cohete para el paquete aviónico. La aviónica indica al cohete dónde está, adónde va y cómo llegar. Imagínala como la mente de la nave.

Hay dos equipos, uno de *hardware* y otro de *software*, cada uno de ellos con siete personas. Todos los días, cada equipo se reúne frente a un pizarrón blanco de piso a techo que ocupa toda una pared. Igual que el de WIKISPEED, también éste se divide en columnas: Pendientes, En proceso, Terminado. Las columnas contienen sólo lo que el equipo debe hacer en este *sprint*. Las tareas van de trabajar con uno de entre media docena de proveedores de tableros de circuitos especiales a planear cómo va a entenderse el acelerómetro con el resto de la nave. El *Scrum Master*, el individuo a cargo del proceso, formula tres preguntas a cada miembro:

1. ¿Qué hiciste ayer para ayudar al equipo a terminar este *sprint*?
2. ¿Qué harás hoy para ayudar al equipo a terminar este *sprint*?
3. ¿Qué obstruye el avance del equipo?

Eso es todo. Ésa es toda la reunión. Si tarda más de quince minutos, estás mal. Gracias a esta reunión, el equipo entero sabe exactamente cómo marcha todo en el *sprint*. ¿Todas las tareas serán terminadas a tiempo? ¿Es posible ayudar a otros miembros a superar obstáculos? No hay asignación de tareas desde arriba; el equipo es autónomo: se asigna *a sí mismo* sus tareas. Tampoco hay informes detallados a la dirección. Cualquier integrante de la dirección o de otro equipo puede llegar, examinar el pizarrón de Scrum dedicado a la aviónica y saber exactamente cómo va todo.

Así, cuando el primer equipo de Scrum quiso saber cómo podía asemejarse a los All Blacks, se documentó acerca de cómo trabajaban los

mejores equipos. Una de las ventajas del desarrollo de *software* es que la situación inicial de este ramo era tan mala y se desperdiciaba tanto dinero en él —miles de millones cada año— que se dedicó mucho tiempo a estudiar por qué, así que había datos acerca de todo.

Uno de los que dedicaron años a examinar cómo se hacían las cosas en el campo del *software* fue Jim Coplien, de los legendarios Bell Labs de AT&T. Coplien, llamado «El Capi» por sí mismo y los demás, pasó años escudriñando cientos de proyectos de *software* para descubrir por qué una reducida minoría de ellos habían salido bien en tanto que la mayoría había terminado en desastre. A principios de los años noventa se invitó a Coplien a analizar un proyecto de Borland Software Corporation, sobre un nuevo producto de hoja de cálculo llamado *Quattro Pro* para Windows. En el marco de ese proyecto se había creado un millón de líneas de código de *software*. Producirlas había implicado treinta y un meses y ocho personas. Esto significa que cada uno de los miembros de ese equipo había producido mil líneas de código por semana. Ningún otro equipo de esta especie había trabajado nunca tan rápido y Coplien quería saber cómo lo había conseguido.

Con ese fin, representó gráficamente todos los flujos de comunicación al interior del equipo: quién hablaba con quién, dónde fluía la información y dónde no. Este diagrama es una herramienta clásica para detectar cuellos de botella o acaparadores de información. Cuanto más exhaustiva es la comunicación —cuanto más saben todos de todo—, más rápido es el equipo. Básicamente, la medida que se deriva de este análisis calcula cuánto saben todos lo que necesitan para hacer su trabajo. Borland obtuvo en esto la calificación más alta en la historia: noventa por ciento. La mayoría de las empresas rondan el veinte.

¿Cómo podíamos alcanzar esa exhaustividad en nuestro equipo? Lo que impide la exhaustividad de la comunicación es la especialización, el número de roles y nombres de éstos en un grupo. Si la gente tiene un epíteto especial suele hacer sólo aquello que parece convenir a su categoría. Y para proteger el poder de ese rol tiende a aferrarse a conocimientos específicos.

Nosotros nos deshicimos de los epítetos. Llamé a todos y les dije que rompieran sus tarjetas de presentación. Si alguien quería presentar un epíteto profesional en su currículum, podía hacerlo, aunque sólo para uso externo. Donde estábamos, donde se hacía de verdad el trabajo, todos éramos únicamente miembros de un equipo.

El otro ingrediente de la «fórmula secreta» del equipo de Borland fue que todos debían asistir a la reunión diaria del equipo para hablar de su

desempeño. Reunir a todos era clave, porque daba al equipo la oportunidad de organizarse alrededor de retos. Si alguien se estancaba en un problema —si el acelerómetro no se entendía con el altímetro—, todos comprendían que ese impedimento podía bloquear el *sprint* entero y tomaban cartas en el asunto para resolverlo a la brevedad.

En Borland, la reunión diaria duraba al menos una hora. Esto me pareció mucho tiempo, así que examiné lo básico que debía comunicarse en esa reunión y di con las tres preguntas.

Fue así como pusimos en operación la reunión diaria. Teníamos ciertas reglas. La reunión se efectuaba todos los días a la misma hora y todos los miembros debían asistir a ella. De no estar presente todo el equipo no podía haber comunicación plena. Y la hora de la junta no importaba, siempre que fuera la misma todos los días. La cuestión era dotar al equipo de un ritmo regular.

La segunda regla era que la reunión no podía exceder de quince minutos. Debía ser breve, directa y al grano. Si algo requería más análisis, lo señalábamos y nos volvíamos a ver luego de la reunión diaria. La idea era obtener la información más procesable y valiosa en el menor tiempo posible.

La tercera regla era que todos debían participar activamente. Para contribuir a esto, sugerí que debíamos permanecer de pie. Esto invitaba a conversar y escuchar y abreviaba las reuniones.

Por eso llamamos a éstas la *parada diaria* o el *scrum diario*. Pero sea *como* fuere que las llames, deben ser siempre a la misma hora, con las mismas tres preguntas, todos de pie y de no más de quince minutos.

El problema que encuentro más a menudo es que la gente tiende a usar la parada diaria para rendir un informe personal —«Hice esto… Haré aquello…», y luego el que sigue—, cuando lo óptimo es que sea un *team back* como los del futbol americano. Un receptor abierto podría decir: «Tengo problemas con tal jugador de la línea defensiva», a lo que un bloqueador ofensivo repondría: «Yo me haré cargo. Abriré esa línea». O bien, el mariscal de campo podría decir: «Nuestra ofensiva está topando con pared; sorprendámoslos con un pase a la izquierda». La idea es que el equipo hable brevemente sobre cómo aproximarse a la victoria, es decir, al final del *sprint*. La pasividad no es mera pereza; también afecta el rendimiento del resto del equipo. Una vez detectada, se le debe eliminar de inmediato.

Quiero equipos dinámicos, que salgan de la reunión diaria sabiendo qué es lo más importante que deben lograr ese día. Si alguien oye decir a otro que una tarea le llevará un día, quizá él sepa cómo hacerla en una hora trabajando

en común. Quiero equipos que salgan de esa reunión diciendo algo como «Persigamos eso, hagámoslo». El equipo debe *querer* ser grande.

Siempre digo a equipos grandes y pequeños lo siguiente: «¿Quieren ser unos mediocres? ¿Ésa es su motivación en la vida? Porque eso es algo que podemos decidir nosotros, no algo preestablecido». Un equipo tiene que reclamar su grandeza.

En Easel, en el primer equipo de Scrum, implementamos la parada diaria en el tercer *sprint*. Habíamos planeado cuatro semanas de trabajo para ese *sprint*, casi la misma carga de trabajo que el mes anterior. Pero la acabamos en una semana, una mejora de cuatrocientos por ciento. Ese primer viernes, todos los integrantes del equipo se miraron entre sí y dijeron: «¡Vaya!». Supe entonces que quizá había topado con algo importante.

Una y otra vez

Esa mejora se incorporó a Scrum desde ese tercer *sprint*. Es la meta de diseño de Scrum. He visto equipos muy disciplinados aumentar ocho veces su productividad. Por eso Scrum es tan revolucionario. Puedes hacer más con menos, el doble de trabajo en la mitad de tiempo. Recuerda, además, que el tiempo no sólo es importante en los negocios. También es la materia de la que está hecha tu vida, así que desperdiciarlo es, en efecto, una forma lenta de suicidio.

Scrum altera tu manera de concebir el tiempo. Luego de participar durante cierto lapso en *sprints* y paradas, dejas de ver el tiempo como una flecha lineal dirigida al futuro para entenderlo más bien como algo fundamentalmente cíclico. Cada *sprint* es una oportunidad de realizar algo totalmente nuevo; cada día, una oportunidad de mejorar. Scrum alienta una visión holística del mundo. Quien se compromete con él valorará cada instante como un ciclo de vida y aliento que no cesa de repetirse.

Siempre me ha inquietado que remodelar una casa pueda consumir tanto tiempo. Mi esposa y yo nos recordábamos uno a otro que tal proyecto tardaría el doble y costaría el doble de lo que pensábamos, y eso si teníamos suerte. Estoy seguro de que has oído las mismas historias de horror que yo: el remozamiento de la cocina que supuestamente llevaría dos semanas y terminó tardando seis, obligando así a la familia a consumir comida rápida durante más de un mes; la reparación eléctrica que se prolongó tres veces más de lo previsto; la minucia que acabó resultando interminable. Pues bien, hace un

par de años mi amigo Eelco Rustenburg, también adepto a Agile, me contó en una cena que había decidido remodelar su casa de cabo a rabo. Acometería todas las habitaciones, haciendo una nueva instalación eléctrica, incorporando nuevos aparatos y dando a todo una nueva capa de pintura. Planeaba tardar únicamente seis semanas.

Todos reímos y empezamos a obsequiar a Eelco *nuestras* trágicas historias de remodelación.

—¿Seis semanas para toda una casa? —pregunté riendo—. Imposible. Eso mismo se llevó la remodelación de mi *cocina*, aunque me prometieron dos semanas. Vivirás en un hotel el resto del año.

—No —replicó él—. Mi casa estará lista a tiempo y dentro del presupuesto. Lo haré usando Scrum.

Esto me entusiasmó: la idea de utilizar Scrum en un terreno muy distinto al del *software*. Encontré a Eelco seis meses después y le pregunté cómo le había ido. «Muy bien», contestó. «Seis semanas exactas. Aunque el caso de mi *vecino* es otra historia».

He aquí lo que sucedió. Eelco decidió que los contratistas trabajaran como un equipo con Scrum. Organizó proyectos semanales que debían llegar a la columna de Terminado y en el tráiler del contratista estacionado frente a su casa puso un pizarrón blanco de Scrum lleno de papeletas adhesivas con tareas. Cada mañana reunía a los carpinteros, electricistas, plomeros o quien fuera necesario para el *sprint* de esa semana y revisaba lo hecho el día anterior, lo que se haría ese día y qué podía impedirlo.

Eelco asegura que esto hizo que los contratistas concibieran el proyecto y se comunicaran sobre él de otra manera. Plomeros y carpinteros acordaban cómo ayudarse para trabajar más rápido. La escasez de materiales se detectaba antes de que obstruyera todo avance. Pero, añadió Eelco, lo más importante fue que las paradas eliminaron la dependencia. En todo proyecto de construcción se dedica mucho tiempo a esperar el fin de una parte del trabajo para que la siguiente pueda empezar, fases que, además, suelen implicar diferentes conjuntos de habilidades: instalación eléctrica y tapizado de las paredes, por ejemplo. La parada diaria juntaba a todos en un mismo lugar, donde deducían rápidamente cómo trabajar en equipo. Dejaban de ser individuos con habilidades por separado para convertirse en un equipo resuelto a llevar a Terminado todas las tareas de una casa.

Y funcionó. Seis semanas después el proyecto estaba concluido. Eelco y su familia volvieron a casa, felices de la vida. Cuando me lo contó, no lo podía creer, pero lo felicité por tener tan buenos contratistas. «Espera», me

dijo, «ahí no acaba todo». Un vecino quiso hacer lo mismo en su propia casa. Ambos vivían en una antigua sección de los Países Bajos y sus casas habían sido construidas al mismo tiempo, con los mismos planos. Habiendo visto lo bien que los contratistas trabajaron en la casa de Eelco, el vecino supuso que podía repetir ese acto de magia.

Así, contrató a los mismos trabajadores, pero esta vez demoraron *tres meses*. Misma gente. Mismo tipo de casa. Mismo trabajo. El doble de tiempo y, claro, el doble de dinero. La diferencia fue que el vecino no aplicó Scrum. Los problemas que éste hace salir a la superficie no se descubrieron hasta muy tarde. La gente no se coordinó de la misma manera y tuvo que esperar a que otros terminaran su labor para poder comenzar la suya. El vecino de Eelco acabó pagando casi el doble que él, principalmente a personas que habían tenido que esperar que otros terminaran de trabajar.

Piensa en tu empleo. ¿Cuánto tiempo pierdes esperando a que otro termine su labor, a recibir información o por tratar de hacer demasiadas cosas a la vez? Quizá te guste pasar todo el día en tu oficina, pero yo preferiría ir a surfear.

RESUMEN

El tiempo es finito. Trátalo como tal: divide tu trabajo en lo que puedes hacer en un periodo corto, fijo y regular de entre una y cuatro semanas. Y si te contagias de la fiebre de Scrum, llámalo *sprint*.

Muestra o muerte. Al final de cada *sprint*, ten algo terminado y que pueda usarse (para volar, manejar o lo que sea).

Tira tus tarjetas de presentación. Los epítetos de roles son indicadores de estatus especializado. Que te conozcan por lo que *haces*, no por tu profesión.

Todos saben todo. La exhaustividad de la comunicación acelera el trabajo.

Una reunión al día. Cuando se trata de revisiones de equipo, una vez al día basta. Reúnanse quince minutos, vean qué pueden hacer para avanzar más rápido y llévenlo a la práctica.

Capítulo cinco

El desperdicio es un crimen

La esencia de Scrum es el ritmo. El ritmo es muy importante para los seres humanos. Oímos su compás en el repiquetear de nuestra sangre y sentimos sus raíces en los recovecos más profundos de nuestro cerebro. Buscamos patrones, tenemos el impulso de seguir el ritmo en todos los aspectos de la vida.

Sin embargo, los patrones que perseguimos no son necesariamente gratificantes ni están optimizados para hacernos felices. Existen, por ejemplo, los ritmos negativos del adicto y el deprimido. Podrías recorrer los pasillos de casi cualquier edificio de oficinas y ver manifestarse esos patrones negativos. Se les puede encontrar en cualquier parte en que alguien se sienta frustrado porque se le obstaculiza, o calladamente desesperado al darse cuenta de que está atrapado en un sistema impasible, o enojado por ser visto como pieza de una máquina.

Esto forma parte de la experiencia humana. Si te remontas a hace cientos de años, podrías leer los escritos de personas cuya vida, justo como la nuestra, se vio atrapada en un sistema que sentían irremediablemente hostil. Pero, al parecer, durante el siglo XX dominamos esa sensación de agobio. Especialmente en el entorno de los negocios, generamos una despersonalización aguda que parecería haber sido dictada por el destino.

Scrum produce un patrón distinto. Acepta que somos animales de costumbres, que buscamos ritmo, algo predecible, pero también algo mágico y capaz de grandeza. Cuando hice Scrum pensé: «¿Qué pasaría si pudiera volver positivos algunos patrones humanos, en vez de negativos? ¿Diseñar un círculo virtuoso autorreforzado que estimulara lo mejor de nosotros mismos y redujera lo peor?». Al dar a Scrum un ritmo diario y semanal, creo haber tratado de brindar a la gente la posibilidad de apreciar a quien ve en el espejo.

Pero hay escollos. Lo que parece un patrón virtuoso podría terminar siendo un absurdo, mero desperdicio. A eso dedicaré este capítulo: al desperdicio que contamina nuestro trabajo, el cáncer que corroe nuestra productividad, organizaciones, vida y sociedad.

El otro día entrevisté a alguien interesado en un puesto en Scrum Inc. y le pregunté por qué quería trabajar en la compañía. Me contó su historia. Había trabajado en una empresa que editaba libros académicos y recursos auxiliares como cuadernos de trabajo, material didáctico, presentaciones, etcétera. Su función era localizar a especialistas importantes de un campo particular para generar con ellos tales productos. Esto le entusiasmaba. Había estudiado historia, era experto en el periodo colonial estadunidense y tenía la oportunidad de colaborar con algunas de las principales inteligencias de su campo.

«Estuve ahí un año», me dijo, «un año dedicado a desarrollar docenas de productos. Al cabo de ese periodo, evaluamos nuestros logros por primera vez. Y resultó que nada menos que la mitad de mi trabajo del año previo se había desechado. No porque no fuera bueno, sino porque no había mercado o su dirección había cambiado. Seis meses de mi vida total y completamente desperdiciados».

Cierta indignación e ira asomaron entonces en su voz, y determinación después. «Confío en que Scrum no permita que tal cosa suceda, que mi trabajo tenga un propósito, que lo que hago importe de veras».

Podrías creer que éste es un ejemplo extremo. Cincuenta por ciento de desperdicio. Pero es muy común. Cuando llego a una compañía, suelo descubrir un desperdicio de hasta ochenta y cinco por ciento del esfuerzo. Sólo la sexta parte del trabajo es valiosa. Muy en el fondo, mientras repetimos nuestro ritmo cotidiano, sabemos que eso es cierto. Por eso nos reímos, algo nerviosamente, de bromas acerca de la insensatez y el despilfarro inherentes a la vida de una corporación moderna.

Estoy aquí para decirte que esto no debería divertirnos, sino avergonzarnos. Deberíamos lamentar la vida y el potencial que desperdiciamos. En el primer capítulo de este libro introduje brevemente a Taiichi Ohno, de Toyota, citando estas palabras: «El desperdicio es un crimen contra la sociedad antes que una pérdida de dinero». Las ideas de Ohno sobre el desperdicio influyeron profundamente en las mías, así que les dedicaré unos párrafos.

Ohno se refirió a tres tipos de desperdicio. Usó para ellos términos japoneses: *muri*, el desperdicio por irracionalidad; *mura*, el desperdicio por

incongruencia, y *muda*, el desperdicio por resultados. Estas ideas concuerdan por entero con el ciclo PDCA de Deming, al que aludí anteriormente: **P**lan, **D**o, **C**heck, **A**ct (Planea, Haz, Revisa, Actúa). Planea significa evitar el *muri*, Haz el *mura* y Revisa el *muda*. Actúa significa la voluntad, motivación y determinación para hacer todo eso. Describiré esos pasos uno por uno y señalaré qué evitar, desde desperdicio en inventario hasta el de no hacer bien las cosas a la primera, el de trabajar demasiado y el desperdicio emocional de expectativas poco razonables[21].

Haz una cosa a la vez

A menudo oigo a personas alardear de su aptitud para hacer varias tareas al mismo tiempo. Sin duda tú también sueles oírlas. Si acaso no eres tú mismo quien se jacta de ello, conoces a alguien que lo hace: el sujeto que realiza tres proyectos simultáneos, que habla por su teléfono celular mientras maneja, que promueve su habilidad quejándose ruidosamente de todo aquello con que debe hacer malabares a diario. Esta «presunción de ocupación» ya forma parte de nuestra cultura laboral. En descripciones de puestos de trabajo pueden verse ahora requisitos como «Debe ser capaz de gestionar cinco proyectos simultáneos».

La aptitud para la prestidigitación parece sumamente *atractiva*, sobre todo en una época en que la información fluye a través de mil conductos y en que el «Debe hacerse ahora» prolifera. Todos quisiéramos ser superprestidigitadores. Nos decimos que podemos lograrlo, pero no es así. Y cuanto más creemos poder hacerlo, peores somos para eso.

Un ejemplo apropiado es una práctica diaria de multitareas: manejar y hablar por el celular. Las investigaciones son rotundas a este respecto: quienes hablan por el celular mientras manejan —aun si utilizan la variedad de manos libres— tienen más accidentes que quienes no lo hacen. El problema es especialmente alarmante si se considera que, según la National Highway Transportation Safety Administration (Dirección Nacional de Seguridad en el Transporte en Autopistas) de Estados Unidos, en todo momento ocho por ciento de los automovilistas de ese país están hablando en su celular.

Esto es lo que nos ha legado la multitarea.

He aquí una cita de mi artículo preferido sobre este tema:

> Aun si los participantes miran objetos en su entorno, a menudo no los «ven» mientras hablan por celular, porque su atención se desvía del entorno y se dirige a un contexto cognitivo interior asociado con la conversación telefónica[22].

Cierto: la gente mira un objeto, el auto de enfrente o el árbol en que está a punto de estrellarse, y *no lo ve*. Aun así persiste en manejar y hablar por teléfono.

Puedo leer tu mente en este momento. Estás pensando: «Bueno, hay quienes no pueden hacerlo. Pero yo soy un ejecutivo dinámico, o una mujer inteligente, y sí puedo». Sin embargo, la bibliografía es muy clara: si crees ser bueno para eso, en realidad eres peor que otros. La University of Utah, la cual ha hecho muchas investigaciones interesantes en esta área, preguntó a algunas personas si creían ser aptas para hacer varias cosas a la vez, como hablar por su celular mientras manejaban, y luego las puso a prueba para ver si estaban en lo cierto. He aquí las conclusiones de los investigadores:

> La percepción de aptitud para multitareas resultó exagerada; de hecho, la mayoría de los participantes juzgaron estar por encima del promedio en tal aptitud. Estas estimaciones tienen escaso fundamento en la realidad. Parece ser, así, que las personas más propensas a incurrir en multitareas y usar un celular mientras manejan son justo las de opiniones más exageradas sobre sus grandes habilidades[23].

El autor principal de este estudio, David Sanbonmatsu, declaró en enero de 2013 a *Shots*, el blog de NPR: «La gente no hace multitareas porque sea buena para eso, sino por ser muy distraída. Se le dificulta inhibir el impulso de hacer otra cosa». En otras palabras, no sabe concentrarse. No puede evitar hacer muchas actividades a la vez.

Aunque quizá debería decir «nosotros», A todos nos pasa lo mismo. Es difícil que no sea así. Lo que hay que recordar es que realizar multitareas es una tontería. Me gustaría que hicieras un pequeño ejercicio, que acostumbro aplicar en mis cursos de capacitación. Es muy sencillo, pero revela el hondo impacto de la concentración y el flujo. Demuestra lo penosas que son para tu cerebro las multitareas y cuánto te retrasan aun si crees que te aceleran. Revela sencillamente que dicha práctica es un *desperdicio*.

Escribe los números 1 a 10, los numerales romanos 1 a 10 (I, II, III, IV, etcétera) y las letras *A* a *L*. Fíjate cuánto tardas en hacerlo, tratando de

avanzar lo más pronto posible. La primera vez, escribe el número arábigo, luego el romano y al final la letra, así:

1	*I*	*A*
2	*II*	*B*
3	*III*	*C*

Trabaja por renglones. Fíjate cuánto tardas. Yo lo haré contigo. ¡Ya está! Tardé treinta y nueve segundos. Ahora, en vez de ir por renglones hazlo por columnas: primero todos los números arábigos, luego los romanos y por último las letras. También yo lo haré. Diecinueve segundos. Mediante el solo hecho de cumplir mis tareas una por una, en vez de pasar de un contexto a otro, reduje mi tiempo a la mitad.

«De acuerdo, Sutherland», te oigo decir, «eso está bien para el celular y listas absurdas, pero yo dirijo una empresa. Tengo que hacer muchas cosas al mismo tiempo, lograr que mis equipos atiendan cinco proyectos simultáneos. Debo seguir siendo competitivo. No puedo darme el lujo de no serlo».

Aquí vuelve a ser útil la inmensa cantidad de investigaciones sobre proyectos de *software*. Como recordarás, esas investigaciones se hicieron porque cada año se desperdiciaban cientos de millones de dólares y los productos eran cada vez peores. Como buenos ingenieros, los expertos en ese campo se pusieron a examinar datos y a medirlo todo. En una obra clásica de desarrollo de *software*, *Quality Sotfware Management*, de Gerald Weinberg, aparece una gráfica muy elocuente[24]:

Número de proyectos simultáneos	Porcentaje de tiempo disponible por proyecto	Pérdida por cambio de contexto
1	100%	0%
2	40%	20%
3	20%	40%
4	10%	60%
5	5%	75%

La columna «Pérdida por cambio de contexto» es desperdicio puro. Cierto: si ejecutas cinco proyectos al mismo tiempo, nada menos que setenta y cinco por ciento de tu trabajo no va a ninguna parte; tres cuartas partes de tu día

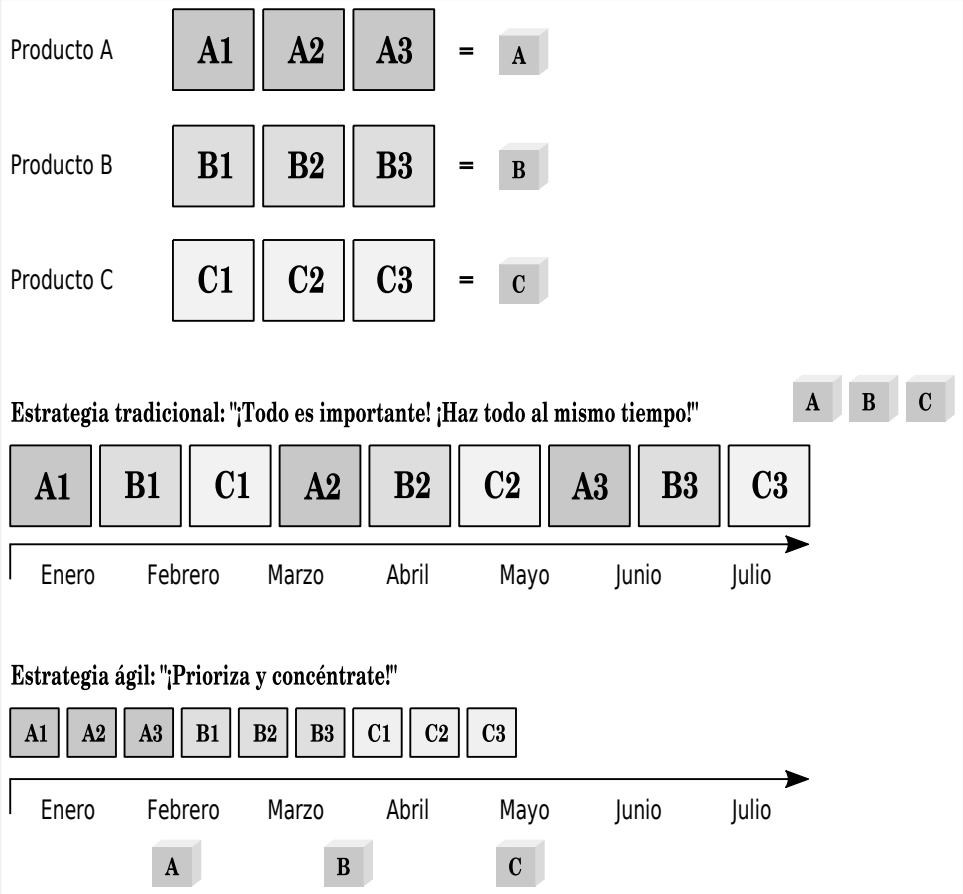
echados al caño. Por eso no pudiste escribir esos renglones y columnas con igual rapidez. Esto es producto de las limitaciones físicas del cerebro.

El científico Harold Pashler lo demostró a principios de los años noventa; lo llamó «Interferencia de doble tarea». Hizo unos experimentos muy sencillos. Puso a un grupo de personas a hacer algo muy simple, como oprimir un botón si se encendía una luz. Dio a otro esa misma tarea y otra más, como oprimir otro botón según el *color* de la luz intermitente. Pero en cuanto se añadió otra tarea, por simple que fuera, el tiempo implicado se duplicó. Pashler teorizó la existencia de un cuello de botella del procesamiento; que la gente sólo podía pensar en una cosa a la vez. Consideró que meter un proceso en la memoria y sacar otro para ejecutar una nueva tarea implica cierto esfuerzo. Y cada vez que cambias de tareas este proceso consume tiempo[25].

Por lo tanto, no lo hagas. Concéntrate en una sola cosa. Si hablas por el celular, aun de algo tan simple como de pasar a comprar leche, literalmente no verás el auto de adelante. Tu cerebro no puede procesar esas dos cosas al mismo tiempo. Investigaciones recientes con resonancia magnética han permitido obtener representaciones gráficas del cerebro en el momento de pensar. Los datos demuestran que para poder pensar en dos cosas a la vez es necesario que cada uno de esos procesos ocurra en uno de los hemisferios del cerebro. Aun así, los escáneres indican que el pensamiento no sucede en forma simultánea, sino que el cerebro pasa de una tarea a otra de manera serial. Básicamente, existe una función de control, así que no puedes discutir contigo mismo con mucho ímpetu[26].

Volvamos al trabajo. ¿Qué significa eso para la ejecución de proyectos? Examinemos un equipo representativo. Este año ha decidido realizar tres proyectos. Llamémoslos *A*, *B* y *C*. Este equipo planea todo el año diciendo que avanzará un poco en un proyecto, luego un poco en otro y después un poco en el último, así que su calendario lucirá de esta manera:

PRIORIZACIÓN ENTRE PROYECTOS



Intentar hacerlo todo al mismo tiempo —la estrategia clásica— supondrá concluir esos tres proyectos a fines de julio. Pero abordar el agregado al modo de Scrum, pasando a Terminado cada proyecto uno por uno, permite minimizar el costo del cambio de contexto y concluir para principios de mayo.

Esos proyectos no cambian de magnitud, ni en cuanto a lo que implica ejecutarlos, sino sólo respecto al principio de hacer una *sola* cosa antes de continuar, lo que consume poco más de la mitad de tiempo. La *mitad*.

¿Y la otra mitad? Eso es desperdicio puro. No produce nada. No se ahorra un solo dólar. No se implementa innovación alguna. Es mero desperdicio de vida humana. Trabajo sin propósito.

Tal es el costo de hacer varias cosas a la vez. Vivimos en un mundo con demandas múltiples sobre nuestro tiempo. La gente quiere de nosotros cosas diferentes: recibimos una llamada telefónica importante, los hijos regresan de la escuela, el jefe entra a nuestra oficina. Pero debes estar *consciente* del costo del cambio de contexto. Es real y debes tratar de minimizarlo.

Si te ocupas de algo complicado —como redactar un informe, crear una presentación, desarrollar una pieza de *software* o escribir un libro—, tienes en

tu mente un objeto muy complejo. Debes tomar en cuenta docenas de factores, recordar qué has hecho, dónde quieres ir y qué impedimentos puedes enfrentar. Hacerlo es muy difícil. ¿Y qué sucede si te interrumpen o si tienes que pasar rápidamente a otro proyecto, así sea sólo un momento? Adivinaste: esa arquitectura mental laboriosamente construida se viene abajo. El solo hecho de regresar al mismo estado de conciencia puede suponer *horas* de trabajo. Tal es el costo. Por tanto, minimiza ese desperdicio tratando de completar de *una vez* las tareas que requieren un tipo específico de concentración. Sitúalas en bloques de tiempo en los que puedas apagar tu teléfono y poner un letrero de «No molestar».

Algunas investigaciones demuestran incluso que las multitareas no sólo te hacen perder tiempo, sino que también te embrutecen. Un estudio efectuado por la University of London en 2005[27] (con pocas personas y sin revisión colegiada, pero útil de todas formas) midió lo torpe que pueden volverte las multitareas. El psiquiatra Glenn Wilson probó el IQ de cuatro hombres y cuatro mujeres en condiciones tranquilas y de distracción (timbres de teléfonos, recepción de correo electrónico). Durante las pruebas, midió la conductancia de la piel de sus sujetos, así como su ritmo cardiaco y presión arterial. Curiosamente, su IQ promedio bajó más de diez puntos cuando se distraían, y el de los hombres más que el de las mujeres (quienes, por alguna razón, quizá estén más habituadas a distraerse).

Hacer a medias es no hacer en absoluto

Como ya dije, Scrum tomó muchas de sus ideas del modelo japonés de manufactura codificado en el libro clásico *The Toyota Production System* de Taiichi Ohno. En Estados Unidos, este modelo se interpretó como manufactura «saneada». Básicamente, la idea es eliminar todo desperdicio posible en la fábrica. Y aunque la mayoría de nosotros no perseguimos mejorar el flujo de trabajo de una planta automotriz, algunas de las ideas de ese sistema son aplicables a cualquier clase de labor.

Un concepto que quiero tocar aquí es el de «trabajo en proceso» o «inventario». La idea es que tener muchas cosas regadas que no sirven para nada constituye un desperdicio. Esas cosas, sean puertas de automóviles o artilugios de cualquier especie, cuestan dinero y si están en la fábrica quiere decir que grandes sumas de dinero están atadas a un inventario que no se necesita por lo pronto. Esto cambia tu manera de ver las cosas que están *en*

proceso. Por ejemplo, si todo lo que una compañía automotriz tiene es un montón de coches a medio hacer, ha invertido en ello mucho dinero y esfuerzo, pero no ha creado nada realmente valioso. En la manufactura saneada, la idea es minimizar la cantidad de cosas a medio hacer dispersas por ahí.

La eficacia de esta noción es aplicable a todo tipo de trabajo. Citemos un ejemplo sencillo que casi todos los adultos casados de este planeta entenderán: la lista de pendientes del hogar. En cualquier semana dada, la mía suele contener de diez a veinte deberes, de repintar el baño a comprar comida para el perro, de pagar la hipoteca a barrer las hojas del jardín. De esto está hecha la vida diaria, ésta es la fricción que se deriva de ser miembro pleno de la sociedad. Esa lista puede atacarse de varias maneras, pero lo peor que puedes hacer es tratar de cumplir cinco cosas al mismo tiempo. Eso es multitareas y quizá no termines ninguna, lo que te deja con trabajo en proceso.

Imagina (o recuerda, si no eres afortunado) tener cinco tareas *parcialmente* concluidas. Pintaste una pared del baño, dejaste la comida del perro en la cajuela, elaboraste el cheque de la hipoteca pero no lo enviaste por correo y juntaste las hojas pero no las metiste en bolsas. Invertiste esfuerzo, pero no creaste nada valioso. Esto último ocurrirá cuando las cubiertas protectoras y latas de pintura estén fuera del baño, el perro haya comido, el banco haya recibido su dinero y el jardín esté limpio. Hacer algo a la mitad es, en esencia, no hacer nada.

Como ya dije, en Scrum hay un ritmo de trabajo. En cada repetición, o *sprint*, el equipo intenta acabar varias cosas. Pero ese Terminado implica un producto completo, entregable, que el cliente pueda usar. Si al final del *sprint* algo quedó a medias, estarás peor que si no hubieras hecho nada. Has gastado recursos, tiempo y esfuerzo sin llevar nada a un estado susceptible de entrega. Tienes un auto a medio hacer. Quizá habría sido preferible producir algo más chico, que realmente funcione.

Otra manera de ver el trabajo en proceso o inventario es como inventario físico. Tomemos el caso de los automóviles. Tener muchos sin vender en un lote es un problema para un fabricante. Pero no tener ninguno para su venta también lo es. Así, cada fabricante y distribuidor ejecuta un cuidadoso acto de equilibrio. Quiere producir vehículos suficientes para disponer de existencias, pero no tantos que supongan invertir grandes cantidades en cosas que no se venden.

Permíteme ponerle cifras a eso. En diciembre de 2012, General Motors (GM) comenzó a despedir personal en algunas de sus plantas en Estados Unidos. ¿Por qué? Había fabricado demasiados coches. A fines de noviembre de ese año tenía 245 853 *pickups* estacionadas en lotes en todo el país. Eso representaba camionetas de carga para ciento treinta y nueve días. A un precio promedio, representaba también unos siete mil quinientos millones de dólares. Miles de millones. Mucho dinero, en este caso en forma de camionetas, pero dinero al fin, ahí nada más sin vender. Así, la compañía empezó a cerrar plantas, despidiendo a empleados justo antes de la Navidad.

¿Un inventario para cuántos días *debe* tener una compañía automotriz? El estándar de la industria es sesenta días, menos de la mitad de lo que GM tenía. Piénsalo. Cuando compras comida para tu perro no es necesario que te surtas para seis meses. Ocuparías mucho espacio en tu alacena, y tal vez ese mes ya no te quedaría dinero para el cheque de la hipoteca.

Quizá pienses: «Bueno, ellos *fabricaron* esas camionetas de carga; las terminaron, ¿no? No son cosas a *medio hacer*; ¿cuál es el problema?». El problema es que demasiado inventario es casi lo mismo que trabajo en proceso. Si atas mucho valor a cosas que no están *rindiendo* valor, no contarás con esos recursos para realizar otras cosas, como comercializar más, hacer más promoción de ventas o explorar nuevas ideas. Debes tener inventario; la clave es minimizarlo.

Tareas que no se terminan y productos que no se usan son dos aspectos de lo mismo: esfuerzo invertido sin resultados positivos. Evítalo.

Hazlo bien a la primera

El doctor James Womack, fundador del Lean Enterprise Institute del MIT y autor de numerosos libros sobre manufactura saneada, cuenta una magnífica historia sobre los peligros del «retrabajo» en su obra clásica *The Machine That Changed the World* (La máquina que cambió el mundo). Womack y su equipo pasaron años viajando por el mundo para investigar el mayor empeño manufacturero nunca antes emprendido por seres humanos: la fabricación de automóviles. Querían saber por qué algunas compañías los producían más rápido y con menos defectos que otras. Hoy todo fabricante racional practica lo que Womack decidió llamar *manufactura saneada*, pero entonces las cosas eran distintas.

Una de las principales diferencias entre los fabricantes radicaba en el mercado de autos de lujo. En Japón, compañías como Toyota, Honda y Nissan invertían un promedio de 16.8 horas en hacer un auto de lujo. Diecisiete horas después de la entrada de partes en un extremo de la fábrica, del otro emergía un Lexus. Y había treinta y cuatro defectos por cada cien vehículos. No estaba nada mal.

En Europa, el caso era diferente. Compañías como Mercedes-Benz, Audi y BMW tardaban cincuenta y siete horas en fabricar un auto y obtenían 78.7 defectos por cada cien vehículos.

¿Por qué los europeos demoraban tanto? ¿Y por qué generaban tantos defectos? BMW no se distingue precisamente por coches malos. He aquí el porqué: en las plantas de Toyota, cualquier trabajador puede detener la línea cuando surge un problema. En ese momento, todos se congregan donde se detuvo la línea y no para gritarle al que la paró, sino para resolver el problema, sea cual fuere. No quieren que en el otro extremo salgan autos con cosas por remediar. Resuelven el problema una vez y para siempre. De no hacerlo, ese defecto podría aparecer en cientos de vehículos.

En las fábricas europeas de autos de lujo las cosas se hacían de otro modo. Al final de la línea de producción, docenas de individuos de bata blanca iban y venían tratando de resolver problemas. Se cercioraban de que la puerta emitiera al cerrar ese golpe metálico BMW, o que el motor ronroneara con el tono correcto. Confirmaban que todas las partes engranaran apropiadamente. No se veían a sí mismos como fabricantes, sino como artesanos, operarios haciendo algo bello. Esto es fabuloso cuando la producción es baja, pero cuando haces millones los costos se acumulan. Como informa Womack en su estudio:

> La planta alemana invertía más esfuerzo en resolver problemas recién creados que el que la japonesa dedicaba a producir un auto casi perfecto a la primera[28].

Sí, leíste bien: los alemanes destinaban más tiempo a reparar un auto recién terminado que el que los japoneses dedicaban a hacer uno completo. Hay una razón de que Toyota se haya convertido en el fabricante de autos número uno del planeta: los hacía bien a la primera.

Pero las cosas no siempre quedan perfectas al primer intento. Somos humanos; cometemos errores. La forma en que manejas esos errores puede tener gran efecto en lo rápido que haces las cosas y en su nivel de calidad. En Toyota, como ya dije, cualquier trabajador puede detener la línea de

producción. La idea es mejorar continuamente el proceso y que el momento indicado para resolver un problema es justo cuando se le detecta, no después.

Hace unos años fui a California a hablar con el personal de desarrollo de Palm. Esta compañía hizo algunos de los primeros asistentes digitales personales que ahora llamamos teléfonos inteligentes. Y llevaba registro automático de todo lo que hacía. Una de las muchas cosas que medía era cuánto tiempo implicaba corregir un error, es decir, cuánto tardaba un desarrollador de *software* en resolver un problema que él mismo había introducido en el sistema. La computadora registraba esto automáticamente, en toda ocasión.

Supongamos que al tratar de integrar código de Matt en el sistema, los empleados de control de calidad detectaban un error. Renuente, como la mayoría de los desarrolladores, a corregir ese código de inmediato, Matt se comprometía a ocuparse de eso después. Primero escribía *nuevo* código.

En casi todas las compañías pruebas como ésa no ocurren siquiera el mismo día. Pueden pasar semanas o meses antes de probar todo el código y sólo entonces se descubrirían los problemas. Pero Palm hacía pruebas automáticas diarias de todo su código, así que sabía de inmediato cuándo había un problema.

Puesto que examinaba a todos sus «Matts» —cientos de desarrolladores—, la compañía decidió analizar cuánto implicaba remediar un error si se le corregía de inmediato o semanas después. Recuerda que el *software* puede ser demasiado complicado, así que ¿cuál crees que era la diferencia entre ambas prácticas?

De veinticuatro veces. Si un error era atacado el mismo día en que se le cometió, corregirlo llevaba una hora; tres semanas después, *veinticuatro*. Así el error fuera grande o pequeño, simple o complejo, corregirlo suponía siempre veinticuatro veces más tiempo tres semanas después. Como es de imaginar, pronto se pidió a todos los desarrolladores de esa compañía probar y corregir su código el mismo día.

Ya he escrito mucho sobre los límites de la mente. Sólo podemos recordar unas cuantas cosas; sólo podemos concentrarnos en una cosa a la vez. Esta otra tendencia —la de que reparar cosas se dificulta al paso del tiempo— constituye una limitación similar. Cuando trabajas en algo, creas todo un espacio mental a su alrededor. Conoces todas las razones de lo que haces. Tienes en la cabeza una estructura muy complicada. Recrearla una semana después es *difícil*. Debes recordar todos los factores que consideraste al tomar una decisión, recrear el proceso mental que te llevó a ella. Tienes que ser

como eras, incursionar otra vez en una mente que ya no existe. Hacer eso lleva tiempo, *mucho* tiempo. Veinticuatro veces más de lo que llevaría si resolvieras el problema en el momento de detectarlo.

Estoy seguro de que has vivido esta experiencia en tu trabajo y quizá hayas aprendido desde niño la lección: *hacer bien las cosas a la primera*. Lo que los datos añaden ahora es que si cometes un error —y todos los cometemos—, debes corregirlo en cuanto lo adviertas, de lo contrario, lo pagarás caro.

Trabajar de más sólo crea problemas

Cuando Scott Maxwell, fundador de la sociedad de capital de riesgo OpenView Venture Partners, trabajaba como consultor en McKinsey & Company, a principios de la década de 1990, recibió una exhortación que le pareció extraña. Jon Katzenbach, entonces consejero de esa compañía y hoy autor de varios libros y director del Katzenbach Center de Booz Allen Hamilton, le dio algunos consejos que nunca olvidaría. Le contó que en sus inicios, en los años setenta, en McKinsey todos trabajaban los siete días de la semana. Ésa era la cultura de la empresa, lo que se esperaba de ellos. No trabajar tanto se habría visto como que no ponías de tu parte ni contribuías al éxito del equipo.

Por causas religiosas, Katzenbach sólo trabajaba seis días a la semana. Y descubrió algo: que aunque trabajaba menos, hacía *más* que los empleados —entonces no se contrataba a mujeres ahí— que laboraban todos los días. Así, decidió hacer la prueba de sólo cinco días a la semana. Y notó que hacía más todavía. «Trabaja demasiado», le dijo a Maxwell, «y harás menos». Añadió que siempre había querido reducir su semana laboral a cuatro o hasta tres días, para ver qué pasaba, pero dudaba que la compañía lo aceptara.

Maxwell y los demás jóvenes consultores se burlaron entonces de esa idea. *¿Trabajar menos tiempo? ¿No era eso holgazanear*? Pero él no olvidó nunca esa idea mientras progresaba en su carrera, y como director general y fundador de OpenView Venture Partners empezó a invertir en compañías tecnológicas, algunas de las cuales practicaban Scrum. Enterado de que yo era el inventor de este enfoque y vivía en la misma ciudad que él, un día me invitó a desayunar. Ante una taza de café y *croissants*, me contó que los equipos de desarrollo de una compañía en la que había invertido aumentaron su productividad en veinticinco a treinta y cinco por ciento luego de

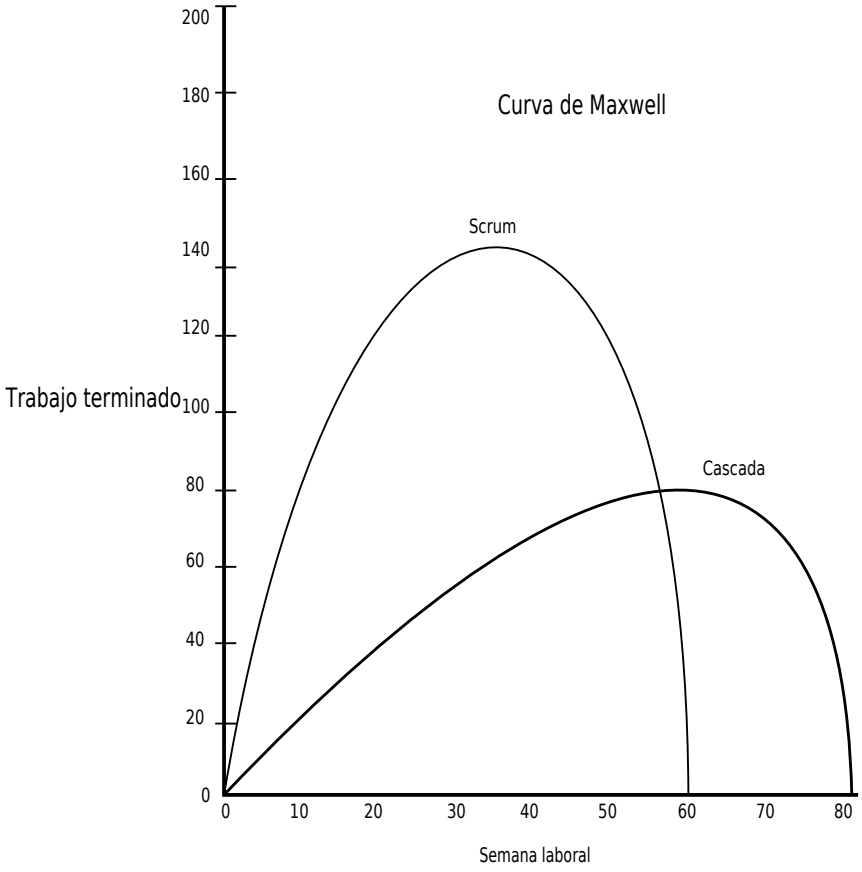
implementar Scrum, lo cual le impresionó mucho. Mi respuesta instantánea: «¿Veinticinco a treinta y cinco por ciento? ¡Lo han de estar usando mal!».

Maxwell decidió aplicar Scrum en todas las áreas de OpenView. Empleados de inversión, gente de investigación, la alta dirección, personal administrativo, todos fueron integrados a equipos de Scrum. Y entonces sucedió una de las grandes cosas que Scrum hace posible: OpenView descubrió *cómo* trabaja la gente, no cómo *dice* trabajar.

OpenView era en ese tiempo como muchas otras oficinas dinámicas. En su cultura había arraigado la expectativa de que los empleados trabajaran hasta tarde y los fines de semana. Eran sujetos enérgicos, ambiciosos. Pero estaban exhaustos, deprimidos y desmoralizados. Las condiciones eran tan severas que algunos no aguantaban y se iban.

Pero cuando los equipos de esa empresa comenzaron a trabajar con Scrum, Maxwell notó un cambio en la productividad. Trabajar más tiempo no permitía generar más productos. Un día me llevó a su oficina y dibujó esta curva en un pizarrón blanco:

DUPLICA LA PRODUCCIÓN REDUCIENDO LA CARGA DE TRABAJO



El eje *y* es productividad y el *x* horas de trabajo. La cresta de productividad empieza a descender en poco menos de cuarenta horas a la semana. Armado de estos datos, Maxwell comenzó a mandar temprano al personal a su casa.

«Tardaron un rato en entender que hablaba en serio», dice, «pero finalmente aceptaron mi manera de pensar».

Comenzó diciendo a sus empleados que trabajar hasta tarde no era señal de compromiso, sino de fracaso. «No es porque yo quiera que tengan una vida equilibrada», les decía, «es porque de esta manera harán más».

Así que no más noches, no más fines de semana. Cuando el personal sale de vacaciones se espera que realmente lo haga, no que revise el correo electrónico o se reporte a la oficina. La idea es que si no puedes tomar un descanso sin tener que confirmar que todo marcha bien en la oficina, no diriges bien a tus equipos.

«Muchas compañías no practican [los límites al horario de trabajo]», dice Maxwell. «Pero hay una correlación directa. Haces más, eres más feliz y alcanzas más calidad». No hay vuelta de hoja: trabajar menos permite hacer más con mayor calidad.

Maxwell dice que esa curva es distinta para personas diferentes y aun para la misma persona en momentos diferentes de su vida. «Conforme maduro y ejerzo papeles distintos he notado que llego a mi productividad máxima en menos horas que hace veinte años», dice. Condición física, dieta, asuntos personales y otros factores tienen que ver en ello, piensa. Pero también sabe que rinde más conforme ha madurado y reflexionado en cómo trabajar. «Puedo atacar cada vez más oportunidades de gran relevancia».

¿A *qué* se debe que si trabajas menos hagas más? No parece tener sentido. Maxwell dice que quienes trabajan demasiado empiezan a cometer errores, cuya corrección, como ya vimos, puede implicar más esfuerzo que crear algo. Quienes trabajan de más se distraen más y distraen a otros. Pronto toman malas decisiones.

La intuición de Jon Katzenbach resultó ser cierta. Las inquietantes evidencias revelan que tenemos una capacidad muy limitada para tomar decisiones y que cuanto más energía perdemos y menos descansamos, más se reduce esa capacidad.

En abril de 2011, un grupo de investigadores israelíes publicó una investigación notable sobre toma de decisiones en los *Proceedings of the National Academy of Sciences of the United States of America*. Su artículo, titulado «Extraneous Factors in Judicial Decisions» (Factores extraños en

decisiones judiciales), estudiaba más de mil resoluciones de ocho jueces israelíes que presidían dos comités de libertad condicional. Esas resoluciones atañían a delincuentes israelíes tanto judíos como árabes, lo mismo hombres que mujeres. Los delitos iban de malversación de fondos y agresión a violación y homicidio. La gran mayoría de las decisiones que los jueces examinaban eran solicitudes de libertad condicional[29].

Parece fácil, ¿no? Aquéllos eran jueces apreciados que se servían de sus muchos años de experiencia y sabiduría para tomar decisiones cruciales que afectaban no sólo a los presos y sus víctimas, sino también al bienestar de la comunidad. Cada día oían entre catorce y treinta y cinco casos.

Si hubieras sido uno de esos presos, ¿de qué habría dependido que se te liberara o no? ¿De que te hubieras arrepentido de verdad? ¿De tu reforma y conducta en prisión? ¿De la severidad de tu crimen? No. Del tiempo transcurrido desde que el juez había consumido un sándwich.

Los investigadores analizaron en qué momento tomaban los jueces sus decisiones, si otorgaban clemencia y cuánto tiempo había pasado desde que comieron un bocadillo. Si apenas habían empezado a trabajar, regresaban de una pausa para comer algo o volvían de almorzar, tomaban decisiones favorables en más de sesenta por ciento de los casos. Este índice se reducía a cerca de cero en el momento de la siguiente pausa.

Básicamente, justo después de una pausa breve, los jueces llegaban con una actitud más positiva y tomaban decisiones más indulgentes. Exhibían más imaginación y capacidad de ver que el mundo y la gente podían cambiar, podían ser distintos. Pero una vez agotadas sus reservas de energía, tomaban cada vez más decisiones inclinadas hacia el orden establecido.

Estoy seguro de que si se hubiera preguntado a esos jueces si estaban ciertos de haber tomado siempre buenas decisiones, se habrían ofendido. Pero los números, y los sándwiches, no mienten. Cuando no nos quedan reservas de energía tendemos a tomar malas decisiones.

Este fenómeno se ha llamado «fatiga del ego». La idea es que tomar cualquier decisión implica un costo de energía. El agotamiento resultante es raro; no te sientes cansado físicamente, pero tu capacidad para tomar buenas decisiones disminuye. Lo que cambia es tu autocontrol, tu capacidad para ser disciplinado, reflexivo y visionario.

Esto quiso demostrarse con un experimento muy sugestivo. Un grupo de investigadores quería saber cómo afectaba al autocontrol la toma de decisiones. Reunieron así a una serie de soldados rasos de la investigación psicológica —estudiantes universitarios—, a algunos de los cuales les

hicieron tomar muchas decisiones. Específicamente, se les presentaron diferentes productos y se les pidió escoger entre ellos. Se les dijo que debían pensarlo bien porque recibirían un regalo al final del experimento según sus preferencias. A los demás estudiantes no se les pidió tomar decisiones[30].

Al grupo de prueba se le hicieron preguntas como si prefería velas con aroma de vainilla o de almendra, cuál marca de champú y qué clase de caramelos. Después se le sometió a una prueba clásica de autocontrol: cuánto tiempo podía mantener una mano en agua helada.

Sea cual fuere el recurso consumido en tomar decisiones, también la autorregulación se desgasta. Los estudiantes que habían tomado todas esas decisiones sobre productos no pudieron mantener la mano en agua helada tanto como el grupo de control, que no había tomado decisiones.

Hay un número limitado de decisiones formales que puedes tomar en un día; si tomas más mermas tu aptitud para regular tu conducta. Empiezas a cometer errores, a la larga errores serios. Como se advierte en la curva de Maxwell, esas malas decisiones tienen un efecto en la productividad. En consecuencia, vuelve a casa a las cinco. Apaga tu celular el fin de semana. Ve una película. Y quizá, sobre todo, come un sándwich. Al no trabajar tanto, harás más y mejor.

Scrum pide a sus adeptos abandonar la mentalidad de medir meras horas. Las horas representan de suyo un costo. En cambio, mide la *producción*. ¿A quién le importa cuántas horas trabajó alguien en algo? Lo relevante es que se entregue rápido y sea de gran calidad.

Sé razonable

Los tres tipos de desperdicio identificados por Taiichi Ohno hacen que la gente trabaje más duro y más tiempo de lo necesario. Acabo de señalar por qué eso es mala idea, pero reconocer las diversas clases de desperdicio que Ohno llamó *muri*, o «Irracionalidad», es quizá la mejor palanca para el cambio.

La primera es «Absurdo». Debes dar a tu equipo metas desafiantes, impulsarlo a alcanzar más, pero *no* hacer que persiga metas absurdas e imposibles.

La segunda es «Expectativas poco razonables». ¿Cuántas veces no has oído presumir a alguien de que gracias a sus heroicos esfuerzos se salvó un proyecto? Esto suele celebrarse con palmadas en la espalda, vítores y

felicitaciones, pero a mí me parece una falla fundamental del proceso. Un equipo que depende de actos heroicos regulares para cumplir sus fechas límite no trabaja como debería. Pasar de una crisis a otra cansa y no permite una mejora razonada y continua. Es la diferencia entre un vaquero que llega a caballo a rescatar a la dama y un disciplinado pelotón de la Marina que despeja la zona de muerte.

Ohno llamó al tercer tipo de desperdicio «Sobrecarga». Es el tipo de conducta que Scott Adams satiriza normalmente en su tira cómica, *Dilbert*. Incluye políticas onerosas de la compañía, reportes inútiles que tienen a la gente llenando formularios porque sí y reuniones sin sentido que quitan tiempo y no generan nada valioso.

Aunque Ohno no mencionó un cuarto tipo de desperdicio, a mí me viene a la mente el «desperdicio emocional». Surge cuando una compañía tiene entre sus filas a un idiota, alguien que gusta de complicarle la vida a la gente y ponerla nerviosa. Los idiotas suelen justificar su conducta diciendo que intentan que la gente trabaje mejor. Pero lo cierto es que sólo satisfacen los aspectos negativos de su personalidad y nada menoscaba más la aptitud de un equipo para sobresalir.

No seas un idiota, ni permitas, induzcas ni aceptes esta conducta en otros.

Flujo

En un mundo teóricamente perfecto no habría procesos, reuniones, formularios ni reportes. En cambio, se crearía *justo* lo que el cliente quiere, aun si éste no sabe aún qué es. Cualquier «proceso» es desperdicio y esto incluye a Scrum.

Pero no vivimos en un mundo perfecto y los malos procesos están tan arraigados en nuestra mente que, como alternativa, necesitamos el más ligero de ellos que tenga mayor influencia en nuestro trabajo. Scrum dirige nuestra mente a la eliminación del desperdicio absurdo que parece formar parte del trabajo. He intentado lograr que ese proceso sea el marco menos perturbador posible que mantenga concentrada a la gente.

Lo que realmente necesitas en tu trabajo es un «flujo» natural. En las artes marciales o la práctica de la meditación, cuando te sientes unido a un movimiento éste ha dejado de ser un esfuerzo; es energía que fluye con donaire a través tuyo. Cuando ves a grandes bailarines o cantantes sientes que

se abandonan a una fuerza superior mientras permiten que su arte circule por ellos. Todos deberíamos pretender llegar a ese punto en nuestro trabajo.

Pero como te dirá el maestro de kung fu, el monje, el bailarín o la estrella de ópera, en la raíz del flujo está la disciplina. No puede haber movimiento de más —nada extraño—, sólo aplicación concentrada de capacidad humana. El desperdicio es todo lo que te distrae de eso. Si empiezas a concebir el trabajo en términos de disciplina y flujo bien podrías hacer algo increíble.

RESUMEN

La multitarea te embrutece. Hacer más de una cosa al mismo tiempo te vuelve más lento y peor en ambas. *No lo hagas*. Si crees que esto no se aplica a ti, te equivocas.

Hacer a medias es no hacer. Un auto a medio hacer utiliza recursos que podrían usarse en crear valor o ahorrar dinero. Cualquier cosa «en proceso» cuesta dinero y energía sin rendir nada.

Hazlo bien a la primera. Cuando cometas un error, corrígelo al instante. Deja lo demás y ocúpate de él. Corregirlo después puede consumir veinte veces más tiempo, o más, que si lo corriges ahora.

Trabajar demasiado sólo complica las cosas. Trabajar mucho tiempo no permite hacer *más*, sino *menos*. Resulta en fatiga, lo que induce errores y esto te obligará a corregir lo que acabas de hacer. Más que trabajar hasta tarde o los fines de semana, hazlo entre semana a un ritmo sostenible. Y tómate unas vacaciones.

No seas irracional. Las metas desafiantes motivan, las imposibles deprimen.

No a los actos heroicos. Si necesitas que un héroe haga las cosas, tienes un problema. El esfuerzo heroico debe entenderse como un error de planeación.

Basta de políticas absurdas. Toda política que *parece* ridícula probablemente lo *es*. Los formularios, reuniones, aprobaciones y normas absurdos son sólo eso: absurdos. Si tu oficina parece una caricatura de *Dilbert*, ponle remedio.

Fuera idiotas. No lo seas, ni permitas esa conducta. Quienquiera que cause caos emocional, inspire miedo o temor o degrade o subestime

debe ser parado en seco.

Busca el flujo. Opta por la manera más tersa y sin contratiempos de hacer las cosas. Scrum consiste en permitir el mayor flujo posible.

Capítulo seis

Planea realidades, no fantasías

«Jeff, tenemos un problema».

Así comienzan muchas de mis conversaciones telefónicas. Cuando se siente acorralada, la gente toma el teléfono para llamarme. Esta vez era Mark Landy, jefe de arquitectura de *software* en Medco. Muchos estadunidenses que surten sus recetas médicas por correo tratan con esta empresa. Cuando recibí esa llamada, Medco era una compañía de *Fortune 100* con ingresos por treinta y ocho mil millones de dólares, la farmacéutica más grande de Estados Unidos, con decenas de miles de empleados, a quienes la dirección acababa de poner al borde del precipicio.

Recibí dicha llamada en diciembre de 2006. En julio de ese mismo año, el presidente de Medco, Kenny Klepper, había anunciado a Wall Street su más reciente idea. Mark Landy la describió así: «Queremos convencer a cada vez más personas de optar por recibir sus recetas por correo. Pero algunas barreras nos lo impiden», como la impresión de inconveniencia. Pero, me dijo Landy, había formas de salvar esas barreras. «Cuando vas a una farmacia tu experiencia es muy poco clínica. Entregas tu receta, firmas un documento que dice que no quieres consultar al farmacéutico y te vas. Nosotros podemos mejorar esa experiencia».

Una de las cosas que querían hacer era poner a contestar teléfonos a farmacéuticos que no sólo conocieran las medicinas referidas, sino también todas las recetadas a un paciente específico. Esto era particularmente importante si el paciente padecía una afección crónica como diabetes o enfermedades cardiacas, justo el caso de ochenta por ciento de las personas bajo tratamiento médico regular. Y la mayoría de estas personas —sobre todo si son de edad avanzada— toman seis o más medicinas al mismo tiempo, algo que sus médicos —especialistas de diferentes campos de la salud— no siempre saben.

«Los médicos no [siempre] comparten información entre sí. Pero como nosotros somos la farmacia, sabemos más que ellos y en tiempo real, [aun] antes que el seguro médico», me dijo Landy.

Ésta era entonces la idea de Klepper: «Pongamos farmacias especializadas en cinco puntos del país: la farmacia de lo cardiaco, lo diabético, lo asmático, etcétera, y capacitemos a farmacéuticos asignados a esos sitios para que conozcan las interacciones entre medicamentos, efectos secundarios, etcétera. Como estos farmacéuticos tendrán una visión completa de la condición del paciente, podrán informar a los médicos de posibles contraindicaciones. Es probable que un diabético tenga sobrepeso y problemas hepáticos. Así, metabolizará sus medicinas de otra manera. Si un médico nuevo le receta algo para la presión, el farmacéutico de Medco podría llamarlo para recomendar una revisión hepática del paciente y un probable ajuste de su dosis».

La meta era atraer nuevos clientes a Medco, que atendía sobre todo a empresas y seguros médicos. Gracias a esas nuevas farmacias, o Centros de Recursos Terapéuticos, los clientes ahorrarían dinero, no necesariamente reduciendo el costo de sus recetas, sino sus costos médicos generales, que aumentan cuando no se toman medicamentos en forma apropiada o éstos no interactúan bien entre sí, al menos en una persona en particular. Más todavía, Medco *garantizaría* esos ahorros. Si un cliente no ahorraba la cantidad proyectada por la compañía, ésta aportaría la diferencia.

Para decirlo suavemente, a Wall Street *le gustó* la idea. Muy buena, ¿no? Ahorrar dinero y ofrecer mejor atención a la salud. Más clientes, más ventas. Beneficio mutuo. Sólo que había un pequeño problema. Aunque Klepper había consultado a sus gerentes para confirmar que la idea era técnicamente posible, no había obtenido detalles sobre cuánto *tardaría* la implementación del plan. La gente que lo haría realidad se enteró de él luego de que el presidente de la compañía prometiera a Wall Street que, a como diera lugar, el nuevo sistema estaría en operación el 7 de julio de 2007.

Cumplir esa fecha era de suma importancia para Medco, porque aunque había sido la primera empresa en ofrecer farmacias automatizadas de pedidos por correo, no era en absoluto la única y sus competidores estaban muy ansiosos. Por desgracia, Medco tenía que superar muchos obstáculos. Por ejemplo, gran parte del *software* del que dependía para dirigir sus robots era muy anticuado. En las cinco enormes plantas de Medco, ocupadas por cuatro mil farmacéuticos que procesaban recetas, los robots se hacían cargo de recoger píldoras mientras otros las envasaban y mandaban por correo, y todos

estos sistemas tenían que entenderse entre sí con una precisión total, pues de lo contrario alguien podía morir.

La idea era que el ambicioso plan de Klepper permitiera a Medco poner al día sus añejos sistemas y mantenerse un paso adelante de sus competidores. Pero la compañía tardaría seis meses en saber que no podría ejecutar a tiempo ese plan. Sus cálculos indicaban que, en el mejor de los casos, tendría listo el sistema al menos un año después de lo previsto, tal vez más. Fue entonces cuando me llamaron.

Por qué tardaron *seis meses* en saber que no podrían terminar a tiempo es algo que merece consideración. No fue porque no hayan sido listos o no dispusieran de los equipos indicados o la tecnología correcta. Tampoco porque no trabajaran con ahínco o no fueran competitivos. Medco no habría podido ser la compañía más grande de su sector si hubiera padecido esas deficiencias.

Fue porque cometió un error básico. Creyó que podía planearlo todo. Dedicó meses de esfuerzo a hacer detallados planes que parecían convincentes, se tradujeron en bonitas gráficas e incluían pasos muy precisos que por lo general describían una ficción, no la realidad.

Como ya dije, el acto mismo de planear es tan seductor, tan atractivo que la planeación termina siendo más importante que el plan y éste más importante que la realidad. Jamás olvides que el mapa *no* es el terreno.

Cuando un equipo se reúne para planear un proyecto suele haber electricidad en la sala: una sensación de posibilidad, de mundos nuevos por descubrir y nuevas ideas con que experimentar. Ésta es ciertamente una de las sensaciones más intensas en la vida.

Pero entonces llega el momento en que la inspiración se vuelve cálculo y parte de esa energía se disipa. La gente comienza a ponderar: «¿Cómo podemos llegar del punto *A* al punto *B*? ¿Y cuánto tiempo implicará hacerlo?».

Lamentablemente, esta fase de cálculo puede ser un proceso de entra basura/sale basura. Los involucrados pueden ser muy inteligentes, pero rara vez *se dan cuenta* de que lo que incluyen en sus gráficas suelen ser meras ilusiones.

Cuando Mark me terminó de explicar la situación de Medco repuse: «¡Vaya que tienen un problema!». Luego de una pausa, añadí: «Pero apuesto que podemos resolverlo».

Justo antes de Navidad volé a Nueva Jersey para pasar un día en la empresa a fin de determinar el alcance del problema. No era trivial. Había

alteros de hojas en las que se esbozaban requerimientos, requisitos legales, todo tipo de informes, fases-puertas y aseguramiento de la calidad. En alguna parte yacía oculto lo que en verdad había que hacer, pero nadie tenía un plan de cómo *llevarlo a cabo*.

Tras reunirme con el personal clave, llamé a Brent Barton, instructor de Scrum con quien ya había trabajado en otros proyectos. «Brent», le dije, «te necesito y a quien puedas conseguir para principios de enero. Tenemos una misión justo a nuestra medida».

Más tarde Barton describiría la Medco que halló al llegar como una compañía «estancada». Había tantos intereses y personas en pugna que no se hacía nada. El primer día nos reunimos con siete grupos, cada uno de ellos a cargo de una parte del proyecto, y ninguno mostró interés en probar algo nuevo. Sin embargo, dice Barton ahora, «podíamos darnos el lujo de decir: "¡Al diablo!". Como consultor puedes usar el miedo como aliado. Cuando topábamos con alguna resistencia, simplemente decíamos: "Está bien; sigan haciendo las cosas a su manera, dejen todo como está y entreguen tarde". Pero ellos replicaban: "No, no está bien"».

Lo primero que hicimos fue reunir a todos los actores clave, a toda la gente que estaría a cargo del trabajo en una sala de juntas. Barton les había pedido imprimir todos los documentos que describieran las actividades del proyecto. No queríamos correos electrónicos, sino documentos en papel.

Confluimos en una sala grande, de quince metros por lado y sin ventanas, como misteriosamente parecen ser siempre este tipo de salas. En medio había una mesa en la que amontonamos todos los documentos que la gente llevó. La pila era de al menos sesenta centímetros de alto.

«¿Cuántos de ustedes han *leído* esto?», pregunté.

Silencio.

«Usted firmó este documento», dije a uno de los gerentes. «Tiene su firma. ¿No lo leyó?».

Incómodo silencio otra vez.

Yo no quería cebarme en él, pero es un hecho que, en un proyecto tras otro, la gente corta y pega texto en documentos, pero nadie lee esos miles de páginas. No puede. Ése es el asunto: la gente ha establecido un sistema que la obliga a aceptar una fantasía.

Barton y yo sacamos tijeras, cinta adhesiva, barras de pegamento y papeletas adhesivas. Todo lo que debes saber para *hacer esto* lo aprendiste en el jardín de niños.

«Esto es lo que haremos», dijo Barton, «revisaremos estas pilas de papel, recortaremos todo lo que debe hacerse para ejecutar el proyecto y lo pegaremos en la pared».

Un par de horas más tarde, cientos y cientos de papeletas cubrían las tres paredes de la sala. En la mesa había quedado más de cincuenta por ciento de aquella torre de sesenta centímetros. Duplicación, plantillas, boletines. Total y absoluto desperdicio.

Dije entonces a los equipos: «Ahora debemos calcular cuánto trabajo implicará cada una de estas papeletas». No cuánto tiempo, sino cuánto *trabajo*.

Más adelante detallaré las mejores maneras de hacer esto, ya que los seres humanos somos pésimos para calcular cargas de trabajo. En ese momento enseñé a los equipos un método rápido y desaliñado, la mejor de las malas maneras de hacerlo y ellos lo siguieron.

Un buen rato después, en la pared estaba todo lo que debían hacer para llevar a cabo el proyecto, dividido en tareas manejables. Y habían calculado cuánto esfuerzo implicaría cada una. Estaban emocionados. Una ilegible pila de papel se había convertido en labores comprensibles. Como reza el viejo refrán: «¿Cómo puedes comerte un elefante? Una mordida tras otra».

Un aspecto clave que hicimos con cada papeleta fue escribir no sólo la actividad por realizar, sino también cómo sabríamos que estaba terminada. Así incorporamos todos los requisitos de la FDA (Food and Drug Administration, Dirección de Alimentos y Medicinas), el aseguramiento de la calidad y los reportes de procesos que los equipos debían cumplir. Simplemente establecimos que, para completar una tarea específica, tenían que alcanzarse tales o cuales metas. Incorporamos esto al proyecto en el nivel de las tareas, a fin de no esperar a que todo estuviera terminado para descubrir que no habíamos cumplido un reglamento federal o una medida interna de calidad. De esta manera, todos los miembros del equipo, no sólo el personal de requisitos, tenían que satisfacer tal nivel de calidad antes de pasar a la tarea siguiente. Así se elimina en un proyecto una increíble cantidad de retrabajo. Llamo a una norma por ser cumplida «Definición de terminado». Todos saben si algo se acabó o no; hay normas claras que cualquier tarea debe satisfacer.

Al ver tantas papeletas en la pared todos tuvieron una sensación de logro. Por fin sabían qué tenían que hacer.

«Bueno», dijo Barton, «¿qué deberíamos hacer primero?».

Hablaron unas cinco personas.

«¿Y después?».

Otras cinco con ideas diferentes.

«¿Y luego?».

Se trataba de que hicieran algo que a veces nadie quiere hacer: *enumerar el trabajo en orden de prioridad*. La gente suele decir que todo es importante. Pero lo que Barton preguntó fue: «¿Qué es más valioso para el proyecto? Hagamos eso primero».

Al final teníamos seis filas de papeletas en las paredes, cada una de diferente color en representación de un equipo distinto. Las listas cubrían las tres paredes de la sala. Supe entonces que al fin podríamos empezar.

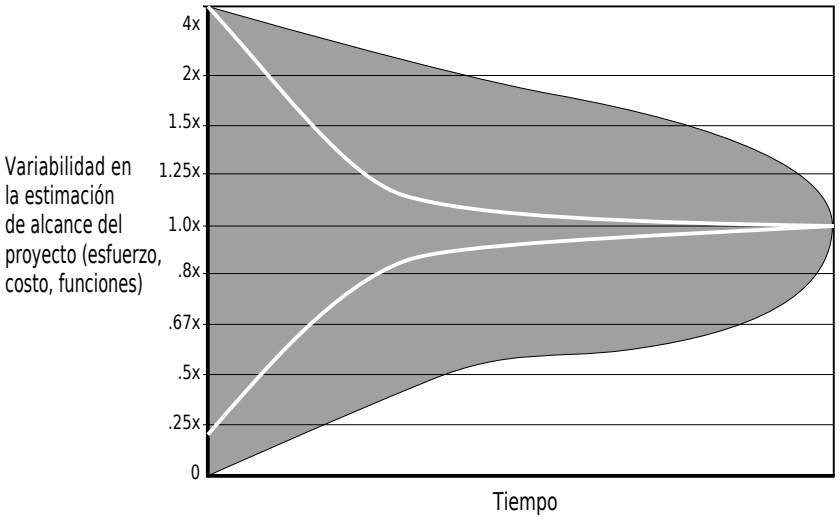
Planeación de una boda

Parecerá demasiado simplista, pero permíteme ilustrar los pasos de este proceso sirviéndome de un ejemplo de menor escala: una boda. Una boda formal es un proyecto con muchas actividades por cumplir en una fecha particular; y como sabes —o sabrás, si decides hacerlo—, si ya estás casado todo saldrá mal y absorberá cuatro veces más el esfuerzo que previste.

Claro, también puede ser al revés: algo que pensaste que llevaría horas podría despacharse en quince minutos. La pregunta acuciante es: ¿por qué somos tan malos para calcular cuánto tiempo consumirá algo?

¡Y vaya que somos malos! Volveremos a esa boda más adelante, antes déjame presentarte una gráfica con uno de los mejores nombres en la historia de los diagramas, el «Cono de incertidumbre».

EL CONO DE INCERTIDUMBRE



Esta gráfica indica que las estimaciones iniciales del trabajo pueden ir de cuatrocientos por ciento más de lo previsto a setenta y cinco por ciento menos, un rango de error de ocho veces. Conforme el proyecto avanza y más cosas se resuelven, las estimaciones se ajustan cada vez más a los hechos, hasta desaparecer y dejar únicamente la realidad.

Regresemos a Medco. Dedicó meses a planear sus esfuerzos: cómo sería el producto, cuánto tiempo absorbería. Pero aun después de todo es probable que, como demuestran las investigaciones, haya errado en hasta un factor de cuatro en cualquier dirección. Por eso soy de la opinión de que la planeación tipo cascada es una manera absurda de hacer las cosas.

«Está bien, Sutherland», puedo oírte decir, «somos muy malos para estimar, pero tengo que hacer *algo*, ¿de acuerdo? Debo tener un plan». Tienes razón. Pero la clave es ir afinando el plan *a lo largo* del proyecto, no dejarlo terminado desde el principio. Planea en detalle apenas lo suficiente para cumplir el próximo incremento de valor y calcula el resto del proyecto en líneas generales. En Scrum, al final de cada repetición tienes algo valioso que puedes ver, tocar y mostrar a los clientes. Puedes preguntar a estos últimos: «¿Es esto lo que quieren? ¿Resuelve al menos una parte de su problema? ¿Vamos en la dirección correcta?». Si la respuesta es no, cambia tu plan.

¿Cómo?

Volvamos a la boda. Lo primero por hacer es elaborar una lista de todo lo que compone una boda exitosa. Podría ser ésta:

- Novia y novio
- Flores
- Invitaciones
- Iglesia
- Salón para la recepción
- Comida
- Oficiante
- Vestido
- Anillos de bodas
- Música (DJ o en vivo)

Lo siguiente es ordenar por prioridad esos elementos, lo cual depende de cada quien. Cada novia y novio ven el mundo a su manera. Pero el otro día pregunté a mi amigo Alex cómo había ordenado su lista y hela aquí:

- Novia y novio
- Oficiante

- Anillos de bodas
- Salón para la recepción
- Invitaciones
- Comida
- Música
- Vestido
- Flores
- Iglesia

El objetivo de este ejercicio es determinar cuáles son las cosas más importantes y ocuparse primero de ellas. Para Alex, la comida y la música tenían mayor relevancia que casarse en una iglesia o las flores. Es esencial disponer de estos datos, porque si de repente comienzas a chocar con la fecha o con restricciones de costos sabes dónde empezar a cortar: al final de la lista. Abundaré en este tema en el capítulo 8.

En Medco, la lista cubría las tres paredes de una inmensa sala de juntas y constaba de cientos de tareas a cargo de seis equipos. Pero el concepto era el mismo: organiza por valor, cualquiera que éste sea: valor comercial en el caso de Medco, valor de la felicidad de la novia en el de una boda.

El tamaño importa, pero sólo relativamente

Una vez en poder de tu lista de actividades ordenada por prioridad debes saber cuánto esfuerzo, tiempo y dinero implicará el proyecto. Como ya señalé, los seres humanos somos pésimos para esto, pero buenos para la evaluación *relativa*, consistente en comparar un tamaño con otro. Piensa, por ejemplo, en cómo distinguir entre camisetas chicas, medianas y grandes.

Mi ejemplo favorito de evaluación relativa es el de «Puntos perros». Hace unos años, mi amigo Mike Cohn, una de las principales figuras del pensamiento Agile, tuvo que vérselas, como yo, con qué hacer para que sus proyectos estuvieran listos a tiempo y dentro del presupuesto, y cómo calcularlos. Amante de los perros, pese a lo cual su esposa le prohibía tener un solo can, dio en preguntar a sus equipos a qué «perro» correspondía por su tamaño cada parte de un proyecto. Enumeraba muchas razas, como éstas:

- Labrador
- Terrier
- Gran danés

- Poodle
- Salchicha
- Pastor alemán
- Setter irlandés
- Bulldog

Y luego preguntaba: «¿Este problema es un salchicha o un gran danés? Si es un salchicha, este otro debe ser un labrador, ¿no?». Los equipos revisaban así todas las funciones por desarrollar y evaluaban de qué tipo de perro se trataba. Mike decía entonces: «Demos a cada raza un valor numérico, para que sea más fácil. Asignemos al salchicha el uno y al gran danés el trece. Por tanto, el labrador será el cinco y el bulldog el tres»[31].

Tú podrías hacer lo mismo con la lista de pendientes que acabamos de elaborar para una boda. Hallar un buen salón implicará un poco de investigación, informarse de precios, visitar lugares. Es algo complicado, así que considerémoslo un problema tamaño pastor alemán, un cinco. ¿Novia y novio? Sencillo: los dos tienen que presentarse. Esto es un salchicha, un uno, porque para hacerlo bastará una llamada telefónica. Las invitaciones, en cambio, son muy complicadas. Hay que hacer la lista de los invitados de los novios, conseguir las de sus madres, elegir el papel e imprimir y rotular las invitaciones. Es mucho trabajo, un gran danés, trece. O quizá dos gran danés. Si es muy grande, tal vez debas dividirlo en piezas manejables, como conseguir los nombres de los invitados por una parte y tratar con el impresor por la otra. Cada uno de estos proyectos es de tamaño bulldog, ¿no?, un tres. La rotulación sería un pastor alemán, cinco, y así sucesivamente.

Esto es evaluación relativa, comparar tareas entre sí. No todos usamos perros para hacerlo, pero quizá notaste un patrón en los números que yo asigné: 1, 3, 5, 8, 13. Cada número de esta serie es la suma de los dos anteriores. Este patrón se conoce como *serie de Fibonacci* y hay una razón de que la utilicemos: está presente en todo.

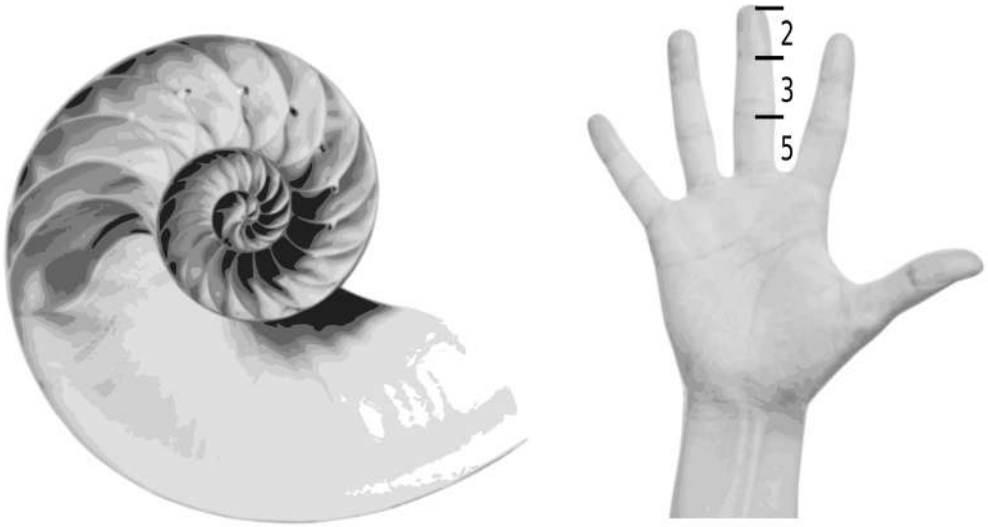
Tal serie reproduce la forma en que está dispuesta la naturaleza, sea en la concha de un molusco, las ramas de un árbol, las protuberancias de una piña o los pétalos de un cono de pino. Aparece en la coliflor y en las circunvoluciones del cerebro humano. Es igual así examines el rizo de una hoja de helecho o la forma de una galaxia. Es uno de esos fenómenos que, al pensar en ellos, resultan sumamente extraños.

Este fenómeno tiene un nombre: sección o proporción áurea. Los humanos la hemos incorporado en edificios y en el arte, del Partenón de Atenas a la Gran Mezquita de Kairuán, Túnez. La hemos usado para decidir la

forma y el tamaño de las páginas de un libro y las proporciones de los naipes. Estamos programados para gustar de las proporciones. Para los propósitos de este libro, baste saber que nuestra especie comprende profundamente las proporciones de la serie de Fibonacci. Las llevamos en los huesos.

Serie de Fibonacci: Por todas partes

- La serie de Fibonacci es un patrón en el que el número siguiente de la serie es la suma de los dos anteriores, por ejemplo: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55…
- Ubicua en los sistemas naturales, así que los seres humanos tenemos milenios de experiencia con ella.



Los números de la serie de Fibonacci están lo bastante separados entre sí para advertir fácilmente su diferencia. Es fácil optar por un extremo u otro. Si alguien asigna a algo un cinco y otro un ocho, vemos intuitivamente la diferencia. ¿Pero entre un cinco y un seis? Esto es muy sutil, más de lo que nuestro cerebro puede registrar.

En la medicina es bien sabido que para que un paciente pueda reportar una mejora en un síntoma debe ser de más de sesenta y cinco por ciento. Nuestra mente no funciona en incrementos leves. Percibimos mejor los saltos de un estado a otro y no saltos tenues, sino marcados.

Usar la serie de Fibonacci para calcular el tamaño de tareas permite hacer estimaciones que no sean cien por ciento exactas. Nada es precisamente un cinco, ocho o trece, pero emplear estos números nos brinda la posibilidad de

recoger opiniones sobre el tamaño de una tarea usando el mismo rasero y, por tanto, de formar un consenso.

Calcular grupalmente de esta manera nos ofrece una estimación más precisa que la que podríamos obtener en forma individual.

El oráculo de Delfos

Ahora ya sabemos que somos buenos para comparar cosas y cuál es la mejor proporción por utilizar en esa tarea. Pero ¿cómo realizarla? Una lista de actividades en orden de prioridad es muy útil, pero ¿cómo saber qué historia representa un 5 y cuál un 8; cuál un goldie y cuál un schnauzer? Y aun si alguien tiene una buena idea, ¿cómo confirmar que sus estimaciones concuerdan con las de los demás? ¿Qué tal si no ha tomado en cuenta algunos factores clave?

No es de sorprender que éste no sea un problema nuevo. La gente se las ha visto con él desde hace décadas. Un inconveniente es que miembros diferentes de un equipo saben cosas diferentes, pero otro se conoce como el «efecto tren». Sin duda has estado en reuniones en las que éste ocurre. Alguien propone una buena idea y todos hablan de ella. Y aun si al principio la rechazas, terminas por aceptarla porque el grupo lo hace. Todos aprueban un curso por seguir porque en ese momento parece muy buena idea, pero luego resulta ser un completo fracaso. Y cuando sondeas a la gente sobre esa decisión, suele suceder que todos tenían sus reservas, pero que no las expresaron porque vieron que los demás estaban muy entusiasmados. La gente supone que si los demás aceptan algo sus dudas son absurdas o están mal informadas y no quiere parecer tonta frente al grupo. El pensamiento grupal no es una falla individual, sino humana.

En la bibliografía especializada este efecto se ha explicado como «cascada informativa». Tal como afirman Sushil Bikhchandani, David Hirshleifer e Ivo Welch, autores del artículo «A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades» (Una teoría de la moda, la costumbre y el cambio cultural como cascadas informativas): «Una cascada informativa ocurre cuando, habiendo observado los actos de quienes lo preceden, un individuo juzga óptimo seguir la conducta del anterior sin considerar su propia información»[32].

Un ejemplo excelente dado por estos autores es la presentación de un artículo para su publicación en una revista. Supongamos que un primer editor

lo rechaza, tras de lo cual el autor lo propone a otra revista. Enterado del primer rechazo, es muy probable que el editor de esta última lo rechace también. Y si hay una tercera, su editor, al tanto de los dos rechazos anteriores, tiene aún más probabilidades de hacer lo propio. La gente parte del supuesto de que los demás emiten juicios razonables, *aun si contradicen los suyos propios*. Esto no está bien. Cuando te formas una idea de cuándo entregarás un proyecto multimillonario —o si terminarás todo a tiempo para el día de tu boda—, es crucial que uses tu propio criterio y que emplees otras estimaciones para mejorar la tuya, no para remplazarla.

Otro problema frecuente es lo que se conoce como «efecto halo». Tal cosa ocurre cuando un rasgo de algo influye en la forma en que la gente percibe otros rasgos no asociados con aquél. Esto fue empíricamente estudiado por primera vez en 1920 por Edward Thorndike. En su artículo clásico «A Constant Error in Psychological Ratings» (Un error constante en evaluaciones psicológicas), Thorndike pidió a oficiales calificar a sus soldados según varias cualidades, físicas, intelectuales, de liderazgo, de personalidad, etcétera, tras de lo cual examinó cómo una serie de ellas afectaba la calificación de otra, al grado de correlacionarse estrechamente entre sí. Si el físico de alguien merecía una alta calificación, lo mismo ocurría con sus habilidades de liderazgo. Y con su inteligencia. Y con su carácter. Esta investigación ha sido confirmada por estudios complementarios a lo largo de los años, corroborando así que, por ejemplo, si alguien es bien parecido, todos suponen que también es listo y digno de confianza[33].

Pero el efecto halo se extiende más allá de la belleza física; puede aparecer en cualquier parte. Algunos investigadores han señalado que, por ejemplo, las organizaciones no gubernamentales (ONG) suelen ser tratadas como fuerzas para el bien aun si no lo son, que compañías automotrices producirán un auto «halo» para dar a una línea entera una buena impresión, o que el iPod de Apple concedió a todos los productos de Apple una pátina de fábula.

Como en el caso del efecto tren, la gente que atiende al «halo» no examina los datos reales; más bien, gravita hacia algo con lustre positivo. Tampoco ésta es una falla de la voluntad; nuestra naturaleza es así. Combatirla de frente es absurdo; sería como combatir la gravedad.

Sin embargo, tú puedes enfrentar esto con inteligencia. En la década de 1950 se pidió a la Rand Corporation contestar ciertas preguntas aterradoras, propias de la guerra fría. Invocando en su terminología el oráculo de Delfos, la sacerdotisa que predecía el futuro, Norman Dalkey y Olaf Helmer

publicaron en 1963 un artículo insulsamente titulado «An Experimental Application of the Delphi Method to the Use of Experts» (Una aplicación experimental del método de Delfos para el uso de expertos), con la útil referencia «Memorandum RM-727/1-Abridged» (Memorándum RM-727/1-Abreviado). En ese artículo declararon su intención de hacer preguntas evitando que las opiniones de una persona afectaran las de otra. Así, reunieron a un grupo de expertos: cuatro economistas, un especialista en vulnerabilidad física, un analista de sistemas y un ingeniero eléctrico. Y se propusieron aplicar la opinión de expertos a la selección, desde el punto de vista de un planificador estratégico soviético, de un sistema industrial estadunidense como objetivo óptimo y a la estimación del número de bombas A requeridas para reducir la producción de municiones en un monto prescrito[34].

O para decirlo en términos llanos: la idea era preguntar cuántas armas nucleares necesitaban los rusos para impedir a los estadunidenses hacer las suyas. Esto sucedía en un momento en que un conflicto nuclear se juzgaba no sólo posible, sino también factible de ganar.

La cuestión es que ni Dalkey ni Helmer querían que sus expertos se influyeran entre sí. ¿Qué tal si uno era jefe de departamento en una gran universidad y otro un profesor modesto de un pequeño instituto tecnológico? ¿Cómo impedir que los falsos supuestos de una persona estropearan las opiniones de otra?

Para tal efecto, aquellos dos investigadores hicieron una serie de encuestas anónimas. Ningún experto sabía quiénes eran los demás; sólo tenían que entregar sus estimaciones. Luego de cada cuestionario, el dúo investigador tomaría las respuestas —y los datos en que éstas se basaban— y las haría circular en el grupo sin ningún detalle que revelara la identidad de los participantes. Remoje y repita.

Así, del primer cuestionario resultó que el número de bombas necesarias para alcanzar una certeza de cincuenta por ciento en la destrucción de la industria armamentista estadunidense iba de cincuenta a cinco mil. Cuando Dalkey y Helmer analizaron las respuestas creyeron hallar aspectos comunes en las estimaciones: la vulnerabilidad de varios blancos, la capacidad de recuperación de diversas industrias, las reservas iniciales, etcétera. Preguntaron entonces a los expertos si ese desglose era correcto y qué otra información habían empleado para contestar.

Obtuvieron toda clase de respuestas, desde la solidez de las fábricas hasta la diferencia entre vulnerabilidad física y económica y el plazo óptimo de

manufactura de varios componentes.

Dalkey y Helmer circularon estos datos entre los expertos y preguntaron: «¿Cuántas bombas se necesitan?». Esta vez la escala fue de entre ochenta y nueve y ochocientas. Después repitieron el procedimiento y luego una vez más. Los resultados eran cada vez más cercanos entre sí. La escala se redujo al final a entre ciento sesenta y siete y trescientas sesenta armas nucleares.

Reducir una escala de estimaciones extremadamente amplia de diez mil a doscientos por ciento es una herramienta muy eficaz para los responsables de políticas públicas. Les permite obtener un consenso entre expertos sin preocuparse por los sesgos. Esta herramienta es tan eficaz que Rand sigue usándola hoy en día. Como ejemplo reciente está un ejercicio Delfos de 2011 para analizar el conflicto en Afganistán y calcular las posibilidades de éxito de Estados Unidos. Por si te interesa, las perspectivas no resultaron buenas.

Póker de planeación

La ventaja de Delfos es que toma una amplia serie de opiniones, intenta eliminar todos los sesgos posibles y con juicios informados, pero anónimos, reduce las opiniones a una estimación generalmente aceptada. Lo malo, para nuestros propósitos, es que el procedimiento es muy lento. Cuando me reuní con los equipos de Medco no había tiempo para encuestas anónimas. Aquellos cientos de tareas tenían que evaluarse en horas, no en días, menos aún en semanas.

Por fortuna, hay una manera rápida y exacta de reunir estimaciones. Se llama Póker de planeación.



La idea es muy sencilla. Cada persona recibe un mazo de naipes con los números de Fibonacci: 1, 3, 5, 8, 13, etcétera. Luego, cada tarea por calcular se pone sobre la mesa. Todos tiran entonces la carta que creen que representa la cantidad de esfuerzo correcta y la ponen bocabajo. Después, todos las voltean al mismo tiempo. Si la secuencia resultante es de números sucesivos (como un cinco, dos ochos y un trece), el equipo los suma, saca el promedio (en este caso, 6.6) y pasa a la tarea siguiente. Recuerda que hablamos de *estimaciones*, no de cálculos rígidos. Y de estimaciones sobre partes reducidas del proyecto.

Si la distancia entre dos números es de más de *tres* cartas, quienes tiraron los naipes respectivos explican sus razones, tras de lo cual se juega otra ronda de póker de planeación. De lo contrario, se promedian las estimaciones, lo que dará números aproximados a los que dieron los estadísticos en la Rand Corporation.

He aquí un ejemplo: supongamos que estás pintando una casa y debes calcular el tiempo que llevará pintar la sala, la cocina y dos alcobas. Haces esto con un equipo con el que ya has pintado habitaciones antes. Primero evalúan las dos recámaras: todos calculan un tres. No hay desacuerdo; todos han hecho esto antes y juzgan muy sencillas las alcobas. Después, el equipo evalúa la sala. Es una habitación grande pero simple; los cálculos van de cinco a trece, con un promedio de seis. Tampoco esta vez es necesario

discutir. A la cocina se le asigna un tres, un ocho, un trece y un cinco. Quien tiró el tres alega que es una habitación pequeña, con menos espacio de pared que las recámaras. Quien tiró el trece replica que lo que consumirá más tiempo será cubrir gabinetes y mostradores y que para pintar las pequeñas áreas resultantes tendrán que usar brocha, no rodillo. El equipo lanza nuevas cartas. El tres ha pasado a ser ocho y los demás mantienen sus números. Éstos están cerca entre sí, se suman y promedian y se pasa a la tarea siguiente.

Este sencillísimo método permite evitar todo condicionamiento, como los efectos tren y halo, y a la vez que compartir entre los miembros del equipo conocimientos sobre una tarea particular.

Es crucial, sin embargo, que la estimación corra a cargo del equipo que llevará a cabo el trabajo, no de expertos «ideales».

Aprendí esto a la mala cuando trabajé para GSI Commerce, empresa de comercio electrónico en Pennsylvania más tarde adquirida por eBay. GSI diseña las tiendas en línea de compañías como Levi’s, Toys «R» Us, Major League Baseball y Zales Diamonds. No son proyectos modestos. Y GSI es muy buena para esto.

Pero tuvo la idea, que en su momento pareció espléndida, de que en vez de que cada equipo hiciera sus cálculos, la tarea se asignara a los mejores estimadores de la compañía (los más listos del salón), quienes conocían los proyectos y la tecnología y sabían qué había que hacer. Así, esos expertos evaluaron algunos proyectos: éste debía tardar tanto, aquel otro tanto más y así sucesivamente. El plan era evaluar ochenta proyectos multimillonarios, tanto para los clientes como para los equipos a cargo del trabajo. Parecía razonable, ¿no?

Pero distó tanto de serlo que el experimento se detuvo a medio camino, tras evaluar cuarenta proyectos. Esto me recordó aquellos estudios sobre medicinas que se cancelan porque éstas matan a los pacientes en vez de curarlos. Las estimaciones eran tan inexactas que resultaron inútiles. Nada se entregó a tiempo. Los clientes estaban molestos. Los equipos, desmoralizados. Un desastre absoluto. Los gerentes pidieron entonces a los equipos de trabajo hacer las evaluaciones. Y, ¡quién lo iba a decir!, esta vez los cálculos fueron más acordes con la realidad.

Lo que aprendí de esto fue que sólo las personas a cargo del trabajo saben cuánto tiempo y esfuerzo implica éste. Quizá su equipo sea muy bueno en algo pero pésimo en otra cosa. Tal vez dispongan de un experto útil en un área particular, pero ningún miembro conozca un área diferente. Como ya

dije, cada equipo es único; cada uno posee un ritmo propio. Obligarlos a adoptar procesos uniformes es una receta para el desastre.

No hay tareas, sólo historias

Cuando enumeras cosas por hacer es tentador elaborar una lista como la de la boda de Alex: iglesia, flores, oficiante, comida, etcétera. El problema es que si asignas cualquiera de esos elementos a un equipo no íntimamente involucrado en las consecuencias de las decisiones entre rosas blancas y margaritas podrías no obtener los resultados que buscas.

¿Cuántas veces se te ha encargado en el trabajo hacer algo cuya razón no entiendes? Alguien te pide determinar cuánto cambiaron las ventas de un mes a otro en la región *A* examinando las tiendas de más de cincuenta y cinco metros cuadrados. Lo haces, pero sin saber *para qué*. Por eso mismo podrías proporcionar el tipo de datos equivocado, malinterpretar la cuestión o molestarte por recibir una encomienda tan pesada. O si eres el gerente, podría asombrarte que tu personal no capte de inmediato que estás pensando cerrar las tiendas chicas para abrir otras grandes.

El problema es que no recibes o das información suficiente para hacer bien un trabajo. Los seres humanos pensamos en forma de relatos, de anécdotas. Es así como comprendemos el mundo. Poseemos un entendimiento íntimo de personajes, deseos y motivaciones. Pero nos metemos en problemas cuando intentamos extraer partes específicas de la línea argumental principal y tratarlas fuera de contexto.

Lo primero en lo que debes pensar al considerar una tarea es en el personaje o papel, como un cliente, una novia, un lector, un empleado. ¿Para quién se hará esta tarea? ¿Por el cristal de quién debemos ver el mundo al hacer este trabajo, tomar esta decisión o terminar esta parte?

Luego debes pensar en el *qué*: qué quieres hacer en primer término. Por lo general nos detenemos en este punto, pero esto es apenas la mitad del proceso.

Por último, debes pensar en la motivación. ¿*Por qué* esta persona quiere tal cosa? ¿De qué servirá esto y cómo complacerá a este cliente en particular? En cierto modo, ésta es la clave. La motivación lo colorea todo.

Mi ejemplo favorito al respecto es un meme de internet de hace unos años: una foto del capitán Jean-Luc Picard, del USS *Enterprise*, que rezaba al pie: «Como capitán de una nave espacial, me gustaría que la función de diario

de navegación aplicara la fecha en forma automática». Si lo piensas bien, esto tiene sentido. ¿Nunca te has preguntado por qué, en el futuro remoto, el capitán de una nave espacial tendría que decir la fecha al hacer una entrada en su diario? «Diario del capitán. Fecha: 4671.7 El planeta Marte es hermoso desde el espacio…» *Nosotros* no tenemos que hacerlo cuando elaboramos una entrada de blog. ¿Por qué él sí?

Pero la pregunta clave que esa foto no responde es *por qué*. ¿Por qué Picard quiere esa funcionalidad? ¿Para qué servirá? ¿Sólo para mantener las entradas en orden cronológico o para algo más serio? ¿Esos diarios deben ser inalterables para servir con fines de auditoría a investigadores de escena del crimen de la armada espacial? Estas dos implementaciones son muy distintas, una muy simple, otra robusta. El equipo debe saber qué quiere hacer, momento en el cual podría pensar en hacerlo de otra manera, con información más relevante en la que quizá el capitán ni siquiera ha pensado pero que sería muy útil.

A menudo, las necesidades cambiarán según la persona involucrada. Imagina una historia con este final: «… Necesito un auto para ir a trabajar». Si esta frase empezara con «Puesto que todos los días tengo que hacer un largo viaje a la oficina…» y no con «Como agricultor en la zona desértica de Dakota del sur…», acabarías con una interpretación muy distinta acerca de cuál es el vehículo ideal.

Así, antes de ordenar por prioridad lo que debes hacer define al personaje, usuario o cliente: el individuo que usará lo que vas a producir. Debes conocer sus gustos, aversiones, pasiones, entusiasmos, frustraciones y alegrías; comprender sus motivaciones. ¿Cómo da a conocer lo que quiere ese tipo de personaje? ¿Por qué necesita un auto? ¿Qué se va a hacer con el diario del capitán?

Esto influirá asimismo en tu evaluación de las cosas. Una simple función de calendario es fácil. Una fecha inalterable con propósitos legales sería un tanto más complicada.

Escribe historias cortas

Cuando escribas tus historias, sin embargo, no olvides que deben ser cortas y precisas para poder evaluarlas. Imagina una historia sobre Amazon.com: «Como cliente, necesito la librería en línea más grande del mundo para poder comprar el libro que quiera cuando quiera». Sí, esto resume a Amazon, pero

es demasiado amplio para poder hacer algo al respecto. Debes desglosarlo lo más posible.

Podrías escribir historias como éstas acerca de una librería en línea:

> «Como cliente, me gustaría poder hojear libros por género, para poder encontrar los que me gustan».
>
> «Como cliente, me gustaría meter un libro en un carrito para comprarlo».
>
> «Como gerente de producto, me gustaría poder rastrear lo que compra un cliente para ofrecerle libros específicos con base en compras pasadas».

Estas historias son las que pueden interesar a un equipo. Pueden inducir una conversación sobre cómo implementarlas. Son tan específicas que pueden llevarse a la práctica, pero no prescriben la *forma* de hacerlo. Recuerda que el equipo decide cómo hacer el trabajo, pero *qué* hacer se decide con base en su valor de negocios. La serie completa de historias que podrían componer la idea de una librería en línea se conoce como «epopeya», una historia demasiado grande para ser puesta en ejecución, pero que incluye historias menores que constituyen en conjunto una idea.

Tim Stoll es uno de esos sujetos cuya carrera abarca un amplio espectro de acontecimientos, con la mira puesta en lograr que los equipos trabajen más rápido. Fue médico de las fuerzas especiales estadunidenses destacado en Iraq y Afganistán, contratista de la CIA y policía perseguidor de delincuentes violentos, y ahora es instructor de Scrum. *Siempre* lo ha sido, dice, incluso cuando dirigía misiones de las fuerzas especiales.

«En operaciones especiales», indica, «no las llaman historias, sino cursos de acción, pero son lo mismo».

He aquí una de las pocas historias que Stoll puede contar públicamente sobre una misión de las fuerzas especiales, una misión médica en Laos. «Teníamos dos epopeyas. La primera era un curso de instrucción médica: capacitar a fuerzas locales en medicina de guerra. La segunda era una operación para retirar minas sin detonar».

En su calidad de médico, Stoll estaba a cargo de la primera epopeya. Dice que, antes de cumplir la misión, pensó en qué debía lograr y cómo debía ordenar las subhistorias. Y comenzó con ideas que embonan bien con el marco de Scrum.

«Como médico de fuerzas especiales, debo enseñar fisiología básica a mis alumnos, para que conozcan el cuerpo humano».

Añade que, al comenzar a escribir sus historias, supo que debía empezar ahí. Sus alumnos debían saber dónde estaban los huesos para poder aplicar primeros auxilios. «Enseñaría primero los huesos largos, luego los cortos y más tarde las muñecas, tobillos, tendones y ligamentos». Sólo después de cubrir las historias básicas podría ocuparse de cómo fijar huesos, destapar vías respiratorias y detener hemorragias.

Tras escribir esas historias, pudo ver qué necesitaba para sustentar sus objetivos de enseñanza. Necesitaba un esqueleto. Folletos en inglés y laosiano. Y después dividió todo en repeticiones o *sprints*. «Dos días de vuelo a Laos. Una semana de preparativos. Luego, dos repeticiones de seis semanas de instrucción. Teníamos que hacer pasar a nuestros estudiantes de paramédicos básicos a intermedios. Y lo logramos».

Lista y terminada

Cuando escribes historias o elaboras una lista de pendientes es importante que te hagas dos preguntas: ¿la historia está lista?, ¿cómo sabrás que ya está terminada?

Usemos como ejemplo la historia de Stoll:

> Como médico de fuerzas especiales, debo enseñar fisiología básica a mis alumnos, para que conozcan el cuerpo humano.

Yo uso siempre un recurso nemotécnico para saber si una historia está lista. Fue creado por Bill Wake, especialista en diseño de *software*. Wake dice que para que una historia esté lista debe cumplir los criterios INVEST («Invierte»):

Independiente. Debe ser practicable y completable por sí sola. No debe depender inherentemente de ninguna otra.

Negociable. Hasta que no esté terminada, debe ser posible reescribirla, llevar integrada la tolerancia a cambios.

Valiosa. Tiene que aportar valor a un cliente, usuario o interesado.

Estimable. Debe ser posible evaluarla.

Suficientemente corta. Debe ser corta para poder calcularse y planearse fácilmente. Si es demasiado grande, reescríbela o divídela en historias más pequeñas.

Totalmente comprobable. Debe poder pasar una prueba que confirme que ha sido completada. Escribe la prueba antes de ejecutar la historia.

La historia de Stoll es independiente: él puede cumplir su misión sin tener que considerar, por ejemplo, el combustible que se necesitará para que los alumnos lleguen al campamento en helicóptero. Es negociable: enseñar fisiología es la historia que él cree que debe ejecutarse, pero si al llegar a su destino descubre que los estudiantes ya poseen esos conocimientos, puede cambiar su enfoque de enseñanza. Es valiosa: los estudiantes adquirirán conocimientos prácticos y aplicables del cuerpo humano. Es corta: cubre la anatomía básica, no cómo hacer una operación a partir de la anatomía enseñada. Y es comprobable: él conoce la información por impartir y puede aplicar una prueba a sus alumnos para ver si aprendieron de verdad.

Cada historia perseguida debe implicar una «Definición de estar lista» (como «¿Cumple los criterios INVEST?») y una «Definición de terminada» (como «¿Qué condiciones debe satisfacer, qué pruebas pasar, para que se le pueda considerar concluida?»). En proyectos reales hemos descubierto que si las historias están listas, el equipo duplicará la rapidez de implementación, y si están terminadas al final del *sprint*, los equipos podrán duplicar nuevamente su rapidez. Éste es uno de los trucos necesarios para poder hacer el doble de trabajo en la mitad de tiempo.

Planeación del *sprint*

En Scrum, todos y cada uno de los *sprints* deben anticiparse en lo que se llama la reunión de Planeación del *sprint*. Ésta congrega a todos los participantes, quienes analizan la lista de historias por ejecutar y dicen: «¿Qué podemos hacer en “este” *sprint*? ¿Están listas estas historias? ¿Podrán terminarse para el final de esta repetición? ¿Podremos mostrarlas entonces al cliente y comprobar que poseen valor real?».

La clave para responder a estas preguntas está en lo rápido que avanza el equipo.

Conoce tu velocidad

Ahora podemos contestar al fin la pregunta acerca de cuándo estará terminado el trabajo, porque sabemos cómo medir lo que hace el equipo. Todos conocemos las historias, las cosas que deben hacerse. Y las hemos evaluado: ésta es un ocho, aquélla un tres, etcétera. Iniciamos entonces nuestro primer *sprint* de, digamos, una semana de duración. Finalizada la semana, contamos todas las historias terminadas, sumamos los puntos asignados y este número nos dice lo rápido que avanzamos, nuestra velocidad. Una vez con este dato, es posible examinar cuántas historias nos faltan y cuántos puntos representan para poder saber cuándo acabaremos.

Asimismo, una vez que conoces tu velocidad puedes saber lo más importante en Scrum: qué te impide avanzar *más rápido*. Qué te impide *acelerarte*. En el capítulo anterior me ocupé del desperdicio, de lo que te retrasa. Ahora verás si realmente te *has* librado del desperdicio.

Volvamos a Medco, donde iniciamos este capítulo. Luego de que evaluamos todo el trabajo, me reuní con la alta dirección responsable del proyecto. Asistieron a esa reunión varios vicepresidentes que eran gerentes generales de unidades de negocios y un vicepresidente sénior.

Una vez que nos sentamos a la mesa de la sala de juntas, el vicepresidente sénior dijo tener una sola pregunta:

—¿Cumplirán la fecha original? —inquirió, golpeando la mesa con la palma de la mano.

—No lo sé —respondí—. Pero batiremos la fecha ajustada que sus colaboradores le dieron, o de lo contrario usted recuperará su dinero.

—¡Eso no basta! ¿Cumplirán la fecha original?

—No puedo saberlo *hoy*. Tenemos que poner en marcha a los equipos para ver cuán rápidos son. Pero le diré una cosa: en seis semanas le informaré de nuestra fecha de entrega y no le va a gustar. Pero —añadí, antes que pudiera interrumpirme— también le daré entonces una lista de lo que se interpone en el camino de sus equipos, lo que les impide cumplir la meta de julio que usted prometió a Wall Street. Una lista de impedimentos. Y usted tendrá que encargarse de eliminarlos lo más pronto posible.

Soltó una risotada.

—¡Impedimentos! Perfecto, Jeff. Trabajé un tiempo en Toyota.

Yo también reí y le dije:

—Bueno, entonces parece que éste es un buen proyecto.

Así supe que él conocía la taxonomía del desperdicio de Taiichi Ohno y sabía cómo funcionaban las cosas; que librarse del desperdicio era la clave para acelerar a los equipos.

Luego de medir su velocidad en tres *sprints*, los equipos habían pasado de veinte a sesenta puntos por *sprint*, lo que me permitió saber cuándo terminarían el proyecto. Dada su velocidad, y estando entonces a principios de marzo, necesitarían para concluir otros diecinueve *sprints* de dos semanas cada uno: el 1.º de diciembre.

Esto no fue del agrado de la dirección. No bastaba. Era el 1.º de julio o nada. Todo dependía de esto.

Entregué entonces un memo con una lista de doce impedimentos que iban de no facultar a la gente para tomar decisiones a onerosos requisitos técnicos, de personas que no asistían a las reuniones a cosas simples como no contar con todos los miembros de un equipo de trabajo en la misma sala. Había problemas de proceso, personalidad y procedimiento, el tipo de cosas endémicas a toda corporación.

Esa clase de impedimentos pueden parecer insuperables. ¿Cuán seguido has contemplado tu centro de trabajo y pensado: «Lo hacemos de esta manera, lo hacemos siempre de esta manera, pese a que todos sabemos que es absurda»? Pero, por alguna razón, la gente juzga imposible cambiar la cultura corporativa. Antes yo creía lo mismo, en especial tratándose de grandes compañías con cultura y políticas anquilosadas.

Medco me enseñó que estaba equivocado y jamás volveré a pensar igual. Aquel vicepresidente sénior llegado de Toyota envió nuestro memo a todo el personal un lunes. Junto a cada impedimento aparecía el nombre de un gerente. El jueves siguiente ya habían desaparecido todos los obstáculos. Tal vez para sentirse motivada a cambiar, a veces la gente necesita sentir un arma en la cabeza, pero eso demostró lo que puede hacerse si hay voluntad para llevarlo a cabo (o si se tiene a cargo a un sujeto procedente de Toyota). Nada está escrito en piedra. Cuestiónalo todo.

Al final del siguiente *sprint*, la velocidad de los equipos había aumentado cincuenta por ciento. La nueva fecha de entrega era el 1.º de septiembre. Seguían siendo tres meses después de lo previsto, pese a que los equipos habían pasado de veinte a noventa puntos por *sprint*, ¡un aumento de más de cuatrocientos por ciento!

Pero eso aún era insuficiente.

Barton y yo reunimos a todos, de ingeniería a mercadotecnia, analistas de negocios, personal de requisitos y de la gerencia. Y todos tenían miedo.

Temían por sus puestos y carreras si no lograban su cometido. Entonces dije:

—Haré tres preguntas:

1. ¿Hay algo que podamos hacer de otra manera para avanzar más rápido?

    - —Bueno —contestó el jefe de ingeniería—, a la mitad del *sprint* anterior los empleados de seguridad de informática desactivaron un puerto de internet, así que nuestros equipos en la India y Brasil no pudieron hacer nada.
    - —Deberíamos resolver eso, ¿no? —repuse yo, incrédulo.
    - El director de ingeniería volteó a ver al de informática, sentado en el otro extremo de la mesa. Acordaron que podían reducir en un mes más la entrega del producto. Pero aún quedaban dos meses de retraso.

2. ¿Podemos librarnos de algunos Pendientes? ¿Encargar algunas cosas a otros equipos?

    - A nadie se le ocurrió una buena idea sobre esto.

3. ¿Podemos dejar de hacer algunas cosas? ¿Reducir en cierta medida el alcance del proyecto?

    - La primera reacción fue que esto era imposible, que ya habían reducido al mínimo los requerimientos.
    - —Entonces dediquemos la tarde a reducir esto —dije—. Cada tarea tendrá que luchar por su vida.

Tardamos varias horas en hacerlo, pero conseguimos quitar otro mes a la fecha de entrega.

—Bueno —dije en ese momento—, todavía tenemos un mes de retraso. Si no se nos ocurre otra cosa, tendremos que decirle a la dirección que no lo lograremos.

—¡No! —exclamaron ellos—, nos despedirán a todos. Repasemos esas tres preguntas.

Propuse que nos reuniéramos con el equipo directivo. El problema no era sólo nuestro, también suyo y podía ayudarnos.

Fue una reunión corta. La dirección analizó la situación y dijo:

—Debemos terminar el 1.º de julio. ¿Podríamos lanzar inicialmente el programa en una sola fábrica? ¿Un solo centro? ¿O en un par de ellos? ¿Esto

funcionaría?

Hubo algunos carraspeos, murmullos y modificaciones de algunas cosas, pero finalmente se resolvió que era posible reducir las funciones necesarias y cumplir la fecha de julio de 2007 que el presidente había prometido a Wall Street.

Al final de esa reunión, el vicepresidente sénior dijo simplemente:

—Cantemos victoria. Llámenos si tienen algún problema.

Ver el precio de las acciones de Medco ese verano fue un espectáculo increíble. Cuando comenzamos a montar la infraestructura, las acciones empezaron a subir y no habían dejado de hacerlo para cuando terminamos. ¿Cuánto? Muchos miles de millones de dólares, de veinticinco a más de cincuenta dólares por acción en menos de un año. Wall Street había decidido que la compañía seguiría creciendo, atraería nuevos clientes y mantendría el liderazgo en su campo. En retrospectiva, debí haber pedido un porcentaje de la capitalización de mercado, no simples honorarios.

Años después Medco se valió de Scrum para producir lo que llamó Medco 2.0. Cada parte de la compañía fue reestructurada de raíz. Nuevas fábricas, nuevos robots, nuevos procesos, más automatización. Mark Landy, ya director técnico de la compañía, dice que sin la experiencia de los Centros de Recursos Terapéuticos no lo habrían conseguido. «No nos habrían permitido hacerlo en toda la empresa. Pero ya teníamos la confianza de toda la organización: desarrollo, operaciones, finanzas, clínica. Fuimos capaces de forjar una nueva cultura». Y eso, dice, es lo más importante de Scrum: que cambia la cultura en la que la gente trabaja, lo cual puede asustar a algunos. En efecto, la compañía tuvo que deshacerse de empleados que no libraron el cambio, dice Landy; y no porque fueran incompetentes, sino porque acaparaban información y conocimientos en su beneficio, para seguirse creyendo indispensables antes que ayudar al equipo y la compañía. Cambiar esa cultura, sin embargo, es lo que permite que emerja la excelencia verdadera.

RESUMEN

El mapa no es el terreno. No te enamores de tu plan. Casi seguro es erróneo.

Planea sólo lo que necesitas. No intentes proyectar todo con años de anticipación. Planea apenas lo suficiente para mantener ocupados a

tus equipos.

¿Qué tipo de perro es esto? No evalúes en términos absolutos, como horas; está demostrado que los seres humanos somos pésimos para eso. Evalúa las cosas relativamente: a qué raza de perro o tamaño de camiseta (CH, M, G, EG, EEG) corresponde el problema, o emplea la secuencia de Fibonacci, de uso más común.

Consulta el oráculo. Usa una técnica ciega como el método de Delfos para evitar sesgos de condicionamiento como el efecto halo o tren, o el mero pensamiento grupal.

Planea con póker. Usa el póker de planeación para evaluar rápidamente el trabajo por hacer.

El trabajo es una historia. Piensa primero en quién recibirá valor de algo; luego, en qué es esto, y después en por qué lo necesita. Los humanos pensamos en forma de relatos, así que ofrece uno: «Como X, yo quiero Y, para que Z».

Conoce tu velocidad. Cada equipo debe saber con precisión cuánto trabajo puede hacer en cada *sprint*. Y tiene que saber cuánto puede aumentar esa velocidad trabajando con más inteligencia y eliminando barreras que lo retrasan.

Velocidad x tiempo = entrega. Una vez que sabes lo rápido que avanzas, sabrás cuán pronto llegarás a la meta.

Fija objetivos audaces. Con Scrum no es tan difícil duplicar la producción o reducir a la mitad el tiempo de entrega. Si lo consigues, tus ingresos y el precio de tus acciones también se duplicarán.

Capítulo siete

Felicidad

La gente quiere ser feliz. No feliz de manera complaciente, pusilánime, sino activa. Thomas Jefferson, entre muchos otros, exaltó la felicidad que se deriva de la búsqueda. Buscar algo parece ser lo que nos hace felices. *Bien practicado*, Scrum hará felices a los trabajadores, los clientes, los gerentes y los accionistas (por lo general en ese orden).

La verdadera felicidad no llega fácilmente. Una vez conocí a un alpinista que me vendió una foto de la cima de los Himalaya con el ocaso del sol al fondo. La tomó poco después de haber llegado solo a la cumbre del Everest, hacia el final de la tarde. Parecía imposible regresar al campamento base antes de que oscureciera. Si no lo conseguía, estaba seguro de que moriría congelado. La elocuencia de la foto reflejaba lo que había sentido al escribir el que creyó sería su último mensaje, donde decía que se sentía feliz de haber llegado a la cima pese a que quien leyera esa nota lo encontraría muerto.

Si charlas con alpinistas acerca de una expedición, descubrirás que no hablan mucho de la experiencia de ascender a una cumbre, sino más bien de las bajísimas temperaturas, las ampollas dolorosas, la mala comida, las terribles condiciones y el equipo insuficiente. Y te dirán que, tras el júbilo de haber llegado a la cúspide, viene por lo común una sensación de chasco (a menos que persista la experiencia de la cercanía de la muerte). Lo lograron. Su esfuerzo dio fruto. Pero si les preguntas en qué instante se sintieron más felices, te dirán que en los momentos de prueba, de tener que llevar al límite el cuerpo, la mente y el espíritu. *Es entonces* cuando son más felices, cuando experimentan dicha auténtica. Y quieren experimentarla otra vez. Se diría que nadie en su sano juicio se sometería voluntariamente a eso *dos veces*. Pero los alpinistas parecen incapaces de parar, desafían una cima tras otra en pos de hallar la satisfacción en la búsqueda de la cumbre siguiente.

Lo interesante es que la mayoría de las culturas no están hechas para premiar y fomentar ese tipo específico de felicidad. El profesor Tal Ben-Shahar impartía el curso más popular en la Harvard University, psicología positiva. En su libro *Happier* (Más feliz), escribe: «No se nos premia por disfrutar el viaje, sino por su *consumación* exitosa. La sociedad premia resultados, no procesos; arribos, no viajes».

Sin embargo, nuestra vida diaria se compone principalmente de viajes. No escalamos cimas todos los días, ni obtenemos resultados grandiosos o bonos abundantes. La mayor parte de nuestros días se nos va en perseguir metas, cualesquiera que sean. En una compañía, la meta podría ser lanzar un gran producto, mejorar un poco con él la vida de la gente o resolver algún problema que aqueja al mundo. Pero si sólo obtenemos recompensas por los resultados, no por los procesos, seremos desdichados.

A principios de los años ochenta, cuando dejé la academia por el mundo de los negocios, se me puso a cargo de docenas de programadores de computadoras sumamente desdichados. Sus proyectos siempre se atrasaban y excedían el presupuesto, y eso si daban resultado. Su ánimo era tan negativo que la energía en la sala deprimía a todos. Su proceso estaba tan maltrecho que era imposible tener éxito. He dedicado los treinta últimos años de mi vida a atacar justo ese problema.

Me di cuenta de la importancia de la felicidad al organizar mi primer equipo con Scrum. Me percaté entonces de que debía ocuparme del estado emocional del equipo tanto como de su estado mental. Para un piloto de caza de West Point, eso implicó un ajuste enorme. Estaba acostumbrado a que las cosas se planearan. Siendo clínico y científico, me costaba trabajo entender que para potenciar a la gente, para cambiar su vida para bien yo tenía que cambiar. En el curso de ese primer esfuerzo de Scrum comprendí que la verdadera grandeza se funda en la alegría. Y que ser feliz es dar el primer paso al éxito.

Si todo esto parece un poco New Age o da la impresión de que voy a decirte que te sientes frente a una hoguera a cantar «Kumbayah» debes saber que cuando me inicié como asesor de nuevas empresas, los capitalistas de riesgo con quienes trabajaba me creían un *hippy* de San Francisco. En su visión del mundo, potenciar a la gente no funcionaría nunca. Claro, ahora soy consultor sénior de sociedades de capital de riesgo que suelen tratarme como oráculo. Cuando la gente tiene un problema difícil, pide al oráculo la solución. No necesariamente espera que la respuesta tenga sentido. Sólo la prueba y, para su sorpresa, casi siempre surte efecto.

Esto se debe a que la felicidad es crucial para tu empresa y, en realidad, predice mejor las utilidades que la mayoría de los parámetros del director financiero. En este capítulo expondré lo importante que es la felicidad para tus ganancias y cómo conseguirla, medirla y aplicarla. Esto es felicidad con rigor.

Tal vez me hice una persona mejor al desarrollar Scrum, lo cual nos ha hecho más felices a mi familia y a mí. Pero como hombre de negocios y científico, me gustan los datos duros.

Felicidad es éxito

Las investigaciones son muy claras: a la gente feliz le va mejor, en el hogar, el trabajo y la vida. Gana más, tiene un empleo mejor, obtiene un título universitario y vive más. Esto es notable. Casi universalmente, esa gente es mejor en lo que hace.

La gente feliz vende más, gana más, cuesta menos, tiene menos probabilidades de abandonar su empleo, es más sana y vive más. O como se dijo en un artículo publicado en 2005 en el que se hizo un metaanálisis de doscientos venticinco estudios con cerca de doscientos setenta y cinco mil participantes:

> La felicidad conduce al éxito en casi todos los terrenos de la vida, como matrimonio, salud, amistad, participación comunitaria y creatividad, y en particular en nuestro trabajo, carrera y negocios[35].

Ese metaanálisis demostró que la gente que se sentía feliz tenía más probabilidades de conseguir entrevistas de trabajo, ser evaluada positivamente por sus supervisores, exhibir desempeño y productividad superiores y ser mejor como gerente.

Pero lo más interesante es que resulta lógico que a la gente feliz le vaya mejor; el éxito la hace feliz, ¿no? Falso. De acuerdo con ese mismo metaanálisis, «un estudio tras otro confirman que la felicidad *precede* a resultados importantes e indicadores de prosperidad».

Así es. La gente no es feliz por ser exitosa, sino exitosa porque es feliz. La felicidad es una medida *predictiva*. Y el desempeño aumenta aun si la gente es sólo un poco más feliz. No es necesario cambiar radicalmente la vida de

alguien para hacerla feliz, al menos por un tiempo. Aun un poco de felicidad produce resultados mucho mejores. La gente no necesita ser delirantemente feliz, como el día de su boda; basta con que lo sea un poco más. Claro que hacerla más feliz tiene un gran efecto. Pero el mensaje que quiero transmitirte es simple: *incluso los gestos pequeños pueden tener una influencia enorme*. Scrum se centra en tomar esas pequeñas cosas y convertirlas sistemáticamente en un andamiaje para el éxito. Haz las cosas una por una y cambiarás el mundo.

Te voy a proporcionar un juego de herramientas para medir tu felicidad y la de tu equipo, compañía y familia, así como la de cualquier otra organización a la que pertenezcas. Eso es lo que hace Scrum. Olvídate de ejercicios para generar confianza y créala todos los días. Y mídela. No basta con *pensar* que la gente es feliz, debes ser un científico al respecto y cuantificar la felicidad, compararla con el desempeño. Si algo no luce bien hay un problema. Es estupendo que vayas al bar con tu equipo a reforzar los vínculos entre ustedes, pero esto no le sirve de mucho a la compañía si tales vínculos no se traducen en mejor desempeño. Yo me junto con muchas personas para divertirme. Con mi equipo me gusta que el aspecto social vaya directo al desempeño. Y así sucede.

Cuantificar la felicidad

¿Qué nos hace felices a nosotros, nuestros empleados y nuestros compañeros de equipo? ¿Cómo convertir esa felicidad en más productividad e ingresos? ¿Hay un método mejor que el de repartir caramelos todos los días?

Sí lo hay. Pero antes volvamos a Toyota y la cruzada de Taiichi Ohno para eliminar el desperdicio, lo cual desembocó en la idea de la mejora continua. El objetivo no es alcanzar cierto nivel de productividad y quedarse ahí, sino examinar constantemente tus procesos para mejorarlos siempre. La perfección es inalcanzable, desde luego, pero cada incremento en esa dirección cuenta.

Así como el trabajo debe dividirse en partes manejables y el tiempo en piezas también manejables, la mejora debe fragmentarse paso por paso. En japonés, mejora se dice *kaizen*. ¿Qué pequeña mejora puede hacerse de inmediato para que las cosas sean de más calidad?

En Scrum esto se sabe al final de cada *sprint*, en lo que yo llamo la «retrospectiva del *sprint*». Una vez que el equipo ha mostrado lo que hizo en

el *sprint* más reciente —aquella cosa «terminada» que puede enviarse a los clientes en busca de realimentación—, se reflexiona en qué salió bien, qué pudo haber salido mejor y qué puede hacerse mejor en el siguiente *sprint*. ¿Qué mejora en el proceso puede implementar de inmediato el equipo?

Para ser eficaz, esta reunión requiere cierto grado de madurez emocional y una atmósfera de confianza. La clave es que no se busca a quién culpar; se examina el proceso. ¿Por qué las cosas resultaron así? ¿Por qué no pudimos hacerlas de otra manera? ¿Qué pudimos hacer más rápido? Es crucial que, *como equipo*, la gente asuma la responsabilidad de su proceso y resultados y busque soluciones *como equipo*. También debe tener fortaleza para sacar a colación los asuntos que le molestan en una forma orientada a la solución, no acusatoria. Y el resto del equipo debe tener madurez para oír esos comentarios, aceptarlos y buscar una solución en vez de ponerse a la defensiva.

La reunión de retrospectiva es la parte «Revisa» del ciclo de Deming Planea-Haz-Revisa-Actúa. La clave es llegar al paso «Actúa», a ese *kaizen*, que cambiará el proceso y lo mejorará la próxima vez. No basta con decir cómo te sientes; debes ser capaz de actuar.

En mi opinión, la mejor forma de juntar todo esto es lo que llamo la «medición de la felicidad». Se trata de una forma sencilla pero eficaz de llegar a lo que *kaizen* debe ser, pero también de que *kaizen* haga más feliz a la gente. He aplicado esta técnica con excelentes resultados.

He aquí cómo funciona: al final de cada *sprint*, cada miembro del equipo responde a unas cuantas preguntas:

1. En una escala de 1 a 5, ¿cómo te sientes con tu papel en la compañía?
2. En esa misma escala, ¿cómo te sientes con la compañía en general?
3. ¿Por qué te sientes así?
4. ¿Qué te haría más feliz en el siguiente *sprint*?

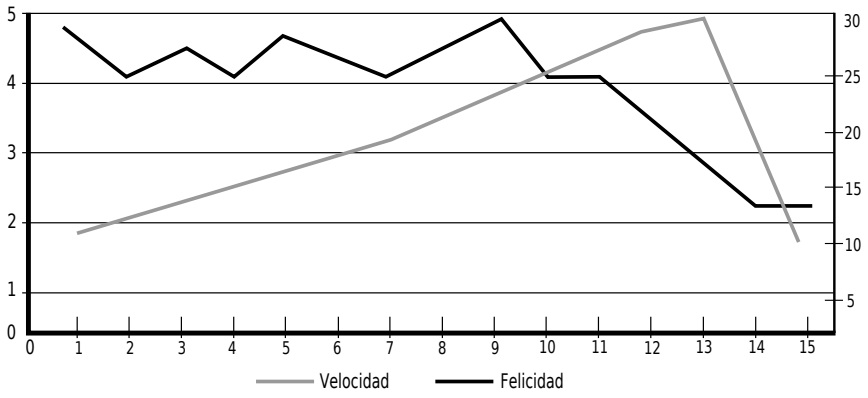
Eso es todo. Tarda unos cuantos minutos. Los integrantes del equipo se turnan para contestar, lo que genera conversaciones muy ilustrativas. Esto suele resultar en que el equipo alcance más rápido un *kaizen*. Este método expone qué es lo más importante para cada miembro y lo que considera más importante para la compañía.

La pieza crucial es que el equipo convierta esa gran mejora en lo más importante por hacer en el siguiente *sprint, con pruebas de aceptación*. ¿Cómo puedes probar que hiciste esa mejora? Antes debes definir el éxito de

modo práctico y concreto, para que en la próxima retrospectiva del *sprint* resulte fácil ver si el equipo logró ese *kaizen*.

Hace unos años decidí ampliar mi compañía, Scrum Inc., para convertirla en consultoría de Scrum con todos los servicios. Rastreamos entonces nuestra velocidad y descubrimos que en cada *sprint* de una semana acabábamos unos cuarenta puntos de historias. Cuando puse en práctica la medición de la felicidad, lo primero que emergió fue que las historias sobre nuestros usuarios no eran muy buenas. No estaban lo bastante listas, no tenían una definición de Terminado y eran muy vagas. Trabajé en ello y empezamos a obtener mejores historias. Pero en el siguiente *sprint*, las historias seguían sin ser muy buenas; nuestras cifras de felicidad no fueron altas. En el tercer *sprint* surgió otro problema, lo atacamos y desapareció. En unas cuantas semanas, nuestra velocidad pasó de cuarenta a ciento veinte puntos por *sprint*; triplicamos nuestra productividad preguntándonos simplemente qué haría más feliz a la gente. En consecuencia, nuestros clientes fueron más felices y nuestros ingresos se dispararon. Todo lo que tuve que hacer fue empezar a preguntar al equipo: «¿Qué los haría más felices?» y dárselo.

Con esos datos hicimos una gráfica y advertimos cosas muy interesantes. Como director general, me fijo en lo que pasará a futuro con nuestros ingresos, crecimiento y productividad. Descubrí entonces que, a diferencia de los indicadores financieros, el parámetro de la felicidad era *predictiva*. Los financieros analizan el pasado, pero cuando preguntas a la gente qué la hace feliz, se proyecta al futuro. Y cuando piensa en lo feliz que está en la compañía, proyecta cómo cree que le irá a ésta. Obtienes de este modo indicaciones de un problema *antes* de que surja. Y si prestas atención en lo que tu equipo dice, puedes actuar y resolver el asunto antes de que sea un problema. En esta gráfica, por ejemplo, un descenso en felicidad precede en semanas a uno en velocidad o productividad. Si sólo consideraras la productividad, no sabrías que hay un problema hasta caer en un precipicio. Pero si ves una caída grupal en felicidad, aun si la productividad está aumentando, sabes que debes atacar un asunto y pronto.



Vuelve visible todo

¿Qué es lo que *realmente* hace feliz a la gente? Lo mismo que hace grandes a los equipos: autonomía, maestría y propósito. O para decirlo más largamente, la posibilidad de controlar tu destino, la sensación de que estás mejorando y saber que persigues algo más allá de ti. No obstante, también hay pasos fáciles y concretos que la gerencia puede dar para conseguir que la cultura de una compañía aliente esas cualidades.

Un elemento de Scrum que suele preludiar la consecución de autonomía, maestría y propósito es la transparencia. La idea es que no debe haber conspiraciones secretas, agendas ocultas, nada detrás de la cortina. Demasiado a menudo, en las compañías no está claro qué hacen todos o en qué forma la actividad diaria de cada persona promueve las metas de la organización.

Cuando inicié Scrum, dediqué mucho tiempo a pensar en las leyes que un buen amigo mío contribuyó a introducir en la legislatura de Colorado, las leyes «Sunshine». Éstas establecen que en todas las reuniones públicas debe permitirse el acceso a cualquier persona, que todos los documentos deben estar a disposición del público y que nada ha de ocurrir a puerta cerrada, nada debe quedar oculto. A esto se debe que, en Scrum, cualquier individuo pueda asistir a una reunión. Cualquier interesado puede observar una parada diaria o estar presente en una revisión.

Yo quería que todo fuera visible, lo cual puede ser inquietante para algunos. PatientKeeper es una compañía que desarrolla aplicaciones portátiles para médicos y hospitales. Cuando me contrató, convertí de inmediato toda el área de ingeniería en un departamento con Scrum. Dije a los desarrolladores que todos serían enterados de todo. Estaban tan acostumbrados a que los

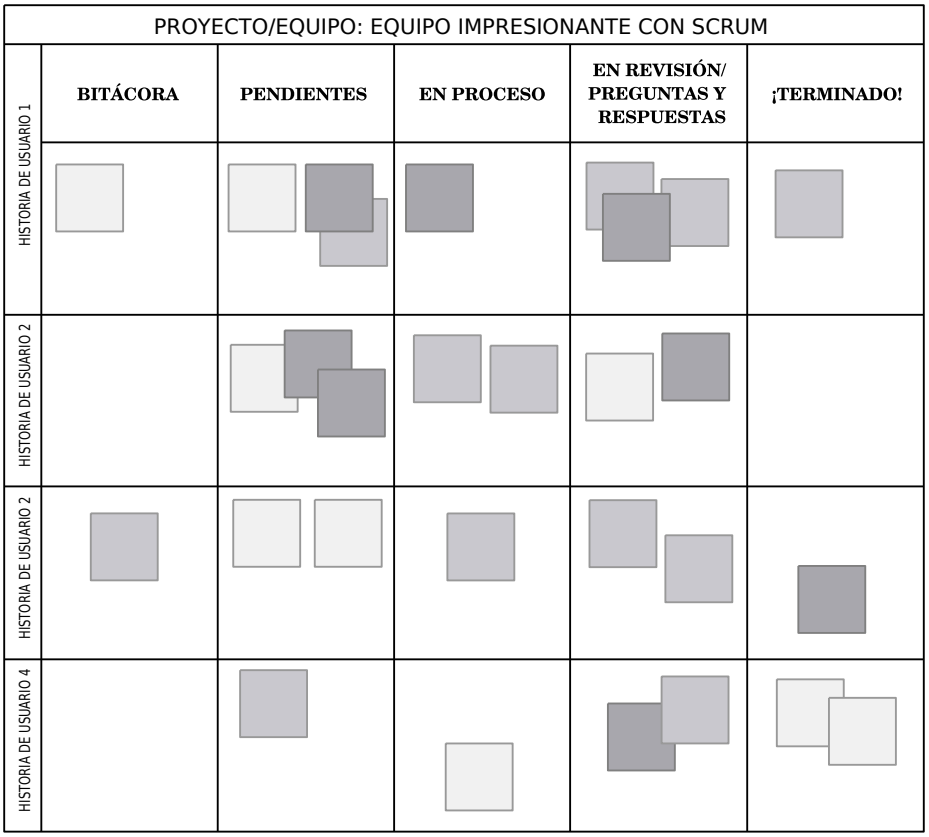
indicadores se usaran para perjudicarlos que el nuevo nivel de transparencia les hizo temer que se abusara más de ellos.

«Confíen en mí», les dije, «esto no servirá para afectarlos o castigarlos, sino para hacer mejor las cosas».

Como ya dije, no me interesa el desempeño individual, sino el grupal. Puedo duplicar la productividad de un equipo en un mes, pero ¿la individual? Esto podría tardar un año. ¿Y muchos individuos? ¿Una división entera? ¿Una compañía? Podría ser eterno. Así, uso la transparencia para concentrarme en mejorar al equipo. He descubierto que, por lo general, éste puede ocuparse de cuestiones de desempeño individual. Sabe qué hacen sus miembros, quién contribuye, quién perjudica, quién engrandece al equipo y quién lo daña.

Así, en Scrum todo es visible. En mis compañías, cada sueldo, cada dato financiero, cada gasto está a disposición de todos. Nunca he entendido por qué alguien querría mantener eso en secreto, salvo para promover su agenda individual o tener infantilizada a la gente. Quiero que el asistente administrativo pueda entender el estado de pérdidas y ganancias y saber *con exactitud* que lo que él hace contribuye a eso. Quiero que todos en la compañía se unan a un mismo propósito. Atomizar a la gente en silos informativos retrasa a todos. Además, genera recelo y desconfianza. Divide a una compañía en los grandes que saben todo y los peones que sólo ejecutan segmentos de una agenda misteriosa que no pueden entender. Tonterías. Si no puedes confiar en que la gente que contratas estará a tu lado en tus actividades has empleado a la gente equivocada y erigido un sistema condenado al fracaso.

La representación visual más elocuente de esta idea, la cual hallarás en toda sala de trabajo en equipo con Scrum en el planeta, es el pizarrón de Scrum.



Hay *software* para medir todo tipo de cosas y que te da todo tipo de medidas y análisis, pero el pizarrón de Scrum es apenas un montón de papeletas adhesivas en un pizarrón blanco. Hay tareas de tres niveles: Pendientes, En proceso y Terminado. Cuando alguien firma una historia, todos saben quién trabaja en ella y cuándo se le consuma. Además, como las papeletas del pizarrón representan lo que debe hacerse en un *sprint*, todos saben exactamente qué está haciendo el equipo.

Puesto que éste sabe qué se hizo ya y qué está por hacerse aún, puede regularse. Sabe qué tiene que hacer, puede ver si un colega está en dificultades, si una historia ha estado demasiado tiempo en la columna En proceso. Puede organizarse para resolver problemas que se vuelven obvios una vez que todo es transparente.

En PatientKeeper, la transparencia a la que los desarrolladores inicialmente temieron rindió dividendos. Gracias a que todo el trabajo era transparente, podían coordinar deberes entre distintos equipos. Todos sabían exactamente qué hacían los demás en todo momento. Podían apoyarse unos a otros si alguien tropezaba con un obstáculo. Tal vez un desarrollador ya había resuelto un problema que otro enfrentaba, ¡aun si no estaban en el mismo equipo! La productividad de PatientKeeper aumentó más de cuatro veces.

Hacíamos versiones de producción de *software* para empresas cuarenta y cinco veces al año. Y no era una actualización de Angry Birds, sino productos que se usaban en grandes hospitales y de los que dependía la vida de muchas personas. Pero gracias a que éramos transparentes en todo, podíamos llevar el producto al mercado más rápido que nadie más en el mundo entero. Esto es lo que puede hacer la luminosidad.

Tiempo después de mi paso por PatientKeeper, un nuevo equipo directivo decidió que Scrum no era la mejor manera de administrar. ¿El resultado? El lanzamiento de productos se redujo de cuarenta y cinco a dos al año y los ingresos de cincuenta a veinticinco millones de dólares al año, mientras que el desgaste, antes menor de diez por ciento, se disparó a más de treinta. Retornar a la conducta corporativa tradicional volvió mediocre el desempeño de una compañía antes grandiosa.

Dar felicidad

Zappos es una empresa que considera que la felicidad es básica en su cultura. Esta página en internet de enorme éxito convenció a la gente de hacer algo que muchos creían imposible: comprar zapatos en línea. Su director general, Tony Hsieh, escribió un libro sobre esto, *Delivering Happiness* (Dar felicidad), en el que refiere la singular cultura de Zappos, fundada en crear momentos «¡Guau!» para los clientes. Resulta que para hacer felices a los clientes necesitas tener gente feliz del otro lado del teléfono.

Al hablar con los ejecutivos de Zappos, una palabra que aparece con frecuencia es *relación*. Sus propias investigaciones demuestran que cuanto más se relacionan sus empleados en el trabajo, más felices son y, al parecer, también más productivos e innovadores. De este modo, los ejecutivos de la compañía se propusieron deliberadamente crear relaciones, no sólo en los equipos o departamentos, sino en toda la compañía. Y no sólo entre personas del mismo nivel, sino también de niveles diferentes, desde vicepresidentes hasta empleados de cuentas por cobrar.

Lo hacen esto por medios simples y complejos. Por ejemplo, promueven físicamente los encuentros casuales. Su edificio tiene muchas salidas, pero están cerradas, salvo una, lo que obliga a la gente a entrar y salir por ella. La idea es que hacer que los empleados tropiecen unos con otros creará y nutrirá sus relaciones.

Hay otro ejemplo de la manera en que Zappos integra a la gente a su cultura. Cada empleado, de almacenistas a directores, tiene que pasar por lo que Christa Foley, gerenta de recursos humanos, llama «campamento de entrenamiento». Durante cuatro semanas, cada empleado es educado a toda velocidad en la forma de trabajar de la compañía, aunque también en el modo en que opera su cultura. Éste es en realidad el segundo filtro de su proceso de contratación. Aun después de obtener un puesto, debes demostrar que puedes asimilar su cultura.

Los resultados, dice Foley, son extraordinarios. «Las relaciones que [los empleados] forjan durante el campamento los acompañan a todo lo largo de su carrera». El campamento es intenso a propósito: la gente debe presentarse a las siete de la mañana, trabajar duro, cumplir fechas límite y pasar pruebas. Pero funciona. Quienes se someten al campamento se mantienen en mutuo contacto no sólo meses sino años, y organizan reuniones y parrilladas para seguir unidos.

«Se vuelven una familia extensa», dice la ejecutiva de Zappos Rachel Brown. «Llevan a casa a sus amigos del trabajo, conviven con ellos».

Otra forma en que Zappos mantiene felices a sus empleados es dándoles la oportunidad de aprender y crecer. La compañía casi siempre prefiere ocupar vacantes con empleados propios. Si, por ejemplo, se anuncia una vacante en recursos humanos y de ello se entera alguien de contabilidad a quien siempre le ha interesado un puesto como ése, esta persona podría recibir «capacitación». Esto da al empleado la oportunidad de saber si realmente le gusta ese trabajo y al gerente la de ver si encaja en el equipo. La compañía también ofrece cursos gratis impartidos por empleados: introducción a las finanzas, codificación para principiantes, de todo. Zappos quiere que la gente crezca en la compañía.

Como dije en el capítulo tres, acerca de los equipos, a la gente le gusta crecer; quiere ser mejor en lo que hace y descubrir en qué más puede mejorar. La idea es que dominar su trabajo motive a la gente. Darle la oportunidad de saber dónde encaja ayuda a Zappos a mantener a sus empleados felices, emocionados y comprometidos.

Para quienes llevan mucho tiempo en una carrera muy tradicional, esta cultura puede ser una bocanada de aire fresco. «Toda mi carrera previa a Zappos consistió principalmente en reclutar», dice Foley. Era un trabajo rutinario, afirma, y la fastidió. Llegar a Zappos la revitalizó. Fue su cultura, dice. «Esto hace que venir a trabajar me emocione».

Eso es lo que Zappos quiere, lo que cualquier compañía debería querer. Y es lo que yo quiero: que a la gente le guste ir a trabajar. Esto supone un cambio en la mentalidad, de trabajar en una compañía a trabajar en *mi* compañía. Y ésta es una mentalidad que a algunas personas se les dificulta aceptar. Lo mismo que el motivo de que Zappos se concentre en ascensos internos. Ha descubierto que la gente de fuera, en especial de alto nivel, tiene problemas para adaptarse. «Somos una combinación de emprendedores e innovadores», dice Foley. Pero eso es sólo la mitad del asunto. «La otra mitad es la colaboración». La compañía quiere gente que trabaje en común y mantenga relaciones en toda la organización. A veces esto no encaja en la cultura corporativa estándar. Un alto ejecutivo me dijo: «Mi puesto no tiene ningún nombre específico. Creemos que como grupo podemos hacer las cosas mucho mejor».

En casi toda compañía hay gerentes a los que les gusta dirigir su área sin transparencia ni colaboración. Crean una dinámica de «nosotros contra ellos». Trazan fronteras y casi puede verse a las diferentes divisiones conjurar unas contra otras como salidas de una maquiavélica corte medieval. Imagina cuánto más productiva sería una compañía si todos trabajaran hacia una meta común. Imagina una compañía que todos concibieran como suya, donde cada día fuera una oportunidad de mejorar, de hacer algo mejor, de aprender algo nuevo. En cambio, la mayoría de ellas establecen un entorno en el que la gente se ocupa más de politiquerías que de generar beneficios.

En Zappos, si no encajas en el equipo y la cultura, no encajas en la compañía. Su tasa anual de deserción es de doce por ciento y la mayor rotación, asegura, ocurre en su centro de atención telefónica. Eso se debe a que despide a personas a las que no les apasiona dar buen servicio a los clientes. Zappos ve a esas personas como su rostro público y sus estándares son altos. Es flexible en muchas cosas, pero no en ésa.

He visto desplegarse en equipos esa misma dinámica. Un miembro de un equipo podría tener conocimientos o habilidades especializadas y acapararlos como un avaro. Los ve como una posesión que garantiza su empleo. Mediante sus retrospectivas y transparencia, Scrum exhibe casi de inmediato este tipo de comportamiento. Vuelve obvio dónde están los obstáculos, dónde el desperdicio. Cuando yo dirijo una compañía, digo a la gente con hábitos de «avaro» que no puede darse el lujo de tomar como rehenes a su equipo y a la compañía, que puede cambiar de mentalidad o irse a trabajar a otra parte.

Zappos ha descubierto que cuanto más alto sea el nivel de un nuevo empleado, más arraigadas son sus ideas y, por tanto, más tiene que trabajar

para deshacerse de sus viejas maneras de hacer las cosas. Scrum da a la gente un marco para lograrlo. Ofrece una estructura para que toda la organización se dirija a una meta común. Sus pilares son la transparencia, el trabajo en equipo y la colaboración. Muchas compañías abrazan ahora esta filosofía; las que no, pierden inevitablemente ante las que sí lo hacen.

Las ventas de Zappos pasaron de un millón seiscientos mil dólares en 2000 a más de mil millones en 2008, una tasa de crecimiento de ciento veinticuatro por ciento anual durante ocho años consecutivos. No sé tú, pero yo creo que éste es un argumento muy convincente para hacer feliz a la gente. Y Scrum es un juego de herramientas que puedes usar para lograrlo.

Rompe la burbuja de la felicidad

Algo que la felicidad no es —al menos del tipo al que me refiero— es complacencia. Es más bien lo opuesto: compromiso positivo y apasionado. Como dice Christa Foley, de Zappos, la felicidad es lo más alejado de la pasividad. «Me gusta venir a trabajar. En vez de [inducirte a ser] complaciente, nuestra cultura positiva y edificante te hace trabajar con más empeño». Esta compañía debe eliminar a personas que creen que trabajar en un lugar agradable significa no hacer nada. Quiere gente que se sirva de la alegría como motivación.

Y no es la única. La *Harvard Business Review (HBR)* dedicó su número de enero-febrero de 2012 al tema de la felicidad. Descubrió que

> la única ruta para tener empleados felices que también beneficia a los accionistas es la sensación de realización que resulta de hacer bien un trabajo importante. Debemos aspirar no sólo a hacer «felices» a los empleados, sino también a lograrlo ayudándolos a alcanzar grandes cosas. En suma, debemos ganarnos la vehemente defensa de la misión y el éxito de la compañía por parte de los empleados ayudándolos a ganarse la vehemente defensa de los clientes[36].

Esa defensa vehemente tiene beneficios tangibles. Los empleados felices se presentan a trabajar, se esfuerzan más y no sólo no dejan la compañía, sino que además atraen a otros como ellos que comparten la misma motivación. En su artículo para el citado número de *HBR*, Gretchen Spreitzer y Christine

Porath decidieron no llamar «feliz» a esa gente debido a las connotaciones de complacencia de esta palabra. En cambio, la llamaron «próspera». Establecieron que ese tipo de personas rinde dieciséis por ciento más que sus iguales, tiene ciento veinticinco por ciento menos fatiga, es treinta y dos por ciento más comprometido y está cuarenta y seis por ciento más satisfecho con su trabajo. Toma menos días de licencia por enfermedad, hace menos citas con el médico y tiene más probabilidades de ser ascendido[37].

Esas personas «prósperas» comparten aquello sobre lo que he escrito en este capítulo: cada una es vital y apasionada e intenta perfeccionar su oficio, así pertenezca a la tripulación de un avión o sea ayudante de camarero en un restaurante. ¿Qué pueden hacer las compañías para crear una atmósfera en la que la gente prospere? Los gerentes pueden alentar la autonomía permitiendo a las personas tomar decisiones sobre su trabajo. Y pueden cerciorarse de que sus empleados estén al tanto de todo lo que ocurre, porque, como dicen aquellas investigadoras, «trabajar en un vacío de información es tedioso y poco inspirador». Los gerentes también deberían tener cero tolerancia a la descortesía y no permitir nunca que un empleado contamine la cultura corporativa con abusos o faltas de respeto. Por último, deben dar realimentación rápida y directa.

Scrum brinda a la gente todo eso: está hecha para *hacer* que ocurra, en especial la realimentación directa, la cual sucede todos los días en la reunión de parada diaria y es iluminada por la retrospectiva de *sprint* y la medición de la felicidad.

Cabe hacer una advertencia. Es posible —tanto, que he dedicado mucho tiempo a estudiarlo— que se desarrolle una «burbuja de la felicidad». Esto suele ocurrir después de que el equipo ha alcanzado un gran éxito o incrementado su productividad, principalmente por usar Scrum. Se ha organizado y está orgulloso de su progreso. Entonces la complacencia puede aparecer. Sus miembros se dicen: «Hemos mejorado tanto que ya no necesitamos mejorar más». Llegan a una meseta de productividad y muy pronto su trabajo desmerece. Pero son tan buenos que viven por un tiempo en una burbuja de felicidad que los libra de verdades desagradables. No se dan cuenta de que la mejora continua significa justo eso: permanencia. Cuando era piloto de caza, solía decirse que había que renunciar luego de tres mil horas en la cabina, porque te vuelves complaciente y podías perder la vida. Aunque un equipo complaciente puede correr menos riesgos en los negocios, aun así el desempeño permanente del equipo está en peligro.

Esta actitud complaciente suele revelarse en un comentario como éste: «*Merecemos* dejar de esforzarnos, nos lo hemos ganado». O bien, algunas personas valoran tanto su espíritu de equipo y felicidad que no quieren ponerlo en riesgo. O temen el cambio y creen que si lo que tienen funciona, ¿para qué alterarlo?

Puesto que es aquí donde pueden descomponerse las formas de Scrum, las «burbujas de la felicidad» son una de mis mayores preocupaciones. Lo he visto una y otra vez: un equipo puede hacer todo lo que Scrum enseña —priorización, realización de tareas una por una, interfuncionalidad, rituales de revisión— sin mejorar por ello. Será mucho mejor que antes de que adoptara Scrum y sus éxitos lo demostrarán, pero se habrá dormido en sus laureles. Dice: «Ya no es *necesario* que mejoremos».

Esto me recuerda al equipo estadunidense de basquetbol que participó en las Olimpiadas de 2004. Tenía jugadores de primera —LeBron James, Tim Duncan y Allen Iverson, por nombrar unos cuantos— y Estados Unidos contaba con una historia no sólo de triunfos, sino también de predominio en ese deporte, en particular desde que se permitió la participación de jugadores profesionales. Los basquetbolistas estadunidenses sabían que eran los mejores, salvo que resultó que *no lo eran*. Perdieron más partidos que ningún otro equipo olímpico de su país en su especialidad. Perdieron incluso ante Lituania. Su orgullo y complacencia fueron su ruina. Vivían en la burbuja de la felicidad.

¿Cómo romper esa burbuja antes de que tus jugadores avergüencen a su país en una transmisión en vivo por televisión frente a miles de millones de personas? El primer paso es estar consciente del problema, por lo cual recomiendo que los equipos midan su velocidad en cada *sprint*. Debe saber cuál es su índice de cambio. Si no hay crecimiento positivo, sé que debemos intensificar esfuerzo. Y la persona de la que dependo para esto es el *Scrum Master*. Él debe ser capaz de ver el problema y planteárselo al equipo. Es crucial que alguien haga las preguntas difíciles. Y para eso necesitas un «bufón».

> Me maravillo del parentesco que tienes con tus hijas; ellas me quieren azotar si digo la verdad; tú quieres azotarme si miento, y a veces soy azotado por guardar silencio[38].

El bufón es la persona que hace preguntas desagradables o dice verdades incómodas. No siempre es fácil convivir con este tipo de gente, porque se le

suele considerar problemática o ajena al equipo, pero debe ser cultivada y aprovechada.

Quizá el mejor ejemplo de esto es uno que todos conocemos: el cuento clásico «El traje nuevo del emperador», de Hans Christian Andersen. Como recordarás, había una vez un emperador al que le gustaban tanto los ropajes finos que tenía una capa diferente para cada hora del día. Si se quería saber dónde estaba, el primer lugar donde buscar era el vestidor. Un día llegaron a verlo unos estafadores y le dijeron que tenían una fina tela misteriosa que las personas indignas de reinar no podían verla. Pidieron la seda más exquisita, que fingieron coser, cosiendo en cambio el aire, mientras aquellas telas preciosas iban a dar a sus alforjas. Un día, el emperador fue a supervisar su progreso y no vio nada. Recordando que sólo los dignos de reinar podían ver la tela, dijo que era la más fina que hubiese visto nunca. Preguntó a sus consejeros y ellos juraron que aquél era el tejido más prodigioso del mundo. El día previsto, los estafadores vistieron cuidadosamente de nada al emperador, quien recibió comentarios tan favorables de su corte que decidió desfilar por la ciudad, para mostrar a su pueblo ese traje fabuloso.

Seguro te acuerdas del final de la historia: nadie dijo nada sobre la desnudez del emperador para que no se le considerara indigno. Así, la procesión real recorrió la avenida hasta que un pequeño exclamó: «¡Pero si no lleva nada puesto!». Su padre quiso hacerlo callar, pero entonces, primero en un murmullo y luego a gritos, la gente confirmó: «¡No trae nada puesto!». Aunque temía que estuviera en lo cierto, el emperador continuó la procesión y sus cortesanos lo siguieron, sosteniendo una cola inexistente.

El bufón es ese niño, la persona capaz de ver que la verdad aceptada no pasa de ser una ilusión consensada y que el emperador no viste traje alguno. Así que si tienes uno o dos bufones, consérvalos.

Hay otras formas de romper la burbuja de la felicidad —inyectando sangre nueva o por intervención de la dirección, por ejemplo—, pero en el fondo todas son iguales, pues enfrentan al equipo con una realidad que quizá no quiere ver. Por fortuna, con Scrum todo es transparente: cuánto produce el equipo, la calidad de su trabajo, cuán satisfecho está el cliente. Una de las virtudes de Scrum es que hace rápidamente visible lo desagradable. En contraste, los equipos y organizaciones tradicionales pueden acercarse alegremente al despeñadero y preguntarse qué pudo marchar mal. Esperan demasiado a obtener realimentación práctica del mercado y unos de otros.

Feliz ahora, feliz mañana

Los psicólogos, incluido Ben-Shahar, de Harvard, dicen que una forma de analizar cómo ve el mundo la gente es preguntarle qué la hace feliz hoy y si eso mismo la hará feliz mañana. Creo que éste es un cristal útil para mirar a las personas en el entorno del trabajo.

Todos solemos corresponder a uno de cuatro tipos de gente, de acuerdo con Ben-Shahar. El primero, el «hedonista», es alguien que hace lo que lo vuelve feliz ahora. ¿Mañana? «Que el mañana se ocupe de mañana. Yo disfrutaré el hoy». Veo mucho esta clase de comportamiento en nuevas empresas: un montón de personas en el garage proverbial haciendo cosas porque es prestigioso y divertido hacerlas. Pero no se presta mucha atención a crear un producto sustentable. Muy poca energía mental se invierte en pensar cómo funcionará aquello en un mes y menos todavía en un año.

Y lo que suele suceder es que quienes invierten en esas empresas se preocupan, así que contratan a un montón de gerentes para que vigilen a los *hackers*. De repente, estos últimos descubren que el mundo que tanto disfrutaban antes es un horror ahora. Ya existen toda suerte de reglas, pruebas e informes. Eso es horrible hoy y ellos creen que lo será siempre. Llamemos «nihilistas» a estos individuos.

Luego están los sujetos llegados para dirigir una compañía. Son los que están dispuestos a invertir semanas de ochenta horas (y a forzar a otros a hacerlo también), porque creen que más tarde se les ascenderá y serán más felices. Claro que cuando los ascienden sólo obtienen una nueva serie de dolores de cabeza que les requieren más tiempo. Les gusta la «carrera de ratas».

El cuarto tipo de personas es aquel que Scrum trata de identificar y alentar: el individuo que trabaja en algo divertido hoy, pero con la mira puesta en un mejor futuro y que está convencida de que eso será divertido siempre. Esta clase de personas raramente experimentan fatiga o desilusión. Están libres de los sentimientos negativos por el trabajo de los hedonistas, los nihilistas y los gerentes adictos a la carrera de ratas que se empeñan en hacer que todos acaten la disciplina.

Scrum promueve una mentalidad galvanizadora. Al hacer trabajar a todos en común, el equipo ayuda al hedonista a ver al frente, convence al nihilista de que hay un futuro sin quejumbres y dice a los gerentes aferrados a una interminable carrera de ratas que en realidad hay una mejor manera de hacer las cosas.

Por eso implementé en mi compañía la medición de la felicidad. Ésta ayuda al equipo a ayudar a sus miembros a ser personas mejores. Elimina las causas de infelicidad sistemática, cuidadosa e incrementalmente. Faculta a la gente a cambiar y aporta un incentivo para hacerlo.

¿Recuerdas el error fundamental de atribución? Cuando estás rodeado de necios, no busques malas personas; busca los malos sistemas que las premian por actuar de ese modo. Y después usa la medición de la felicidad para remediarlo.

En la preparatoria o la universidad, muchos de nosotros estudiamos la jerarquía de las necesidades del psicólogo estadunidense Abraham Maslow. Esta jerarquía mostraba, en forma de pirámide, las necesidades de las que se ocupan primero los seres humanos y luego las que se vuelven más apremiantes una vez satisfechas las inferiores. En la base de la pirámide están las necesidades fisiológicas: aire, agua, alimento, ropa y techo. Si no tenemos estas cosas, no podemos comenzar a pensar siquiera en nada más. La capa siguiente es la seguridad, no sólo física y financiera, sino también de buena salud. Es importante tener cierto acceso a la atención médica. Curiosamente, muchas personas se quedan ahí, pese a que la capa siguiente contiene algo que definitivamente necesitamos como seres humanos pero que la sociedad suele ignorar: amor y pertenencia, las relaciones de las que se habla en Zappos. Sobre ella está la necesidad de autoestima y respeto a los demás. Y en la cima de la pirámide está la necesidad de realizar el pleno potencial personal.

Esta capa superior era la que más interesaba a Maslow y en la que se centra Scrum: ayudar a la gente a alcanzar su crecimiento y realización personal. Los individuos que llegan alto en esta pirámide no sólo son más felices y se sienten más realizados, sino que también son más eficientes e innovadores. Y son capaces de producir grandeza.

Casi puedo verte asintiendo con la cabeza en este momento, porque todos conocemos visceralmente esa pirámide, aun si a algunos no nos la han explicado nunca. El truco es subir a esas grandes alturas y disponer después de un medio para medir con *precisión* la influencia que ejerces. Si tienes un negocio, quizá midas la grandeza por ingresos y crecimiento. Si intentas curar a los enfermos, tal vez midas la grandeza por el número de quienes sobreviven. Si tratas de cambiar el mundo, quizá la midas por *cuánto* lo has transformado. Si sólo intentas cumplir tu lista de pendientes en el hogar, tal vez la midas por cuántas tardes de fines de semana tienes libres para ir a pescar.

No basta ser feliz. La felicidad debe aprovecharse para producir resultados. Todos los elementos de Scrum confluyen para ayudar a una persona a hacer justo eso. ¿El verdadero truco? Prioridades. Hablaremos de ellas enseguida.

RESUMEN

Es el viaje, no el destino. La felicidad verdadera está en el proceso, no en el resultado. Solemos premiar sólo resultados, pero lo que *realmente* debemos honrar es a la gente que se esfuerza por ser grande.

La felicidad es lo de hoy. Te ayuda a tomar decisiones más inteligentes. Además, cuando eres feliz también eres más creativo, menos propenso a dejar tu trabajo y más a cumplir más de lo que nunca previste.

Cuantifica la felicidad. No basta sentirse bien; debes medir esa sensación y compararla con tu desempeño. Otros indicadores ven atrás; la felicidad es una medida que mira al futuro.

Mejora cada día y mídelo. Al final de cada *sprint*, el equipo debe tomar una pequeña mejora, o *kaizen*, que lo haga más feliz. Y esto debería ser lo más importante por cumplir en el *sprint* siguiente.

El sigilo es veneno. Nada se debe mantener en secreto. Todos deben saberlo todo, y esto incluye sueldos e información financiera. La confusión sólo sirve a la gente que vela por su interés propio.

Haz visible el trabajo. Ten un pizarrón que muestre todo el trabajo pendiente, lo que está en proceso y lo que ya se terminó. Todos deben verlo y actualizarlo cada día.

La felicidad es autonomía, maestría y propósito. Todos queremos controlar nuestro destino, mejorar en lo que hacemos y perseguir un propósito que nos trascienda.

Rompe la burbuja de la felicidad. No seas tan feliz como para dormirte en tus laureles. Asegúrate de medir la felicidad de acuerdo con el desempeño; si hay divergencias, preparate para actuar. La complacencia es enemiga del éxito.

Capítulo ocho

Prioridades

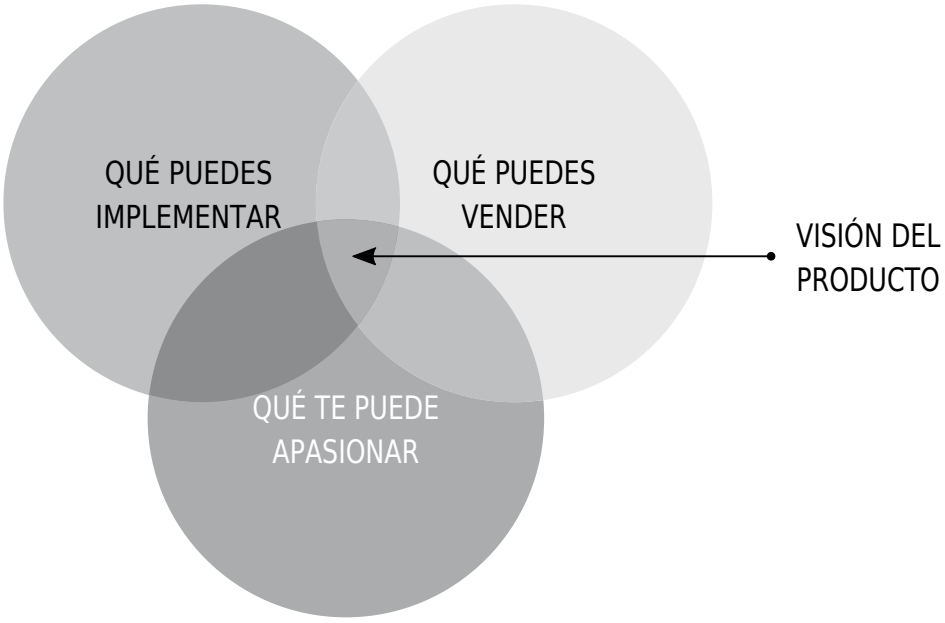
Conocí hace unos años a Scott Maxwell en Johnny's Luncheonette, en Newton Center. Ya te he hablado de él. Fundó OpenView Venture Partners y es quien comprendió que trabajar más tiempo crea más problemas de los que resuelve. He trabajado casi ocho años con OpenView y las compañías de su cartera de clientes, y en todas ellas hemos visto drásticos aumentos de productividad. Sin embargo, Scrum no se reduce a hacer que los equipos avancen más rápido. También tiene que ver con una influencia mayor, lo que, en el caso de la gente de capital de riesgo, adopta una forma muy simple: utilidades. Si una compañía no gana dinero, no tienes una empresa de éxito; tienes un pasatiempo.

No puedo decirte cuántas empresas he visto con grandes ideas y un producto excelente, algo que entusiasma a todos, que parece llenar un nicho del mercado y que aparentemente *debería* tener éxito de tan bueno que es. Pero pese a megadosis de imaginación, inspiración y trabajo intenso la gente que hace el producto no puede saber cómo ganar dinero con él.

¿Cuál es la diferencia entre Pets.com y Zappos? Ambas vieron un segmento del mercado en el que la gente gastaba miles de millones de dólares al año. Ambas vieron una manera de ofrecer productos en línea más fácil y económicamente. Una de ellas se volvió emblemática del exceso de empresas de internet y del derroche de millones y millones de dólares; la otra vale en la actualidad más de mil millones. Ambas tuvieron visión; lo que Pets.com *no* tuvo fue una noción de sus prioridades. No sabía qué hacer en qué momento.

Me gusta mostrar a la gente el diagrama de Venn que se presenta en la página siguiente.

EL RESPONSABLE DEL PRODUCTO DEBE EQUILIBRAR MÚLTIPLES ATRIBUTOS DEL PRODUCTO



Cada compañía debería reflexionar en este diagrama. Si sólo te centras en lo que puedes hacer, quizás termines haciendo algo que nadie quiere, aun si te apasiona. Si sólo te concentras en lo que puedes vender, podrías ofrecer cosas que en realidad no puedes hacer. Si sólo haces lo que puedes vender pero no te apasiona, terminarás trabajando empeñosamente para crear mediocridad. Pero en el centro, en ese punto grato, está una visión fundada en la realidad: una visión con una posibilidad real de convertirse en algo grande. En este capítulo te enseñaré cómo conseguirlo. Los capítulos anteriores se han ocupado de cómo hacer las cosas más rápido y mejor; éste capítulo tratará de cómo hacer que «lo más rápido y mejor» trabaje en tu beneficio: cómo alcanzar la grandeza.

Scott Maxwell dice que el verdadero poder de Scrum radica en sus Pendientes listos, priorizados y evaluados. Por eso implementó ese enfoque en su grupo de capital de riesgo y por eso lo considera una ventaja competitiva fundamental.

Pendientes: qué hacer y cuándo

Lo primero que debes hacer al poner en ejecución Scrum es elaborar una lista de Pendientes. Puede constar de cientos de elementos, o contener las pocas cosas que debes resolver primero. Por supuesto, necesitas una idea clara de qué quieres tener al final del trabajo. Podría ser un producto, una boda, un

servicio, una nueva vacuna o una casa pintada. Podría ser cualquier cosa, pero una vez que tienes una visión, debes considerar qué implicará hacerla realidad.

Una compañía con la que trabajé recientemente fabrica sistemas automatizados de calefacción, ventilación, electricidad y plomería de edificios, todo junto. Uno de sus nuevos productos es un sistema automatizado para el hogar. Esa compañía fabricó un sistema capaz de controlar todos los aspectos de una casa, desde la apertura del portón hasta el control de costos de calefacción y el encendido de luces, todo desde un dispositivo portátil. Así, sus ejecutivos se reunieron para hacer una lista de lo que necesitaban para hacer posible ese sistema: interruptores, controladores, interfaces, sensores, protocolos de comunicación… No las reglas y piezas específicas, sino todas las «historias» que necesitarían.

Escribieron cosas como «En mi calidad de propietario de una casa, me gustaría poder ver quién llama a mi puerta para abrirla únicamente a quien quiero que entre». Escribieron historias sobre la apertura de la puerta de la cochera, el encendido del HVAC y el control de las luces. Y no dejaron de escribir hasta tener una lista de todo lo que creían que el sistema debía hacer para resultar una adquisición atractiva.

La lista terminó constando de cientos de elementos. Ése es un sistema grande y complicado. La idea en que se basan los Pendientes es que deben contener todo lo que el producto podría incluir. En realidad no vas a fabricarlo todo, pero necesitas una lista de lo que podría incluir esa visión del producto.

La clave, sin embargo, es lo que decides hacer primero. Las preguntas que debes hacer son: ¿cuáles son los elementos con mayor efecto de negocios, los más importantes para el cliente, los que pueden producir más dinero y los más fáciles de hacer? Debes percatarte de que hay muchos elementos en esa lista que nunca conseguirás, pero has de conseguir primero los que rinden más valor con menos riesgo. Dado el desarrollo y cumplimiento incremental de Scrum, comienza por las cosas que generarán ingresos inmediatos, librando de riesgos el proyecto. Y haz eso en el nivel de funciones. Empieza a ofrecer valor a tus clientes en cuanto sea posible. Necesitas algo que esté completamente Terminado, algo que puedas enseñar. Podría ser una pequeña parte del gran proyecto, pero debe estar demostrablemente Terminada. Si vas a pintar una casa, quizá lo Terminado en primera instancia sean todas las molduras de la sala.

En el desarrollo de productos existe una regla empírica eficaz, confirmada una y otra vez. Ya me he referido a ella: ochenta por ciento del valor radica en veinte por ciento de las funciones. Piensa en esto un momento. En todo lo que compras, la mayor parte del valor —la mayor parte de lo que la gente necesita — está en sólo una quinta parte de lo fabricado. En el caso que nos ocupa, la compañía analizó esa inmensa lista de elementos que su sistema de automatización para el hogar podía incluir, y supo —lo supo de verdad— que los clientes sólo querían veinte por ciento. El truco de Scrum es saber cómo hacer primero ese veinte por ciento. En el desarrollo de productos tradicional, los equipos no saben cuál es ese veinte por ciento hasta que entregan el objeto completo. Esto quiere decir que nada menos que ochenta por ciento de su esfuerzo es desperdicio. Y ya sabes qué pienso del desperdicio.

¿Y si pudieras empezar ofreciendo cosas cinco veces más rápido que la competencia, con cinco veces más valor? Sería un éxito inicial total.

La compañía automatizadora tomó una enorme lista de funciones ante la que sus miembros se preguntaron: «¿Qué haremos mañana? ¿Qué es lo más importante para el cliente? ¿Cómo podemos ofrecerle valor más rápido que nadie más?». Como dice Scott Maxwell, la parte difícil no es saber *qué* quieres, sino qué *puedes lograr*. Esto es cierto así construyas una casa o fabriques un auto, escribas un libro o un videojuego o recojas los elementos de una escena del crimen o basura. Deduce qué puede ofrecer más valor con menos esfuerzo y haz eso de inmediato. Luego identifica el siguiente incremento de valor y después el siguiente. Antes de lo que crees, habrás creado u ofrecido algo con resultados reales y demostrables. La clave es disponer el trabajo por orden de prioridad.

¿Cómo hacerlo? Primero necesitas a alguien que sepa cuál es la visión y dónde radica el valor. En Scrum esa persona se llama «responsable del producto».

El responsable del producto

En Scrum hay únicamente tres papeles. Eres parte del equipo y haces el trabajo, o eres el *Scrum Master*, ayudando al equipo a saber cómo hacer mejor el trabajo, o el responsable del producto. Este último decide qué hacer. Está a cargo de los Pendientes, su contenido y, sobre todo, el orden de éste.

Cuando formé el primer equipo con Scrum, en 1993, no tenía un responsable del producto. Yo formaba parte del equipo de liderazgo y tenía

muchas otras responsabilidades aparte de determinar qué debía hacer el equipo en cada *sprint*. Realizaba labores gerenciales y de mercadotecnia, trataba con los clientes e ideaba la estrategia. Pero supuse que podría manejar los Pendientes en el primer *sprint*. Sólo tenía que confirmar que dispusiera de «historias» y funciones suficientes en las que el equipo trabajara en el siguiente *sprint*. El problema fue que, después del segundo, introdujimos la reunión de parada diaria. Nuestra velocidad aumentó cuatrocientos por ciento en el *siguiente sprint*, así que el equipo acabó en una semana lo que creímos que nos llevaría un mes. ¡Ya no había más Pendientes! Creí que tendría un mes para crear más «historias». Un gran problema, claro, pero había que atacarlo. Entonces concebí el papel de responsable del producto y las cualidades que se necesitarían para desempeñar esa función en forma adecuada.

Me inspiré a este respecto en el jefe de ingeniería de Toyota. En esta organización, un jefe de ingeniería es responsable de una línea de productos completa, como la de Corolla o Camry. Para ejercer esta función, quienes la asumen deben echar mano del talento de grupos especializados en ingeniería de carrocería, o chasís, o sistema eléctrico y demás. El jefe de ingeniería debe apoyarse en todos esos grupos para crear un equipo interfuncional capaz de fabricar un auto. Fuera de Toyota, todos creen que los ya legendarios jefes de ingeniería (o *shusas*, como se les llamó originalmente) son líderes omnipotentes del «estilo Toyota». Y en cierto sentido lo son. Pero no tienen autoridad. Nadie responde a sus órdenes; más bien, ellos responden a las de sus grupos. La gente puede decirles que están equivocados, así que deben asegurarse de tener razón. No destinan a nadie evaluaciones de desempeño, ascensos ni aumentos. Pero deciden la visión del auto y cómo se fabricará, mediante la persuasión, no por coerción.

Yo quise incorporar esa idea en Scrum. John Shook, del Lean Enterprise Institute, comenzó una vez su descripción del papel del jefe de ingeniería citando el manual de liderazgo del cuerpo de Marines de Estados Unidos:

> La responsabilidad de liderazgo de un individuo no depende de autoridad. […] La arraigada suposición de que la autoridad equivale a responsabilidad es la causa de muchos males organizacionales. Los malos entendidos en torno a este asunto son endémicos y problemáticos, y están tan fijos en nuestra conciencia que ni siquiera nos percatamos de ellos[39].

Al reflexionar en mi época de West Point y Vietnam, me descubrí aceptando que el liderazgo no tiene nada que ver con la autoridad. Tiene que ver más bien con —entre otras cosas— el conocimiento y con ser un líder de servicio. El jefe de ingeniería no puede *decir* que algo ha de hacerse de tal o cual forma. Debe convencer, lisonjear y demostrar que su propuesta es la correcta, la *mejor*. Por lo general esta función corre a cargo de alguien con treinta años de experiencia. Yo quería eso en Scrum, pero sé que muy pocas personas poseen tal nivel de habilidad y experiencia. Así, dividí el papel en dos, otorgando al *Scrum Master* el *cómo* y al responsable del producto el *qué*.

Aun en aquellos primeros días de Scrum, yo sabía que necesitaba a alguien estrechamente relacionado con el cliente. El responsable del producto debía ser capaz de proporcionar al equipo realimentación del cliente en todos y cada uno de los *sprints*. Tenía que dedicar la mitad de su tiempo a hablar con las personas que compran el producto (para conocer su opinión sobre el lanzamiento incremental más reciente y su valor) y la otra mitad a crear con el equipo los Pendientes (para mostrarle lo que los clientes valoraban y lo que no).

Recuerda que el «cliente» puede ser el consumidor general, un gran banco, tu esposo o alguien que necesita la vacuna contra el rotavirus y depende de ti para conseguirla. El cliente es cualquier persona que obtendrá valor de lo que tú haces.

Pero yo no quería a un gerente. Quería alguien en quien el equipo creyera y confiara cuando ordenara los Pendientes por prioridad. Así, conseguí al sujeto más inteligente de mercadotecnia; no de ingeniería, sino de mercadotecnia. Y fue así como Don Rodner se convirtió en el primer responsable del producto. Él conocía el producto que estábamos haciendo no desde un punto de vista técnico —aunque sabía lo suficiente de eso para comunicarse con los ingenieros—, sino desde el punto de vista del cliente. ¿Qué necesitaba la gente que usaría el producto? Cuando elijas a un responsable del producto opta por alguien que pueda meterse en la mente de quien obtiene valor de lo que haces. Como dice un amigo mío: «Mi esposa es la perfecta responsable del producto: sabe justo lo que quiere. Yo me limito a ponerlo en práctica».

El responsable del producto no sólo necesita una gama de habilidades más amplia que el *Scrum Master*, sino que también debe apegarse a una serie de normas diferente. El *Scrum Master* y el equipo están a cargo de lo rápido que avanzan y de la mayor rapidez que pueden alcanzar. El responsable del producto lo está de traducir la productividad del equipo en valor.

Al paso de los años, he reducido a cuatro las características esenciales de un responsable del producto:

Uno: debe conocer el terreno. Por esto entiendo dos cosas: comprender el proceso que el equipo recorre para saber qué puede hacerse e, igualmente, qué no. Pero también conocer el *qué* para saber cómo traducir lo que puede hacerse en valor verdadero y significativo. Esto podría ser un sistema de cómputo que ayude al FBI a atrapar terroristas o un método de enseñanza para mejorar el aprovechamiento de los alumnos de escuelas públicas. El responsable del producto debe conocer bien el mercado para saber qué marcará una diferencia.

Dos: debe tener autoridad para tomar decisiones. Así como la dirección no ha de interferir con el equipo, el responsable del producto debe disponer de margen para tomar decisiones sobre cuál será la visión del producto y qué debe hacerse para realizarla. Esto es importante porque el responsable del producto se halla bajo presión de muchos interesados, tanto internos como externos, y debe ser capaz de mantenerse firme. Debe estar a cargo de los resultados, pero se le ha de permitir tomar decisiones.

Tres: debe estar a disposición del equipo para explicar qué se debe hacer y por qué. Aunque está a cargo en definitiva de los Pendientes, ha de mantener un diálogo constante con el equipo. A menudo la experiencia de este último determinará las decisiones que el responsable del producto ha de tomar. Este individuo debe ser confiable y congruente y estar a disposición del equipo. Sin acceso a él, el equipo no sabrá qué hacer o en qué orden llevarlo a cabo. Depende del responsable del producto respecto a «la visión» y, asimismo, la inteligencia del mercado sobre qué es importante. Si el responsable del producto no está a disposición del equipo, todo el proceso puede venirse abajo. Por ello rara vez recomiendo como ejecutores de esta función a directores generales u otros altos ejecutivos, quienes no cuentan con el tiempo que el equipo necesita.

Cuatro: el responsable del producto debe hacerse cargo del valor. En un contexto de negocios lo que importa son los ingresos. Yo mido a un responsable del producto por los ingresos que produce por «punto» de esfuerzo. Si el equipo produce cuarenta puntos de trabajo a la semana, debo medir cuántos ingresos crea cada punto. Pero la medición de valor podría ser cuántos éxitos tiene un equipo. Sé de un equipo de representantes de la ley que medía el valor por el número de arrestos de criminales buscados que hacía cada semana, y de iglesias que usan Scrum y miden su éxito por lo bien que sirven a su comunidad y si crecen o no. La clave es decidir cuál es la

medida de valor y hacer que el responsable del producto asuma el deber de ofrecer cada vez más de él. En Scrum, este tipo de medida es fácil de observar a causa de la increíble transparencia del método.

Todo esto es mucho pedir a una sola persona y por eso en proyectos grandes suele haber un *equipo de responsables del producto* que se ocupan de todas las necesidades. Abundaré en esto más adelante. Primero, para visualizar lo que el responsable del producto debe hacer, me gustaría que te imaginaras en la cabina de un F-86 Sabre al lado del «Comandante Demandante» John Boyd, quien está a punto de entrar en feroz batalla en el espacio aéreo de la península de Corea.

Observa, oriéntate, decide, actúa

Durante la guerra de Corea, el combate aéreo tuvo lugar principalmente entre F-86 Sabres estadunidenses y MiG-15s de factura rusa. El MiG era más rápido, más maniobrable, con mayor proporción propulsión-peso y era en todos los órdenes un avión superior. En el papel, los MiG-15 debían haber arrasado con los pilotos estadunidenses. En cambio, fueron derribados en una proporción de 10:1. El empeño de saber qué pasó determinó el futuro de la guerra en general y fue crítico en el desarrollo de Scrum.

John Boyd es sencillamente el mayor piloto de caza de la historia, aunque jamás abatió a un enemigo en combate. Cumplió únicamente veintidós misiones en Corea antes del armisticio y en ese entonces se precisaba de treinta como piloto para ser considerado un «tirador». Fue después de la guerra, como maestro de la USAF Weapons School, en la base de la fuerza aérea de Nellis, en el sur de Nevada, que dejó huella. En una rama del ejército que valora la frecuente rotación de personal, Boyd pasó como instructor un número de años sin precedente: seis.

Los pilotos de caza no son personas humildes. Cuando aparecen en Nellis, se les considera ya los mejores pilotos de la USAF y exhiben cierta petulancia. Boyd sabía cómo destruir infaliblemente el ego de un piloto para que aprendiera lo que él debía enseñarle. Los invitaba a volar y hacía que el alumno lo hiciera en sus «seis», justo detrás de él, la mejor posición en una batalla aérea. Luego le instruía trabar combate. Sin falta, en menos de cuarenta segundos asestaba un disparo mortal cerca del alumno, mientras en el radio gritaba «¡Armas! ¡Armas! ¡Armas!». Esto era antes de que hubiera rayos láser, computadoras y armas simuladas. Ese grito hacía saber al alumno

que estaba muerto. El éxito infalible de Boyd en esta maniobra le valió un segundo apodo, «Cuarenta Segundos» Boyd.

Su otro sobrenombre era «Comandante Demandante», mote que se ganó a causa de sus enérgicas declaraciones. Casi siempre las emitía con la cara a menos de diez centímetros de quien lo contradecía mientras le hundía dos dedos en el pecho. Infaliblemente, en esos dos dedos reposaba un puro Dutch Masters encendido. Según la leyenda, en ocasiones —por accidente, estoy seguro— prendía fuego a la corbata de su adversario. No mostraba contrición en tales momentos, sirviéndose de cualquier arma en su arsenal para ganar una discusión.

Boyd tenía la habilidad de percibir todo el espacio de batalla. Como lo dijo en un relato oral:

> Me veía en una bola inmensa —*dentro* de ella— y podía visualizar todas las acciones que ocurrían ahí mientras, por supuesto, no dejaba de maniobrar. […] Podía visualizar esas imágenes desde dos puntos de referencia. Cuando combatía en el aire, *podía verme como un observador objetivo mirándome a mí mismo y a todos a mi alrededor*[40].

Este tipo de conciencia, la capacidad para advertir una esfera completa de cielo y ver desenvolverse los acontecimientos, dio forma a sus teorías militares y reescribió a la larga la manera de librar guerras de Estados Unidos.

Cuando Boyd dejó la Weapons School, decidió estudiar ingeniería, y mientras lo hacía creó un modelo de desempeño de las aeronaves que describía el combate aéreo en términos de relaciones de energía. La teoría de la maniobrabilidad de la energía (ME) toma en cuenta la energía cinética y potencial de una nave en cualquier situación, su altitud, velocidad relativa de vuelo y dirección, y qué tan rápido puede cambiar cualquiera de esas variables. Esta teoría se incorporó más tarde al modelo de la mayoría de los aviones de combate, lo que condujo directamente al desarrollo del F-15 y el F-16, los cazas dominantes de los últimos cuarenta años.

Según la teoría de Boyd, el problema era que el MiG-15 debía haber arrasado con el F-86 Sabre. No era lógico que no hubiese sido así. De acuerdo con su biógrafo, Robert Coram, Boyd cayó en trances frecuentes durante varios días mientras intentaba explicárselo. Estaba seguro de que su teoría era correcta, pero ¿cuál era el motivo de la proporción de abatimiento de 10:1 obtenida por los pilotos estadunidenses? ¿Su preparación? Esto sólo podía explicar una parte de ello. ¿La táctica? Tal vez, pero tampoco de este factor

podía derivarse un resultado tan disparejo. Por fin reparó en que los pilotos estadunidenses veían mejor y actuaban más rápido. No gracias a una cualidad inherente a ellos, sino a decisiones simples de diseño. El Sabre tenía una cubierta de burbuja, mientras que la del MiG constaba de múltiples riostras y hojas de cristal que bloqueaban la visión del piloto. El F-86 tenía asimismo controles de vuelo de propulsión totalmente hidráulica, en tanto que el MiG sólo tenía controles hidráulicamente *asistidos*. Se sabía que sus pilotos hacían pesas para tener fuerza suficiente en el torso para maniobrar la nave.

En consecuencia, los pilotos estadunidenses podían ver los MiG primero y luego, críticamente, proceder con base en esa información más rápido que los chinos y norcoreanos. La batalla no se decidía por lo que las máquinas podían hacer, sino por la rapidez con que la observación se traducía en acción. El MiG actuaba, la nave estadunidense respondía y mientras el piloto de aquél trataba de reaccionar, el estadunidense podía actuar. Respondía tan pronto a cada nueva posición del MiG que éste, tecnológicamente más avanzado, terminaba siendo presa fácil.

El mismo fenómeno ocurrió en Vietnam cuando yo estuve ahí. Para entonces ya había aviones diferentes en la liza, el MiG-21 y el F-4, pero también esta vez la visibilidad superior del F-4 venció a la superior maniobrabilidad del avión de factura soviética. Como diría Boyd, su innovación más famosa llevó a los pilotos «dentro del ciclo de decisión del enemigo».

Este discernimiento se volvió fundamental para el ejercicio de la guerra. Y es justo para lo que yo diseñé Scrum: para permitir que el responsable del producto tome decisiones rápidamente, a partir de realimentación en tiempo real. Al recibir aportaciones constantes de quien obtiene valor de lo que tú haces —sea la persona que hace clic en el botón Buy (Comprar) en Amazon, los parroquianos de tu iglesia, los niños en un aula o alguien que se prueba un vestido—, estás en posición de ajustar incesantemente tu estrategia y de triunfar más pronto.

Esta idea responde un tanto al ridículo nombre de «ciclo OODA», siglas de **O**bserva, **O**riéntate, **D**ecide y **A**ctúa. Y aunque puede parecer gracioso al oírlo, es mortal en la guerra y los negocios. Introducirse en el ciclo de alguien lo somete a confusión y duda. Reacciona de más y de menos. Como dijo Boyd en una instrucción que impartió a otros oficiales, «sobrevive quien puede manejar más rápido la razón de cambio»[41].

«Observa» es muy obvio; consiste en mirar la situación mientras se desenvuelve. Pero esto no es tan sencillo como parece. Boyd lo describió

como salir de ti para adquirir una visión panorámica, no sólo la abarcada por tu perspectiva.

«Oriéntate» no se refiere exclusivamente a dónde estás, sino también a qué resultados puedes ver, el menú de opciones que creas para ti. De acuerdo con Boyd, factores de esa creación son la herencia genética, las tradiciones culturales, las experiencias previas y, desde luego, las circunstancias en desarrollo. De este modo, la Orientación refleja no sólo cómo ves el mundo y tu lugar en él, sino también qué mundo eres *capaz* de percibir.

La mezcla de Observación y Orientación conduce a una «Decisión» que deriva en «Acción». En ese momento, el ciclo se reinicia con la Observación de los resultados de tus acciones y las de tu adversario o, en el mundo de los negocios, con la de la reacción del mercado.

Al rendir un incremento funcional, Scrum concede al responsable del producto la posibilidad de ver cuánto valor genera ese incremento y cómo reacciona la gente a él. Luego, a partir de esa información, puede hacer modificaciones en lo que hará el equipo en el siguiente *sprint*. Esto forma un ciclo de realimentación constante que acelera la innovación y la adaptación y permite al responsable del producto calcular cuánto valor se ofrece. (En los negocios medimos eso con dinero. Si pinto una casa, podría medirlo con habitaciones terminadas). Así, el responsable del producto puede ajustar las cosas sobre la marcha a un mundo en cambio permanente.

Quizá sea difícil imaginar lanzamientos incrementales de productos o proyectos que no parecen tener valor hasta que se les concluye. Por ejemplo, ¿cómo hacer lanzamientos incrementales de un automóvil? ¿O de un videojuego de cien millones de dólares? La clave es determinar qué fragmentos poseen valor, el suficiente para conseguir realimentación sobre ellos y reaccionar en tiempo real.

Tomemos los autos como ejemplo. Toyota desarrolló el Prius, desde su concepción hasta su entrega, en quince meses, más rápido que cualquier otro vehículo que haya hecho. Aunque el equipo que diseñó y fabricó ese coche no empezó a venderlo antes de acabarlo, pronto comenzó a hacer prototipos para que el jefe de ingeniería pudiera «patear las llantas» y ver si el equipo seguía la dirección correcta. Esta rapidez en la elaboración de prototipos, consistente en hacer vehículos totalmente funcionales antes de lanzarlos y en mejorar tales prototipos una y otra vez hasta tener el producto que se desea vender a los consumidores, puede derivar en cambios increíblemente veloces. La clave es no tener un diseño totalmente establecido desde el principio, sino hacer un prototipo funcional y ver en qué puede mejorar. Ya que se ha mejorado, se

procede a hacer el siguiente prototipo y a mejorarlo. La idea es que cuanto más pronto tengas realimentación real, más rápido podrás hacer un mejor coche.

WIKISPEED, a la que me referí en el capítulo cuatro, produce prototipos completos de sus vehículos *cada semana*. Y los vende. Estas transacciones no ocurren en un mercado masivo —el equipo de WIKISPEED aún no está preparado para eso—, pero hay adeptos tempranos dispuestos a desembolsar veinticinco mil dólares por esos prototipos iniciales. Cuando piensas en fabricar algo, no supones que no podrás ofrecer algo valioso hasta el final. En cambio, tratas de pensar en el producto mínimo viable. «¿Qué es lo menos que puedo hacer sin dejar de ofrecer algo de valor a un cliente?».

Los videojuegos son otro buen ejemplo. Hoy en día, cada vez más desarrolladores permiten que adeptos tempranos paguen acceso «alfa» anticipado. De esa manera, reciben realimentación de sus fans más devotos antes de que el juego esté en plena operación. Esto les permite ver cómo reacciona la gente, en vez de tratar de adivinar cómo lo hará.

Según la industria a la que perteneces o de la organización que diriges, podría ser difícil hacerte a la idea de lanzamientos incrementales. De ser necesario, si acaso no puedes poner algo frente a un cliente *externo*, identifica a un cliente *interno* —el responsable del producto, por ejemplo— que pueda actuar en sustitución del público. Enséñale a tu cliente interno cualquier cosa que despierte realimentación útil: fragmentos de un plan de expansión inmobiliaria, el diseño de actualización de una fábrica, la reconstrucción de un sistema de frenos, una campaña de servicio voluntario o lo que sea. La idea es darte la oportunidad de inspeccionar y adaptar. Una compañía u organización que no puede reaccionar a nuevas condiciones, competidores o gustos está en dificultades.

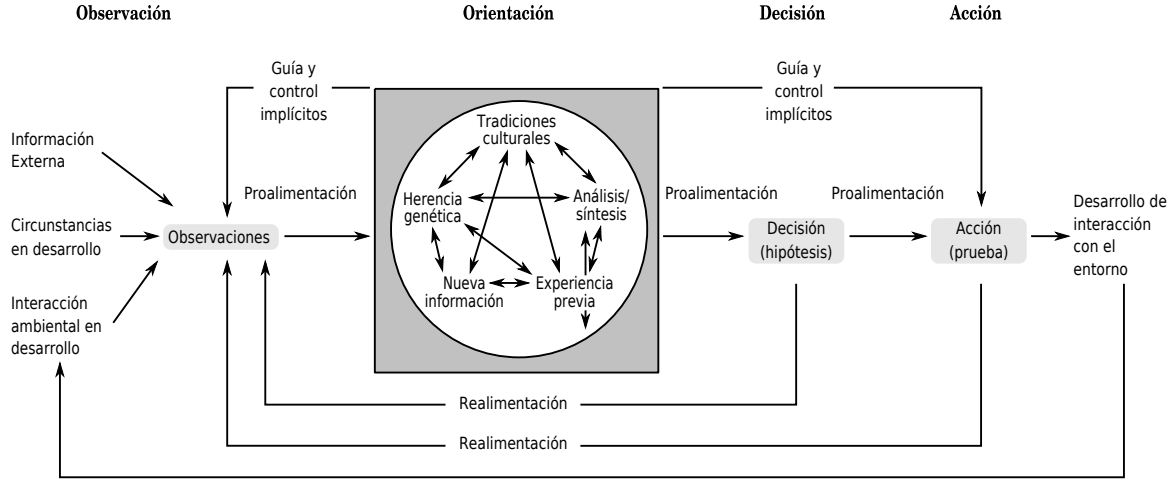
Boyd lo explica de esta manera:

> Debemos entrar en el *tempo* o ritmo del otro, donde lo abatiremos. […] Hemos de tener una imagen o foto en nuestra cabeza, que llamamos orientación. Luego debemos tomar una decisión sobre lo que vamos a hacer y después ejecutarla. […] Tras considerar la acción [resultante], además de nuestra observación, surgen nuevos datos, nueva orientación, nueva decisión y nueva acción *ad infinitum*. […] Orientación no es sólo el estado en que te encuentras; también es un proceso. Siempre te estás orientando. […]

> Un pequeño mundo, bello y ceñido, donde nada cambia. […] [Las criaturas que viven en él son] dinosaurios y morirán. La clave es no ser un dinosaurio. Si estás en una condición de equilibrio, estás muerto. […] El mensaje de fondo es simple: no hay salida. […] Así es esto, chicos[42].

Así es esto, chicos. Como dije en el capítulo uno, tienes frente a ti una decisión radical: renovarte o morir. Si no te introduces en el ciclo de decisión de la competencia, ella se introducirá en el tuyo. Como dijo Boyd, «lo que debo hacer es embrollar al adversario. […] Entonces puedo sumirlo en confusión y desorden y causarle parálisis». No sé tú, pero yo preferiría estar en el extremo emisor de eso, no en el receptor.

CICLO OODA



Lo primero es lo primero

Tienes entonces un responsable del producto que actualiza constantemente los Pendientes, ordenando las cosas por hacer y concluir. Cuando tienes un centenar de elementos, ese proceso de ordenamiento puede complicarse pronto. La clave es saber cómo ofrecer el máximo valor tan rápido como sea posible. Puede haber millones de formas de disponer esos Pendientes, pero necesitas la que brinda lo más pronto posible el veinte por ciento de funciones con ochenta por ciento del valor. Es casi seguro que tu primer cálculo en el primer *sprint* sea incorrecto, pero será tu mejor estimación en esas condiciones.

Sin embargo, ése es sólo tu primer cálculo. Tras el primer *sprint*, una vez que completes el ciclo OODA y entregues algún producto a los clientes, cambiarás ese orden, comprendiendo que es preferible otro acomodo.

Y seguirás así, actualizando y repriorizando continuamente los Pendientes en cada *sprint*, tendiendo al orden que ofrece más rápido valor. Quizá nunca alcances el orden absolutamente perfecto, pero debes perseguirlo paso a paso, *sprint* por *sprint*.

La clave es recordar que ese orden siempre está en estado de flujo. El orden correcto una semana no lo será a la siguiente. Tu entorno habrá cambiado. Tú habrás aprendido nuevas cosas. Habrás descubierto que algunas son fáciles y otras difíciles. Así, la alteración constante del orden de los Pendientes sucede en cada *sprint*. La clave es admitir la incertidumbre, aceptar plenamente que tu imagen de momento del orden y el valor sólo es relevante en un lapso particular. Volverá a cambiar, una y otra vez.

Un mal hábito que una compañía podría adoptar, a causa del cambio constante de las necesidades del mercado y de que los gerentes no saben con exactitud dónde reside el valor máximo, es priorizarlo todo. Todo es de alta prioridad. El adagio por tener en mente es de Federico II de Prusia, más tarde llamado «el Grande»: «Quien defiende todo no defiende nada». Al no concentrarte en tus recursos y en tu energía mental, los reduces a la irrelevancia.

Hace unos años cumplí setenta y lo celebré en Normandía, Francia. Fui a conocer entonces la famosa playa en la que mi padre desembarcó durante la invasión del Día D. Con marea baja, la playa de Omaha parece tenderse en declive a lo largo de varios kilómetros antes de hundirse en el distante mar, un tramo de arena aparentemente interminable. Subir esa pendiente húmeda y larga frente a las armas alemanas debe haber requerido un valor que sólo es posible imaginar. Recorrer las tumbas de los miles que murieron ese día impone silencio y respeto. Pero cuando me puse a leer acerca de las defensas alemanas, comprendí que una de las razones del éxito del desembarco estadunidense fue que los alemanes olvidaron la admonición de Federico el Grande. Las fintas de los aliados los habían confundido tanto que desplegaron sus fuerzas en toda la costa de Francia. En consecuencia, los aliados pudieron aislar cada unidad germana, derrotándola y pasando a la siguiente. Los alemanes no priorizaron como debían, por lo que lo perdieron todo.

Lanzamiento

Tienes tus prioridades. Sabes dónde radica ochenta por ciento del valor. ¿Cuándo entregarás tu producto? Aquí es donde Scrum puede producir valor en forma radicalmente más rápida. Cada vez que haces algo debes ponerlo lo más pronto posible en manos de quienes van a usarlo. Tienes que hacerlo aun antes de producir el veinte por ciento de las funciones, y hacerlo con algo que ofrezca al menos un *poco* de valor. Llamo a esto producto mínimo viable (PMV). Eso es lo que debes mostrar al público por primera vez. ¿Cuán eficaz debe ser? Bueno, debe funcionar, aunque a alguien que haya trabajado en él le parezca bochornoso. ¡Debes presentarlo en público *lo más pronto posible*! Esto te brindará la realimentación necesaria para potenciar tu ciclo de decisión y priorización. Es la versión 0.5. Una cámara que puede tomar fotos pero no enfocar. Un comedor con dos sillas. La distribución de una vacuna a cinco entre el centenar de pueblos a los que quieres socorrer. Algo casi risiblemente incompleto.

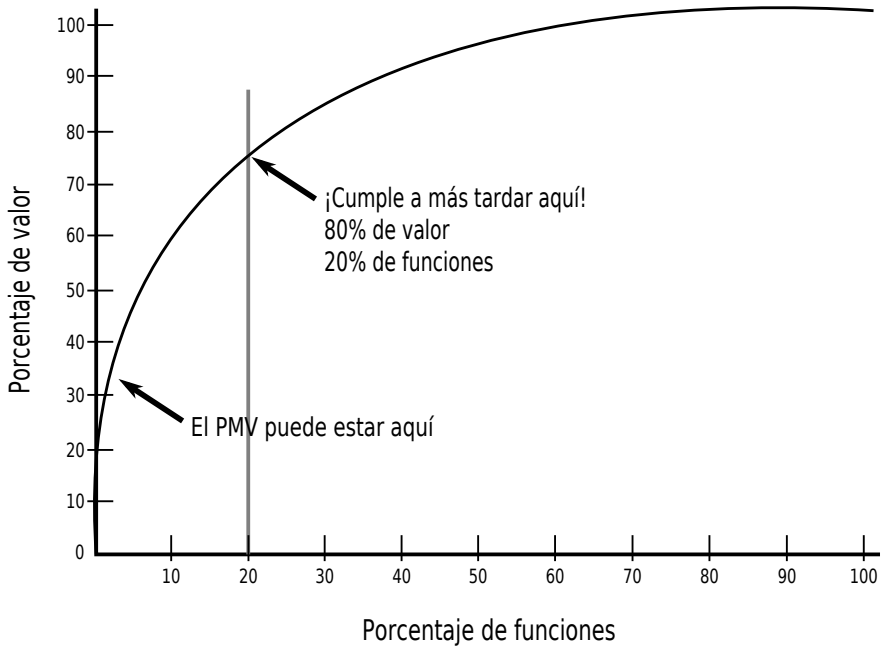
Pero lo que eso consigue para ti es *realimentación*. El cuerpo de la cámara es incómodo de asir porque el botón del obturador está en un lugar extraño. La madera de las sillas no coincide con la de la mesa. Acabas ofendiendo a los ancianos del pueblo a causa de un paso en falso totalmente evitable. Comete pronto estos errores, con el menor daño posible.

Luego, al hacer el lanzamiento oficial o la presentación de un gran programa, ya habrás ajustado y descubierto lo que la gente realmente valora. En nuestro ejemplo de la cámara, tal vez resulte que los fotógrafos *dijeron* que disponer de un modo apaisado y poder compartir fotos en Facebook eran aspectos igualmente valiosos, pero cuando la *usaron* no emplearon nunca el modo apaisado y en cambio siempre querían publicar fotos en Facebook.

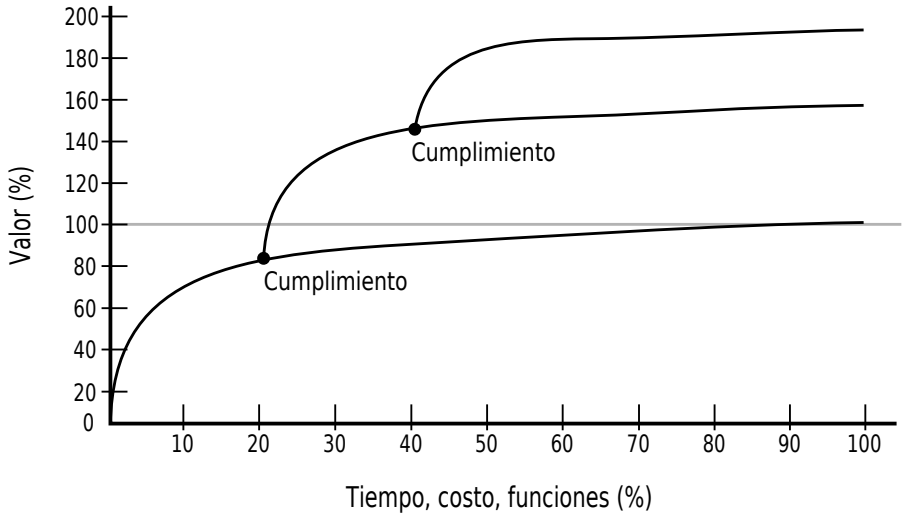
Esto te permite producir primero las funciones que ellos valoran y lanzar tu producto cuando sólo has hecho veinte por ciento del trabajo. Sabes que no es perfecto, pero se acerca. Cada hora dedicada a pulir la manzana es una oportunidad de valor perdida.

Lo maravilloso de este proceso es que es iterativo: sólo «remoja y repite». Una vez que la gente tenga tu producto, servicio o cambio, te dirá cuáles son las *siguientes* cosas más valiosas. Desarrolla entonces veinte por ciento de eso y vuelve a presentarlo. Y así sucesivamente.

CURVA DE VALOR: CUMPLIMIENTO RADICALMENTE *MÁS RÁPIDO*



COMPOSICIÓN DE VALOR: CUMPLIMIENTO RADICALMENTE *MEJOR*



Dado este proceso de lanzamientos incrementales, cuando hayas creado la mitad de las funciones de tu producto o proyecto inicial habrás lanzado doscientos por ciento del valor, en la mitad de tiempo. Éste es el verdadero poder de Scrum. Es como puede modificar fundamentalmente no sólo tu manera de trabajar, sino también de vivir. No te obsesiones con cumplir una lista completa de cosas —con todo y el fregadero de la cocina—; céntrate en cumplir lo más valioso: lo que la gente de veras quiere o necesita.

Esto me recuerda historias de Iraq o Afganistán. Un pelotón estadunidense llega a una ciudad, ve a su alrededor y dice: «Esta gente cría pollos.

Hagámosle una planta procesadora». Invierte entonces millones de dólares en la construcción de una avanzada planta de pollos. Pero no toma en cuenta que ahí casi no hay electricidad regular, o que la mayoría de la gente es analfabeta y no será fácil capacitarla para manejar el equipo. Entonces alguien va a la ciudad y pregunta a los lugareños: «¿Qué les sería *realmente* útil?», y ellos contestan: «Un puente para cruzar el río, para no tener que perder medio día yendo al cruce más cercano para ir al mercado». Ese puente cuesta un centenar de dólares. Parece mucho menos impresionante que una gran planta. No semeja nada espectacular al hablar de él con los jefes en Washington. Pero para esa gente es infinitamente más valioso que el magnífico edificio con máquinas que ya se están oxidando.

Otra cuestión que vale la pena mencionar es que a veces terminas antes. Supongamos que estás haciendo un superreloj con alarma de nueva generación para Alarm Clock Inc. Tienes una lista de docenas de funciones: reloj, repetición, botón, temporizador, alarma fuerte, radio, base para iPhone, GPS y demás. Pero como eres un hábil responsable del producto, priorizas lo que la gente desea: una alarma fácil de programar, volumen suficiente, un radio y una pantalla lo bastante vívida para poder verla ya sea que la habitación esté iluminada o a oscuras. Y cuando tu equipo termina eso, te percatas de que creó el reloj con alarma más elegante que haya habido nunca. Es el iPod de Apple de los relojes con alarma. Es bonito y hace una cosa muy, muy bien. Así que en vez de poner a tu equipo a incorporar funciones adicionales, lanzas ese reloj y te pones a trabajar en el proyecto siguiente. El equipo puede proporcionar más valor haciendo otra cosa.

Dinero por nada y modificaciones gratis

Al principio de este libro conté la historia del proyecto Sentinel del FBI. Como recordarás, un contratista externo gastó cientos de millones de dólares en producir *software* que no funcionó. Una de las mayores fuentes de exceso de costos en ese caso —y en casi todos los contratos, sean para producir *software*, aviones o edificios— son los honorarios por las modificaciones. La acumulación de esos honorarios es en realidad el modelo de negocios de un montón de contratistas del gobierno. Subvalúan un proyecto, sabiendo que obtendrán ganancias gracias a solicitudes de modificaciones. Cuando se elabora el contrato de un proyecto de varios años de duración, cuyos requerimientos se han plasmado en bonitas gráficas, resulta tentador decir:

«Bueno, esto lo cubre todo». Entonces el contratista replica: «Estoy aceptando esto y sólo esto. *Si quiere modificaciones, le costarán*». Esta facturación *a posteriori* se ha vuelto centro de tantos costos que compañías y agencias ya han establecido consejos de control de modificaciones. Desde el punto de vista de los costos, esto tiene sentido. Limita el número de modificaciones y limitarás el costo asociado a ellas.

Pero lo que los timoratos no perciben es que montan un sistema diseñado para negar a la gente lo que desea. Tratan de limitar costos, pero al hacerlo impiden el aprendizaje, la innovación y la creatividad. Si emprendes un proyecto y de pronto te das cuenta de que el valor real, ese veinte por ciento, no reside en las funciones que dispusiste —sino en otra serie de cosas que descubriste al hacer el trabajo—, la gestión de proyectos tradicional está hecha para detenerte, está hecha para impedir el rápido suministro de valor.

Además, ¡el esfuerzo de «ejercer firme control» no funciona! Aun si los consejos de control de modificaciones intentan limitarlas, su necesidad es tan grande que a menudo se ignora a aquéllos. Sin las modificaciones, el proyecto carecería de valor. A regañadientes, entonces, esos consejos las permiten y el costo del proyecto aumenta. Luego aparece otro cambio por hacer. Y después otro más. Pronto, el proyecto excede su presupuesto en millones de dólares y se retrasa uno, dos o cinco años.

Por eso se me ocurrió la idea de modificaciones gratis. En un contrato estándar de precio fijo, tu di sencillamente que los cambios son gratis. Enumera todas las funcionalidades que esperas; por ejemplo, si vas a fabricar un tanque, necesitas que pueda andar a ciento veinte kilómetros por hora y disparar diez rondas por minuto, con asientos para cuatro, AC, etcétera, etcétera: todo lo que crees que precisará. El fabricante revisa esa descripción y dice: «Hmm, yo diría que fabricar ese motor equivale a cien puntos, el mecanismo de carga a cincuenta, los asientos a cinco», etcétera, hasta llegar al final de la lista. Tendrá entonces un número fijo de puntos por cada función. Pero luego, en cada *sprint*, el cliente, quien en este escenario está contractualmente obligado a trabajar muy de cerca con el responsable del producto, puede cambiar por completo las prioridades. Cualquier elemento o función de los Pendientes puede ser movido a otro lugar. ¿Y nuevas funciones? Fácil: sólo elimina de la lista de pendientes funciones de magnitud equivalente. ¿Ahora quieres un sistema guiado por láser? Bueno, eso equivale a cincuenta puntos de trabajo; para compensar esta adición, eliminemos cincuenta puntos de funciones de baja prioridad al final de la lista de Pendientes.

Algunas compañías han llevado a un nuevo nivel esta idea de sólo entregar al cliente funciones de alto valor. Hace unos años supe de un desarrollador con Scrum que obtuvo un contrato de diez millones de dólares para producir *software* para una gran compañía constructora. Ambas partes convinieron un horizonte temporal de veinte meses. Pero la compañía con Scrum insertó una cláusula: si la empresa constructora quería cancelar el contrato en cualquier momento, podía hacerlo; sólo tendría que pagar veinte por ciento del valor del contrato restante. Básicamente, si el *software* funcionaba como la constructora quería, ésta podía detener la producción adicional por parte de la empresa con Scrum.

El desarrollador de *software* inició sus *sprints*, de un mes de duración. Luego del primer mes, el cliente redirigió parte del esfuerzo de aquél para obtener más valor. En el segundo, igual. Después del tercero, dio por terminado el contrato, tomó el *software* y lo puso en operación. Ya tenía el valor que necesitaba.

Hagamos unos cálculos aquí para ver cómo todos ganaron. En los tres primeros meses del contrato, el cliente pagó 1.5 millones de dólares a la empresa con Scrum. Cancelar anticipadamente el contrato le impuso el pago de veinte por ciento de los 8.5 millones restantes, es decir 1.7 millones. Así, pagó 3.2 millones por una pieza de *software* que había creído que le costaría diez y la obtuvo diecisiete meses antes de lo previsto.

Pero no fue el único que ganó. La compañía con Scrum licitó el contrato con un margen de utilidad esperado de quince por ciento. Así, en esos tres primeros meses invirtió 1.3 millones de dólares en desarrollo, pero recibió 3.2. Su margen de utilidad pasó de quince a sesenta por ciento, un incremento de cuatrocientos por ciento en ganancias. Ahora, ya desocupados sus desarrolladores, puede pujar por otros proyectos. Esto no es sólo buenos negocios, también es una estrategia de retiro anticipado.

La empresa pudo hacer esto gracias a los componentes de Scrum. Al gestionarse como unidad interfuncional, el equipo era capaz de acelerarse pronto y, por tanto, de brindar más valor más rápido. Al final de cada *sprint*, un incremento del producto estaba ya en Terminado. Funcionaba. Podía usarse de inmediato. En cada *sprint*, el responsable del producto podía repriorizar los Pendientes con base en la realimentación del cliente. Y una vez creado valor suficiente para éste, todos dejaron de trabajar. Así es como Scrum armoniza los intereses de todos: del equipo, el *Scrum Master*, el responsable del producto, el cliente y la compañía. Todos trabajan por la misma meta y con la misma visión: *ofrecer valor real lo más pronto posible*.

Soy un firme convencido de las situaciones de beneficio mutuo, y ganar más generando mejores productos a menor precio me parece un excelente negocio.

Riesgo

La gestión del riesgo está en el centro mismo de toda empresa de éxito. Scrum te permite reducir los riesgos de fracaso. Los tres tipos más comunes son el riesgo de mercado, el técnico y el financiero. O para decirlo de otra manera: «¿La gente necesita lo que estamos haciendo? ¿Realmente podemos hacerlo? ¿Podremos vender lo que hagamos?».

Ya he escrito mucho sobre el riesgo de mercado. Scrum te ayuda a minimizarlo enfatizando el cumplimiento incremental. Te permite poner más rápido un producto frente a los clientes. Y al obtener realimentación pronto y seguido, puedes hacer pequeños cambios en el acto en lugar de verte obligado a grandes cambios una vez que has invertido millones de dólares y te das cuenta de que lo que hiciste no es lo que el cliente quiere. Lo que hiciste suele ser lo que él dijo desear al principio del proceso, pero lo cierto es que la gente no sabe qué quiere hasta que puede probar algo. Un alto grado de asesoría de negocios gira alrededor de quiebras rápidas. Yo prefiero pensar en entregas rápidas.

El riesgo técnico es interesante. La pregunta de si en verdad es posible hacer lo que el cliente desea es difícil, sobre todo si vas a hacer algo físico que requiere plantas, herramientas y una inversión inicial.

¿Te acuerdas de la compañía con el sistema automatizado para el hogar? Abordó su proyecto haciendo lo que se conoce como «ingeniería concurrente basada en conjuntos». Esto significa «hacer diferentes prototipos para ver cuál funciona mejor antes de pasar a la producción propiamente dicha». Por ejemplo, aquella empresa sabía que necesitaba una cámara para que los clientes pudieran ver quién tocaba a la puerta y dejar pasar a un visitante accionando un timbre. La parte más cara de la cámara, y que requería el mayor tiempo de entrega, era la lente. ¿Debía ser de plástico, vidrio o cristal? ¿Cuál de estos materiales resiste cualquier clima? ¿Cuál rinde la mejor imagen? ¿Cuánto cuesta manufacturar cada elemento?

En vez de tomar la decisión de antemano y pasar de lleno a la manufactura, la compañía produjo tres lentes totalmente funcionales y las comparó. Como sólo quería resolver la pregunta sobre la lente y tenía que hacer eso primero a causa de los largos tiempos de entrega de manufactura,

probó cada lente usando un montaje de cámara en laptop. Resultó que el vidrio era el que mejor satisfacía los criterios. Pero, decisivamente, la compañía pudo emitir ese juicio tras haber visto algo que funcionaba. La elección no se basó en conceptos teóricos, tenía algo que podía ver y tocar. Una vez que respondió a esa pregunta, pudo proceder al diseño del bastidor de la lente y los procesadores que manipularían la imagen. Priorizando la decisión sobre la lente, la compañía se ahorró varios millones. Apple es famosa por hacer esto con todos sus productos, elaborando a menudo una docena de prototipos totalmente funcionales antes de organizar una selección para ver cuál es el mejor. Esto permite expresar pronto ideas diferentes todavía en ausencia de una gran inversión.

El riesgo financiero es la causa de la quiebra de muchas empresas. Hacen algo fabuloso, pero no lo pueden vender por una suma suficiente para obtener ganancias. Un ejemplo clásico es el periodismo en línea y la desaparición del diario impreso. Cuando la internet hizo explosión en los años noventa, los periódicos ansiaban incorporar su contenido en ella. Algunos directores supusieron que, en línea o fuera de ella, la gente pagaría por anunciarse, así que ofrecieron gratis su contenido. El problema fue, por supuesto, que los anunciantes querían pagar mucho menos por los anuncios en línea que por los impresos, pese a que el costo de producción de contenido seguía siendo el mismo. Otros intentaron levantar muros de pago en torno a su contenido, pero tantas páginas ofrecían noticias gratis que se vieron obligados a hacer lo mismo. Enviar reporteros a lugares de los hechos es costoso. Los resultados pueden verse en el cierre de salas de redacción en todas partes.

La idea de ofrecer contenido o un servicio gratis y ganar dinero en publicidad sigue imperando hasta la fecha en nuevas empresas tecnológicas. Los emprendedores ven el caso de Facebook o Google y dicen: «Puedo hacer lo mismo». El problema es que no hay muchos Facebooks ni Googles. En los primeros días de la internet, cuando el espacio en línea permitió por primera vez a las compañías dirigirse a segmentos de clientes particulares, la «hiperconcentración» se juzgaba valiosa. Pero al surgir cada vez más plataformas para facilitar eso, tal capacidad ha perdido valor.

Otra causa del colapso financiero de compañías es pagar de más por adquirir clientes. Un ejemplo son las compañías de cupones diarios, como Groupon y Living Social. En sus inicios, estas empresas adquirieron clientes fácil y rápidamente. Pero tras ampliar su alcance y elevar sus gastos generales, cada vez les resultó más costoso atraer anunciantes adicionales y

más personas dispuestas a comprar un cupón. Los resultados pueden verse en la valuación de estas compañías.

Lo que Scrum hace por las empresas es contestar pronto la pregunta clave: «¿Ganaremos dinero haciendo esto?». Al poner rápido frente a los clientes lanzamientos incrementales, descubrirás qué valoran y qué están dispuestos a pagar. Y si tus primeras conjeturas resultan equivocadas, puedes hacer cambios. Lo más que puedes perder es tiempo y energía en los pocos *sprints* que invertiste, en oposición al costo multimillonario de tender una infraestructura enorme y complicada sólo para descubrir que, aunque a la gente le encantó tu producto, no le gustó tanto como para pagar el costo de hacerlo.

He aquí lo que harás mañana

Bueno, ¿qué harás mañana para implementar Scrum donde trabajas? El primer paso es reunir una lista de Pendientes y un equipo. Piensa en la visión que tienes de tu producto, servicio o lo que sea, y empieza a desglosar lo que debes hacer para realizarla. No necesitas mucho. Basta con Pendientes para una semana. Mientras los miembros del equipo celebran sus reuniones de parada diaria y ejecutan su primer *sprint*, podrás acumular Pendientes suficientes para mantenerlos ocupados en los dos *sprints* siguientes. Sin embargo, nunca pierdas de vista tus Pendientes, porque a medida que tu equipo se acelere, empezará a producir más de lo que creíste posible.

Luego, como responsable del producto, elabora una guía de caminos que te permita ver adónde te diriges. ¿Cuánto crees poder terminar este trimestre? ¿Dónde quieres estar este año? Cabe recordar que esto es sólo una instantánea en el tiempo, así que no planees de más, sólo calcula. No estás haciendo un contrato de compromisos obligatorios; simplemente deja asentadas tus ideas acerca de dónde estarás en un tiempo. Créeme: el panorama cambiará. Tal vez radicalmente.

La razón de planear así es crear transparencia en la organización. Si tienes un equipo de ventas, debe saber en qué funciones estás trabajando para poder empezar a promoverlas. El liderazgo debe tener una idea sobre la fuente de ingresos, así como sobre cuándo y cuánto. El mensaje clave es que todo debe hacerse a descubierto. Cualquiera debe poder ver dónde está tu producto en cualquier momento. Debe poder ver avanzar a Terminado las historias en el pizarrón de Scrum. Debe poder ver puntos gráficos de historias contra el

tiempo y una línea estable en dirección a cero, o nada, para saber cuántos puntos de historias hizo tu equipo en el más reciente *sprint* y cuántos calculas que hará en el siguiente. Has de saber que, como responsable del producto, se te evaluará con base en los ingresos y los costos.

Descubrirás pronto, sobre todo si trabajas en un sitio con múltiples equipos, que tendrás que empezar a formar un equipo de responsables del producto para poder generar Pendientes suficientes para los equipos. Podrías tener un responsable del producto a cargo de la estrategia y la interacción con el cliente, y otro táctico, que decida qué harán los equipos en cada *sprint*.

Lo importante, sin embargo, es empezar, así que comienza de una vez. Scrum está diseñado para que puedas echar a andar un equipo en un par de días. Consigue tu lista de Pendientes, planea tu primer *sprint* y lánzate. No es necesario que dediques mucho tiempo a la planeación, reflexión, meditación, declaraciones de misión o proyecciones a cinco años. Deja esto a la competencia y que muerda el polvo. Y sobre la marcha, ¿por qué no haces un mundo mejor? Te diré cómo en el capítulo siguiente.

RESUMEN

Haz una lista. Revísala dos veces. Elabora una lista de todo lo que quizá deba hacerse en un proyecto y luego disponla en orden de prioridad. Coloca los puntos con mayor valor y menor riesgo al inicio de los Pendientes y luego los siguientes y los siguientes.

Responsable del producto. Esta persona traduce la visión en Pendientes. Debe conocer la propuesta de negocios, el mercado y el cliente.

Un líder no es un jefe. Un responsable del producto establece lo *que* se debe hacer y *por qué*. *Cómo* se realizará y *quién* está a cargo de ello lo decide el equipo.

El responsable del producto. Este individuo debe conocer el terreno y tener autoridad para tomar decisiones definitivas. Debe estar disponible para responder a preguntas y estará a cargo del suministro de valor.

Observa, Oriéntate, Decide, Actúa (OODA). Percibe el panorama estratégico total, pero actúa tácticamente con rapidez.

Miedo, incertidumbre y duda. Es mejor dar que recibir. Entra en el ciclo OODA de la competencia y envuélvela en su confusión.

Obtén dinero por nada y haz las modificaciones gratis. Crea cosas nuevas sólo en tanto produzcan valor. Accede a intercambiarlas por cosas que requieren un esfuerzo igual. Lo que al principio creíste necesitar no es nunca lo que en verdad precisas.

Capítulo nueve

Cambia el mundo

Scrum tiene sus orígenes en el mundo del desarrollo de *software*. Pero ya se ha extendido a muchas otras áreas. Diversas empresas lo usan para todo, desde fabricar naves espaciales hasta gestionar la nómina y ampliar los recursos humanos, y surge ya también en muchos otros terrenos, de finanzas a inversión, del entretenimiento al periodismo. Suele asombrarme que un proceso del que fui pionero en 1993 para contribuir al desarrollo de *software* haya demostrado ser universalmente aplicable. Scrum acelera el esfuerzo humano, sea cual fuere.

De hecho, ya lo he visto aparecer en los lugares donde menos se habría imaginado, abordando los más espinosos problemas de la humanidad. Piensa en algunos de ellos. Por ejemplo, la gente que vive en la pobreza, la cual es no sólo degradante, sino que además genera un sinnúmero de males sociales, de crimen y corrupción a destrucción y guerra. Después está nuestro sistema educativo, que les está fallando a los estudiantes en todo el mundo. En vez de enseñar habilidades para el siglo XXI, abrumamos a nuestros jóvenes con formas de enseñanza y aprendizaje creadas en el siglo XIX. Otro elemento desconcertante que viene a la mente es el gobierno, que se ha paralizado de muchas formas, fundado en ideas surgidas hace cientos de años que ya no parecen encajar con nuestro modo de vida.

Es fácil escandalizarse ante las más recientes noticias de gente que muere en África, la violencia en las escuelas o las poses interminables de la gente en el poder. A veces todo esto parece demasiado. Pero esos problemas, los difíciles, son justo los que Scrum está diseñado a atacar. En cada uno de esos casos, la gente está recurriendo a Scrum para ayudar a resolver tales problemas y, lo mismo que en el mundo de los negocios, está teniendo mucho éxito.

Educación

En cierto sentido, las comunidades dormitorio son iguales en todo el mundo. Ubicadas a unos kilómetros de grandes metrópolis, la gente se muda ahí para comprar casas económicas, formar una familia y mandar a sus hijos a la escuela sin muchos de los problemas de la gran ciudad.

Alphen aan den Rijn es una ciudad de esa clase, situada en el oeste de los Países Bajos, entre Leiden y Utrecht, quizá a cuarenta y cinco minutos de Amsterdam. Al aproximarse a ella en auto un día de clases, todo el flujo vehicular va en la dirección opuesta, al trabajo. Granjas lecheras y molinos de viento viejos y nuevos cubren el campo.

En la ciudad, el tránsito es casi exclusivamente de bicicletas. Cientos de ellas se dirigen a la escuela pública local de educación media, Ashram College. Como la ciudad, ésta es asimismo muy representativa de las escuelas holandesas. Tiene mil ochocientos estudiantes, de entre doce a dieciocho años de edad. Holanda analiza pronto a sus estudiantes, dividiéndolos entre programas vocacionales inferiores, destinados a producir peluqueros, mecánicos y secretarias; programas vocacionales superiores, que dirigen a los chicos a enfermería, administración e ingeniería, y programas universitarios, para los destinados a medicina, derecho o la investigación. Los chicos de las capas inferiores pueden incorporarse a la fuerza de trabajo a los dieciséis años, mientras que los de las superiores podrían pasar buena parte de su veintena en la universidad y la educación profesional. Cada capa requiere cursos básicos comunes, aunque a cada grupo se le enseña por separado. En Ashram se imparten las tres capas. Y una de las materias fundamentales es la que Willy Wijnands imparte a estudiantes en todos los niveles en esa escuela: química.

Estoy seguro de que tienes recuerdos de la materia de química en la preparatoria: mesas de laboratorio en filas muy derechas frente al maestro, en la parte delantera del salón, y quizá una semana de disertaciones seguida por algunos días de trabajo en un problema práctico con un compañero, la elección del cual era a menudo estratégica y muy enfática. Quizá la química te gustaba, tal vez te aburría hasta las lágrimas, y puede que *Breaking Bad* te haya dado una nueva apreciación de la posible recompensa económica de la buena técnica de laboratorio y la importancia de elegir el compañero correcto. Cualquiera que haya sido tu experiencia, una vez que el maestro comenzaba a hablar de enlaces covalentes o algún otro concepto oscuro, es probable que hubiera un casi audible *clic* cuando tus compañeros y tú miraban a la ventana,

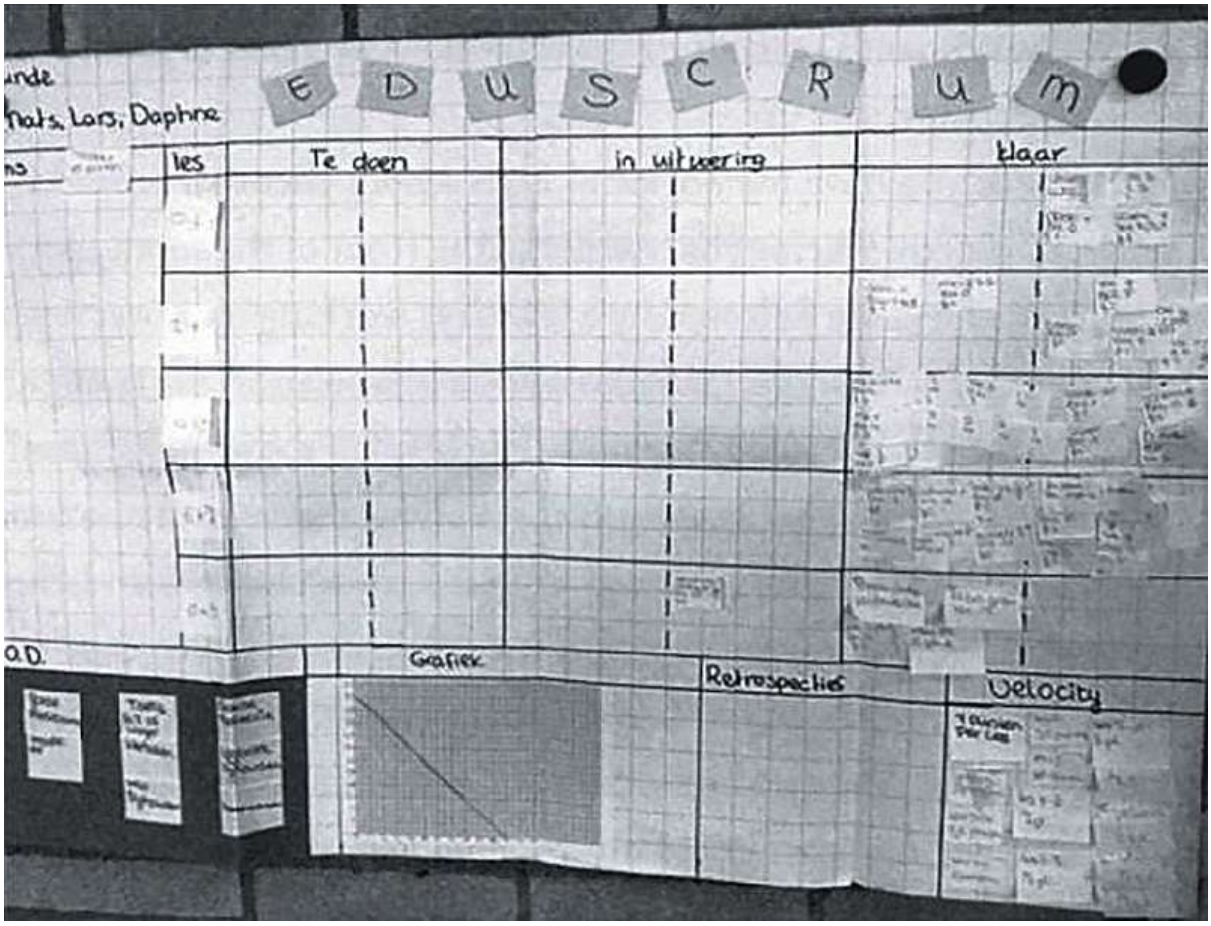
garabateaban monitos o pensaban en el guapo chico o chica de la segunda fila. Admitámoslo: en las aulas estadunidenses, cuando la química gobierna, suele seguirle la ensoñación.

Pero eso no sucede en las clases de Wijnands. «Mira», dice cuando sus alumnos irrumpen en el aula y se precipitan a sus pupitres, curiosamente sin sentarse. «Yo no hago nada». Son las ocho y media de la mañana de un miércoles normal de septiembre y el salón de Wijnands no se parece a ningún otro. Los pupitres no forman filas frente a la parte delantera del salón. En cambio, están dispuestos de tal forma que cuatro alumnos componen un equipo y se miran entre sí.

En vez de sentarse al comenzar la clase, estos chicos sacan una enorme hoja de papel cubierta de papeletas adhesivas, la fijan en la pared y se reúnen en torno a ella. La hoja está dividida en grandes columnas. *Alle items* a la izquierda. Luego *Te doen*, después *In uitvoering* y por último *Klaar*. Como cabe suponer, tales expresiones significan «Todos los elementos», «Pendientes», «En proceso» y «Terminado».

Al final de las columnas hay cuatro encabezados adicionales: DT, o definición de Terminado; *Grafiek*, que apunta a la gráfica de reducción, la cual muestra el avance hacia la meta, y, finalmente, Retrospectiva y Velocidad, donde los alumnos miden cuántos «puntos» obtienen en cada lección. Sus *sprints* suelen ser de cuatro o cinco semanas de duración, y concluyen con una prueba.

Frente a sus pizarrones de Scrum —o *flops*, como les dicen en holandés (derivación de *flip chart*, «rotafolios»)—, los estudiantes determinan qué lecciones consumarán hoy. Mueven de Pendientes, *Alle Items*, a *Te Doen* las papeletas que creen poder cumplir y se ponen a trabajar. De nueva cuenta, como Wijnands gusta decir, él no hace nada. Los alumnos abren sus libros y se ponen a estudiar. Pero quizá lo más importante es que se enseñan entre sí. Wijnands recorre el salón examinando los pizarrones de Scrum y las gráficas de reducción. Ocasionalmente, identifica un sitio en el que los alumnos tienen un problema, explica rápidamente un concepto difícil o toma al azar una historia de la columna *Klaar* e interroga a cada alumno al respecto, para confirmar que todos entiendan los conceptos. De no ser así, regresa la papeleta a *Te doen*. Resulta que parte de la definición de Terminado es que todos entiendan el material.



Los alumnos disponen para sí de una parte única del pizarrón de Scrum: «definición de diversión». El trabajo no sólo debe terminarse; ellos tienen que divertirse haciéndolo. Las tres pruebas son Confianza, Humor y un término intraducible, *Gezelligheld*, equivalente a «acogedor», «sociabilidad», «diversión» o «agrado», o a ver a un amigo luego de una larga ausencia, o pasar tiempo con nuestros seres queridos, o sencillamente pertenecer. En realidad, ésa me parece una manera perfecta de describir la sensación de apoyo, gozo, esperanza, alegría, comodidad y emoción de estar en un equipo muy bueno.

«No hay necesidad de que seas policía», dice Wijnands. «Ahora tenemos otra manera de ejercer el control de los alumnos. Ellos hacen todo, ¡hasta asignarse tareas por hacer en casa!». Cada equipo sabe en qué parte del material se ubica, las fechas en que tiene que cumplir pasos intermedios y si todos sus miembros deben trabajar fuera de clase para aprender a tiempo todo el material. «Se organizan solos; desarrollan maneras de estudiar más rápidas e inteligentes. Un equipo comenzó por la prueba y trabajó en sentido inverso. Un grupo de chicos de once años. “Eso no está bien”, les dije. Se les

descompuso la cara». Wijnands esboza una sonrisa contagiosa. «Pero después añadí: "¡Está excelente!"».

Scrum, o eduScrum, como lo llama Wijnands, se les presenta a los alumnos el primer día de clases. Lo primero que hacen es formar equipos, equipos interfuncionales. Cada estudiante se califica en varias categorías, de valor a gusto por las matemáticas, tomar en cuenta los sentimientos de los demás y «avanzar directo a la meta». Luego se les pide formar equipos interfuncionales, con todas las habilidades necesarias para aprender el material. Esto, dice Wijnands, les enseña algo tan importante como la química: a trabajar y apreciar a personas con talentos diferentes a los suyos.

Tim Jansen tiene diecisete años. Éste es su último año de preparatoria. Lleva tres practicando Scrum y está a punto de entrar a la universidad, donde piensa estudiar química. Parece un nerd típico. Listo, quizá no muy desarrollado socialmente. «Aprendo más rápido que los demás», dice. «Pero al trabajar en común, mejoras. Aprendo mejor explicando las cosas a otros». Se vuelve hacia Gudith Zwartz, sentada al otro lado de la mesa. «Ella sabe que me puede preguntar sobre contenido y yo puedo preguntarle sobre organización. Ella ordena las cosas mejor que yo».

Gudith parece muy diferente a Tim: esbelta, bonita, de cabello rubio. «Terminas por conocer mejor a tus compañeros. Sabes quién es bueno para qué».

«Scrum ayuda a los extraños a relacionarse con las otras partes del grupo», dice, mirando a su igualmente bonita y refinada amiga Maneka Bowens. «A veces tú eliges al equipo y otras ellos te eligen a ti. Te enteras de que son mejores que tú en algunas cualidades».

Ese tipo de aprendizaje, dice Wijnands, forma parte de la idea: volver conscientes las habilidades inconscientes. Las habilidades que es posible probar en un examen distan de ser las únicas importantes. Ayudar a los estudiantes a aprender a identificar y valorar diferentes fortalezas en sí mismos y los demás es una habilidad del siglo XXI. Eso es algo que todos debemos aprender.

Tras elegir equipos, se enseña a los alumnos a evaluar no por horas o días, sino por puntos. Estiman entonces cada pieza del material que deben aprender usando la evaluación relativa inherente a la serie de Fibonacci, jugando el póker de planeación. Willy explica muy sencillo la idea de los puntos.

—Ignoren todas las medidas que les han enseñado. No hay medidas absolutas. Si yo peso cincuenta puntos —dice, y señala a una delgada preparatoriana—, ¿cuántos pesas tú?

—Mm, ¿cuarenta? —aventura ella.

—¡Vaya, gracias! Creí que serían unos veinte.

Al final de cada serie de lecciones, los equipos hacen una retrospectiva, preguntándose: «¿Qué salió bien? ¿Qué pudimos hacer mejor? ¿Cómo puede mejorar el equipo?».

Esta atención a los equipos, dice Wijnands, a veces sorprende a los padres. Cuenta el caso de una madre que llamó y dijo que su hija había hecho todo el trabajo. ¿Por qué se le obligaba a cargar con los demás?

«Le dije que la muchacha debía tener el valor de decirles a los otros que hicieran más. Lo hizo y las calificaciones de los exámenes subieron. La madre me volvió a llamar, para darme las gracias. Los alumnos no sólo deben aprender a trabajar solos; también en común».

La energía en las aulas de Ashram es notoria y se traduce en resultados. En el sistema escolar holandés, las calificaciones van de 1 a 10, y 5.5 se considera una calificación aprobatoria aceptable. En las clases de Willy, un 7 es aceptable. Y los alumnos cumplen esa línea de base. El año pasado, dice Wijnands, los resultados en exámenes subieron más de diez por ciento.

Willy se enteró de Scrum por su yerno, que trabaja en una gran compañía técnica en los Países Bajos en la que se utiliza. Willy ha sido maestro por cerca de cuatro décadas y dice que había buscado esto todo el tiempo: un método que enseñe a los chicos a aprender solos y a valorar sus habilidades y las de los demás. Asimismo, a divertirse haciéndolo.

Algo importante acerca de Scrum es que raramente permanece mucho tiempo como «encendido-apagado»; está hecho *a escala*. En las escuelas de los Países Bajos, por ejemplo, eduScrum no depende de un solo individuo, por bueno que sea como profesor, como Wijnands. Aunque quizá haya comenzado con Willy y él pueda haber convencido a algunos de sus compañeros profesores de química en Ashram de hacer la prueba, ahora está creciendo. Apoyada por la comunidad de negocios, ahora existe en los Países Bajos una eduScrum Foundation que forma a profesores y educa a escuelas acerca de Scrum. Esta fundación ha capacitado ya a setenta y cuatro profesores, de todas las materias, en veinte escuelas. Piensa crecer a razón de sesenta profesores y quince escuelas al año. En cinco años, esto significará trescientos maestros y setenta y cinco escuelas más. Buen comienzo. Yo conocí a algunos de esos profesores, de todo el país, y me dijeron que esto es el nuevo Montessori. Lo ven como un *movimiento*.

Pero esto no sucede únicamente en los Países Bajos. En Arizona hay una escuela pública para alumnos rurales pobres de familias de indios americanos

que emplea Scrum. En algunas universidades se le empieza a enseñar ya. En la Harvard Business School se construyó un aula llamada Innovation Lab, donde toda la instrucción se basa en equipos. Y como me dijo el profesor Takeuchi de esa escuela, para enseñar a trabajar en equipo se usa Scrum.

Mientras estuve en Ashram, conversé con algunos de sus alumnos. Cuando pregunté si había dudas, un chico levantó la mano.

«No puedo creer que usted haya diseñado esto para el *software*», dijo, «parece perfectamente diseñado para preparatorias».

Sentí que los ojos se me llenaban de lágrimas al mirar a ese joven. Luego supe que era autista. Antes de Scrum, él era apático y resignado. Scrum le dio la oportunidad de avanzar, de disfrutar la escuela, de ser una persona mejor y más completa. Hace años, mientras trataba de hacer mejoras en compañías de *software*, yo no sabía que también estaba creando algo que podía contribuir a mejorar la vida de la gente.

Pero así es. Y quizá no lo haya hecho nunca más eficazmente que en la Uganda rural.

Pobreza

Uganda es uno de los países más pobres del mundo. Más de la tercera parte de su población vive con menos de 1.25 dólares diarios. La gran mayoría de los ugandeses reside en áreas rurales donde la pobreza es endémica y la gente lucha por subsistir cultivando pequeñas parcelas familiares. Muchos de esos sitos son muy remotos, a días a pie desde la ciudad comercial más cercana. Las familias sufren enviando a sus hijos a la escuela, ya que sus manos son necesarias para ayudar en la granja. Las mujeres en particular desertan pronto. La esperanza de vida es de cincuenta y tres años. La mortalidad infantil es de más de cinco por ciento de niños nacidos vivos y seis mil mujeres mueren cada año de complicaciones en el embarazo. La vida de un agricultor rural en Uganda no es fácil.

La Grameen Foundation es una derivación del Grameen Bank, de Muhammad Yunus, ganador del Premio Nobel, pionero en microfinanzas para los muy pobres en Bangladesh. Esa fundación se dedica a ayudar a los pobres del mundo a salir de la miseria, no con donativos sino aprovechando las menospreciadas fortalezas de los indigentes. En Uganda decidió intentar justo eso, dando a los pobres la posibilidad de compartir y aumentar sus conocimientos.

Para hacerlo, reclutó a mil doscientas personas de áreas rurales pobres, personas a las que llamó Trabajadores Comunitarios del Conocimiento (TCC). Esa fundación ya había desarrollado aplicaciones móviles para microfinanzas y pagos y decidió dar a dichos trabajadores del conocimiento no sólo información bancaria, sino también información que pudiera servirles en su vida diaria, lo que, en el caso de Uganda, significa aplicada a la agricultura. La fundación brindó acceso a buenas prácticas agrícolas dando a tales trabajadores teléfonos inteligentes y transmitiendo la información por ese medio.

Steve Bell, del Lean Enterprise Institute y *Scrum Master* certificado, visitó recientemente dos pueblos remotos y describió cómo funciona el sistema. Hay una asamblea de agricultores, en la que todos permanecen de pie en un campo. Uno de ellos llevó una planta con una plaga. Los TCC buscaron rápidamente fotos en el teléfono hasta encontrar una de una planta con esa plaga particular. Así, se dispuso al instante del tratamiento específico contra ella, el cual no requería pesticidas ni productos químicos costosos y que el agricultor podía aplicar de inmediato.

Bell dice que la rápida transmisión de información práctica es de suyo eficaz, pero que esa aplicación también enlaza a los agricultores con otros en toda Uganda. Usando esta conectividad, pueden reportar con precisión cuántos cultivos venden en la ciudad comercial más próxima, a menudo a varios días de camino. Antes, los agricultores estaban a merced de intermediarios que se aprovechaban de su desconocimiento del mercado para fijar los precios que ellos querían. Ahora los agricultores saben cuánto ganan los intermediarios.

Bell me contó el caso de una mujer que le dijo que, gracias a la información agrícola, pudo duplicar su rendimiento. Pero la información del mercado también le permitió duplicar sus precios. Antes, ella obtenía trescientos chelines por bushel, pero tras enterarse de que el precio normal era de mil chelines por bushel, pudo negociar uno de seiscientos. El doble de rendimiento, el doble de ganancias, misma cantidad de trabajo. Esto es lo que Scrum está diseñado a hacer y la forma en que le sirvió a ella.

Eric Kamara dirige el grupo de tecnología de la oficina en Kinshasa de la Grameen Foundation. Su grupo se sirve de Scrum para desarrollar sus aplicaciones. Kamara afirma que cada vez que un grupo solicita una serie de funciones, su equipo las califica en una escala de 1 a 7 en tres preguntas:

1. ¿Qué tan importante es este trabajo para la misión de ayudar a los pobres?

2. ¿En qué contribuirá esta función al trabajo de los TCC?
3. ¿Se cuenta con el apoyo de socios para la función? (Esta fundación prefiere trabajar con socios como la Gates Foundation en lugar de hacerlo sola.)

Esto permite a Kamara priorizar el trabajo usando criterios específicos. Antes de Scrum, dice, la gente pedía todo al mismo tiempo. Y dados los limitados recursos de una organización no lucrativa, la fundación no podía hacerlo todo, así que el resultado era no hacer nada. Ahora, en cada *sprint* los grupos que necesitan ciertas funciones se presentan a explicar lo que quieren y en forma transparente ven cómo su función se enfrenta a otras. Esto ayuda a un grupo que cuenta con recursos limitados a determinar qué tendrá mayor relevancia.

Tal como lo he visto en otras partes, este tipo de trabajo se extendió pronto al resto de las oficinas de Kinshasa, afectando literalmente la manera de efectuar sus labores rutinarias. Antes, aquella oficina tenía la clase de reuniones semanales que todos temen: una actualización de estado de varias horas de duración en la que se exponían y lamentaban problemas pero en la que se hacía muy poco. La reunión era interminable y todos salían de ella insatisfechos. A menudo, el único resultado era echar la culpa a alguien en vez de buscar soluciones. Ahora, dice Kamara, cada equipo cuenta con un pizarrón de Scrum. Antes de la reunión, problemas y obstáculos saltan a la vista. Hoy en día, al director de la oficina le basta con dar una vuelta para ver al instante dónde se han bloqueado u obstaculizado cosas con sólo revisar el estado de los pizarrones de Scrum.

Al hablar con personas del ámbito de las organizaciones no gubernamentales, una queja frecuente es que sus filas están repletas de individuos que comparten propósito y compromiso pero que carecen de disciplina. Scrum puede tomar la pasión de la gente y, dándole claridad en cuanto a lo que debe priorizar, aprovecharla.

Es fácil elaborar una propuesta de *negocios* de Scrum. Si lo aplicas, ganarás más, mucho más. Harás el doble en la mitad de tiempo. Pero la promesa más llamativa para la humanidad es para la gente que ha dedicado su vida a ayudar a los más pobres entre los pobres. Si Scrum puede ayudar a los individuos que han trabajado en los márgenes a obtener el mismo efecto, se habrá dado un paso enorme hacia la consecución de un amplio bien social.

No sólo ese «bien» llegará más pronto, sino que además será mensurable. Scrum da a la gente la posibilidad de medir fácilmente su progreso. En la Grameen Foundation se cuenta con lo que se conoce como Índice de Progreso en Salir de la Pobreza. Éste mide la eficacia de cada programa. Tras hacer una

encuesta, es posible ver el efecto en poblaciones rurales de esos Trabajadores Comunitarios del Conocimiento dotados de teléfonos celulares. Experimentar con diferentes maneras de hacer las cosas. Ayudar a la gente a *innovar* su salida de la pobreza.

Me sorprende ver a Scrum volver a sus raíces. Cuando lo inicié, me inspiré en el Grameen Bank y otras instituciones de microfinanzas, así como en la manera en que ayudaban a equipos de pobres a trabajar en común para ayudarse a salir de la miseria. Reunían un equipo de tales personas y pedían a cada una presentar un plan de negocios que explicara qué harían con veinticinco dólares. Una podía querer comprar una carreta para vender fruta en la plaza. Otra, adquirir una máquina de coser para hacer vestidos y venderlos. Sólo cuando se habían pagado todos los préstamos del equipo se les prestaba más dinero. Sus miembros se reunían cada semana para ver cómo ayudarse entre sí. Los resultados eran impresionantes. Al principio, la mujer de la máquina de coser pudo ganar lo suficiente para dar de comer a sus hijos. Semanas después pudo permitirse comprarles zapatos. Luego pudo mandarlos a la escuela. Varios ciclos después ya tenía una pequeña empresa y pudo empezar a construir su casa. Dije entonces a los programadores de *software* con quienes trabajaba: «Estos pobres no tienen zapatos, pero pueden impulsar su salida de la pobreza. Ustedes tienen zapatos pero no *software*. Ellos han encontrado una manera de trabajar en común para salir de la miseria. ¿Ustedes están dispuestos a hacer lo mismo?». Fue así como nació Scrum.

Las organizaciones no lucrativas son sólo un área en la que podemos innovar el bien social. ¿Y la forma en que nos organizamos? ¿Y el gobierno?

Gobierno

Gobierno es no sólo cómo organizamos la esfera pública —cómo conseguimos caminos, policía, tribunales y DMV—, sino también cómo formalizamos lo que somos como personas. Es una codificación de lo que creemos ser. En Estados Unidos, las aspiraciones fundamentales de la población se recogen en un documento firmado por un grupo de rebeldes que seguramente habrían sido colgados por separado si no se hubieran «rolado» juntos: la Declaración de Independencia. Redactada por terratenientes aristocráticos, idealistas y esclavistas, ese documento recogió asombrosamente un concepto radical acerca de qué tipo de pueblo querían ser los estadunidenses de la época de la independencia.

> Sostenemos que estas verdades son evidentes; que todos los hombres son creados iguales, que han sido dotados por su Creador de ciertos derechos inalienables, como la vida, la libertad y la búsqueda de la felicidad. Que para garantizar esos derechos se instituyen gobiernos entre los hombres, los cuales derivan sus justas facultades del consentimiento de los gobernados.

En la era moderna es difícil advertir cuánto se alejaban de la norma esas palabras. Aunque las ideas de la Ilustración ya habían empezado a propagarse, en ese tiempo no había democracias. El gobierno se imponía desde arriba, por el derecho divino de los reyes y el poder de las armas. Grandes imperios regían buena parte del mundo, no sólo Gran Bretaña, sino también Francia, Austria, Rusia y Turquía. La idea de que los individuos estaban dotados de derechos, no de que los poderosos se los otorgaran, era, para decirlo suavemente, revolucionaria.

La «república» fue un tipo de gobierno emergida de esos ideales. Igual que el robot de Rodney Brooks aprendió a caminar, Estados Unidos se levantó trabajosamente sobre sus pies, tropezó, cayó y ocasionalmente siguió el camino equivocado. Pero esos ideales inspiraron revoluciones en el mundo entero y hoy la mayoría de las grandes potencias están gobernadas, al menos en la forma, por las personas a las que dicen representar.

El problema es, por supuesto, que luego de más de doscientos años de desarrollo burocrático, en la estructura misma del gobierno se han incrustado intereses permanentes que dificultan escuchar la voz del pueblo. La corrupción —sea en la pequeña escala de los burócratas que aceptan sobornos a cambio de servicios, o en la grande de bancos enormes que acumulan riquezas privatizando ganancias y socializando pérdidas— es resultado de la falta de transparencia y la centralización del poder en manos de unos cuantos.

En la mayoría de las capitales del mundo ha prosperado una clase cortesana que constituye el gobierno permanente. Contratos se otorgan, dinero se gana y poder se confiere por «aquellos que conoces», no por «lo que aportas». Esto es particularmente evidente en la forma en que políticos, generales y poderosos burócratas se rotan entre el gobierno y la industria. El número de generales de cuatro estrellas que encabezan a los contratistas de la defensa, o de senadores que se convierten en cabilderos, o de exfuncionarios que dirigen grupos comerciales es impresionante.

Pero como enfaticé en el capítulo tres, es inútil buscar malas personas; busca en cambio malos sistemas. Censurar no hace más que demorar, así que

no caigamos en la trampa fácil del error fundamental de atribución. Hagamos en cambio una pregunta que realmente puede cambiar las cosas: «¿Qué serie de incentivos motiva el mal comportamiento?». Dudo seriamente que cualquiera de los bandidos de Beltway se considere una mala persona; apostaría que en su mayoría son sujetos bienintencionados. Es el sistema el que les ha fallado, y nosotros. Pero ¿cómo podemos cambiar eso? ¿Cómo alentar la transparencia, las prioridades y la responsabilidad? Conoces la respuesta: con Scrum.

Comencemos a miles de kilómetros al oeste de Washington, en la capital del estado de Washington, Olympia. Ahí, los dos últimos gobiernos —primero republicano, ahora demócrata— han abrazado lo que ellos mismos llaman un gobierno saneado. El gobernador actual, Jay Inslee, dijo en una entrevista de campaña en el otoño de 2012: «Gran parte de lo que el gobierno del estado hace es tomar decisiones. Queremos hallar la manera de tener menos papeles en el escritorio»[43].

El plan del gobernador consta de cinco puntos que podrían haberse extraído de cualquier plataforma de campaña: 1) un sistema educativo de «clase mundial», de preescolar a universidad; 2) una «economía próspera»; 3) convertir a Washington en líder nacional en energía sustentable y medio ambiente limpio; 4) comunidades sanas y seguras, y 5) un gobierno eficiente, eficaz y responsable.

No son metas revolucionarias. Son lo que la gente *debería* esperar de su gobierno. El hecho mismo de que parezcan lugares comunes es un indicador de su importancia. Un lugar común, después de todo, es sencillamente una verdad tan repetida que se ha vuelto trillada. Pero lo *diferente* del gobierno de Inslee es la forma en que lo hace. Su equipo llamó a su nuevo método SMART: específico, mensurable, asequible, relevante y con límite de tiempo (Specific, Measureable, Attainable, Relevant, Time-Bound). En otras palabras, desea usar Scrum. Y lo hace.

La oficina del director de Información del estado de Washington es responsable no sólo de qué tecnología se adquiere, sino también de cómo se hace. Consta de veinte personas que deben confirmar que no ocurran grandes fallas de informática con un costo de decenas de millones de dólares. Entre tanto, ese departamento se ocupa de actualizaciones de informática para las partes del gobierno que hacen todo, desde emitir licencias de manejo hasta distribuir beneficios de desempleo y regular la pesca y las reservas naturales. En 2012 supervisó ochenta solicitudes por un total de más de cuatrocientos

millones de dólares. Y emite normas y orientaciones para varios organismos sobre cómo poner en marcha las políticas gubernamentales.

Para hacerlo, usa Scrum. Las paredes de los cubículos de sus oficinas han desaparecido, para formar equipos con Scrum. Michael De Angelo, subdirector de Información, dice que intentan emitir cada semana políticas prácticas e implementables para los departamentos estatales.

«Estamos actualizando nuestro proceso de cómo deben presentar las agencias sus planes de inversión. Nos hemos fijado la meta de cambiar una cosa cada semana. Estamos adoptando un enfoque incremental. Cada semana tenemos un producto potencialmente embarcable, que las agencias pueden experimentar de cerca. Tienen así algo tangible». «Producto embarcable» significa en su caso cambios prácticos a políticas. No necesariamente tiene que ser una cosa; basta con que sea algo, lo que fuere, que cree valor.

En vez de tratar de elaborar un documento enorme en el que se prevea cada parte del proceso de financiamiento, han decidido hacerlo pieza por pieza. Quieren producir mejoras en el gobierno estatal cada semana.

La reacción, dice De Angelo, ha sido variada. Hay mucho temor de no tener un producto perfecto. En agosto de 2013 explicó: «Justo la semana pasada hicimos un cambio en la manera en que los clientes nos buscan. Pero hay mucha documentación en la que sigue apareciendo el viejo método: en nuestra página en internet, documentos, ese tipo de cosas. Estaban aparte todas las demás cosas que debíamos cambiar [primero]. Pero decidimos no esperar y hacerlo. Pondremos al día la documentación en el siguiente *sprint*. La opción es no ofrecer algo mejor durante meses… robarle valor a la gente».

La otra cosa que la oficina del director de Información está haciendo es tratar de promover Scrum en toda la burocracia estatal. Por eso han cambiado sus procesos por Scrum: para poner el ejemplo, para poder hablar por experiencia. Los beneficios son sumamente grandes para no aprovecharlos.

Pero hay obstáculos. De Angelo afirma que se han percatado de que, en algunos casos, el método en cascada está inscrito en la ley estatal. Cambiar eso puede ser difícil. El estado de Washington financia cosas en ciclos de dos años. «Tienes que pedir trozos grandes. No puedes decir: “Agregaremos valor hasta que nos digan que nos detengamos”», dice. «El gobierno quiere ver [que] eso va a costar tanto [y que] obtendremos tanto valor en ese marco temporal, para poder hablar de eso con los ciudadanos, aunque sepamos que es mucho más ineficiente».

Parte del problema es que en Estados Unidos, en los niveles tanto federal como estatal, las legislaturas se dividen en comités. Un grupo de legisladores

se ocupa de la educación, otro de la delincuencia, otro más del presupuesto y aún otro más de los servicios sociales. «Están fracturados. Nunca ven el cuadro general», dice Rick Anderson, consultor de agencias estatales y de gobiernos de condados y ciudades en Washington, Oregon, California y Hawai. Anderson ha trabajado con legislaturas, y sostiene que aunque el cambio puede tardar en llegar, tendrá que ocurrir.

Cree que debería comenzarse por establecer metas basadas en el desempeño.

«Bueno, Agencia X, aquí están tus metas y aquí los resultados esperados. Una vez que los obtengas, podrás empezar a elaborar leyes basadas en resultados», dice.

En un mundo puesto al día y movido por Scrum, en vez de aprobar un plan específico para construir un puente sobre un río, un órgano legislativo diría al departamento de autopistas: «Necesitamos que *X* número de personas atraviesen este canal en *Y* cantidad de tiempo con *Z* costos. Cómo hacerlo es cosa de ustedes». Esto daría pie al descubrimiento y la innovación.

Por el contrario, la norma en nuestros días son proyectos de construcción que exceden su presupuesto en cientos de millones de dólares. ¿La razón? Conforme las cuadrillas avanzan en el proyecto descubren nuevos problemas y nuevas formas de resolverlos. En vez de sofocar ese tipo de innovación con consejos de control de modificaciones y gran cantidad de reportes deberíamos alentarlo.

Pero ¿y los ideales con los que comencé esta sección, en los que una sociedad se define mediante un documento? ¿Una constitución, digamos? Bueno, un país decidió que la manera de desarrollar una constitución que verdaderamente representara la voluntad de la gente era usar Scrum.

En 2008, una crisis financiera totalmente evitable sacudió al mundo. Grandes bancos perdieron el control de los precios, apoyándose crecientemente en la aceptación de cada vez más deudas incobrables. Uno de los países más afectados fue Islandia. Bancos privatizados se habían escindido ahí del gobierno y corrido grandes riesgos en los mercados financieros. Y como dicen en Wall Street, si no sabes quién es el bruto en la sala, entonces lo eres tú. En este caso, Islandia fue el bruto. La suma de dinero que pidió en préstamo era excesiva para un país tan pequeño. Finalmente, los bancos tenían valoraciones doce veces mayores que el presupuesto nacional. Cuando todo se vino abajo, el «milagro económico islandés» quedó hecho trizas.

En una expresión de indignación, los ciudadanos de Reikiavik tomaron las calles y golpearon ollas y cazuelas frente al Althing, el parlamento. El

gobierno que había supervisado las prácticas financieras se desplomó en lo que se conoció como la «revolución de las ollas y las cazuelas». El gobierno dimitió y un nuevo liderazgo prometió una nueva constitución.

Para elaborar esa constitución, algunos funcionarios decidieron ser abiertos y tratar con el pueblo. Formaron un comité constitucional que decidió usar Scrum. El comité se reunía cada semana, decidía acerca de una sección del documento y la daba a conocer públicamente cada jueves. Después reunía comentarios de la gente por medio de Facebook y Twitter. En unos meses tenía un nuevo documento con el apoyo abrumador de la sociedad islandesa. Era una nueva expresión de cómo ésta se veía a sí misma.

Lamentablemente, los poderes que se habían beneficiado del fraude financiero atacaron de nuevo. Luego de solicitar una demora tras otra —de confundir, quejarse y actuar contra la voluntad de la gente—, un nuevo parlamento compuesto por los mismos partidos que autorizaron la destrucción de la economía de Islandia decidió ignorar la nueva constitución. Una demanda clave de la revolución fue negada. Por ahora, al menos.

El mundo está cambiando y quienes se benefician del sigilo y el engaño pronto descubrirán que les quedan pocos sitios para esconderse. Scrum está cambiando el mundo a su alrededor, y aunque pueden lanzar una acción de retaguardia, el cambio es inevitable. El enfoque de Scrum es tan rápido, transparente y sensible a los deseos de la población que derrotará en definitiva a los políticos que se interpongan en su camino.

Renovarse o morir.

Cómo trabajaremos todos algún día

Ya me he referido en este libro al concepto *Shu Ha Ri* de las artes marciales. Las personas en el estado *Shu* siguen fielmente las reglas, para conocer las ideas en que se basan. Aquellas otras en el estado *Ha* empiezan a crear su propio estilo dentro de las reglas, adaptándolas a sus necesidades. La gente en el estado *Ri* está más allá de las reglas; encarna los ideales. Ver a un verdadero maestro en el estado *Ri* es como ver una obra de arte en movimiento. Sus acciones parecen imposibles, pero eso se debe a que el maestro ha convertido una filosofía en cuerpo, una idea en realidad.

Todo lo cual preludia el hecho de que en Scrum *hay* ciertas reglas, que bien harías en aprender y trascender. Las he incluido en el apéndice de este libro y escrito un capítulo tras otro sobre por qué existen esas reglas,

alentándote, eso espero, a aplicarlas en tu vida personal, tu compañía y tu comunidad. La paradoja de dichas reglas, sin embargo, es que eliminan fronteras y generan libertad, y para muchos la libertad puede ser aterradora.

Una compañía que ha aprendido a dejar en libertad a sus empleados *y* a optimizar la innovación es Valve. Examinar esta empresa es contemplar cómo podríamos inevitablemente organizarnos todos, sea para hacer mejor *software*, sacar a la gente de la pobreza, planear una boda o reparar una casa.

Formada en la década de 1990 como compañía de videojuegos con éxitos revolucionarios como Half-Life y Portal, Valve se financia sola y es dueña de toda su propiedad intelectual. La casi totalidad de sus más de trescientos empleados se sitúan en una sola torre de oficinas en Bellevue, Washington. Tiene más de cincuenta millones de clientes y gana cientos de millones de dólares al año. Además, en ella nadie está a cargo.

El origen de Valve es nada menos que Microsoft. Hoy en día Microsoft es una compañía muy diferente, pero en los años noventa era el epítome de la corporación vertical. Todos se definían por su distancia en la pirámide corporativa respecto del fundador y director general, Bill Gates, entonces el hombre más rico del mundo y hoy uno de los más ricos aún.

Greg Coomer es uno de los integrantes del grupo que fundó Valve. Trabajó para Gabe Newell, quien dirigía un grupo de desarrollo en Microsoft. Coomer describe cómo esa hiperatención en la estatura se manifestaba en las herramientas que usaba la gente. «En Microsoft había un accesorio de Outlook llamado Org Chart. Y en todos los correos que recibías, si hacías clic en él veías el lugar que el remitente ocupaba en la compañía. A cuántos clics de Bill estaba, cuántos subordinados directos tenía, si eran enemigos o amigos: todo esto podía descubrirse a partir de su puesto en el organigrama».

Coomer dice que si reducías la imagen, podías ver una pirámide gigantesca con Gates en la cima. Si la amplificabas, había un montón de pirámides más pequeñas. «Había pirámides por todos lados».

Menos en el grupo de Gabe Newell. Compuesto por un centenar de personas, todas ellas dependían directamente de él. «Esto destacaba visualmente en la app Org Chart», dice Coomer. «No tenía sentido. Y causaba problemas políticos, porque Gabe no tenía el número indicado de gerentes ni la estructura correcta». La reacción de la compañía era casi como la de glóbulos blancos en ataque masivo contra una infección. Claro que ahora Microsoft tiene ya tres mil empleados que trabajan en equipos con Scrum y transita hacia veinte mil. Pero en aquel entonces esa «infección» debía ser eliminada.

Así, Newell, Coomer y otros se fueron y formaron su propia compañía, Valve. Hace unos años, Coomer trató de elaborar un manual para los empleados que explicara cómo opera Valve. El documento no enumeraba grados salariales ni si los cristales estaban cubiertos por la flexible cuenta de egresos. Más bien, era un intento por transmitir los valores de Valve.

«Calculé que se necesitarían de nueve a dieciséis meses para que la gente interiorizara el estilo Valve de hacer las cosas. Pero pasó más tiempo para que la gente se sintiera potenciada», dice Coomer. El documento perseguía serenar rápidamente al personal, pero Coomer y los demás fundadores batallaron con las palabras, porque no querían dar la impresión de que la explicación procedía de lo alto. La primera sección es «Bienvenido a Planolandia»:

> Ésa es nuestra manera corta de decir que no tenemos gerencia y que nadie «responde a las órdenes» de nadie más. Tenemos un fundador/presidente, pero ni siquiera él es tu jefe. Esta compañía es para que tú la dirijas hacia oportunidades y lejos de riesgos. Tienes autoridad para dar luz verde a proyectos. Tienes autoridad para lanzar productos.
> Una estructura plana elimina todas las barreras organizacionales entre tu trabajo y el cliente que disfruta de él. Toda compañía te dirá que «el cliente es el jefe», pero aquí esa afirmación tiene peso. No hay ningún papeleo que te impida determinar por ti mismo qué quieren nuestros clientes y dárselo.
> Si piensas: «¡Vaya!, eso es mucha responsabilidad», tienes razón[44].

He aquí cómo comienza un proyecto en Valve: alguien decide iniciarlo. Eso es todo. Piensa en cuál es el mejor uso de su tiempo y energía, qué servirá más a la compañía y el cliente, y lo hace. ¿Cómo consigue que otros trabajen con él? Los convence. Si esos otros creen que es buena idea, se unen al equipo, o «conspiración», como se le llama en Valve. La totalidad de los cientos de escritorios de Valve tienen ruedas. Cuando la gente empieza a trabajar en común en un proyecto, vota literalmente con sus escritorios, optando por una configuración nueva.

Coomer describe la manera en que esto operó en el caso del producto Big Picture. Uno de los principales productos de Valve es su plataforma Steam, que brinda videojuegos y *software* a los usuarios. Los juegos tanto de Valve como de terceros se basan en esa plataforma. Ésa es la forma dominante en

que hoy se lanzan juegos para PC. Pero Coomer recuerda que, en cierto momento, él y otros temieron haber llegado ya a todos los clientes posibles, más de cincuenta millones.

«Nos pusimos a pensar cómo estaban creciendo nuestra compañía y Steam, y examinamos cuál creíamos que sería el límite en el número de clientes a los que podíamos llegar. Queríamos llegar a la gente en otros lugares, en su sala, en sus dispositivos móviles, en todo».

Conversó entonces con otras personas: diseñadores, otros individuos. Los convenció de que era buena idea inventar algo que operara en televisores, teléfonos y tabletas, y crearon la idea de Big Picture, un medio de ofrecer videojuegos para esas plataformas. Pero la gente a la que Coomer había convencido no tenía las habilidades necesarias para hacer ese producto. Sabían cómo querían que fuera, pero no tenían la capacidad de implementarlo.

«Comenzamos a burlarnos entonces de cómo podía ser y luego hicimos una película de lo fantástico que sería, que usamos después para reclutar gente en el proyecto. No sabíamos codificar, de modo que teníamos que reclutar gente que lo hiciera».

Y así fue. Un año después lanzaron el producto. ¿Quién decidió cuándo lanzarlo? La gente que trabajó en él. ¿Quién decidió que era bueno? La gente que trabajó en él.

«Cuando una compañía se ha optimizado en torno a la innovación, usualmente cambia en forma fundamental, eliminando estructuras y jerarquías internas, toda estructura interna», dice Coomer. Valve opera en esa forma todo el tiempo. No esperan a que una crisis los obligue a cambiar; cambian *constantemente*. Ése es el modo diario en que dirigen la compañía. Dice su manual:

> Valve no es reacio a toda estructura organizacional; éstas brotan en muchas formas todo el tiempo, pasajeramente. Pero surgen problemas cuando la jerarquía o divisiones codificadas del trabajo no han sido creadas por los miembros del grupo, o cuando esas estructuras persisten durante largos periodos. Creemos que dichas estructuras comienzan inevitablemente a servir a sus necesidades, no a los clientes de Valve. La jerarquía empezará a reforzar su estructura contratando personas que se ajusten a su forma, añadiendo personas para ocupar puestos de apoyo subordinados. Sus miembros son inducidos también a asumir conductas en pos de beneficios que aprovechen la

estructura de poder, en vez de dedicarse a aportar valor a los clientes[45].

Podría parecer que Valve es vulnerable a polizones —personas que quieren sacar provecho del sistema—, pero la revisión colegiada es constante. Claro que la gente es la que decide en qué trabajar, pero si no puede convencer a otros de que su idea es buena, quizá no lo sea. Coomer dice que en vez de darse el *lujo* de contar con alguien que les diga qué hacer, tienen un grupo de iguales que les dice qué piensa de lo que han decidido hacer.

Éste no es un sistema perfecto. Ninguna organización humana lo es. Pero en Valve los intereses del personal suelen ser planteados primero por los miembros del equipo en conversaciones entre sí. Pueden recurrir a los demás para efectos de consulta. Esto puede resultar en realimentación, una brusca medida correctiva o hasta despido. Pero es una decisión en equipo.

La excepción ocurrió en 2013, cuando Valve desarrolló un problema que su sistema no era muy apto para manejar. Por primera vez contrató de golpe a un numeroso grupo de personas. Habiéndose decidido una expansión al *hardware* y los dispositivos móviles, no se tenían las habilidades para ello. Así, se contrató a muchas personas para atacar ese problema.

Pero contratar en forma simultánea a tanta gente no aclimatada al estilo Valve de hacer las cosas causó problemas. Había reductos de empleados que no tomaban decisiones al estilo tradicional de Valve. *Decían a los demás qué hacer* y no cumplían los elevados estándares de la empresa. Normalmente, otros miembros del equipo no habrían tolerado esa conducta, pero como todos los integrantes de ese grupo eran nuevos, sus compañeros no tenían confianza suficiente en la manera de actuar de Valve.

«Entonces, un grupo de gente básica de Valve, con cierta antigüedad, actuó para proteger los valores de la empresa, aunque tuvo que actuar *fuera* de esos valores para hacerlo», dice Coomer. La compañía despidió al mismo tiempo a una docena de personas. Al hablar con Coomer, se advierte que él sigue considerando eso un fracaso. Lo describe casi como una reacción biológica, parecida, curiosamente, a la forma en que Microsoft actuó con los fundadores de Valve: organismos que atacan a invasores extraños para proteger al conjunto.

«Hemos hablado mucho de lo que para las metas que nos hemos fijado significa que hayamos tenido que actuar para sacar a esa gente», reflexiona. «Y de cómo podemos evitarlo en el futuro. Y de no tener que depender de un grupo que lleva mucho tiempo en la compañía». Se detiene un momento y luego dice, seguro: «Dentro de un año lo habremos resuelto».

Hay fe en lo que han hecho. Han buscado sistemáticamente maximizar la libertad, capacidad y creatividad humana. Aunque ha habido tropiezos ocasionales, su manera de operar es muy eficaz como para que no se le reproduzca una y otra vez.

«Ésta es una innovación capitalista tan impresionante como muchas innovaciones industriales que cambiaron la naturaleza del trabajo», dice. «Es tan útil y exitosa que no hay manera de que no sea una fuerza de cambio en el mundo».

¿Usan Scrum? Bueno, dice Coomer, basta recorrer el pasillo para ver gran cantidad de pizarrones blancos, con ruedas, cubiertos de papeletas adhesivas. Pero no *obligan* a la gente a usarlo; le permiten decidir qué proceso es el indicado para ella. Y como en casi todo, Coomer y los demás fundadores se abstienen de decir a otros qué hacer. Sin embargo, muchos empleados de Valve han decidido que, dada la opción de hacer cualquier cosa, prefieren Scrum. Y esto es suficiente para mí.

Aún hay pocas compañías como Valve, aunque cada día aparecen más. Y no sólo en el ramo del *software*. Morning Star, líder global en procesamiento de jitomates, no tiene jefes. Cada empleado negocia con otros los roles y responsabilidades, sea acerca de ventas, conducción de camiones o la ejecución de ingeniería compleja. En cualquier compañía, primero debes lograr que los empleados se liberen, para aceptar después la responsabilidad que eso implica.

O, como lo dijo Funkadelic en 1970, «Libera tu mente… Tu trasero te seguirá».

¿Qué no podemos hacer?

El cinismo es quizá una reacción racional a la desesperación, pero es uno de los estados humanos más corrosivos. Los años iniciales de este siglo han estado llenos de elementos que engendran cinismo: guerras insensatas disfrazadas de patriotismo, terrorismo nihilista encubierto de fe, codicia envuelta en rectitud ideológica, cortesanos políticos ambiciosos que sólo persiguen sus egoístas fines.

El cínico suspira y dice: «Así es el mundo. Los seres humanos somos esencialmente corruptos y egoístas; pretender otra cosa resulta ingenuo». De este modo justifica las restricciones y racionaliza los límites.

En las dos últimas décadas me he sumergido en la bibliografía del origen de la grandeza. La asombrosa respuesta es que, fundamentalmente, los seres humanos *quieren* ser grandes. La gente quiere hacer algo audaz; hacer un mundo mejor, aun en forma modesta. La clave es deshacerse de lo que la obstruye, quitando los impedimentos a su transformación en lo que ella es capaz de ser.

Eso es lo que hace Scrum. Fija metas y resuelve sistemáticamente, paso a paso, cómo cumplirlas. Más todavía, identifica lo que nos impide alcanzarlas.

Scrum es el código del anticínico. No es desear un mundo mejor o rendirse al que ya existe. Es una manera práctica y factible de poner en marcha el cambio. Sé de proyectos con Scrum destinados a repartir vacunas entre niños en riesgo, y de otros dirigidos a construir casas a menor costo, eliminar la corrupción menuda, atrapar criminales violentos, acabar con el hambre y mandar gente a otros planetas.

Nada de lo anterior es un deseo fantasioso; son, en cambio, planes ejecutables. No te confundas; esos planes tendrán que ser inspeccionados, adaptados y modificados a cada paso, pero están en *movimiento*. En todo el mundo ocurren ya repeticiones rápidas que nos lanzan a un mundo mejor.

Espero que eso sea lo que hayas obtenido de este libro: la certeza de que es posible; de que puedes cambiar las cosas, de que no tienes por qué aceptar las cosas como son.

> Todos los hombres sueñan, pero no igual. Los que sueñan de noche en los polvosos recovecos de su mente, despiertan de día para descubrir que eso fue sólo vanidad; pero los que sueñan de día son peligrosos, porque pueden hacer realidad sus sueños con los ojos abiertos, volverlos posibles.
>
> —T. E. Lawrence, *Los siete pilares de la sabiduría*[46]

No escuches a los cínicos que te dicen lo que no se puede hacer. Sorpréndelos con lo que se puede.

RESUMEN

Scrum acelera todos los esfuerzos humanos. El tipo de proyecto o problema no importa; Scrum puede servir para cualquier esfuerzo por mejorar desempeño y resultados.

Scrum para la escuela. En los Países Bajos, un número creciente de maestros usa Scrum para impartir sus cursos de preparatoria. Ven una mejora casi inmediata en resultados de exámenes, de más de diez por ciento. Y atraen a toda clase de estudiantes, desde los destinados a la educación vocacional hasta los dotados.

Scrum para la pobreza. En Uganda, la Grameen Foundation usa Scrum para proporcionar datos agrícolas y comerciales a agricultores rurales pobres. El resultado: el doble de rendimiento y el doble de ingresos para algunas de las personas más pobres del planeta.

Rompe tus tarjetas de presentación. Deshazte de todo distintivo profesional, de todos los gerentes, todas las estructuras. Da a la gente la libertad de hacer lo que considera mejor y la responsabilidad para hacerse cargo de ello. Los resultados te sorprenderán.

Agradecimientos

Ningún proyecto es resultado del trabajo de una sola persona; es producto de un equipo y este libro no es la excepción.

Quiero dar primero las gracias a mi hijo, J. J. Sutherland. Fue él quien sugirió que escribiéramos juntos un libro sobre el notablilísimo viaje por el que Scrum me ha llevado en unos cuantos años. Él quería un descanso tras una década de correr de una guerra y un desastre a otros para NPR y pensó que contar la historia de cómo surgió Scrum, por qué da resultado y cómo ha cambiado al mundo sería no sólo importante, sino también entretenido. El texto que tienes ahora en las manos narra mi historia, pero es producto de muchas horas con mi hijo, quien llevó las palabras a la página.

Howard Yoon, el más hábil de los agentes literarios, nos pidió pensar en grande, vasto y lejos. Su discernimiento, consejos, sabiduría y simple y astuto conocimiento práctico no sólo hicieron posible este libro, sino que también lo llevaron a una escala completamente distinta.

No es común tener la oportunidad de trabajar con un verdadero maestro en su oficio y yo tuve la increíble suerte de disponer de esa oportunidad con Rick Horgan, del Crown Publishing Group. Su tacto, diestro y consumado, hace sencillamente que las cosas *mejoren*. Y consigue que todo parezca fácil. Me quito el sombrero; gracias de verdad.

El primer responsable del producto Alex Brown, Joe Justice y el resto del equipo de Scrum, Inc., compartieron ideas decisivas, energía ilimitada y vasta experiencia.

Gracias también:

A los profesores Hirotaka Takeuchi e Ikujiro Nonaka, cuyo trabajo dio origen a la idea de Scrum y quienes son desde entonces mis buenos amigos.

A mi cocreador, Ken Schwaber, cuya irascible obstinación me ayudó a dar forma a Scrum y convertirlo en la fuerza que es hoy.

Y, sobre todo, a mi esposa, Arline, quien ha estado a mi lado desde el principio y, como ministra unitaria-universalista, introdujo Scrum en muchas iglesias. Hizo un mundo mejor cuando nos enseñó cómo aplicar Scrum a una organización entera.

Gracias, por último, a los cientos de miles de *Scrum Masters*, responsables de productos y equipos de todo el planeta que realmente viven Scrum cada día. Ustedes hacen de Scrum la fuerza vibrante y positiva que es en el mundo y nunca dejan de sorprenderme con lo que han logrado con él.

Apéndice

IMPLEMENTACIÓN DE SCRUM: CÓMO EMPEZAR

Ahora que ya has leído este libro, he aquí, en pocas palabras, cómo iniciar un proyecto con Scrum. Ésta es una descripción del proceso en líneas muy generales, pero debería ser suficiente para comenzar. Este libro fue escrito para darte el *porqué* detrás de Scrum. Esta sección te proporcionará, en forma abreviada, el *cómo*.

1. Elige un **responsable del producto**. Este individuo es quien posee la visión de lo que vas a hacer, producir o lograr. Toma en cuenta los riesgos y recompensas, qué es posible, qué puede hacerse y qué es lo que le apasiona. (Véase el *capítulo ocho*: **Prioridades**, para más información).
2. Selecciona un **equipo**. ¿Quiénes serán las personas que harán efectivamente el trabajo? Este equipo debe contar con todas las habilidades necesarias para tomar la visión de los responsables del producto y hacerla realidad. Los equipos deben ser pequeños, de tres a nueve personas por regla general. (Véase el *capítulo tres*: **Equipos**, para más información).
3. Elige un ***Scrum Master***. Ésta es la persona que capacitará al resto del equipo en el enfoque Scrum y que ayudará al equipo a eliminar todo lo que lo atrasa. (Véase el *capítulo cinco*: **Desperdicio**, para más información).
4. Crea y prioriza una **bitácora del producto**. Se trata de una lista de alto nivel de todo lo que debe hacerse para volver realidad la visión. Esta bitácora existe y evoluciona durante el periodo de vida del producto; es la guía de caminos hacia éste. En un momento dado, la bitácora del

producto es la visión definitiva de «todo lo que el equipo podría hacer, en orden de prioridad». Hay sólo una bitácora del producto; esto significa que el responsable del producto debe tomar decisiones de priorización en todo el espectro. El responsable del producto debe consultar tanto a todos los interesados como al equipo para cerciorarse de que representa lo que la gente necesita y lo que se puede hacer. (Véase el *capítulo ocho*: **Prioridades**, para más información).

5. Afina y estima la **bitácora del producto**. Es crucial que la gente que realmente se hará cargo de los elementos de la bitácora del producto estime cuánto esfuerzo implicarán. El equipo debe examinar cada elemento de la bitácora y ver si, en efecto, es viable. ¿Hay información suficiente para llevar a cabo el elemento? ¿Éste es lo bastante pequeño para estimarse? ¿Existe una definición de «terminado»; es decir, todos están de acuerdo en los criterios que deben cumplirse para poder decir que algo está «terminado»? ¿Esto crea valor visible? Cada elemento debe poder mostrarse, demostrarse y (es de esperar) entregarse. No calcules la bitácora en horas, porque la gente es pésima para esto. Calcula por tamaño relativo: pequeño, mediano o grande. O, mejor todavía, usa la serie de Fibonacci y estima el valor puntual de cada elemento: 1, 2, 3, 5, 8, 13, 21, etcétera. (Véase el *capítulo seis*: **Planea realidades, no fantasías**, para más información).
6. **Planeación del *sprint***. Ésta es la primera de las reuniones de Scrum. El equipo, el *Scrum Master* y el responsable del producto se sientan a planear el *sprint*. Los *sprints* son siempre de extensión fija, inferior a un mes. La mayoría de la gente ejecuta en la actualidad uno o dos *sprints* semanales. El equipo examina el inicio de la bitácora y pronostica cuánto puede llevar a cabo en este *sprint*. Si el equipo ha pasado por varios *sprints*, debe considerar el número de puntos que acumuló en el más reciente. Este número se conoce como **velocidad** del equipo. El *Scrum Master* y el equipo deben tratar de aumentar ese número en cada *sprint*. Ésta es otra oportunidad para que el equipo y el responsable del producto confirmen que todos comprenden a la perfección cómo esos elementos cumplirán la visión. Durante esta reunión todos deben acordar asimismo una meta de *sprint*, que todos han de cumplir en este *sprint*.

   Uno de los pilares de Scrum es que una vez que el equipo se compromete con lo que cree que puede terminar en un *sprint*, eso se

queda ahí. No puede cambiar ni crecer. El equipo debe ser capaz de trabajar en forma autónoma a lo largo del *sprint* para terminar lo que pronosticó que podía hacer. (Véase el *capítulo cuatro*: **Tiempo**, y el *capítulo seis*: **Planea realidades, no fantasías**, para más información).

7. **Vuelve visible el trabajo**. La forma más común de hacerlo en Scrum es crear una **tabla de Scrum** con tres columnas: Pendiente, En proceso y Terminado. Notas adhesivas representan los elementos por llevar a cabo y el equipo avanza por la tabla conforme los va concluyendo, uno por uno.

   Otra manera de volver visible el trabajo es crear un **diagrama de finalización**. En un eje aparece el número de puntos que el equipo introdujo en el *sprint* y en el otro el número de días. Cada día, el *Scrum Master* suma el número de puntos completados y los grafica en el diagrama de finalización. Idealmente, habrá una pendiente descendente que conduzca a cero puntos para el último día del *sprint*. (Véase el *capítulo siete*: **Felicidad**, para más información).

8. **Parada diaria** o **Scrum diario**. Éste es el pulso de Scrum. Cada día, a la misma hora, durante no más de quince minutos, el equipo y el *Scrum Master* se reúnen y contestan tres preguntas:

   - ¿Qué hiciste ayer para ayudar al equipo a terminar el *sprint*?
   - ¿Qué harás hoy para ayudar al equipo a terminar el *sprint*?
   - ¿Algún obstáculo te impide o impide al equipo cumplir la meta del *sprint*?

   Eso es todo, en eso consiste la reunión. Si se prolonga más de quince minutos lo estás haciendo mal. Lo que esto hace es ayudar al equipo a saber exactamente dónde se encuentra todo en el curso de un *sprint*. ¿Todas las tareas serán terminadas a tiempo? ¿Hay oportunidades de ayudar a otros miembros del equipo a vencer obstáculos? Las tareas no se asignan desde arriba; el equipo es autónomo: él lo hace. Tampoco se rinden informes detallados a la dirección. El *Scrum Master* se encarga de eliminar los obstáculos, o impedimentos, contra el progreso del equipo. (Véase el *capítulo cuatro*: **Tiempo**, y el *capítulo seis*: **Planea realidades, no fantasías**, para más información).

9. **Revisión del *sprint*** o **demostración del *sprint***. Ésta es la reunión en la que el equipo muestra lo que hizo durante el *sprint*. Todos pueden asistir, no sólo el responsable del producto, el *Scrum Master* y el equipo,

sino también los demás interesados, la dirección, clientes, quien sea. Ésta es una reunión abierta en la que el equipo hace una demostración de lo que pudo llevar a Terminado durante el *sprint*.

El equipo debe mostrar únicamente lo que satisface la definición de Terminado, lo total y completamente concluido y que puede entregarse sin trabajo adicional. Esto puede no ser un producto terminado, pero sí una función concluida de uno de ellos. (Véase el *capítulo cuatro*: **Tiempo**, para mayor información).

10. **Retrospectiva del *sprint***. Una vez que el equipo ha mostrado lo que logró en el *sprint* más reciente —la cosa «terminada» y en posibilidad de enviarse a los clientes en busca de realimentación—, piensa en qué marchó bien, qué pudo haber marchado mejor y qué puede mejorar en el siguiente *sprint*. ¿Cuál es la mejora en el proceso que como equipo pueden implementar de inmediato?

    Para ser eficaz, esta reunión requiere cierto grado de madurez emocional y una atmósfera de confianza. La clave es que no se trata de buscar a quién culpar; lo que se juzga es el proceso. ¿Por qué tal cosa ocurrió de tal manera? ¿Por qué pasamos por alto tal otra? ¿Qué podríamos hacer más rápido? Es crucial que la gente, como equipo, asuma la responsabilidad de su proceso y de sus resultados y busque soluciones también como equipo. Al mismo tiempo, debe tener fortaleza para tocar los temas que le incomodan de un modo orientado a la solución, no acusatorio. Y el resto del equipo ha de tener la madurez de oír la realimentación, aceptarla y buscar una solución, no ponerse a la defensiva.

    Al final de la reunión, el equipo y el *Scrum Master* deben acordar una mejora al proceso que implementarán en el siguiente *sprint*. Esa mejora al proceso, también llamada *kaizen*, debe incorporarse en la bitácora del *sprint* siguiente, con pruebas de aceptación. De esta manera, el equipo podrá ver fácilmente si en verdad implementó la mejora y qué efecto tuvo ésta en la velocidad. (Véase el *capítulo siete*: **Felicidad**, para más información).

11. Comienza de inmediato el ciclo del siguiente *sprint*, tomando en cuenta la experiencia del equipo con los impedimentos y mejoras del proceso.



Jeff Sutherland es militar de formación, egresado de la United States Military Academy, donde se graduó como piloto de combate. Intervino en más de cien misiones en Vietnam. Posteriormente realizó estudios en Stanford y posee un postgrado en la School of Medicine de la University of Colorado.

Es director de Scrum Inc. y consejero senior de Openview Venture Partners, donde asesora a compañías con capital de alto riesgo. Viaja por el mundo dictando conferencias y talleres.

www.scruminc.com

Notas

[1] Dan Eggen y Griff Witte, «The FBI's Upgrade That Wasn't; $170 Million Bought an Unusable Computer System», en *Washington Post*, 18 de agosto de 2006, p. A1. <<

[2] *Status of the Federal Bureau of Investigation's Implementation of the Sentinel Project*, Departamento de Justicia de Estados Unidos, Oficina del Inspector General, informe 11-01, octubre de 2010. <<

[3] *Ibíd*. <<

[4] Taiichi Ohno, *Toyota Production System: Beyond Large-scale Production*, Productivity, Cambridge, 1988. <<

[5] Theodore Roosevelt, «Citizenship in a Republic», discurso pronunciado en la Sorbona de París el 23 de abril de 1910. <<

[6] Hirotaka Takeuchi e Ikujiro Nonaka, «The New New Product Development Game», en *Harvard Business Review*, enero-febrero de 1986, pp. 285-305. <<

[7] Ken Schwaber, «Scrum Development Process», en J. Sutherland, D. Patel, C. Casanave, J. Miller y G. Hollowell, eds., *OOPSLA Business Object Design and Implementation Workshop*, Springer, Londres, 1997. <<

[8] W. Edwards Deming, «To Management», discurso pronunciado en el Mt. Hakone Conference Center, Japón, 1950. <<

[9] Hirotaka Takeuchi e Ikujiro Nonaka, «The New New Product Development Game», en *Harvard Business Review*, enero-febrero de 1986, pp. 285-305. <<

[10] Douglas MacArthur, «The Long Gray Line», discurso pronunciado en West Point, Nueva York, 1962. <<

[11] *Ibíd*. <<

[12] Richard Feynman, Report of the *Presidential Commission on the Space Shuttle Challenger Accident*, Appendix F: Personal Observations on Reliability of Shuttle, informe, 1986. <<

[13] Joby Warrick y Robin Wright, «U. S. Teams Weaken Insurgency in Iraq», en *Washington Post*, 6 de septiembre de 2008. <<

[14] Michael Flynn, Rich Jergens y Thomas Cantrell, «Employing ISR: SOF Best Practices», en *Joint Force Quarterly*, núm. 50, tercer trimestre de 2008, p. 60. <<

[15] Christopher Lamb y Evan Munsing, «Secret Weapon: High Value Target Teams as an Organizational Innovation», *Strategic Perspectives*, núm. 4, 2011. <<

[16] Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, Reading, 1975. <<

[17] Nelson Cowan, «The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity», en *Behavioral and Brain Sciences*, núm. 24, 2001, pp. 87-185. <<

[18] Richard Nisbett, Craig Caputo, Patricia Legant y Leanne Marecek, «Behavior as Seen by the Actor and as Seen by the Observer», en *Journal of Personality and Social Psychology*, vol. 27, núm. 2, 1973, pp. 154-164. <<

[19] Stanley Milgram, «The Perils of Obedience», en *Harper's Magazine*, 1974. <<

[20] Andrew Marvell, «To His Coy Mistress», Poets.org. <<

[21] Taiichi Ohno, *Toyota Production System: Beyond Large-scale Production*, Productivity, Cambridge, 1988. <<

[22] David Strayer, Frank Drews y Dennis Crouch, «A Comparison of the Cell Phone Driver and the Drunk Driver», en *Human Factors*, vol. 48, núm. 2, verano de 2006, pp. 381-391. <<

[23] D. M. Sanbonmatsu, D. L. Strayer, N. Medeiros-Ward y J. M. Watson, «Who Multi-Tasks and Why? Multi-Tasking Ability, Perceived Multi-Tasking Ability, Impulsivity, and Sensation Seeking», en *PLoS One*, 2013, 8(1), e54402.doi:10.1371/journal.pone.0054402. <<

[24] Gerald M. Weinberg, *Quality Software Management*, Dorset House, Nueva York, 1991. <<

[25] Harold Pashler, «Dual-task Interference in Simple Tasks: Data and Theory», en *Psychological Bulletin*, vol. 116, núm. 2, 1994, pp. 220-244. <<

[26] S. Charron y E. Koechlin, «Divided Representation of Concurrent Goals in the Human Frontal Lobes», en *Science*, vol. 328, núm. 5976, 2010, pp. 360-363. <<

[27] Glenn Wilson, The Infomania Study (número resumido) http://www.drglennwilson.com/Infomania_experiment_for_Hp.doc. <<

[28] James P. Womack, Daniel T. Jones y Daniel Roos, *The Machine That Changed the World: The Story of Lean Production*, HarperPerennial, Nueva York, 1991. <<

[29] Avnaim-Pesso, Danziger y Levav, «Extraneous Factors in Judicial Decisions», en *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, núm. 17, 2011. <<

[30] K. Vohs, R. Baumeister, J. Twenge, B. Schmeichel, D. Tice y J. Crocker, *Decision Fatigue Exhausts Self-Regulatory Resources - But So Does Accommodating to Unchosen Alternatives*, 2005. <<

[31] Mike Cohn, *Agile Estimation and Planning*, Prentice Hall, Upper Saddle River, 2005. <<

[32] Sushil Bikhchandani, David Hirshleifer e Ivo Welch, «A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades», en *Journal of Political Economy*, vol. 100, núm. 5, 1992, pp. 992-1026. <<

[33] Edward L. Thorndike, «A Constant Error in Psychological Ratings», en *Journal of Applied Psychology*, vol. 4, núm. 1, 1920, pp. 25-29. <<

[34] Norman Dalkey y Olaf Helmer, «An Experimental Application of the DELPHI Method to the Use of Experts», en *Management Science*, vol. 9, núm. 3, 1963, pp. 458-467. <<

[35] Sonja Lyubomirsky, Laura King y Ed Diener, «The Benefits of Frequent Positive Affect: Does Happiness Lead to Success?», en *Psychological Bulletin*, vol. 131, núm. 6, 2005, pp. 803-855. <<

[36] Gretchen Spreitzer y Christine Porath, «Creating Sustainable Performance», en *Harvard Business Review*, enero-febrero de 2012, pp. 3-9. <<

[37] *Ibíd*. <<

[38] William Shakespeare, *El Rey Lear*, en *Obras completas*, vol. II, traducción de Luis Astrana Marín, Aguilar, Madrid, 1932, acto 1, escena 4. <<

[39] John Shook, «The Remarkable Chief Engineer», Lean Enterprise Institute, 3 de febrero de 2009. <<

[40] Daniel Ford, *A Vision So Noble: John Boyd, the OODA Loop, and America's War on Terror*, CreateSpace Independent, 2010. <<

[41] John Boyd, *New Conception*, 1976. <<

[42] *Ibíd*. <<

[43] Brad Shannon, «McKenna, Inslee Outline Plans to Bring Efficiency to Government», en *The Olympian*, 6 de octubre de 2012. <<

[44] «Valve Handbook for New Employees», Valve Press, 2012. <<

[45] *Ibíd*. <<

[46] T. E. Lawrence, *Seven Pillars of Wisdom: A Triumph*, Cape, Londres, 1973. <<

JEFF SUTHERLAND

SCRUM

EL ARTE DE HACER EL DOBLE DE TRABAJO EN LA MITAD DE TIEMPO



Lectulandia