

La naturaleza del Software

Haciendo un repaso de nuestra historia, me he dado cuenta que llevamos dedicados más de veinte años a la tarea de **construir software** para clientes privados y públicos de toda naturaleza.

Fundamentalmente hemos desarrollado **software de gestión empresarial**, pero no solo, hemos dado soluciones a problemas de toda naturaleza. Desde hace ya más de diez años estamos centrados en la **consultoría tecnológica** y en **diseño de arquitecturas de software**.

Gracias a estos últimos años, empiezo a vislumbrar algunos patrones repetidos en los **éxitos y fracasos**. Creo que es momento de poner en limpio todo lo aprendido en estos veinte años. Por ello, me siento con ganas de deciros cuales son nuestros **diez mandamientos** en los **diseños de arquitectura**:

El software es un negocio y como tal hay que tratarlo

Básicamente, todo lo que hacemos, parte de una necesidad que cubrir, un resultado esperado, un presupuesto asignado que alguien paga y una serie de personas que lo hacen.

Resumiendo, lo que lo primero que tenemos que tener claro en todo proyecto es:

- ¿Qué vamos a resolver?
- ¿Cuanto cuesta?
- ¿Quien lo paga?
- ¿Quién lo hace?
- ¿Cuál es el beneficio?

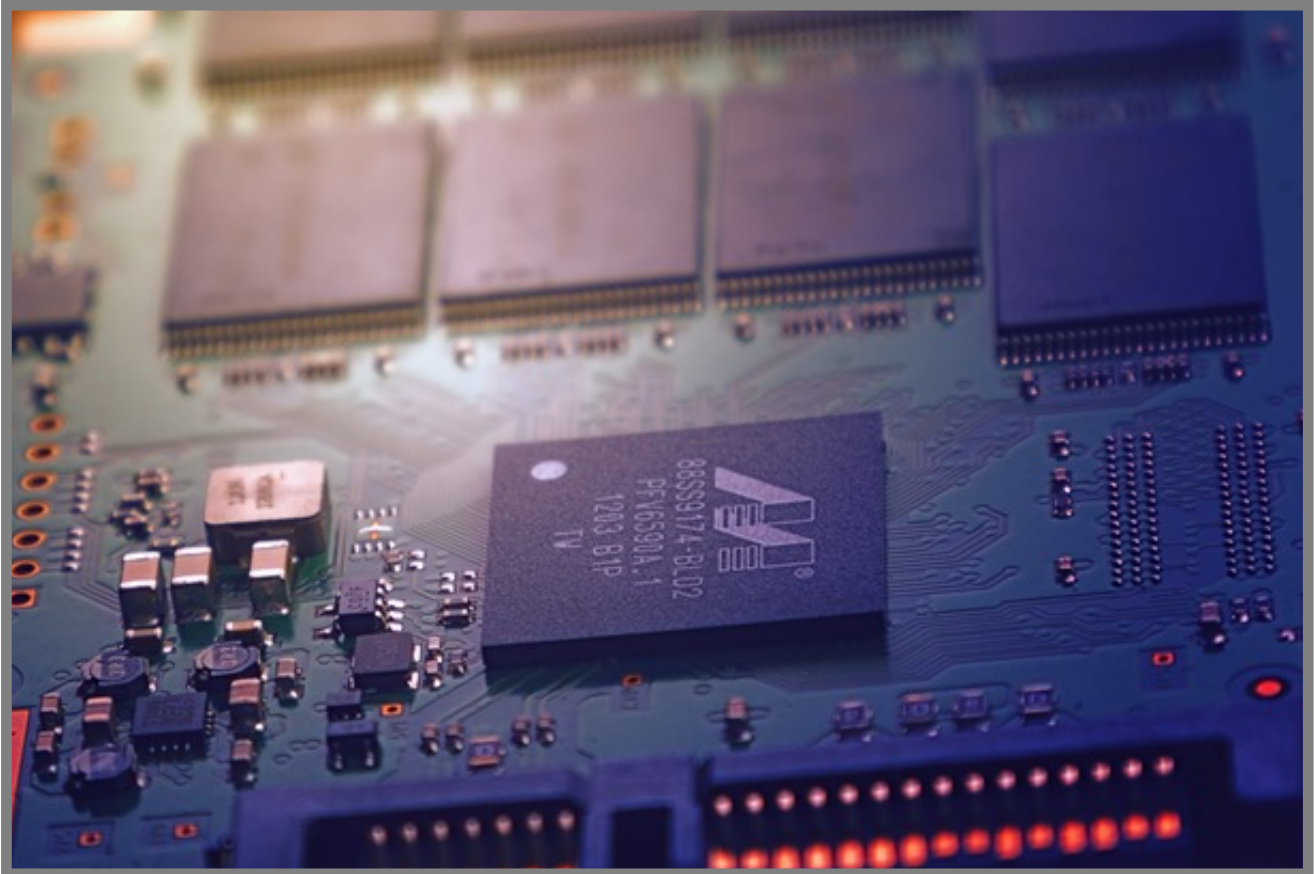


Si alguna de estas preguntas no tiene respuesta directa, ya hemos fracasado.

-

El software es de naturaleza compleja

Sin lugar a dudas, el software es de naturaleza compleja, tienes que aprender a lidiar con esa complejidad. De igual forma, el cliente y todos los agentes intervinientes en el proceso, también tienen que entender esta cuestión.



Si alguien de peso en el proceso productivo no entiende esta cuestión, ya hemos fracasado.

El problema a resolver se puede describir perfectamente en una página

Si el cliente no es capaz de describir el problema en una página, es que no tiene claro cual es su problema.

Si después de que el cliente te lo cuente de forma simple, no eres capaz de plasmarlo en una página, es que no entendiste el problema que el cliente te describió.



Si no somos capaces de describir el problema en una página, ya hemos fracasado.

Las personas mienten

Cuando conversas con el cliente, analista, jefe de proyecto respecto al problema que quieren resolver, no siempre enfocan bien la descripción de su problema.

Normalmente este mal enfoque se produce por:

- Desconocimiento del detalle del negocio.
- Por quererte engañar, mejorando su posición de cara a la negociación.
- Por no poder o querer contarte ciertas realidades (lo que yo llamo sus miserias).
- Por intentar dar soluciones técnicas al problema sin disponer del conocimiento tecnológico necesario.
- ...



Si no descubrimos lo que nos ocultan, ya hemos fracasado.

Simple, pero que resuelva

Para que una solución tecnológica perdure en el tiempo tiene que partir de un pequeño producto que resuelva el problema de forma clara (mínimo producto viable).



Si no la solución a desarrollar no es simple, ya hemos fracasado.

La documentación es una representación de la realidad, no la realidad en si misma

Los documentos que recibes, son una representación de la realidad, no la realidad en si misma. Normalmente, si profundizas un poco verás que la realidad es diferente.

Estos documentos los recibimos de múltiples fuentes; cliente, analista, desarrolladores, operaciones, testing, QA, jefes de proyecto... todos ellos los tienes que poner en duda.



Si no somos capaces de encontrar la realidad subyacente, ya hemos fracasado.

Programar es el arte de introducir nuevos errores

Cuando picas código, estás añadiendo nuevos errores en la ejecución de tu programa. Ser ordenado y aburrido ayuda a minimizarlos.

Cada línea de tu código es importante, una sola línea puede tirar por tierra todos tus esfuerzos.

Un buen diseño de arquitectura permite la introducción de nuevos errores sin provocar el fallo del sistema. Para ello tienes que determinar cual es el centro del mismo y aislarlo del resto de componentes.

Cuando alguien comete un error en el proceso productivo, no le matarás, más bien harás lo contrario. Le contarás las implicaciones de su error, le ayudarás a mejorar sus habilidades, le reforzarás en sus aciertos y todo ello con la intención de que crezca en el proceso. Un buen profesional del mundo del software es difícil de encontrar y muy fácil de perder.



Si no somos ordenados y aburridos, ya hemos fracasado.

Las modas, modas son

Está muy de moda...

- NodeJS/AngularJS
- Cloud/Containers/Micro-servicios
- AgileXP/Scrum
- PMP/ITIL
- DevOps
- ...

Son modas... solo unas pocas perduran, otras pocas mutan y el resto perece.



Aplica las modas en la medida justa, en la medida que sienten bien al proyecto, si nos excedemos, ya hemos fracasado.

Cuantas menos personas conformen el proyecto mejor es la comunicación entre ellas

Cada día más es habitual ver grandes proyectos formados por decenas de personas. A más personas en un proyecto peor es la comunicación entre ellos.

Trocea bien el sistema a desarrollar, que cada grupo se focalice en su parte y evita solapamientos.

Haz responsable del código a los programadores, haz responsable de la arquitectura a los arquitectos, haz responsable de las garantías a QA.



Si no conseguimos un grupo de personas adecuadas para el proyecto, ya hemos fracasado.