



*A veces me siento y pienso...*



*y a veces, nada más me siento*

## Trabajo en Equipo 2

Siguiendo con el tema iniciado en la columna anterior y con la analogía entre equipos deportivos (deportes de conjunto) y equipos de trabajo, sobre todo en el desarrollo de software, intuyo que existen más similitudes que discrepancias.

Cualquier equipo deportivo de prestigio cuenta con un entrenador, un preparador físico, algunos entrenadores para posiciones específicas, jugadores y entrenamiento, mucho entrenamiento. Un factor importante es que siempre intentan tener al mejor jugador de la posición, aunque esto no siempre es posible, pues depende del bolsillo. También cada conjunto tiene un esquema de juego que depende de la idea de juego del entrenador y evidentemente de las capacidades de los elementos de los que se compone. A grandes rasgos podemos pensar que estos son algunos de los factores importantes en un equipo deportivo.

El éxito o fracaso de estos equipos depende de tres factores: de los entrenamientos, de que los elementos que participan en un juego determinado realicen de manera correcta su función y del conjunto al que enfrentan. Un equipo que no entrena, que no ensaya sus jugadas y acciones en situaciones específicas y que durante el partido trabaja (juega) sin coordinación, sin colaboración, sin respetar sus funciones, lo más probable es que vaya de derrota en derrota. Algo que puede parecer una tontería es el tiempo; cada equipo tiene para cada juego y para toda la temporada o torneo un tiempo concreto. No se puede pretender ganar fuera de esa cota.

No hay nada que nos impida pensar y considerar al desarrollo de software como un juego, un juego sui géneris. Quizá lo menos visible es el contrincante, ¿contra quien competimos? La salida sencilla que estará en la mente de muchos es que se compite contra otras compañías, contra otros grupos de desarrollo... es aceptable.

Es un tópico muy socorrido el hecho de que el desarrollo de productos de software es prácticamente una labor de conjunto. Es decir, se necesita un equipo.

Es evidente que de forma inmediata y dado lo que he dicho (escrito en realidad) tendamos a identificar los factores que mencioné al hablar de equipos deportivos.

Lo más probable es que señalemos claramente que la primera aproximación es entre jugadores y desarrolladores (llamémosles así). Intuitivamente sabemos que así como no es posible formar un equipo de puros porteros o delanteros, tampoco es posible tener un grupo de trabajo de solo programadores o analistas. MoProSoft, TSP o RUP (Modelo de Proceso de Software, Team Software Process, Rational Unified Process) proponen que cada miembro del equipo debe tener un papel asignado (role) definiendo con esto las responsabilidades que cada jugador debe cumplir durante el juego. Esperaríamos al menos que cada jugador tuviera las aptitudes para el papel que le fue asignado. Aquí aparece el primer obstáculo, en no pocos equipos de desarrollo a los jugadores se les asigna o se les contrata para un papel para el cual su perfil no da el ancho. En muchas ocasiones las premuras inducen a aventurarse en proyectos con equipos cojos, es decir, faltándoles posiciones que son necesarias. No basta tener cubiertas todas las posiciones, es medular que cada elemento tenga la capacidad y habilidad para desarrollar su o sus funciones de manera adecuada. A veces he conocido equipos de trabajo que su falla es inducida por la debilidad de algunas de sus posiciones, desde el liderazgo hasta el utilero (valiéndome de la analogía futbolera) todos deben saber: *qué hacer, cómo hacerlo, cuándo hacerlo y porqué debe hacerse*. El problema está en muchas ocasiones en el “*cómo hacerlo*” y en el *porqué debe hacerse*. Los equipos entrenan, practican y tienen idea clara de a qué juegan, es decir su estrategia y su estructura (su parado dirían los comentaristas). Entrenamiento, práctica... las claves.

*lfernand@uacj.mx*