Manifiesto de Software a Medida

Lo que NO es el software a medida, y excepciones

Uno [de mis pensamientos] fue el ocurrírseme considerar que muchas veces sucede que no hay tanta perfección en las obras compuestas de varios trozos y hechas por las manos de muchos maestros, como en aquellas en que uno solo ha trabajado. Así vemos que los edificios, que un solo arquitecto ha comenzado y rematado, suelen ser más hermosos y mejor ordenados que aquellos otros, que varios han tratado de componer y arreglar, utilizando antiguos muros, construidos para otros fines.

René Descartes, El Discurso del Método, II

Desarrollar software parece sencillo cuando empiezas a programar. En efecto, cualquier ciencia en sus conceptos más fundamentales, está construida con sencillos ladrillos. Pero: ¿cuántos ladrillos? Allí está la trampa: cuando ves que se trataba de decenas de miles de pequeños ladrillos, puedes ya estar demasiado embarrado para dar marcha atrás. Es entonces cuando nuestra casa de desarrollo hace una propuesta para ampliar el concepto de <<software a medida>>.

Nuestra experiencia en el mundo del software nos ha revelado una tesis poco intuitiva: el tener una herramienta nos arroja a veces ilusiones de control. Pero parece que, en el largo plazo, es más poderoso quien realiza acuerdos estables y duraderos con un proveedor, que quien se casa con una herramienta. Esta tesis será la premisa del siguiente texto.

El histórico de control de cambios de versiones está al final

Aprendamos conceptos de otras industrias "a medida": zapatos y grastronomía.

El software hecho a medida es programación que trata de adaptar herramientas pre-construidas para que se ajusten a la realidad de una empresa, ó en general a una situación específica. Toda adaptación implicará **siempre** un conjunto de algoritmos sobre los cuales no se tiene control, sobre los cuales "se monta" el software. Desde aspectos de tan bajo nivel, como determinar para cuál procesador es compatible una ejecución, hasta aspectos de mucho más alto nivel, como la versión del servidor con la cual son compatibles, en cualquier caso un software está atado a una situación para la cual se ha adaptado. El lenguaje con el cual se construye, incluso, puede no tener disponible el compilador ó intérprete con el cual se ejecuta. Así pues, elaborar un software involucrará siempre un conjunto de algoritmos a cuyo código no necesariamente se tiene acceso, ó si se tiene, no necesariamente es un código de larga duración en el tiempo, siempre tendrá una vida útil.

Igual que cuando enviamos a construir un zapato a la zapatería, el producto tiene una vida útil y no le pedimos al zapatero que nos revele sus métodos sobre cómo elaboró el pegante ó cómo cortó el cuero para construir el zapato. El producto tiene un propósito, y es: calzar a las personas. No es el propósito del zapatero vender un zapato publicando en cada venta todos sus "know-hows" para que cualquier persona pueda, a precio de zapatos hechos a medida, acceder a los secretos que el zapatero ha acumulado por años para construir su negocio y empresa.
De la misma manera, si un negocio de eventos hace un pacto con una empresa de chefs de cocina para construir platos "a medida" todos los días para los diversos eventos que atiende la empresa de eventos, seguramente estará pagando por la construcción de dichos platos con el fin de que los comensales accedan a la mejor comida con las mejores prácticas de salubridad y nutrición que haya acumulado el grupo de chefs a lo largo de los años. No debería ser del interés de la empresa de eventos "acceder a las recetas de los platos a medida para ser independiente de los chefs": en tal caso, por el precio de atender un evento, los chefs estarían entregando sus recetas y secretos construidos y fortalecidos por años. Si realizaran lo mismo con cada empresa de eventos, de hecho las propias empresas de eventos perderían: pues se reduciría la ventaja competitiva y factor diferencial de cada empresa de eventos en la medida en que esos secretos profesionales se vuelvan públicos. Los futuros chefs de cocina accederán fácilmente a esas recetas, con lo cual se reducirá el precio original de esos platos.

En la medida en que los secretos gastronómicos sean públicos, y se libere por tanto la barrera del costo del desarrollo de recetas para futuros chefs, se incrementará la competencia culinaria, y con el incremento de oferta surgirá la competencia por precios -una fuerte tendencia de diversos mercados en la medida en que se llenan de oferta, es la ley de la oferta y la demanda-. Esto generará que la utilidad para los chefs se vea minimizada para el mismo esfuerzo.

La utilidad es la que paga la investigación para generar nuevos productos, con lo cual los chefs individualmente gozarán de menos presupuesto para lograr novedades culinarias ya sea en sabor, en salubridad, en calidad nutricional ó cualquier otro aspecto que mejore el bienestar de las personas.

Así pues, debe distinguirse que algunos productos buscan desde su nacimiento la democratización, mientras que otros productos buscan dar sostenibilidad a la innovación. La innovación puede sostenerse ya sea vendiendo democratización de productos en grandes escalas, ó vendiendo exclusividad en escalas más pequeñas. En ninguno de ambos casos, ni al comprar productos al fabricante de un producto masivo, ni al comprar un servicio de un evento exclusivo con platos especiales, se debe equiparar el acceso al servicio de probar un plato con el acceso a la receta y know-how de cómo se construyó. El comensal sí tiene derecho a preguntar sobre ingredientes ya sea por su propia salud ó incluso para proteger que especies en peligro de extinción no hagan parte de su plato. Pero los detalles del know-how sobre cómo se elaboró la preparación, son independientes del servicio de poder alimentarnos con un plato que fue preparado a nuestro gusto.

La seguridad para la empresa de eventos al pactar con la empresa de chefs, no estará en que la empresa de eventos tenga acceso a los secretos de los chefs, que por demás incluso teniéndolos en texto puede sufrir para conseguir personas con la disposición y la habilidad para replicar las recetas con la misma calidad y características que impartían los chefs creadores de las mismas.

Por el contrario, el énfasis debe hacerse en plantear pactos justos, francos, de larga duración, ya que las relaciones estables convienen a ambas empresas. La independencia de la empresa de eventos radica en que puede cambiar de empresa de chefs; pero sin pretender que, por el costo de contratarles realizaciones de platos, tenga acceso también a llevarse sus secretos industriales.

Las realizaciones de los platos pagan el servicio de que el chef disponga de su tiempo para aplicar sus mejores prácticas profesionales, con la experiencia acumulada por años, con el fin de que el comensal consuma un plato. Por eso se paga un precio de un servicio “a medida”, ya que efectivamente hubo una dedicación exclusiva para lograr un servicio para una persona que pagó por ese derecho. Pero si el comensal desea, ADEMÁS, tener acceso a las recetas del chef, por ello debe estar dispuesto a pagar una suma notablemente superior.

----------*O*----------

Otro caso: bienes inmuebles e inquilino que invierte en una infraestructura.

Veamos otro caso similar. Considérese la situación de un inquilino en arriendo de una casa. Consideremos una situación exigente, de un inquilino siendo un negocio comercial. Un inquilino que tenga un comercio puede invertir en, por ejemplo, modificar una fachada ó cambiar un muro no estructural de una casa, y el propietario de la casa encargarse, bajo su responsabilidad, de dichas modificaciones a gusto del cliente. Esto tiene varias implicaciones y obligaciones para el propietario de la casa, como que debe existir entre ambos algún compromiso para que el cliente garantice que se respetará su permanencia durante un tiempo mínimo en la casa, de tal suerte que su inversión se vea justificada. Así, el cliente puede pedir que, por ejemplo, se permita que su contrato tenga una duración de mínimo 5 años, ó 10 años, por ejemplo. De esa manera, con ese compromiso del propietario, el cliente está garantizando su inversión.

Este pacto permite al cliente el invertir un costo mucho más bajo para adaptar un inmueble a su negocio, permitiendo al cliente el contar con una infraestructura a su medida sin necesidad de comprar una nueva, y alquilándola por un tiempo definido. El cliente está invirtiendo en una adaptación temporal, y lo hace para su comodidad, a cambio de un pacto de estabilidad ó de un pacto más duradero, sin que por supuesto ello implique propiedad sobre la casa. El cliente gana en cuando a que puede poner a andar un negocio con un precio mensual manejable, poniendo a prueba su modelo comercial, sin necesidad de sufrir el costo de comprar una propiedad completa.

Posterior a los anteriores conceptos...

En el mundo del software existen circunstancias adicionales a las expuestas anteriormente. Existen muchas modalidades, escuelas de software a medida, prácticas del mercado. Hemos observado que muchas prácticas comunes no son, a pesar de ser populares, las más convenientes para su propia seguridad, para evitar que su propia empresa pierda dinero ó tenga estabilidad en el largo plazo. Con frecuencia observamos que se cae en la ilusión de dar demasiada importancia a condiciones que no hacen la diferencia para el ejercicio real de la empresa, en detrimento de aspectos que sí podrían generar diferencias importantes. La ausencia de marcos claros genera gran cantidad de dudas sobre cómo proceder en situaciones atípicas, ó también falsas expectativas a lo largo del tiempo, y todo ello distrae del objetivo más fundamental, que es usar la tecnología para brindar estabilidad, bienestar y capacidad de reacción a las empresas.

Nosotros mismos hemos pasado por diferentes metodologías, y hemos podido depurar las prácticas que

consideramos las mejores no sólo para el éxito de un negocio individual, sino también para la economía en general.

Las siguientes pautas buscan facilitar su comprensión acerca de por qué hemos tomado ciertas decisiones, para que comprenda usted que también sirven para su beneficio, y que por otra parte hay ciertas preocupaciones que son típicas pero que realmente no hacen la diferencia para su negocio, permitiendo que sea nuestra compañía la que se encargue de todo lo relacionado con la programación y los cuidados que ésta amerita.

¿Soy absolutamente libre de llevarme mis datos cuando quiera, para donde quiera, de tener copia permanente de los mismos?

Sí. Los datos suyos son de su propiedad privada, y si en algún momento usted desea exportarlos para usarlos en otra parte, tiene el derecho a hacerlo. Esto no incluye la aplicación: el software convierte esos datos en información útil. La aplicación interpreta los datos, y ello se logra adaptando software pre-desarrollado a su situación específica.

¿Yo como cliente recibo el código de su trabajo?

Hay **tres** respuestas a esto:

La **primera** respuesta es: **NO** en el modelo general común que ofrecemos y recomendamos, sería un alto costo adicional innecesario para su compañía y de todas formas no será útil para la gran mayoría de empresas que no se dedican exclusivamente al desarrollo de software. Veamos una metáfora: obviamente es un costo diferente el pagarle a un experimentado chef de cocina para que me prepare un plato, a pagarle un alto costo adicional para que me venda una experimentada y probada receta, donde dicha receta no sólo tiene en cuenta el proceso para lograr un buen sabor final, sino que el chef ha evolucionado detalles como saber mezclar productos para reducir el riesgo de personas alérgicas, tener en cuenta detalles médicos de los comensales para mezclar productos, considerar las mejores prácticas para comensales de colon irritable, identificar ingredientes ó productos con proveedores amigables con el medio ambiente y el buen trato animal, o sustituir unos productos por otros que no cumplan los mejores criterios. Es decir, si bien parece simple y fácil, actividades que desde afuera parecerían sencillos negocios, para un profesional están compuestos de centenares de sutilezas que constituyen un producto completo y que, por esta suma de detalles, aumente confiabilidad por reducir inconvenientes y

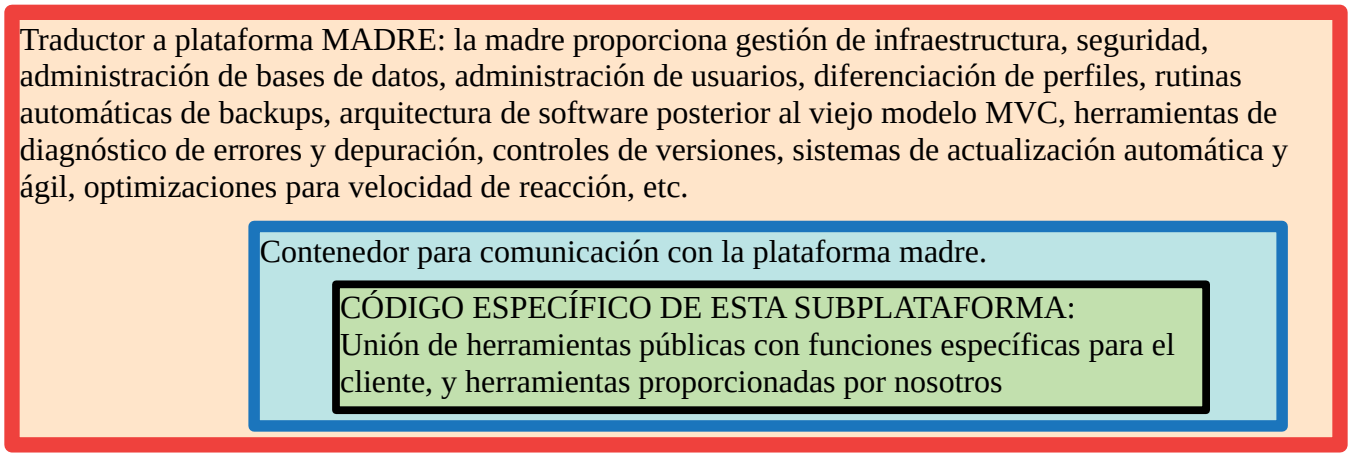
riesgos para usted como cliente, siendo además un producto más probablemente escalable y duradero que si fuera un producto improvisado con poca observación de las minucias.

Puntualicemos algunas razones:

1. Hablaré como el cliente. Si yo soy un cliente que solicita a un cocinero que elabore una comida con platos escogidos por mí, para unos invitados, el propósito es gozar de las mejores técnicas del equipo de cocineros para disfrutar de una cena, no es mi propósito hacerme a las recetas que los cocineros han evolucionado durante años profesionales. Lo segundo es mucho más costoso que lo primero, y no es relevante para el negocio esencial de su compañía, así que generalmente no será útil comprometer sus finanzas con una serie de altísimos gastos posteriores que no serán prácticos ni útiles para su compañía, que de todas formas tiene otro objeto social.
2. Seguridad informática: el software libre de código abierto es una muestra de los peligros de liberar el código fuente, en materia de vulnerabilidades informáticas por el hecho de que se conoce dicho código.

   Imagine que usted está protegiendo un banco. No sería prudente que los delincuentes contaran con un mapa de ese banco que les permita conocer potenciales puntos de entrada no oficiales, corredores, sitios de bienes sensibles, zonas vigiladas, zonas grises con puntos ciegos no vigilados, y demás. Para la protección de su propia seguridad como usuario, ese código fuente (el mapa, en la metáfora) no debería ser público.

La segunda respuesta es: si usted decide pagar el costo adicional de tener el código fuente, y para no alterar el precio original de manera importante, podríamos juntos anticipar una solución opcional que busca ser compatible con otras plataformas. No viene con la plataforma madre nuestra que contiene a su código, sino que en esta modalidad usted pagaría por un código estrictamente desarrollado para usted. Esta modalidad permitiría que el cliente sólo gaste ese dinero cuando realmente lo necesite, ya que el costo puede ser de un orden de magnitud comparable al valor del proyecto original, si bien un poco menor. El valor de la plataforma madre multiplicaría sin necesidad los costos para el cliente, así que solamente aplicaría para el código nuevo específico que antes nosotros no tuviéramos desarrollado, y que hubiera sido desarrollado explícitamente para su proyecto. No tiene mucho sentido someter al cliente a pagar unos cuantiosos honorarios por una situación hipotética, de manera que para aliviar sus temores esta segunda opción ofrece la solución del diagrama de abajo. En esta modalidad se entregaría **únicamente** el código en fondo azul y verde (el contenedor que sirve de capa de comunicación, junto con todo lo que hay en su interior). Esto es el código muy específico de su situación, pero con un contenedor que lo deje listo para “engancharse” con otra plataforma MADRE en caso de ser necesario:



El contenedor es una tecnología que se refiere de manera abstracta a funcionalidades que maneja la plataforma madre. Así, por ejemplo, el código específico le pregunta al contenedor: "¿este usuario, con este perfil, puede realizar esta acción?". El contenedor formula esa misma pregunta al traductor de la respectiva plataforma MADRE.

Así, en la situación hipotética en que el cliente busque contar con este código, en la segunda respuesta, y algún día desee retirarse de ser cliente nuestro: en tal caso, un nuevo equipo de desarrolladores que ya cuente con PLATAFORMA MADRE, sólo tendría que encargarse de elaborar el TRADUCTOR A LA PLATAFORMA MADRE. El resto de funciones internas y externas ya estaría desarrollado. Incluso, se pueden usar soluciones como los "frameworks open source".

Nuestra plataforma se ha esforzado en ofrecer una alternativa en donde uno de sus factores diferenciadores ha sido un esfuerzo en mejorar aspectos de seguridad informática que hemos echado de menos en lo común del mercado; hemos trabajado para buscar mayor agilidad de desarrollo y flexibilidad para diferentes aplicaciones, facilidad de testeo entre otras ventajas; pero el diagrama ilustra que esa plataforma en rojo, en caso de requerirse, podría usted reemplazarla por otra plataforma y ensamblarla con el código en azul. Requerirá un proceso de "empalme" para que otros desarrolladores "enganchen" otra plataforma que represente la parte roja, y se entienda con el subcódigo en azul y verde del diagrama.

Esta separación permitirá que el cliente no se vea obligado a sufrir sin necesidad en este momento los altos costos de la "Plataforma MADRE" ó empezar de ceros, y más aún, le dará una flexibilidad que no tendría en otras plataformas. La base de datos de la presente plataforma se daría de manera separada. Si en algún momento se viera obligado a ello, existen en el mercado incluso los famosos "frameworks" de

código abierto (no quiere decir que sea nuestra recomendación usarlos, pero son muy populares en la comunidad de programadores y nosotros mismos tuvimos la oportunidad de trabajar con ellos hace varios años) como solución de agilidad en la que un tercero podría colaborarle, y reemplazar nuestra plataforma “madre” con alguno de dichos frameworks.

Nuestro servicio incluye la infraestructura para el funcionamiento, para seguridad técnica, y para contener la aplicación madre.

La tercera respuesta es: Sí tiene excepciones . Hay excepciones si su proyecto es pequeño y específico, y por su tamaño es viable desarrollarlo desde ceros para su compañía. Los casos usuales que hemos mencionado en las otras modalidades, describen situaciones maduras en donde un código lleva años de evolución y tienen un cierto tamaño, pero no son el único caso posible en Desarrollo de Software A Medida.

Puede ocurrir que otros desarroladores ó empresas del sector de TICs, ó empresas de ingeniería, necesiten un plugin muy específico, ó un API para su pre-existente plataforma, y requieren el código porque es su herramienta de trabajo, ó porque tienen un departamento de desarrollo que desea aprovecharlo, y lo solicitan explícitamente.

En general, si el proyecto es muy pequeño (menos de 2000 líneas de código), que no lleve años de maduración en múltiples casos, sino que se trata de una adaptación muy concreta en donde no usamos de nuestras plataformas, sino que se trata de un código desarrollado desde ceros para una situación muy específica de su compañía y exclusivamente para la misma, bien puede considerarse una excepción, si su compañía manifiesta explícitamente, desde antes de negociar, el estar dispuesta a pagar el excedente del código fuente. Estos son casos muy particulares y no son lo usual en el desarrollo de software. Por ejemplo, si usted ya tiene paquetes de software, si el desarrollo consiste en unir dos paquetes pre-existentes, y nuestra labor es únicamente la tarea de unión de esos paquetes, naturalmente sí es viable negociar el código a un precio razonable.

En tales excepciones, el acuerdo debe ser respetuoso de la ley 23 de 1982 y demás legislaciones ó decisiones multilaterales relevantes, para garantizar que se fomente en general el respeto a los derechos de autor, el derecho a la reivindicación de paternidad de las obras, y que si existe transferencia patrimonial (no moral) de derechos de autor, ello se realice juiciosamente, diferenciando en lo posible las modalidades contempladas en la ley.

¿Y qué pasa si como cliente deseo cerrar la relación con ustedes de proveeores?

No es nuestra cotidianidad, nuestros clientes **antiguos** no se preocupan jamás por eso, pero si usted es un cliente **nuevo**, puede plantearse esa pregunta. Respondiendo: El presente manifiesto hace parte de los anexos de nuestro contrato, y el siguiente compromiso tiene fuerza contractual, exigible a nosotros con fuerza de ley, bajo la ley 1480 del 2011 de la República de Colombia, artículos 1 al 3, el siguiente compromiso.

Es nuestro compromiso que será posible comprar el subcódigo del diagrama en verde y azul únicamente (el rojo puede sustituirlo con otra subplataforma, ver párrafos de la respuesta 2), por un costo que nunca será superior al que se haya pagado históricamente para el desarrollo que se tenga en un momento dado. Parte de nuestro compromiso es que brindaremos la documentación y mantendremos abierta esta posibilidad para usted, recordando que en el peor de los casos, el costo del acceso al código más la documentación, tendrá como máximo un valor similar al de la realización del proyecto y al dinero que nos hayan pagado hasta ese momento (que será tiempo de desarrollo continuo de código). El entregable en tal caso será únicamente el código que hayamos desarrollado estrictamente para usted, no código que hayamos financiado con otros proyectos. Pero recomendamos que usted sólo pague ese dinero si realmente es necesario, pues no es nuestro negocio ni deseo, ni nuestra práctica común, romper relaciones con nuestros clientes. Para nosotros es mucho mejor negocio y propósito de funcionamiento que nuestros clientes estén contentos con nosotros, y para eso trabajamos. Por el contrario a esta pregunta curiosamente común, en nuestras diversas empresas buscamos que las relaciones no sólo sean duraderas, sino muy amistosas, llenas de riqueza en posibilidades de crecimiento empresarial y personal para los involucrados.

No obstante, para dar gusto a esta inquietud, vamos a pensar en ese peor escenario: ya hemos dicho que el cliente puede enganchar el subcódigo de su proyecto con otras alternativas que puedan servir como plataforma MADRE y reemplazar al contenedor rojo del diagrama.

Pero, para que esta hipotética situación (en esa segunda respuesta no recomendada) con entrega de código se trate de algo más sólido que una ilusión de seguridad, y todavía sin los altos costos de financiar una infraestructura completa, ofrecemos incluir también, durante nuestra operación, envío DIARIO a servicio de almacenamiento en línea suministrado por el cliente para copias de seguridad, de lo siguiente, mientras nosotros aún estemos operando su plataforma:

1. Código específico (en azul y verde)
2. La documentación que se genere para facilidad de otra casa de software que nos suceda
3. Copia de base de datos a servicio de almacenamiento, sin las contraseñas de los usuarios, para

evitar que por eventuales robos de datos en la infraestructura del cliente se desproteja a los clientes de Commercial Cleaning Pros, reduciendo así este punto de fallo de seguridad.

Todo lo anterior con el fin de aliviar las preocupaciones del cliente en caso de que el cliente desee terminar relaciones con nosotros.

Si resultara que una futura casa de software requiriera soporte para adaptar ese código específico a otra plataforma madre (no debería necesitarlo, por la capa de abstracción que es el contenedor y elimina esa necesidad, pero vamos a imaginar que así es), entonces es nuestro compromiso que por un valor equivalente al poder adquisitivo actual (30 de junio de 2020) de U$100 por hora hasta un máximo de diez horas, estaremos dispuestos a asesorar virtualmente a otra casa de software, ya sea en inglés ó en español. La cifra puede ser actualizada y será sostenida por dos años con estas cantidades a partir de la fecha del presente documento (encabezado) siempre que no haya cambios abruptos en las economías de los países en este plazo.

¿Software a medida significa que yo puedo pedirles que hagan lo que yo quiera?

No, hay varias causales de rechazo de una solicitud. Por una parte, usted sabe que todo trabajo tiene un alcance, que no es ilimitado. Pero existen otras causales de rechazo.

El cliente puede participar en indicar algunas de sus preferencias, su visión de cómo debería ser el diseño, en indicar sus preocupaciones y las molestias que espera resolver con el desarrollo, en manifestar sus expectativas de manera clara en lo posible desde el comienzo del proyecto. No participa el cliente en todas las decisiones: existen algunos elementos que limitan la libertad absoluta del software a medida por causales de ética, de técnica, de legalidad, de impacto en el bienestar de las personas, de viabilidad, de alcance, de estándares de iure ó estándares de facto, entre otros.

Algunos criterios son:

- Conflictos éticos
- Respeto a los derechos de Habeas Data de las personas
- Que un desarrollo de software no represente un retroceso conceptual en ingeniería industrial, ó en operatividad.
- Legalidad de las solicitudes
- Solicitudes fuera del alcance de un proyecto, propuesta ó contrato.
- Que incluso dentro del alcance, se solicite un re-trabajo de algo ya realizado sobre lo cual no se objetó a tiempo.
- Que se considere que una decisión lo perjudica a usted ó a su compañía en el largo plazo, ó que

de manera ilegal ó desconsiderada se encuentre que se hace daño a alguien.
- Inconveniente relación costo/beneficio que perjudique un proyecto, ó su impacto sea mínimo en comparación con otros aspectos de mucha mayor importancia.
- Aspectos que son responsabilidad de terceras partes.

¿Software a medida significa que la propiedad intelectual pasa a ser mía?

No. Regresemos al ejemplo del cocinero. Una idea brillante no representa nunca la totalidad de un esfuerzo que soporta por detrás el bagaje previo a su realización.

Si solicito a una empresa de cocineros que elabore una comida para alimentar con ciertos platos a un grupo de personas, ello no implica hacerme acreedor de las recetas que ese equipo humano ha elaborado y probado con los años. La empresa de cocineros me ofrece años de experiencia de platos que han sometido a prueba no sólo en técnicas de preparación, sino que también acumulan criterios para tomar decisiones velando por la salud de los comensales, cuidar de combinaciones peligrosas, selección de materias primas, productos y proveedores que participan en la elaboración, entre otro gran conjunto de secretos de procedimientos que de ninguna manera hacen parte de lo que un comensal ó contratante de los servicios de cocina debería pretender como entregable. Su propósito es darle un servicio de calidad (una comida, en el ejemplo), no se trata de entregar al cliente sus recetas.

Esta pregunta curiosamente ha surgido en algunos clientes en el pasado. Lo que hemos observado en la experiencia de lo que hemos observado entre terceros es:

1) El tener un código fuente no es garantía para usted de que un producto podrá ser transformado, ni le da realmente seguridad práctica, solamente una ilusión que pronto se muestra como inútil. De hecho, en Internet hay muchos paquetes de código libre que gozan de muchísima documentación, comunidades, e historia. Y ahí están, sin que sea trivial su transformación ó adaptación sin mencionar los riesgos de seguridad, incluso cuando se ubica a los pocos y más expertos programadores dispuestos a invertir tiempo de su vida en dichas transformaciones. Es algo que por supuesto hemos también probado en diferentes ocasiones para evolucionar a las ofertas que concebimos para usted como cliente.
2) Algunas tecnologías cuentan con la posibilidad de ingresar módulos externos como "plugins" que se programan extendiendo un núcleo funcional: pero surgen problemas como:
    1. La dificultad de garantizar la seguridad informática con esos plugins que tendrán acceso a los archivos y datos;
    2. Se ha observado que a través de plugins y plantillas de paquetes famosos de código abierto han ingresado vulnerabilidades y ataques informáticos de manera exitosa.

3. Existen vulnerabilidades informáticas publicadas de los paquetes comerciales, y sitios web tanto legítimos como ilegítimos en donde se encuentran colecciones de técnicas para "crackear" dichas vulnerabilidades.
4. La prueba de que esta amenaza es real, es revisar un log diario de diferentes sitios web de bajo tráfico: ya en diferentes microempresas pequeñas, en los logs de sus sitios web, se han detectado decenas de ataques diarios en donde robots automáticos intentan explotar vulnerabilidades pre-identificadas. Las empresas de hosting tienen clara esta realidad e incluso han decidido forzar actualizaciones automáticas de ciertos paquetes famosos de software, con lo cual el software externo reconocido por esas empresas se estará modificando de manera automática, introduciéndose no sólo los parches de seguridad, sino también nuevos puntos de fallo por estar alterando un sistema que ya podía llevar años en funcionamiento y pruebas.
5. Se ha detectado en algunas ocasiones falta de garantía de soporte por parte del fabricante y por parte de la empresa intermediaria de donde se haya descargado ó comprado el plugin ó la plantilla de extensión. Se ha observado que a la hora de buscar soporte ó responsabilidad por un "plugin" comprado para completar un software de código abierto, las empresas intermediarias no necesariamente se responsabilizan del error, y no siempre pueden ubicar al programador que la realizó.
6. Debido a que se trata de un único código fuente evolucionado en una empresa, se ha observado que incluso cuando una gran empresa cliente cuenta con los mejores programadores, por ejemplo en los bancos, al final tiene un único caso de uso. En esta lógca, cada equipo de programadores tiene que volver a re-descubrir los aspectos más básicos, sin que se acumule conocimiento, generando altos costos y tiempos para problemas que puede ya haber resuelto la competencia del mismo sector, y que de todas formas no son parte nuclear de su negocio, sino que simplemente deberían estar ya resueltos. Es una gran desventaja técnica y económica comparada con el caso opuesto, a saber: cuando una plataforma de base evoluciona con muchos estudios de caso.

¿El software a medida será instalado en mis computadores ó mis servidores?

Nosotros ponemos los servidores, y ello es parte del servicio. Parte de las lecciones adquiridas con los años, es que todo software amerita cuidadosas decisiones técnicas sobre la infraestructura, para optimizar los recursos, agilizar los tiempos de reacción, brindar estabilidad y seguridad al cliente. Esto, sumado a los argumentos anteriores ya mencionados en otros puntos de este mismo documento. Adicionalmente, para que pueda aplicar la figura legal de computación en la nube por ejemplo para

exclusiones de IVA, se deben cumplir ciertos criterios técnicos.

No entiendo. Yo pago por un servicio a la medida, ¿no debería yo ser dueño de todo?

El dinero que usted paga no es para realizar una plataforma desde ceros, sino que le ofrecemos una plataforma que hemos desarrollado y adaptado durante varios años a múltiples proyectos de diferente naturaleza, evolucionando con centenares de usuarios que la han puesto a prueba, recibiendo muchas observaciones y optimizándola para diversos fines. Así, antes de usted llegar, se han construido centenares ó miles de versiones, con decenas de miles de problemas resueltos, para ofrecer a usted una adaptación de nuestra plataforma. El empezar con una plataforma que cuenta con años de experiencia le brinda a su empresa mayor estabilidad, seguridad informática, y un acervo de situaciones sutiles que pueden presentarse y sobre las cuales hemos procurado robustecer cada vez más los productos que entregamos a nuestros clientes. “Eso es sencillo”, dira usted, lo cual es un buen pensamiento a la hora de inventar en hombros de gigantes: pero no olvide que es una suma de centenas de miles de cuestiones sencillas.

Más aún, otra razón que se suma a esta tesis, es que en algunos de los desarrollos nuevos, algunas partes son cofinanciadas por nosotros y las otorgamos con un costo mínimo para usted, cuando observamos que es viable y justo, y sobretodo que no estamos perjudicando a otras personas.

Finalmente, observe usted que es nuestro propósito cooperar con el bien de un mercado en general que no sólo nos incluye a usted y a nosotros, sino a una economía en donde las decisiones equivocadas pueden perjudicar el desarrollo y la financiación del desarrollo y aprendizaje de la tecnología para otras personas. Es nuestra responsabilidad, suya y nuestra, tomar decisiones cuidadosas para las futuras generaciones.

Es que yo no quiero estar amarrado a un único proveedor

La respuesta aquí tiene varios puntos:

1. Sus datos son de su propiedad, y usted es libre de exportarlos cuando quiera, a donde quiera.
2. La aplicación tiene como tarea combinar esos datos e interpretarlos para usarlos como información. Usted es libre al tener el acceso a sus propios datos. Esta libertad no incluye el procesamiento de los mismos que realiza la aplicación, salvo con el costo adicional por el sub-

código en fondo azul y verde, que sólo recomendamos gastar si se hace estrictamente necesario.

3. Es más libre quien realiza acuerdos de largo plazo con un proveedor, que quien realiza acuerdos de largo plazo con una herramienta.

La palabra “amarrar” evoca una negativa falta de libertad. Pero en la práctica, el acuerdo con un proveedor significa tener un marco de trabajo que busca lograr una estructura de trabajo que permita agilizar soluciones. A veces perseguimos el tener muchas opciones por encima de tener mucha estabilidad y capacidad de ejecución. Y de todas formas las mejores opciones, si cumplen los criterios de viabilidad, se pueden construir con el proveedor de software con el cual usted tiene un acuerdo de largo plazo.

La lealtad a un proveedor es superior a la lealtad a una herramienta: en el primer caso (lealtad al proveedor), usted tiene la alternativa de cambiar ambos proveedor y herramienta, llevando sus datos a otra parte, pues son de su propiedad; pero en el segundo caso (lealtad a la herramienta) sin un único proveedor que se responsabilice por dicha herramienta, usted debe tantear con ambas cosas (tantear proveedor y tantear herramienta), mientras se distrae del propósito esencial de su negocio, que es diferente del desarrollo de software.

Quien está amarrado a una herramienta que no funciona ó no se puede escalar, es quien sufre, no quien realiza acuerdos de largo plazo con un proveedor.

Examinemos esta preocupación típica. Si el proveedor es justo con sus políticas y cumple un buen papel, procura máxima ética y cumplir buenos principios con sus clientes, el mantener relaciones de largo plazo no debería ser un problema. Pero supongamos que esto usted no lo sabe: usted conoce a una nueva empresa de desarrollo de software, y usted no sabe si este proveedor de software será ético, confiable y tendrá las mejores prácticas técnicas. Esa incertidumbre se compara, entonces, con la alternativa: si usted se organiza para tener varios proveedores que tengan acceso a su software y datos, entonces usted tendrá esa misma incertidumbre con cada nuevo proveedor que contrate, y tendrá que reproducir el mismo riesgo varias veces.

Increíblemente, esta frase de René Descartes la encontramos después de varios años de haber confirmado el siguiente pensamiento. René la descubrió para los arquitectos, modestamente nosotros la habíamos observado para los programadores. Recordémosla:

> *Uno [de mis pensamientos] fue el ocurrírseme considerar que muchas veces sucede que no hay tanta perfección en las obras compuestas de varios trozos y hechas por las manos de muchos maestros, como en aquellas en que uno solo ha trabajado. Así vemos que los edificios, que un solo arquitecto ha comenzado y rematado, suelen ser más hermosos y mejor ordenados que aquellos otros, que varios han tratado de componer y*

arreglar, utilizando antiguos muros, construidos para otros fines.
René Descartes, El Discurso del Método, II

Antes de conocer este pensamiento de Descartes, habíamos notado el siguiente efecto en el software: cuando se posee un código fuente, se debe conseguir y financiar los honorarios de aquel programador dispuesto a realizar intervenciones. Se ha observado que en estas lógicas, ocurre que la empresa poseedora de un código fuente no contrata a un único programador sino que un código fuente empieza a tener participaciones de diferentes programadores con diferentes estilos de programación, técnicas, costumbres, llenando de “pequeños parches” el código, para soluciones “ad hoc”, que terminan estando desconectadas unas de otras, haciendo que el "caos" en el código crezca, y con él la dificultad de modificación. Esto genera que la empresa que por tener <<diferentes proveedores>>, es decir <<diferentes programadores>> que manipulen el código, deba pagar crecientes sumas por modificaciones que serían más económicas por otros métodos, como es el de evolucionar un código a través de muchos proyectos por parte de una empresa de software dedicada a eso.

Vamos a suponer que al final, después de tantear y sufrir costos, pensando con el mayor optimismo, usted finalmente encontrará un proveedor que reúna en un solo proveedor un alto nivel ético y técnico, que respete la confidencialidad de su compañía, y que adicionalmente esté dispuesto a estudiar el código fuente para su compañía. <<*Tenemos a la persona que reúne estas condiciones y está disponible*>>, es su respuesta. Pero regresa usted entonces a una condición peor que su preocupación original. En efecto, ¿qué ocurrirá? Con el paso del tiempo, esta persona de excelentes cualidades humanas y técnicas, evolucionará versiones con UN código con UN caso de uso, que es el de su compañía. Al final, su empresa terminará dependiendo de ambas cosas únicas, programador y herramienta. En ese punto, su opción es exportar los datos a otra nueva herramienta…. Regresando al mismo punto original que ya hubiera tenido al contratar a un proveedor único de software a medida, que de todas formas le permite esa exportación de datos.

Recuerde que:

1. Usted no es el único que ha pagado por un desarrollo de software. Esto significa que el producto se soporta sobre una gran historia de proyectos y desarrollos. Y que recibirá servicio de soporte y sobretodo escalabilidad, por un proveedor experto en software, evitando a usted costos futuros de tantear soluciones ya probadas, reclutar nuevos programadores, y sobretodo reducir riesgos de errores por manipulaciones equivocadas.

2. La confiabilidad y estabilidad de un software a partir de cierto tamaño es algo que se logra con años de evolución de miles de correcciones, con centenas ó miles de usuarios puestos a prueba.

Pero hay excepciones si su proyecto es pequeño y específico, y por su tamaño es viable desarrollarlo desde ceros para su compañía. Esto se explicará en la sección de “excepciones”.

Y el riesgo de la obsolescencia: cómo lo combatimos nosotros.

Suele ocurrir en otras modalidades que un software absolutamente personalizado simplemente se deja instalado en una infraestructura y si bien lo pueden estar usando, no tiene revisiones de fondo ni maduración más allá de actualizaciones básicas, que es seguramente la experiencia que usted teme si se siente identificado con esta pregunta.

Así, un problema típico al desarrollar software personalizado, pero que puede reducirse notablemente, es el siguiente: las tecnologías evolucionan rápidamente y retomar el mantenimiento de una pieza digital personalizada cuesta tiempo, recordar cómo era, cómo estaba hecha, para revisarla, corregirla, hacerle mejoras ó escalarla: incluso si existen buenas prácticas y documentación, el acto de recordar se hace en general más costoso entre más tiempo pasa, y el acto de mantener la compatibilidad de una tecnología es también costoso. Es por ello que hemos procurado generar desarrollos elaborados sobre pilares que tienen una revisión y uso frecuentes, porque su base es común. Parte de nuestro ejercicio ha consistido en tipificar la lógica común de los diferentes proyectos, generalizarla al máximo, para que el ejercicio que usted realiza pueda, potencialmente, ser también aprovechado y puesto a prueba por un negocio totalmente opuesto. Ello permite que ambos negocios revisen y retroalimenten sobre potenciales fallos u obsolescencia que pueda tener un aplicativo, aprovechando el esfuerzo y pruebas de varias personas en situaciones diversas. Y, al ser usado por diversas compañías, si bien no asegura sí incrementa la probabilidad de mantenerse en permanente revisión. En la medida en que las compañías usuarias desean hacer escalamientos de aplicativos, ello genera que componentes comunes sean revisados, con lo cual cuando alguien invierte en software, muchas personas incrementan la estabilidad de sus aplicativos pre-existentes, incluso si no tienen nada que ver con el desarrollo pagado originalmente. Así, usted y lo demás se benefician de la evolución completa de la comunidad.

> ***En términos gráficos:*** *vamos a suponer que somos plomeros, por un instante. Si usted nos encarga construir un acueducto especial en una zona desértica, ello nos obliga a revisar y madurar los sistemas de tubería que tenemos para esos entornos. Construimos el acueducto para usted, a su medida, pero muchas personas pueden beneficiarse de las investigaciones, innovaciones y desarrollos que hagamos en los materiales de tubería, sistemas de construcción, transporte, y todo lo que estuvo alrededor como industria de base para poder construir el acueducto para usted.*

Así, resolvemos ó al menos minimizamos de esta manera un problema que por desgracia existe incluso

en el software abierto: que por ser gratuito genera la ilusión de que siempre estará actualizado, e incluso a veces viene con actualizaciones automáticas. Pero que a su vez está compuesto de pequeños componentes sobre los cuales surgen incompatibilidades dado que una herramienta sufrió actualizaciones y la otra no. Y el ciclo se convierte en estar persiguiendo siempre la última versión actualizada de la totalidad de componentes, ó al menos aquellos que todavía tengan soporte. En el modelo de desarrollo de software aquí presentado, su software evoluciona porque se basa en el mismo núcleo de otros proyectos.

¿Qué pasa si ustedes los proveedores se mueren, si desaparece la empresa, si ocurre un terrible siniestro?

La muerte de proveedores ó de clientes no es el causal más común de cierre de negocios ó fracaso de proyectos. Existen otras causales mucho más importantes y recurrentes, que hemos venido combatiendo a lo largo de los años.

- Por una parte, nuestra garantía y cubrimiento de responsabilidades excluye situaciones de esa índole, es un riesgo que ambas partes debemos asumir.
- Por otra parte, usted tiene acceso a la descarga diaria de sus datos y copia de los mismos, y procuramos pagar con anticipación a nuestros proveedores, intentando que nuestros clientes tengan mayor capacidad de reacción en caso de que nos ocurra un caso fortuito, fuerza mayor, muerte, daños causados por terceras partes, ó en general situaciones que escapen a nuestro control y voluntad.

Historial de versiones de este documento

Versión 0.4: junio 30 de 2020	Se mejoraron redacciones, conceptos, el resumen, se enriquecieron conceptos.
Versión 0.3: abril 5 de 2020	Se agregó sección de “El riesgo de la obsolescencia”, se mejoró la introducción
Versión 0.2: Diciembre 04 de 2019	Se aclararon algunos párrafos, se corrigió redacción. <<Las prácticas que consideramos mejores (...)>>, y por qué el cliente en general no debe asumir el innecesario costo del código fuente. Se agregó la sección de “excepciones” para proyectos muy pequeños y específicos.
Versión 0.1: Noviembre 01 de 2019	