

Software Prediseñado y Leyendas Griegas

Cuenta la leyenda que en la Antigua Grecia había una pequeña posada administrada por un tal **Procusto**. Tan pequeña era que tenía una única habitación con una sola cama para los huéspedes. Siendo estos de talla diversa, **Procusto tenía por costumbre ajustarlos al tamaño del lecho**, estirando a los que resultaban más cortos y recortando las extremidades de los que sobran. El **Lecho de Procusto** simboliza, desde entonces, las situaciones en que las personas deben adecuarse – compulsivamente - a la circunstancia disponible.

La industria del software aplicativo se asemeja bastante a un Lecho de Procusto. En una nota anterior ensayamos una recorrida histórica a lo largo de los últimos treinta y pico de años, en que el software aplicativo pasó por ser **un desarrollo artesanal, un producto normalizado y más recientemente un estándar** susceptible de adecuaciones significativas en su comportamiento en correspondencia de lo requerido por la empresa usuaria.

Es difícil argumentar en contra de la normalización en el mundo de la industria y la tecnología, y seguramente yo seré de los últimos en embarcarme en tal dirección. Buena parte de nuestra cotidianeidad se basa en **principios de intercambiabilidad y conectividad** que son simple consecuencia de la normalización. Que el bombillo eléctrico se enroscará correctamente en el portalámparas nos resulta tan obvio que ni se nos pasa por la cabeza llevar con nosotros el velador para procurar el repuesto. Y en el mundo informático damos por tan descontado que **el accesorio conectado al puerto USB se auto-instalará y funcionará**, como el que al enviar un email el receptor lo podrá recibir, abrir y leer.

'**Lo natural invisibiliza**' decía Enrique Pichon Riviére, y la normalización es tan natural – 'obvia' dirían los adolescentes – que ya no la vemos. Pese a que es un producto bastante reciente ... en plena Revolución Industrial tornillos y tuercas se ajustaban manualmente a lima para que enroscaran el uno en la otra.

De tan natural que es, no pensamos en que **la normalización solo tiene sentido si aporta valor**: sea por **reducción de costos o de tiempos o por incremento de los niveles de seguridad**. Si el producto normalizado no es más barato, se recambia más rápidamente y/o es más confiable que el producto artesanal, deja de tener sentido que el usuario se adecue a lo disponible en lugar de pretender que el producto sea conforme a su necesidad.

Yendo ahora al tema del software aplicativo, nos encontramos con que efectivamente **los proveedores ofrecen y realizan significativas adecuaciones** de su producto Standard a las modalidades particulares de la organización usuaria a través del proceso que llaman de implementación y/o de customización, con las paradojas siguientes:

- a) El **costo de la implementación pasa a ser sumamente significativo**. En numerosos casos, particularmente en los paquetes de clase mundial, puede llegar a tri, cuadro o multiplicar el llamado costo de licencia, que es el del producto en sí mismo.
- b) Los **plazos de ejecución de tales implementaciones devienen sumamente extensos**, de entre 6 meses y hasta 2 años
- c) Las intervenciones de adecuación de las características funcionales del producto **inevitablemente lo fragilizan** ... el soft customizado tiene mayor propensión a fallos que el producto original.
- d) Frecuentemente ante un requerimiento concreto escuchamos un '**no es compatible con el diseño básico**'

Con lo que resulta que finalmente, a un costo que suele ser mayor que el de un desarrollo ad hoc, luego de un plazo también mayor que el de un desarrollo, **tengo el mismo paquete básico que mi competidor directo, lleno de parches que llevará un par de años depurar**. Con el agregado de que las habitualmente obligatorias actualizaciones semestrales o anuales frecuentemente introducen

funcionalidades solicitadas por otros usuarios, que hacen que lo que hasta ayer me funcionaba con precisión de orquesta sinfónica, de pronto deja de existir o empieza a arrojar errores a granel.

¿ Por qué no pensar, entonces, en desarrollos a medida, ahora que existen **herramientas de programación modernas** que drásticamente reducen los tiempos e incrementan los niveles de seguridad ? En mi opinión la línea divisoria debe pasar por el tipo de actividad a modelizar por el soft aplicativo:

- a) **Para actividades sometidas a fuerte normativa regulatoria externa**, tales como todo lo administrativo contable, el software prediseñado puede cumplir acabadamente su cometido, siempre que cuente con herramientas que ya señalamos en otras notas, en particular en cuanto al intercambio de datos por exportación e importación y la inclusión de herramientas de diseño de reportes.
- b) Para **prototipos, simulaciones, presentaciones por única vez**, utilizar los paquetes utilitarios comerciales tales como Excel, Power point, Access, Word
- c) Y para aplicaciones que hacen al **corazón de nuestro negocio**, los que permiten diferenciarnos de nuestro competidor y desarrollar nuestras ventajas, léase **producción, logística de abastecimientos, almacén, política de stocks, programación, diseño, identificación**, pensemos en encomendar el desarrollo de la aplicación acorde con nuestras especificaciones.

Normalmente con un desarrollo a medida bien construido lograremos:

- 1) Que **el soft se adecue a nuestras necesidades**, en vez de adaptar nuestra empresa a lo que da el soft.
- 2) **Partes del producto se pueden empezar a utilizar** mientras aún se está construyendo, y el plazo de desarrollo suele ser menor que el de una simple implementación.
- 3) **Las enfermedades infantiles se depuran más rápidamente**, y no debo cargar con aplicaciones que estaban pensados para otros.
- 4) Sorprendentemente **el costo suele ser menor** que el de licenciamiento más implementación
- 5) Last but not least, los desarrollos se contratan por servicios profesionales que en gran parte son **subsidiarios por programas tipo PACC, Fontar, Fonsoft** y otros.