

Veinte enfermedades comunes en el proceso de desarrollo de software

www.galeon.com

Por **Antonio Quirós**

Construir software no es precisamente una tarea sencilla. Y no digo esto por la complejidad técnica que subyace a la misma, sino por lo difícil que es de gestionar el proceso de crear una solución que satisfaga las expectativas del cliente. Muchas veces comparamos el proceso de desarrollar software al de construir un edificio, pero no hay nada más lejos de la realidad, ya quisiéramos en este mundo trabajar en un entorno tan claro como en el de la construcción. Cuando encargamos una vivienda, tenemos que realizar el proyecto con un arquitecto, los planos describen con bastante precisión cómo será el resultado final del proceso, la memoria de calidades nos dice si la pintura será de un color de otro, si las puertas serán de roble o de pino, la marca y modelo de los saneamientos. Comparemos este entorno con el de un proyecto de desarrollo de software. Con suerte tendremos una documentación razonable; lo normal es que dicha documentación sea ininteligible para el cliente, como mucho tendremos los planos (modelos de datos, de clases, etc.) cuyas claves de interpretación son muy razonables para los doctos, pero nada para los legos. Si se describen los procesos como casos de uso reales, o de cualquier otro modo, nos encontraremos con un contexto muy literario, algo así como si en lugar de darnos el plano de una vivienda nos dieran una descripción donde dijera algo así como "la escalera tendrá 17 peldaños de 24 centímetros de altura, al llegar al descansillo de la escalera nos encontraremos tres puertas a mano derecha y dos a la izquierda, el peso máximo que aguantará la estructura sobre la que descansa la escalera es de dos toneladas, el grano de la pintura no sobrepasará el milímetro de volumen en ningún caso..." Imagínese la cara de nuestro comprador de vivienda tras leer dos mil páginas así de divertidas. Pues eso suelen tener que hacer quienes encargan soluciones de software para que quienes desarrollamos para ellos no terminemos acusándoles de que el producto desarrollado solo tiene lo que pone en la documentación y que si no se la han leído es un problema suyo.

Pero es que no es fácil actuar de otro modo, cómo podemos hacer algo más inteligible el diseño de una aplicación. Un prototipo de la misma sería similar a la maqueta de una casa o al piso piloto de una urbanización, pero en el desarrollo de software construir el prototipo es algo que se aproxima mucho a construir el producto final; habremos tenido como mínimo que pintar todas las pantallas y listados aunque no tengan una funcionalidad clara y eso equivale a un porcentaje muy alto del proceso de desarrollo, por tanto sus costes son altos y los clientes no siempre están dispuestos a sufragarlos.

Por otro lado el cliente suele poner unas expectativas demasiado *novelascas* respecto a lo que espera obtener, más o menos aquello de que dándole a una tecla se acabarán todos sus problemas organizativos. Y claro luego viene la cruel realidad a ponerle al tanto de lo que realmente la aplicación hace. En general hay mucha insatisfacción en todo el proceso por parte de todo el mundo y ello hace que el estrés corra por la venas de los participantes en todo proyecto de desarrollo, estrés que se reparte por igual entre clientes y desarrolladores; ambos incursos en una carrera de inculpaciones mutuas que terminan convirtiendo en un campo de batalla lo que debería ser un proceso de trabajo gratificante para todo el mundo.

Quiere esto decir que el asunto no tenga remedio y que sea una locura meterse a desarrollador o encargar un proyecto de software. No, ni mucho menos. Pero para mejorar este asunto debemos conocer con rigor lo que podríamos denominar enfermedades del proceso de desarrollo de software para, una vez conocidas, saber como curarnos de ellas e intentar que nuestros proyectos de desarrollo ganen en facilidad y rigor y que nuestros clientes alcancen un grado de satisfacción razonable sobre los mismos.

Lógicamente, existen métodos que hacen las cosas más sencillas. Cuanto más organizado y metódico es el proceso de desarrollo más fácil resulta que se logren los objetivos con él planteado. En este artículo trataré primero de desentrañar algunas de las enfermedades

más comunes en el proceso de desarrollo para recetar sobre las mismas algunas soluciones que nos ayuden a superarlas, cosa que espero ir haciendo en sucesivos artículos sobre esta misma temática.

Enfermedad 1: Contratos sin método de gestión alguno

Los contratos están para regular un proceso donde dos o más partes van a relacionarse en un entramado de intereses comunes y cruzados. Vaya por delante que muchos proyectos de desarrollo de software carecen de contrato, alguien vende sin saber muy bien qué vende y alguien compra sin saber muy bien qué compra, además nadie pone en negro sobre blanco cómo va a llevarse a cabo el proceso de gestión del proyecto. La semilla del fracaso está sembrada, si un proyecto así triunfa habrá que felicitar e invitar a cava a las partes.

Pero bien, supongamos que hay contrato. Esto es lo más normal si la solución a desarrollar es grande y las empresas contratante y contratada son serias. La enfermedad no está curada aún por eso. El contrato puede indicar plazos, importes económicos, como mucho describir someramente las funcionalidades del producto a desarrollar y del entorno técnico, puede quizá hablar del equipo de desarrollo y de sus competencias, quizá incluso hable de penalizaciones ante incumplimiento por ambas partes. Pero aún no está completo. El contrato debe describir cómo se realizará el proceso de trabajo, debe existir una metodología de hacer la cosas que esté pactada entre contratante y contratado, de forma que se determinen hitos de revisión y aceptación por las partes para que el proyecto no crezca sin que la semilla del acuerdo esté perfectamente sembrada.

Enfermedad 2: Falta de previsión en los riesgos

Todo proyecto informático implica un importante conjunto de riesgos. Por ejemplo, imaginemos una solución que será usada por personal acostumbrado a antiguos terminales en modo carácter y que ahora deberá enfrentarse a soluciones basadas en navegador y guiadas por el ratón. En ningún caso nadie ha valorado el peligro de que los usuarios rechacen el uso de la aplicación o simplemente mermen su productividad. Si no se tiene en cuenta dicha situación, la aplicación fracasará. Probablemente la empresa desarrolladora haya indicado simplemente al cliente cual sería su entorno de trabajo, probablemente el cliente lo haya pedido así, sin embargo la aplicación fracasa. Alguien debió de tener en cuenta dicho riesgo y ponerlo delante del cliente para que tomara la decisión pertinente. Si no se analizan los riesgos y no existe una transparencia sobre los mismos entre el equipo que desarrolla y el cliente, por más que exista un contrato claro respecto al método de trabajo, la posibilidad de fracaso está servida.

Enfermedad 3: Sin documentación todo son problemas

La base de todo proyecto es la documentación. Existen todo un conjunto de técnicas que nos ayudan a realizar el diseño de las soluciones de software: toma de requerimientos, modelo entidad/relación, diseño de clases, casos de uso reales, etc., etc. La documentación de un proyecto presenta las siguientes características de gran importancia:

- Permite recoger en un fondo documental toda la historia del proyecto de modo que cualquier posible comprobación se facilita.
- Facilita la relación entre los distintos agentes que participan en el entorno de desarrollo: usuarios, analistas, desarrolladores, directores de los usuarios, personal de explotación, etc.
- Ayuda en la resolución de conflictos entre las partes en tanto que representa una fuente fiable donde deben estar recogidos con claridad los requerimientos pedidos por el usuario, las soluciones propuestas por el equipo técnico y todo el material de seguimiento del proyecto.

Aún no hemos descubierto un sistema de notación lo suficientemente certero como para que sea aceptado por todos los agentes de un desarrollo informático (el plano del arquitecto del que antes hablábamos), pero esto no justifica que no se use ninguno. Hay que documentar con claridad usando las técnicas convencionales que la metodología seguida nos prescriba y mantener la documentación con el grado de actualización suficiente.

Enfermedad 4: Un cliente abaratador de costes

Las cosas valen lo que valen y el abaratamiento a ultranza de los costes suele llevar a una situación de pérdida de calidad en el software desarrollado. Muchos clientes toman sus decisiones de forma estrictamente basada en los costes sin tener en cuenta otros criterios. Esto suele ser un error, ya que muchas empresas de desarrollo bajan sus costes a consecuencia de:

- Mantener a un personal poco formado y con las competencias justitas para salir adelante.
- Reducir al mínimo la labor de diseño y dirección de proyecto, de forma que la programación tenga la máxima prioridad, lo que indefectiblemente conduce a la anarquía de la planificación prevista.
- Eliminar la posibilidad de integrar el cambio de requerimientos como parte totalmente unida al proyecto. Los proyectos de software están ligados a una realidad dinámica que cambia mientras el proyecto se está desarrollando y hay que saber integrar dicho dinamismo como parte del propio proyecto.

En general, hay que decir que las empresas que han hecho un esfuerzo desproporcionado por abaratar sus costes suelen encontrarse con un buen conjunto de problemas, ya que las posibilidades de que la aplicación desarrollada no tenga los niveles de calidad adecuados o simplemente no sirva para el fin previsto, son muy altas.

Enfermedad 5: Un proveedor ávido de negocio

En el mundo de las tecnologías de la información, máxime en los tres últimos años de crisis galopante, la gente pasa mucha necesidad. Las empresas intentan conseguir los proyectos sea como sea, sin pararse o no a analizar la rentabilidad de los mismos. Los recursos se retiran con la misma facilidad que se asignan, ya que son necesarios para otros proyectos. La rentabilidad lo preside todo y si este enfermo proveedor ávido de negocio se junta con un cliente enfermo abaratador de costes, la suerte está echada. El fracaso está garantizado.

No quiero decir con esto que tengamos que convertirnos en hermanitas de la caridad. Todos buscamos lo más rentable para nuestra empresa, pero muchas veces lo más *evidentemente rentable* no es lo más rentable. Hay que saber mirar por debajo de la rentabilidad inmediata para ver donde está el truco. Yo, por ejemplo, hace algo de tiempo tuve que retirar un buen recurso técnico de un proyecto algo enquistado y que estaba fuera de rentabilidad, la consecuencia fue que dicho proyecto no funcionó bien de cara al cliente y, debido a su grado de insatisfacción, no nos adjudicó otro proyecto diez veces más grande que aquel que contribuí a fastidiar. La competencia me agradecerá siempre mi búsqueda a ultranza de rentabilidad para mi compañía.

Enfermedad 6: Clientes con unas fuertes expectativas y sin conocimiento acerca de cómo gestionarlas

El asunto de las expectativas que quienes encargan una solución informática mantienen sobre la misma no es un tema baladí. Para demostrar esto solo tenemos que revisar una misteriosa cláusula que suele aparecer en casi todos los acuerdos de licencia, algo así como que la empresa desarrolladora declara que el software hace lo que hace y no lo que el cliente espera que haga.

Es como todo lo del asunto de la ilusión. Cuando nos enamoramos el foco de nuestros amores nos parece poseer un cúmulo de todo tipo de valores éticos y estéticos; casi no comprendemos como no le han hecho Miss o Mister Mundo y/o no le han dado el premio Nobel de la paz. Pero tras unos cuantos meses compartiendo cama, comida, despertares mañaneros, olores, comportamientos extravagantes, etc. el brillo se va. La expectación que levantó se termina y, si realmente no hay un fondo de coincidencia por debajo, la relación se diluye. Pues lo mismo, lo mismito pasa con esto del software. Cuando un cliente busca una solución, normalmente pone unas expectativas desmesuradas en la misma, piensa que le va a resolver más de lo que le resuelve, luego viene el batacazo, darse cuenta que las cosas no llegan a hacerse exactamente cómo el preveía, que con solo darle a la tecla no se resuelve todo, etc.

Por ello hay que protegerse de estas expectativas desmesuradas. Para ello nada mejor que inyectar racionalidad en el sistema. Hay que hablar mucho, la comunicación es la base de

toda buena relación. Hay que hablar, aunque sea para desengañar a nuestro cliente y ponerle las cosas con claridad sobre la mesa. Solo un entorno claro dejará el juego de expectativas en su sitio.

Enfermedad 7: La suficiencia técnica de los desarrolladores

Qué épocas aquellas en que podíamos decirle a un usuario “Oiga, que usted no sabe con quien está hablando, que soy Ingeniero Informático titulado por la Politécnica”. Con ello hacíamos y deshacíamos. Ordenábamos el mundo de las aplicaciones según nuestro criterio indicando qué se podía y qué no se podía hacer. Un cáncer más de los proyectos, la suficiencia técnica de los desarrolladores. Si esto antes podía colar, hoy no cuela de ninguna manera. La cultura informática ha impregnado notoriamente toda nuestra sociedad y hoy la suficiencia técnica lo que más suele encubrir es falta de capacidad para abordar un problema.

Los usuarios de las aplicaciones que desarrollamos no son retrasados mentales a los que hay que engañar con espejitos y abalorios. Solo la franqueza por ambas partes ayudará en un entorno de relación fluida que coopere en la buena gestión del proyecto en lugar de en su destrucción.

Enfermedad 8: Un cliente que lo espera todo de la empresa desarrolladora

Se puede morir de éxito. Cuanto más notoria es la celebridad de la empresa que desarrolla un proyecto, más altas son las expectativas del cliente que lo ha encargado. En muchas ocasiones esto se eleva por encima de lo razonable. Microsoft es muy bueno haciendo software, pero quizá es también la empresa más criticada por sus productos. Todo el mundo espera que hagan más de lo que hacen y que lo hagan mejor, que fallen menos. Si muchos de estos productos hubieran sido realizados por empresas de menos éxito seguro que serían menos criticados, quizá hasta incluso fueran alabados. Ya sabemos que el sentido común es el menos común de los sentidos.

Enfermedad 9: Un cliente dispuesto a poner poco de su parte

Muchos clientes piensan que desarrollar una aplicación consiste sólo en contratar a una empresa para que realice el desarrollo. Nada más lejos de la realidad. Cualquier proyecto de desarrollo de software implica también mucho trabajo para el cliente del mismo. Tiene que organizar sus procesos, transmitir la información a la empresa contratada, participar en todo el proceso de control, facilitar la implantación y seguir políticas activas que faciliten la buena incorporación del nuevo software en la administración. Muchas aplicaciones han fracasado porque las empresas que las contrataban no estaban dispuestas a poner nada de su parte. Solo el trabajo codo con codo de cliente y equipo de desarrollo garantiza el éxito de un proyecto informático.

Enfermedad 10: No controlar la gestión del cambio

Un proyecto de desarrollo no es algo estático. Las tecnologías cambian vertiginosamente, los procesos de negocio igual. Si algo define a nuestro mundo es el cambio vertiginoso y la prisa por integrar dicho proceso. Mientras esto sucede, nuestros proyectos de desarrollo siguen esforzándose en aislar el cambio como un proceso a eliminar. Hacemos completos diseños que, en proyectos largos, tardan años en desarrollarse y que cuando se terminan ya no sirven para nada. Cuando el cliente nos pide que cambiemos algo temblamos e intentamos convencerle de que no debe, de que aquello será muy caro, de que se tambalearán las estructuras de la aplicación, etc. Esto no es bueno para la solución final.

El cambio es un protagonista clave de nuestro mundo y, por tanto, tiene que estar integrado en nuestros proyectos; por ejemplo, uno de los principios fundamentales del modelo de equipo de la metodología MSF de Microsoft es el de esperar los cambios y permanecer ágiles para responder a los mismos.

El cambio tiene que ser parte integrante de todo proyecto de desarrollo, hay que saber integrarlo dentro del mismo y presentar a nuestros clientes una clara alternativa de cómo lo gestionaremos en lo técnico y en lo económico.

Si el cambio es un problema para el equipo de desarrollo de un proyecto, probablemente

dicho equipo de desarrollo sea un problema para el cambio.

Enfermedad 11: Pereza innovadora

La pereza es una mala compañera en cualquier actividad de la vida. Solo quienes permanecen siempre atentos, despiertos y ágiles frente a la realidad son capaces de adecuarse a un mundo que cambia permanentemente delante de nosotros. La pereza innovadora es la causa eficiente que mueve la mala gestión del cambio reseñada en la enfermedad anterior. Quién no conoce al típico programador de Cobol que se ha negado a evolucionar y que no ha aprendido nada nuevo en treinta años. La enfermedad del analista-programador presenta síntomas parecidos. Una vez alcanzado la cúpula de lo que sabes hacer te olvidas de dar saltos cualitativos. ¡Ahí me las den todas! Sé programar en Visual Basic para qué quiero más. Ya, ya..., pero luego viene la programación web, luego los dispositivos móviles, que si ahora se trata de usar el paradigma de la orientación a servicios, luego que si habrá que pasar a analista, que si se es de FP y el cara de mi jefe me ha dicho que para subirme el sueldo que estudie la Ingeniería Técnica. En fin que si se es vago para aprender cosas nuevas y se tiene cierta reticencia al cambio, mejor que nos dediquemos a la agricultura, aunque ni esto ya permanece estable con tanto invernadero y tanta agricultura intensiva y demás. Quizá meterse a religioso y enclaustrarse en un convento, no sé, no sé. ¡Qué stress!

Enfermedad 12: Lo hago como sé y no como quiere el cliente

La pereza innovadora es la causa de esta otra enfermedad común. Tras muchos años desarrollando en un determinado entorno, con unas herramientas concretas, bajo una misma metodología, algunos equipos de desarrollo se muestran poco proclives al cambio. Por ello cuando un cliente demanda una solución que debería ser abordada con herramientas y métodos más modernos, la empresa desarrolladora trata de llevar el asunto a su terreno y proponer soluciones más conservadoras. Esto suele terminar en problema. Si ya cuando abordamos un proyecto corremos el riesgo de que, al terminarlo, la arquitectura elegida haya perdido vigencia, esto se maximiza si arrancamos con entornos anticuados. Al final terminamos una preciosa aplicación obsoleta con la que el cliente nunca termina satisfecho, sobre todo si la contrasta con otras de sus conocidos o, incluso, competencia, mucho más actualizadas.

En fin, que en desarrollo de software siempre hay que estar a la última. No hay que resistirse a la innovación y, desde luego, en ocasiones hay que arriesgarse con versiones recientes, eso sí, siempre que estén lo suficientemente contrastadas en el mercado. Es un mal paso intentar convencer a nuestros clientes de que nuestra tecnología más antigua y fiable es mejor que la más moderna, raro será que logremos su satisfacción con el producto que le desarrollemos.

Enfermedad 13: Equipos de desarrollo que olvidan que la solución final es lo que importa

Muchos técnicos se enfrascan en farragosas discusiones tecnológicas. En ocasiones intentan convencer a sus clientes de que lo que necesitan no es lo pedido por ellos sino otra cosa muy diferente. Además, se suele menospreciar el punto de vista del cliente en el ámbito técnico.

Esta es una grave enfermedad que termina inexorablemente en la muerte del paciente, es decir, el proyecto que se lleva a cabo con equipos proclives a padecerla. El cliente es quien sabe de su negocio, es quien sabe lo que necesita y lo que puede pagar por ello. Los equipos de desarrollo deben calibrar si entrar en un proyecto le es interesante tanto en lo económico como en lo tecnológico, pero una vez aceptado el proyecto hay que luchar a muerte por el mismo, considerando siempre que el punto de vista del negocio es el que regula todo el proceso y que la entrega de la solución final requerida por el cliente es básica para coronar con éxito el proyecto. Muchos desarrollos quedan incompletos, se improvisa en el camino, se toman sendas diferentes a las trazadas originariamente, la solución requerida se pierde de vista en una amalgama extraña de elementos donde la tecnología ha terminado preponderando sobre el punto de vista del negocio. Mal asunto, la tecnología no debe representar un fin en sí misma, sino un medio para ayudar en los procesos de negocio. Quien olvida esto está condenado a fracasar en sus proyectos.

Enfermedad 14: Desarrollo sin diseño

Esta es una grave enfermedad que proporciona un alto grado de minusvalía a las soluciones desarrolladas por los equipos que la padecen. Es como si tuviéramos una persona sin manos o sin cabeza. El desarrollo de una solución es un proceso complejo donde la labor de codificación no siempre representa la parte más importante.

En general, en todo proceso tenemos que hablar de las siguientes fases:

- Contratación y aspectos económicos y legales
- Determinación exacta de requerimientos
- Análisis y Diseño
- Codificación
- Pruebas
- Documentación de usuario
- Implantación

Es raro que en un proyecto serio la fase de codificación suponga más del 40% de la carga total de trabajo, por tanto, puede entenderse que si hacemos bascular dicha carga hasta el 90% como en algunos casos sucede, algo estamos haciendo mal. ¿Quién no se ha enfrentado a esos proyectos en los que tras muchos años de trabajo el cliente nos llama un día y nos pide la documentación del proyecto? Horror, todo el mundo enmudece. Qué decir si nos pide que le indiquemos la metodología o los controles de calidad que empleamos. Ahí morimos si lo único que hacemos es codificar y codificar sin rumbo ni objetivo fijos. Toda construcción necesita una correcta planificación de la misma y, desde luego, el software no es ajeno a este tipo de requerimientos.

Enfermedad 15: Un cliente insatisfecho que pide cambios continuamente

Ya hemos determinado que el cambio es consustancial a los proyectos en el ámbito tecnológico. Esto está claro y debe existir la necesaria predisposición en los equipos; ya hemos hablado de ello. Pero el extremo opuesto es también una grave enfermedad cuyos síntomas debemos remarcar. Lo normal es que los contratos para desarrollar una solución de software lleven integradas las herramientas legales necesarias para solventar las necesidades de cambio, es decir que conforme los cambios de requerimientos supongan carga adicional de trabajo existan las vías adecuadas para que esto pueda tratarse desde una vertiente económica razonable para contratante y contratado. Tanto si esto se ha llevado a cabo correctamente en la fase de contratación como si no se ha hecho, hay muchos clientes que nunca están satisfechos con la solución final y siempre están aportando nuevos cambios a la misma, cambios que no siempre tienen la razonable justificación de negocio sino más bien la de proyectar la vanidad de quien los concibe. Si el cliente paga por ellos y en los contratos está tipificado cómo llevarlos a cabo, la empresa desarrolladora los incorporará sin problema, pero esto terminará causando daño estratégico al proyecto que, quizá no se termine en tiempo debido a dichos cambios y que quizá no obtenga, por tanto, la satisfacción de sus usuarios. El problema se enarca si la gestión del cambio no está tratada contractualmente. Cambiar un control de sitio o el color de una pantalla es como tirar un muro en un edificación y construirlo de nuevo con unos ladrillos diferentes; esto tiene costes en tiempo y esfuerzo que no siempre conviene acometer y el cliente debe también ser sensible a esta situación si, además de un proyecto perfecto, se quiere un proyecto viable. Como siempre, la buena comunicación entre el cliente y el equipo de desarrollo es la base para curar esta enfermedad.

Enfermedad 16: Barrer bajo la alfombra

Cuando hay problemas lo mejor es hablarlos. Barrer bajo la alfombra solo contribuye a ocultar la suciedad, pero no la elimina. Una buena parte del personal técnico que participa en los proyectos tiende a ejercitar el silencio sobre muchos problemas de los que van surgiendo en el día a día, pero los problemas no se solucionan por no hablar de ellos; todo lo contrario, lo normal es que se enarquén. Por ello, ante la más mínima situación problemática que nos encontremos lo razonable es intentar solucionarla con las personas implicadas y, desde luego, reportarla a quien corresponda dentro de nuestra organización. Si no lo hacemos así, la suciedad oculta terminará salpicándonos y algo

que podría haber tenido una escasa importancia se convertirá en un grueso problema de muy difícil solución.

Enfermedad 17: Equipos olvidadizos

Muchos equipos olvidan conscientemente algunos requerimientos de las aplicaciones facilitados por el cliente, normalmente, aquellos de más difícil consecución técnica. Raro es que esto salga bien si se oculta al cliente. Una cosa es hablar con él y convencerle de la falta de viabilidad real de una solución y otra ignorar sus peticiones y engañarle no haciéndolas. Más tarde o más temprano descubrirá el engaño y nos terminará pasando factura. Aunque el proyecto esté terminado y pagado quizá no vuelva a adjudicar ningún desarrollo más a nuestra empresa si los profesionales que le proponemos han tenido este modo de actuar.

Enfermedad 18: Esto no es problema mío

La intervención de los usuarios de una aplicación, de los usuarios expertos en la misma y de la dirección empresarial es básica para que un proceso triunfe. Ante este hecho podemos encontrarnos con varios supuestos. Por ejemplo, usuarios a los que se ha ignorado por la dirección a la hora de transmitir al equipo desarrollador las funcionalidades de la aplicación y que, por tanto, cuando la aplicación se instala son enemigos acérrimos de la misma. Otro aspecto de este mismo problema es la dejación de los usuarios a la hora de asumir sus responsabilidades en la transmisión de la información al equipo desarrollador; o incluso la típica respuesta de *esto no es un problema mío*, cuando planteamos alguna duda. En todo proyecto los usuarios deben estar altamente implicados y su colaboración debe estar garantizada, si esto no es así hemos de ser conscientes de que existe un factor de riesgo notorio en el desarrollo que estamos abordando y que, por tanto, las posibilidades de fracaso serán altas.

Enfermedad 19: Mentir con los recursos

Las compañías que se dedican al desarrollo no lo tienen fácil con la organización de sus recursos. Es difícil mantener la armonía entre el proceso comercial de conseguir recursos y el de producción donde los recursos se van asignando. O se está cargado de proyectos y la correcta planificación de las cosas sale dañada o se está muy escaso y entonces los recursos están ociosos. En todo este proceso es muy fácil caer en la tentación de mentir con los recursos al cliente con el fin de conseguir el proyecto y luego no dedicarlos al asunto según las pautas pactadas. Esto es un juego peligroso, que atenta contra un principio básico de todo proyecto: la claridad en la relación cliente-proveedor. Si no se tiene recursos es preferible intentar convencer al cliente de que espere un tiempo, pero si finalmente tiene que esperar ese mismo tiempo porque hemos ganado el proyecto con falsedad, habremos sembrado la semilla de la discordia y las posibilidades de fracaso en el proyecto serán altas.

Enfermedad 20: Este proyecto es una guerra

Y por fin, la metástasis generalizada. Como consecuencia de una o de varias de las enfermedades reseñadas, llegamos a proyectos donde la guerra declarada entre el cliente y el equipo del proyecto hace que la situación llegue a un punto de difícil retorno. Nadie quiere dar su brazo a torcer y, posiblemente, por fallos cruzados de todo el mundo el desarrollo quede paralizado. La empresa desarrolladora deja de dedicar recursos porque el tema ya está fuera de rentabilidad y piensa que los incumplimientos del cliente son la causa del asunto. Por otro lado el cliente deja de pagar las facturas porque piensa que la empresa desarrolladora no está cumpliendo con sus compromisos. Cuántos proyectos no se han visto abocados a esta situación, cuánto dinero no se ha perdido por una y otra parte con situaciones de este tipo.

En fin, tras el análisis de todas estas enfermedades (no nos engañemos, hay muchas más, podríamos hacer casi la CIE-9 (Clasificación Internacional de Enfermedades) de la OMS respecto a las patologías sufridas por los desarrollos, pero éstas deben bastarnos para abrirnos los ojos respecto a una situación problemática que requiere, desde luego, mucho método de trabajo y mucha comunicación entre cliente y empresa desarrolladora. Sólo si existen ambas bases se hace posible llevar a cabo con ciertas garantías de éxito los proyectos informáticos. Como siempre, la información, el

conocimiento de lo mal hecho y su contraste con lo correcto es la base para mejorar en procesos futuros. Esperemos que este somero catálogo tenga al menos la finalidad de advertir a quienes quieran escuchar y que sea la base de poner soluciones antes de que una enfermedad terminal acabe con nuestros proyectos.