

Cuando el software escribe el software

Este artículo refleja la situación en el momento de su publicación. El contenido puede haber quedado desactualizado desde entonces.

04/08/2021 Cada vez más, el *software* se está convirtiendo en el factor determinante en el desarrollo del automóvil. Marius Mihailovici, Director General de Porsche Engineering Rumanía, analiza el futuro del desarrollo de *software* y explica por qué el trabajo de un programador podría cambiar completamente en los próximos 20 años.

Incluso hoy en día, los coches son en gran medida ordenadores rodantes. Poseen una red de docenas de unidades de control electrónico (ECU), entre 70 y 100 en un vehículo moderno. Estas ECU controlan la inyección de combustible, regulan el funcionamiento de los frenos y supervisan el sistema de aire acondicionado. El siguiente paso serán las HCP (High Performance Computing Platforms), que permitirán integrar en una ECU una potencia de cálculo mucho mayor.

La mayor potencia de cálculo y la integración son necesarias porque el número de líneas de código y la

complejidad de las funciones del vehículo aumentan año tras año. Una cifra puede aclararlo: 100 millones. Es el número de líneas de código de un coche moderno. En comparación, un Boeing 787 Dreamliner solo tiene 14 millones de líneas de código.

También hay muchas líneas de código detrás del sistema de entretenimiento y navegación. A esto hay que añadir la posibilidad de conectar *smartphones* y otros dispositivos al coche, que también es posible solo con un *software* complejo. Y la cosa no acaba ahí. El *software* está asumiendo cada vez más tareas importantes en el coche. Las funciones más importantes hoy y en el futuro son el intercambio de datos con otros usuarios de la carretera y con la infraestructura, la actualización de los vehículos desde la nube y, con el tiempo, incluso la conducción autónoma.

Lograr el equilibrio en el desarrollo de *software* para automóviles se ha vuelto algo muy complicado. Hay que cumplir las normas de seguridad y los requisitos de los clientes en forma de extensas especificaciones. Los procesos de desarrollo convencionales en la industria suelen estar sujetos a un calendario predeterminado que define ciertos avances. Los clientes esperan resultados regulares en fechas concretas.

A esto se añaden los procesos de aprobación oficial conocidos como homologación. Por ejemplo, se estipula que los vehículos no pueden salir al mercado hasta que hayan sido fabricados y validados en una determinada cantidad. Al final, en la industria siempre se trata de pasar de un estado "terminado" al siguiente. Todos estos retos hacen que se requiera un trabajo rápido y flexible. Los objetivos suelen fijarse con poca antelación y cambian rápidamente. Se podría decir que el *software* se desarrolla en un proceso orientado a los resultados.

En Cluj, nuestros programadores trabajan con herramientas que han demostrado su eficacia en el desarrollo de *software* durante 20 años. Utilizamos métodos ágiles siempre que es posible. Se basan en pequeños pasos de desarrollo cuyos resultados se comprueban en series diarias de retroalimentación. Los equipos individuales tienen una gran libertad y trabajan directamente entre sí para apoyarse en sus respectivas capacidades. Fijan con flexibilidad sus propios objetivos cada día. Los jefes de equipo suelen tener aquí solo una función moderadora para no perder de vista el panorama general.

Otro paradigma actual del desarrollo de la programación es la "integración continua". Se trata de un proceso muy automatizado, en el que los elementos se comprueban al final de una jornada de trabajo para ver si están listos para funcionar e integrarse en el sistema general. De este modo, podemos identificar y eliminar rápidamente los errores y problemas. El uso de métodos ágiles y de integración continua garantiza una mayor eficiencia y también ofrece a nuestros clientes un valor añadido: facilita la presentación de resultados provisionales y les da la oportunidad de recibir una rápida respuesta.

Para los próximos cinco años, espero ver un uso creciente de los métodos descritos anteriormente. Nuestro trabajo estará más orientado a los resultados que al tiempo. Es probable que las jerarquías de los equipos pierdan importancia y preveo un mayor grado de equipos planos y autoorganizados con responsabilidades claras. Además, será mucho más frecuente trabajar de forma flexible desde diferentes lugares, basándose en la idea aprovechar la experiencia de las personas estén donde estén.

También asistiremos a una creciente automatización en los próximos años. Cada vez es más frecuente que el propio *software* sea probado por el *software* y no por las personas. Las pruebas manuales desaparecerán por completo.

Los límites entre el coche y el entorno se difuminan

Los avances fuera de la industria del automóvil también nos obligarán a replantearnos las cosas, ya que los coches están cada vez más integrados en la vida digital de sus conductores. Por ejemplo, el *smartphone* se conecta automáticamente al vehículo al entrar en él. El uso de los contenidos, la navegación y la comunicación se fusionan a la perfección.

Este tipo de utilización cambiará el trabajo de nuestros desarrolladores, por la sencilla razón de que los límites entre el "*software* de automoción" puro y otras aplicaciones se están difuminando. Por cierto, esto también requiere una determinada mentalidad por parte de nuestros desarrolladores: empleamos a personas que viven el estilo de vida digital. No solo saben lo que demandan nuestros clientes, también lo que los usuarios de vehículos esperan de sus coches.

Las ECU del futuro influyen en el desarrollo de la programación

Las nuevas arquitecturas de las ECU también están cambiando la forma de desarrollar el *software* del automóvil. Supongo que en el coche del futuro habrá unos pocos ordenadores centrales muy potentes del tipo HCP que, junto con unidades subordinadas más sencillas, controlarán todo el vehículo. Estas unidades centrales también ejecutarán todas las aplicaciones que vayan más allá de las funciones básicas, como el entretenimiento, el tráfico de datos o las aplicaciones de comunicación con los pasajeros. Estos ordenadores centrales, con un verdadero sistema operativo, convierten realmente el coche en un PC sobre cuatro ruedas.

Para los desarrolladores, esto significa que los métodos de su trabajo no varían tanto. Lo que cambia son los sistemas con los que tratan: son más jerárquicos, tienen menos componentes y están controlados por un *software* global. Y, como cualquiera, recibe actualizaciones periódicas. Así que no se desarrollará e instalará una vez, sino que se perfeccionará constantemente, incluso cuando el coche ya está en manos del cliente.

¿Un futuro sin programadores?

La pregunta no es en absoluto absurda: ¿habrá todavía desarrolladores en el futuro? Algunos expertos asumen que, en las próximas dos décadas, la inteligencia artificial (IA) se hará cargo del desarrollo de *software* por completo, incluso en la industria del automóvil. Sin embargo, las opiniones difieren al respecto. Personalmente, me imagino que seguiremos estableciendo el marco y que la IA se encargará de implementarlo. Seguiremos necesitando arquitectos de *software* , ingenieros de requisitos e

ingenieros de *software* para definir lo que la IA o las redes neuronales harán: generar funciones basadas en los requisitos definidos, probarlas automáticamente y corregirlas continuamente hasta que la calidad del *software* alcance el nivel esperado.

Por tanto, los desarrolladores del futuro deberán ser capaces de pensar en términos de sistemas completos y saber cómo utilizan sus vehículos los clientes finales. Cada vez más, su trabajo no solo estará determinado por los requierimientos de los fabricantes, también por los consumidores. Sin embargo, hay algo que no puedo imaginar: un *software* completamente sin "errores". Siempre habrá errores, solo que no tenemos que encontrarlos y arreglarlos nosotros mismos: el futuro nuevo sistema lo hará, supervisado por expertos humanos.

Marius Mihailovici

Marius Mihailovici es Director General de Porsche Engineering Rumanía desde 2016. Anteriormente, dirigió la investigación y el desarrollo en Alcatel-Lucent S.A./Nokia Oyj, donde también trabajó como Director de S*oftware* para proyectos de 2G y 3G. En World Telecom desempeñó labores como ingeniero de comunicaciones.

Información

Texto: Marius Mihailovici

Artículo publicado en el número 1/2021 de la revista Porsche Engineering Magazine.

Consumption data

Taycan Turbo S (Predecessor model)

*Further information on the official fuel consumption and the official specific CO_2 emissions of new passenger cars can be found in the "Leitfaden über den Kraftstoffverbrauch, die CO_2-Emissionen und den Stromverbrauch neuer Personenkraftwagen" (Fuel Consumption, CO_2Emissions and Electricity Consumption Guide for New Passenger Cars), which is available free of charge at all sales outlets and from DAT (Deutsche Automobil Treuhand GmbH, Helmuth-Hirth-Str. 1, 73760 Ostfildern-Scharnhausen, www.dat.de).

Image Sublines

Path: Cuando el software escribe el software/fotos/img_1.jpg
Title: Marius Mihailovici, Director General de Porsche Engineering Rumanía, 2021, Porsche AG
Subline: Marius Mihailovici, Director General de Porsche Engineering Rumanía.

Link Collection

Link to this article
https://newsroom.porsche.com/es_ES/tecnologia/2021/es-porsche-engineering-cuando-programas-crean-escriben-software-25349.html

External Links
https://www.porscheengineering.com/peg/en/

Dr. Ing. h.c. F. Porsche Aktiengesellschaft
Porscheplatz 1
70435 Stuttgart