

On the Role of the Software Architect

Introducción

Este trabajo se basa en la participación en el episodio 616 del podcast *Software Engineering Radio* de **Ori Saporta**, cofundador y Arquitecto de Sistemas de **vFunction**, una empresa dedicada a proveer servicios de “**observabilidad arquitectónica**” en productos de software (centrándose en el análisis de dependencias), siendo una de las primeras a nivel mundial en incorporar el uso de Inteligencia Artificial a su plataforma. Se dedican, por lo tanto, a ayudar a equipos de desarrollo y arquitectos de software a entender mejor la arquitectura de sus productos, buscando mejorar sus diseños.

Diseño constante

La principal idea a combatir, aún presente en muchas compañías, es la de que el proceso de diseño de la arquitectura ha de realizarse al completo previamente al proceso de implementación. Es más común de lo que parece que los arquitectos de software migren constantemente de proyecto en proyecto, limitándose a diseñar la arquitectura en sus inicios, cuando el proceso de diseño y desarrollo es, en realidad, **un proceso vivo y constante**, que se enfrenta constantemente a cambios y adaptaciones, también a nivel arquitectónico. Es importante, por ende, que el arquitecto forme parte activamente del desarrollo, de forma que pueda modificar el diseño para responder a cambios, incluir nuevas funcionalidades o, incluso, llegar a mejores diseños a base de iteraciones. Tratar el software como algo estático hace que este se vuelva complejo y lleva, con el tiempo, a acumular deuda técnica y a un posible colapso, por lo que ha de evitarse. Un buen diseño es dinámico, y en ese aspecto es tarea del arquitecto ser facilitador del cambio. Es importante revisar constantemente el producto, sus usos y requisitos, qué mejorar o adaptar, teniendo en cuenta que un buen diseño hace un año no tiene por qué ser un buen diseño hoy.

Metodologías Ágiles vs Metodologías en Cascada

Con la cada vez mayor aplicación en la industria software de metodologías de desarrollo ágil, que dan mayor independencia a desarrolladores y equipos de trabajo, el papel del arquitecto de software ha pasado, en parte, al de una especie de consultor. De las metodologías en cascada, que solían conllevar grandes procesos de diseño previos a la implementación, hemos pasado al uso de metodologías ágiles, que reducen considerablemente estos o, incluso, los eliminan. Esto lleva a preguntarse si la *agilidad* que nos aportan estas metodologías compensa los problemas de diseño que pueda causar la falta de planificación o no. Para Ori, lo ideal sería encontrar un **punto de equilibrio** que permita ser ágil manteniendo un cierto proceso de diseño previo, que asegure la posibilidad de añadir funcionalidades a la aplicación sin arriesgar su integridad.

El camino del Arquitecto de Software

Nadie empieza su carrera siendo arquitecto, eso está claro. El camino profesional más común para llegar a ocupar este puesto suele partir desde la base del lado técnico del negocio: el desarrollador. Con los años, los desarrolladores ganan experiencia y han de elegir entre dos opciones: especializarse en el uso concreto de una tecnología/metodología o **buscar una visión más amplia del software**. Quienes eligen esta segunda vía, tras un proceso de estudio, son quienes podrán optar a convertirse en arquitectos de software.

Otro camino menos común (pero que es el caso de Ori, el protagonista del podcast) es el de partir desde el lado no técnico del negocio. No obstante, si algo comparte todo aquel que llega

a ser arquitecto, es el **interés en sistemas software complejos** y, algo esencial, **ser buen programador**.

El arquitecto como un puente

El arquitecto de software ha de entender la tecnología sobre la que trabaja y cómo cada pieza de esta interacciona con las demás, pero su trabajo habitual también consiste en tomar decisiones y **conectar dos mundos**: el técnico (*cómo*) y el de negocio (*qué*). El arquitecto ha de servir como un nodo que comunica a clientes, ejecutivos y personal no técnico de una compañía con los equipos de desarrollo. Ha de ser capaz de transmitir dudas, decisiones, cambios e información entre ambos lados, y conocerlos a la perfección, pudiendo actuar de mediador entre estos y asegurando que se entiendan. Es, además, la primera persona a la que se debería buscar en caso de un problema desde el lado de negocio.

Soft skills y trabajo en equipo

Más esenciales, incluso, que las habilidades técnicas, son las llamadas *soft skills*, completamente necesarias para trabajar eficientemente con otras personas. Un buen arquitecto ha de ser alguien capaz de transmitir confianza al resto del equipo y comunicarse eficientemente con cualquier persona que lo necesite, tenga conocimientos técnicos o no. Ha de poder entender a los demás, y guiar al equipo sin imponer sus decisiones, buscando hacer ver su punto de vista y por qué puede haber una mejor forma de hacer las cosas a aquella que otros desarrolladores o individuos hayan tomado (teniendo en cuenta, también, que su solución no tiene por qué ser siempre la mejor). Trabajar en equipo y no ser simplemente un “*jefe*” es crucial para llegar a buenas soluciones.

Documentación y calidad

Históricamente, la figura del arquitecto ha estado estrechamente ligada a la documentación de los proyectos. Comentarios, diagramas, modelos y otras técnicas de documentación son de gran importancia y ayuda, pero su uso en exceso puede llevar a conflictos entre lo documentado/diseñado y lo posteriormente implementado, que **no siempre ha de corresponderse**. Como hemos mencionado ya, el desarrollo software es un proceso dinámico, por lo que la documentación también ha de serlo.

Por otra parte, al referirnos a *calidad* es difícil acotar una definición. Para Ori, en el ámbito del software, la calidad es la habilidad de **entender un problema y llegar a una solución que tenga sentido** y que sirva como una **base** sobre la que seguir trabajando después. Una buena calidad es la piedra angular para construir y expandir grandes sistemas software, haciéndolos sostenibles mediante un buen diseño. Es también deber del arquitecto asegurar que se respeten unas buenas **prácticas y estándares** que aseguren esta calidad.

El buen arquitecto de software

Un buen arquitecto es, en definitiva, alguien capaz de mantener una visión de cómo la arquitectura de un software ha de ser y evolucionar con el tiempo, estando presente en todo su ciclo de vida, siendo capaz de responder a cambios y de cumplir con los requisitos iniciales y futuros. Ser buen arquitecto significa, también, permitir que el software diseñado incorpore nuevas (y buenas, no basta con añadir cosas sin razón) funcionalidades e integraciones. Es entender el problema y llegar a una solución real y efectiva, llevando un buen proceso de desarrollo y haciendo el producto lo mejor posible dentro de las restricciones, y siendo capaz de comunicarlo desde un punto de vista técnico o no.

Ser buen arquitecto consiste, esencialmente, en **diseñar para el cambio**.