

22 de febrero de 2017  
Bogotá D.C.



William Sierra	<a href="mailto:william.sierra@exsis.com.co">william.sierra@exsis.com.co</a>
Carlos Cardona	<a href="mailto:carlos.cardona@exsis.com.co">carlos.cardona@exsis.com.co</a>
Ervid Molina	<a href="mailto:ervid.molina@exsis.com.co">ervid.molina@exsis.com.co</a>

### **The Clean Coder Cap. 9 Gestión del tiempo**

¿Qué estrategia puede utilizar para asegurarse de que no pierda el tiempo destinado para el desempeño de su labor? ¿Cómo puede manejar eficazmente su tiempo?

Los profesionales son conscientes del alto costo de las reuniones. Contemplan los salarios, beneficios, costos de las instalaciones, y demás gastos involucrados en las funciones de secretarías y asistentes. La próxima vez que se encuentre en una reunión, calcule el costo. Podría sorprenderse. La verdad es que las reuniones son necesarias, pero también son grandes pérdidas de tiempo. Hay reuniones que cumplen las 2 premisas. Algunos asistentes pueden encontrarles un valor incalculable; otros pueden encontrarlos redundantes o inútiles.

Por eso hay que aprender cómo declinar educadamente una reunión a la cual es invitado. Decida con cautela a cuales reuniones asiste y a cuáles no, a menos que su participación sea requerida y significativamente necesaria para el trabajo que está haciendo ahora. Una de las tareas más importantes de su superior es mantenerlo fuera de las reuniones. Un buen jefe estará más que dispuesto a defender su decisión de rechazar la asistencia porque esa persona está tan preocupada por su tiempo como usted.

También es cierto que las reuniones no siempre van según lo planeado. Algunas veces usted se encuentra en una reunión que hubiera rechazado si hubiera tenido un poco más de información. Si se encuentra atascado en una reunión que no dispone del buen uso de su tiempo, entonces usted necesita encontrar una manera cortés para salir de la reunión. Por lo que puede preguntar, en un momento oportuno, si su presencia sigue siendo necesaria. Usted puede argumentar que no puede permitirse mucho más tiempo, o preguntar si hay una manera de agilizar la discusión o acortar la agenda.

La razón por la que se dispone a soportar el costo de la reunión es que a veces se necesita a los participantes juntos en una sala para ayudar a alcanzar una meta específica. Para aprovechar el tiempo de los participantes sabiamente, la reunión debe tener una agenda

clara, con tiempos para cada tema y un objetivo definido (Temas a tratar, tiempos asignados y los objetivos que se deben cumplir). Si usted no puede obtener una respuesta clara sobre estas cosas, entonces cortésmente decline su asistencia.

Las reuniones pueden ser guiadas por unas preguntas dadas por los cánones ágiles. Su metodología se basa en que cada participante espera a que el anterior responda las siguientes preguntas:

1. ¿Qué hice ayer?
2. ¿Qué voy a hacer hoy?
3. ¿Qué hay en mi camino?

Eso es todo. Cada pregunta no debe requerir más de veinte segundos, por lo que cada participante no debe exigir más de un minuto. Incluso en un grupo de diez personas, esta reunión debería terminar mucho antes de que hayan transcurrido diez minutos.

Hay un tipo de reunión difícil de sacar adelante, de acuerdo a los cánones ágiles. Esta es la reunión de planificación de una iteración. Pueden realizarse mal, o tomar demasiado tiempo. Se necesita habilidad (vale la pena aprenderla) para que estas reuniones salgan bien.

Este tipo de reunión está diseñada para seleccionar los elementos de respaldo que se podrían implementar en la siguiente iteración. Las estimaciones son dadas por las variables consideradas. La evaluación del valor del negocio debe hacerse. En las organizaciones realmente buenas, las pruebas de aceptación / componente ya estarán escritas, o al menos esbozadas

Otro tipo de reunión es la de la iteración de retrospectiva y demo. Estas reuniones se llevan a cabo al final de cada iteración. Los miembros del equipo discuten lo que salió bien y lo que salió mal. Los stakeholders observan un demo de las nuevas características en funcionamiento. Estas reuniones pueden ser mal llevadas y pueden absorber mucho tiempo, por lo que se deben destinar 45 minutos al final de cada iteración, para realizar la reunión. Se asigna no más de 20 minutos para la retrospectiva y 25 minutos para el demo.

"Cualquier argumento que no pueda resolverse en cinco minutos no puede ser resuelto por argumentos". La razón por la que se prolonga es que no hay evidencia clara que apoye a ninguna de las partes. El argumento es probablemente religioso, en contraposición a los hechos. Los desacuerdos técnicos tienden a salirse de contexto. Cada parte tiene todo tipo de justificaciones para su posición pero rara vez tiene datos. Sin datos, cualquier argumento que no llegue a un acuerdo dentro de unos minutos (entre cinco y treinta) simplemente no llegará a un acuerdo. Lo único que hay que hacer es ir a buscar información.

Algunas personas tomarán actitudes pasivo-agresivas. Ellos estarán de acuerdo sólo para poner fin a la discusión. Se dirán a sí mismos: "Esta es la forma en que lo querían, y ahora van a conseguir lo que querían." Este es probablemente el peor tipo de comportamiento no profesional que existe. Nunca haga esto. Si usted está de acuerdo, entonces participe. Tenga cuidado con las reuniones que son realmente sólo un pretexto para ventilar un desacuerdo y para recoger el apoyo de un lado u otro. Y evitar aquellas reuniones en las que sólo una de las partes está argumentando. Si un argumento realmente debe ser resuelto, entonces pida a cada uno de las partes que presenten su caso al equipo en cinco minutos o menos. Luego haga que el equipo vote. Toda la reunión tomará menos de quince minutos.

### **"Focus-Mana"**

La programación es un ejercicio intelectual que requiere largos períodos de concentración y enfoque. El enfoque es un recurso escaso, como el mana. Después de haber gastado su "focus-mana", usted debe recargarlo haciendo actividades que no requieran concentración por una hora o más.

Los desarrolladores profesionales aprenden a manejar su tiempo para tomar ventaja de su "focus-mana". Se escribe código cuando el "focus-mana" es alto, y se practican otras actividades menos productivas cuando no está. El "focus-mana" también es un recurso decadente. Si no lo usa cuando está, probablemente lo pierda. Esa es una de las razones por las que una reunión puede ser devastadora. Si gasta todo su "focus-mana" en una reunión, no tendrá nada para codificar. Las preocupaciones y las distracciones consumen su "focus-mana" rápidamente.

No hay mucha evidencia sobre esto, pero gran parte del "focus-mana" se recupera después de una buena noche durmiendo. Siete horas de sueño se reflejan en 8 horas de "focus-mana" al día siguiente. De este modo los desarrolladores profesionales administran su horario de sueño para asegurarse de que han recargado su "focus-mana" para cuando lleguen a trabajar por la mañana.

No hay duda de que algunas personas pueden hacer un uso más eficiente del "focus-mana" consumiendo cantidades moderadas de cafeína. Pero tenga cuidado. La cafeína también pone un extraño "jitter"(disparador) en su enfoque. En exceso puede enviar el "focus-mana" en direcciones extrañas y desperdiciar todo ese enfoque en cosas que no valen la pena.

El "focus-mana" puede ser parcialmente recargado por la desfocalización. Un buen paseo prolongado, una conversación con los amigos, un tiempo de observar un paisaje puede ayudar a bombear de regreso el "focus-mana". Algunas personas meditan. Otras personas toman una siesta. Otros escucharán un podcast o un vistazo a una revista.

Hay algo peculiar en practicar disciplinas físicas como las artes marciales, el tai-chi o el yoga. A pesar de que estas actividades requieren un enfoque significativo, es un tipo diferente de enfoque al de la codificación. No es intelectual, es muscular. Y de alguna manera el enfoque muscular ayuda a recargar el enfoque mental. Es más que una simple recarga. Sin embargo, este enfoque puede manifestarse de otras maneras: andar en bicicleta, escuchar música o a la naturaleza, algunos hobbies (carpintería, jardinería, etc.), ejercicios, etc.

Es esencial para el enfoque equilibrar la producción con la entrada apropiada. El desarrollo de software es un ejercicio creativo. Un desarrollador profesional es más creativo cuando está expuesto a la creatividad de otras personas.

Una forma muy efectiva que se utiliza para administrar el tiempo y enfoque es usar la bien conocida Técnica Pomodoro, o también conocida tomates. Es una analogía que sirve para dividir el tiempo entre productividad e interrupciones. Se entiende la productividad como tiempo dedicado al trabajo y que produce resultados. Las interrupciones son actividades que no representan tiempo en el trabajo y que no aportan al cumplimiento de los objetivos.

A veces su cuerpo está en el trabajo, pero su mente y/o corazón no. Puede ser que lo que tenga que hacer le produzca miedo, incomodidad o aburrimiento. Tal vez piense que le forzará a una confrontación o le llevará a un hueco sin fin del cual no se puede escapar. O tal vez simplemente no quiere hacerlo.

Cualquiera que sea la razón, usted encuentra maneras de evitar hacer el trabajo real. Se convence de que existe algo totalmente aparte de su trabajo que es más urgente, y lo hace en su lugar. Esto se llama inversión de prioridad: Aumentar la prioridad de una tarea para posponer tareas verdaderamente importantes. Los profesionales evalúan la prioridad de cada tarea, haciendo caso omiso de sus miedos y deseos personales, ejecutando esas tareas basados en la prioridad de cada una.

La prudencia y experiencia le ayudarán a evitar ciertos callejones sin salida, pero nunca los evitará todos. Así que la verdadera habilidad que necesita es darse cuenta rápidamente cuando está en uno, y tener el valor de retroceder. Esto se conoce como La regla de los agujeros: Cuando se encuentre en uno, deje de cavar.

Como desarrollador se va a encontrar con pantanos. Los pantanos son peores que los callejones sin salida porque siempre se puede ver el camino a seguir, y siempre se ve más corto que el camino de regreso (pero no lo es). El problema es que comenzar un pantano, es como ir por un callejón sin salida: es inevitable. La experiencia y la prudencia pueden ayudar a evitarlos, pero eventualmente tomará una decisión que conduce a un pantano. Los profesionales temen mucho más a los pantanos que a los callejones sin salida. Siempre están al acecho de los pantanos que comienzan a crecer sin límite, y se gastará todo el esfuerzo necesario para escapar de ellos tan pronto sea posible.

Concluimos que los profesionales de software son diligentes en la gestión de su tiempo y su enfoque. Comprenden las tentaciones de la inversión de prioridad y la combaten como una cuestión de honor. Mantienen sus alternativas conservando una mente abierta sobre otras soluciones. Nunca se aferran una solución. Y siempre están al acecho de los pantanos, y los limpian tan pronto como son reconocidos. No hay una visión más triste que un equipo de desarrolladores de software sin freno a través de un pantano cada vez más profundo.