

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

***“El fracaso de un proyecto siempre está determinado por la mala combinación de esos tres conceptos: talento, tecnología y metodología; o lo que es lo mismo, el éxito de un proyecto siempre cuenta con esos tres factores bien equilibrados, pero los tres a la vez y sin la ausencia de ninguno de ellos.”***

Recientemente me pidieron que evaluara el estado de un proyecto software, mezcla de ERP, CRM, etc. El cliente, en este caso el dueño y gerente de la empresa quería conocer el grado de avance del equipo de desarrollo interno responsable del mismo.

La preocupación era evidente: pasaban las fechas previstas y no tenía narices de ver algo que le permitiera suponer que se estaba avanzando en un proyecto que, en definitiva, él estaba financiando y que iba a ser vital para el desarrollo futuro de la compañía.

El equipo de desarrollo, por su parte, se mostraba reacio a mostrar nada y ni siquiera eran capaces de ofrecer una fecha estimada de entrega de la solución. Me pregunto si en software tiene sentido o no *entregar* algo terminado y cerrado a cal y canto. La frustración e inquietud del primero era proporcional a la incertidumbre del equipo de desarrollo: una situación muy complicada, tensa y desagradable para todos los involucrados.

Me bastó una hora de charla con el responsable de la solución para detectar varios de los males endémicos de nuestra profesión. Veinte

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

minutos de mirar el código fueron suficientes para descubrir muchos *debes* que clamaban al cielo; menos tiempo aún necesité para sacar ciertas conclusiones al ver las *evidencias* que el equipo de desarrollo había ido dejando en la herramienta de gestión de proyecto. Demasiadas similitudes con otros proyectos en los que yo mismo había participado. ¿Por qué tanta gente tropezamos siempre con las mismas piedras?.

No me resulta nada agradable cuando me ponen en situación de evaluar el trabajo de un compañero que se gana la vida haciendo lo mismo que tú, con mayor o menor acierto; presuponiendo la buena voluntad y disposición al trabajo, es muy incómodo detectar (y comunicar) las insuficiencias y carencias. Parto de la base de que todo el mundo hace lo que puede y en muchísimas ocasiones no se nos dan suficientes condiciones necesarias para avanzar en el trabajo propuesto de manera correcta y con calidad.

Este es un caso concreto con el que me he encontrado en el último año pero que sin embargo es paradigmático en un gran porcentaje de proyectos software: la mayoría de los que fracasan, lo hacen por tener una combinación no equilibrada de talento, tecnología y metodología. Pero, ¿qué entendemos por estos tres conceptos para el éxito de un proyecto?

No hace falta ser ningún gurú (¿he dicho ya que los gurús o *máquinas* de la computación casi nunca sirven para trabajar en un grupo de desarrollo?), ni ningún *bill gates* ni nada por el estilo: el sentido común se impone. Me pregunto por qué no hay clases durante nuestra formación de sentido común para informáticos. El éxito en la consecución de un

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

proyecto software, más o menos complejo, que involucra a un equipo de desarrollo con roles diferenciados, no está en un único factor determinado, sino en varios y éstos tienen que estar suficientemente equilibrados.

Algunos miembros del equipo pueden ser geniales, súper listos y desplegar talento por los cuatro costados; les gusta todo lo que hacen así como investigar nuevas vías para hacer y mejorar lo que ya funciona. Son los que causan el efecto "*guan, a mí nunca se me habría ocurrido*"; van tan rápido en su forma de causar asombro que no tienen tiempo de documentar mínimamente lo que hacen (total, si ya es genial, para qué), tampoco les gusta que nadie les ponga encima varios *posixs* con las tareas que deben hacer para los próximos días porque eso puede coartar su creatividad. En cuanto leen un artículo con un trabajo estupendo de lo que yo llamo *ingeniería-fontaneril* (o sea, de tocar las tripas de), rápidamente sienten la necesidad de aplicarlo a lo que tienen en ese momento entre manos *de la manera que sea*, aunque entre con calzador. Están, digamos, viciados por su propia genialidad, despliegan un talento avasallador, pero no les pidas que centren en tiro, se ciñan a los estándares tecnológicos definidos y que den visibilidad a su manager, jefe, *scrum master* o lo que sea de lo que están haciendo en ese momento.

Su genialidad y destreza les hacen ser imprevisibles.

Por otro lado, otros miembros del equipo van a un ritmo, digamos, menos acelerado, les cuesta más entender lo que tienen que hacer, meten la pata recurrentemente y para ellos la proactividad es una palabra demasiado compleja. Hacen lo que pueden, tutorizados siempre por alguien de modo

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

que no se les puede pedir que hagan demasiados inventos ni que tengan demasiada iniciativa.

Unos y otros son los extremos contrapuestos del **talento**. Yo siempre asumo que la gente es mucho más inteligente de lo que aparenta y que existen verdaderos diamantes en bruto que ni ellos mismos pueden sospechar que lo son. La falta de talento no suele coincidir en una merma de recursos intelectuales... sino de una falta de motivación, actitud, etc. Como indica la cita, *no suele haber malos alumnos, sino malos maestros*.

Por otra parte, algunos miembros del equipo tienen un dominio de la tecnología que asombra: les basta con ver el síntoma de un problema para detectar dónde está la causa, saben qué trabajo de fontanería tienen que tocar para conseguir un resultado, les gusta eso de *cuanto-mas-intrincado-e-interno-mejor*.

En cambio, otros, *googlean* continuamente cada vez que tienen que implementar una clase abstracta y no saben por qué protesta el compilador, buscan en foros ese nuevo mensaje del IDE que les dice que hay dependencia circulares o ante el problema "mostrar por orden alfabéticamente la lista List<> de" buscan por "mostrar por orden alfabéticamente una lista List<>". Son, en definitiva, gente despierta pero para *otras cosas*. O a lo mejor, estaban más hechos al mundo Java donde sí se movía con extrema facilidad pero ahora se han visto obligados a usar C#, por poner un ejemplo.

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

De nuevo estamos ante el ying y el yang del mismo concepto: **el dominio o no de la tecnología** que se ha decidido usar. No hay que confundirlo con el talento sino con conocer suficientemente bien las herramientas, *frameworks*, arquitecturas, etc. necesarias para el trabajo.

De nada nos sirve tener el suficiente talento y conocer suficientemente bien las herramientas y tecnologías si no tenemos un marco de trabajo ordenado que nos permita avanzar en equipo.

Esta cuestión es clave y se sintetiza en las siguientes dos preguntas: ¿se ha decidido seguir una metodología en el desarrollo de la solución?; si a esta pregunta se responde con un "sí", entonces viene la segunda pregunta, ¿pero de verdad se sigue con igual intensidad y fuerza esa metodología a lo largo de toda la vida del proyecto? Ya el tema en sí de plantear o no el usar una metodología tiene sustancia y contenido para un libro aparte...

Esto es como los coleccionables después del verano o los propósitos bien intencionados del inicio de año: se confunde en ocasiones las intenciones con la realidad, ya que nos jactamos de seguir tal o cual metodología pero no reconocemos que comenzamos muy bien, después flojeamos y cuanto tenemos la más mínima presión de tiempo, lo primero que abandonamos es ¡la metodología!

Sobre esto hablamos largo y tendido en El Libro Negro del Programador, pero dejadme recordaros que existe una realidad muy sutil sólo al alcance de iniciados: cuando la presión se presenta en un proyecto software, lo primero que muere es la metodología, es su primera víctima,

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

¡pero es que seguramente es la falta de metodología la que ha provocado esa misma presión! Entender esto requiere de cierto esfuerzo introspectivo cuando fracasan los proyectos se analizan las causas de esos fracasos.

Implantar una metodología sin la disciplina de aplicarla día a día es no seguir una metodología, así de simple.

Por tanto, vemos que hay tres conceptos que aplican a nuestros desarrollos de software y a continuación nos preguntamos cómo estamos en cada uno de ellos: ¿tenemos talento?, o, ¿la gente percibe que tenemos talento para lo que nos dedicamos?, ¿dominamos la tecnología que se ha decidido usar?, ¿el manager o gestor del proyecto cuenta con la disciplina suficiente para implantar la metodología con igual fuerza en el principio, el durante y el final del proyecto?, ¿nos ceñimos en nuestro rol a esa metodología *siempre*?. ¿Contamos con la necesaria formación y recursos para usar la tecnología del proyecto suficientemente bien?

Veremos que las respuestas a estas preguntas, si parten de la sinceridad, nos darán un "sí, sí, por supuesto en esto y aquello pero no en lo otro", o "pues a mí es que eso de documentar..." o "lo mío es codificar, entro en estado de fluidez y se me pasan las horas volando, pero es que las reuniones de control de proyecto me matan, para mí son pérdida de tiempo", y un largo corolario de posibles combinaciones.

El fracaso de un proyecto siempre está determinado por la mala combinación de esos tres conceptos: talento, tecnología y metodología; o lo que es lo mismo, el éxito de un proyecto siempre cuenta con esos tres

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

factores bien equilibrados, **pero los tres a la vez** y sin la ausencia de ninguno de ellos.

Esto uno tarda años en darse cuenta y lo que es peor, apenas lo tienen en consideración los equipos gestores a la hora de organizar un proyecto. Uno de los fallos fundamentales de un gestor de proyecto es desconocer la fortaleza o debilidad de estos tres aspectos en el equipo que gestiona.

De nada nos sirve tener un talento radiante y natural en resolver problemas software si no dominamos suficientemente la tecnología subyacente que usamos, de poco nos sirve dominar hasta extremos esta tecnología si no tenemos la disciplina de seguir una metodología, la que se haya definido en el proyecto e igualmente, por mucha disciplina metodológica que sigamos, el proyecto no va a terminar bien si no somos capaces de hacer un módulo software sin errores.

¿Qué pasó en el caso que comentaba al principio, aquel proyecto que me pidieron que evaluara?

En ese lamentable caso, fallaron los tres pilares del éxito de un proyecto software: el talento era insuficiente, en forma de un equipo sin la necesaria experiencia en un proyecto similar (no digo que fuesen tontos, ni muchísimo menos, eran y son tíos muy listos y competentes), el dominio de la tecnología a usar era mejorable (en ese caso .NET *framework* y tecnologías relacionadas, *Entity Framework*, etc.) y, para colmo, se comenzó el proyecto con mucha fuerza siguiendo *scrum* pero ésta se fue diluyendo poco a poco (a

## El éxito siempre viene de la unión entre Talento, Tecnología y Metodología

las pocas semanas más bien), al no percibir claramente la rentabilidad metodológica de aplicarla disciplinadamente.

El resultado de todo esto no es sólo un proyecto que fracasa y un equipo frustrado que realmente no termina por entender qué se ha hecho mal y que en realidad ha trabajado con la mejor voluntad del mundo, además hay un agujero económico que alguien tiene que cubrir.

### Puntos clave

---

- Muchos proyectos software fracasan porque no ha existido el correcto equilibrio entre talento, tecnología y metodología.
- En mi humilde experiencia, se suele disponer de mucho talento y conocimiento de la tecnología y casi siempre lo que falla es la correcta implantación de una metodología, cualquiera que sea esta.
- Es preferible tener la suficiente combinación de esos tres aspectos que contar con auténticos *gurús* o llaneros solitarios incapaces de trabajar correctamente en equipo.
- El responsable del equipo debe evaluar estos tres factores y detectar fortalezas y debilidades (aunque reconozco que esto es siempre mucho más fácil de decir que de hacer).