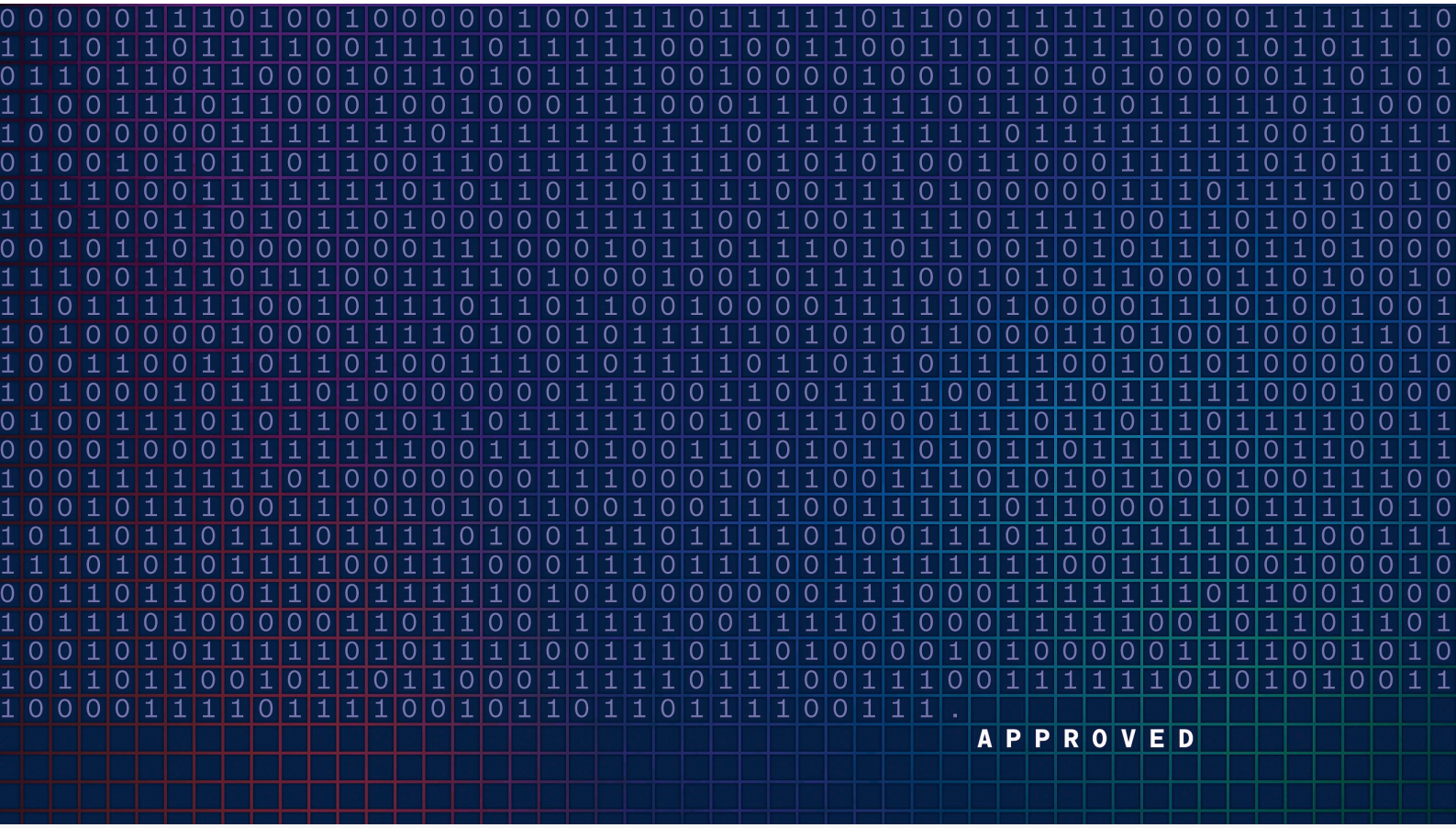

Software seguro: Poniendo el código a prueba

10/09/2024 El software juega un papel clave en la vida moderna, pero también en los vehículos. Por esta razón, los fabricantes de equipos originales (OEM) y los proveedores utilizan métodos y herramientas probados para identificar errores en los programas lo antes posible. Los investigadores ya están trabajando en nuevos enfoques, incluidos algunos basados en inteligencia artificial.

Hoy en día, los vehículos son computadoras móviles con una red de hasta 100 unidades de control electrónico que controlan las funciones del motor y la batería, monitorizan el sistema de aire acondicionado y controlan el sistema de infoentretenimiento. Y se están añadiendo cada vez más funciones inteligentes como el control de crucero adaptativo y las funciones de conducción automatizada. Todo esto solo es posible con la ayuda de software complejo.

A medida que aumenta la complejidad del software, también lo hace el trabajo requerido por los OEM y los proveedores para evitar errores y así garantizar una alta calidad del software. Por un lado, utilizan los procesos estándar de la industria como orientación, particularmente Automotive SPICE (ASPICE) e ISO 26262, así como directrices internas de codificación y calidad que, por ejemplo, prohíben el uso de

funciones propensas a errores en ciertos lenguajes de programación. Con sus procesos de desarrollo, Porsche Engineering logra de manera confiable y reproducible el Nivel 2 de ASPICE. Esto no solo corresponde al estado actual de la técnica y constituye la base esencial para la aprobación en el vehículo, sino que también asegura a los clientes que los errores y las no conformidades pueden ser detectados y remediados en una etapa temprana.

En el desarrollo de software, Porsche Engineering aplica el modelo en V (ver ilustración): A la izquierda, de arriba a abajo, se encuentran los pasos de Requisitos del sistema, Arquitectura del sistema, Requisitos del software y Arquitectura del software. En la base de la V está el diseño del software, que es seguido en el lado derecho de abajo hacia arriba por los pasos de pruebas unitarias, pruebas de integración y pruebas de calificación, así como pruebas de aceptación y uso. "Cada paso a la izquierda corresponde a un paso de prueba a la derecha", explica Stefan Rathgeber, Director de Sistemas de Software de Alto Voltaje en Porsche Engineering. "En las pruebas unitarias, por ejemplo, probamos la unidad más pequeña a nivel funcional, luego siguen las pruebas de componentes un nivel por encima". Para todos los niveles, existen catálogos de pruebas que tienen en cuenta todas las variantes.

Un equipo dedicado de gestores de calidad en Porsche Engineering verifica que todos los pasos del proceso se cumplan y documenten durante el desarrollo del software. Realizan constantemente revisiones para encontrar problemas lo antes posible, porque el esfuerzo requerido para la resolución de problemas y la remediación aumenta considerablemente en las fases posteriores del desarrollo. La imposibilidad de probar todas las constelaciones posibles, por ejemplo, significa que ciertas combinaciones desafortunadas pueden causar problemas.

Las numerosas variantes de vehículos, versiones de equipamiento y actualizaciones de software después de la entrega de los vehículos plantean un desafío especial. "Las variantes de vehículos son un gran tema en este momento", dice Thomas Machauer, Ingeniero Líder en Porsche Engineering. "Se intenta probar solo las diferencias entre las variantes y así encontrar el mejor compromiso entre esfuerzo y calidad".

La profesora Ina Schaefer del Instituto de Tecnología de Karlsruhe (KIT) estudia el tema. Se concentra en la generación inteligente y la priorización de casos de prueba, particularmente para variantes de software. "La combinatoria del sistema individual se potencia aún más por la combinatoria de las variantes", explica. "Por lo tanto, nos hacemos las siguientes preguntas: ¿Qué necesitas probar para cubrir bien el espacio de variantes? ¿En qué orden deben realizarse las pruebas? Si logramos probar solo las diferencias, lograremos un proceso de prueba más eficiente".

Selección inteligente de pruebas

Una tesis doctoral actual en el departamento de Schaefer trata sobre las pruebas inteligentes de diferentes variantes de vehículos cuando se transmite una actualización por aire. "Las actualizaciones complican todo porque los vehículos en el campo tienen versiones de software muy diferentes", dice Schaefer. "En el pasado, el software solo se probaba al inicio de la producción en serie. En el futuro,

tendrás que volver a probar con cada actualización, lo que aumentará enormemente el trabajo involucrado". Esto también plantea preguntas completamente nuevas: ¿Qué necesitas probar en un vehículo individual para asegurarte de que todo sea seguro? ¿Qué efecto tiene la actualización en los componentes del vehículo? Para responder a estas preguntas, el equipo de Schaefer ha desarrollado una herramienta prototipo que examina el espacio de variantes y proporciona una lista de configuraciones a probar. Otras herramientas sugieren la mejor secuencia de pruebas.

Pero incluso la prueba más inteligente no puede cubrir todas las combinaciones matemáticamente posibles de valores de entrada y salida en un software dado. Es por eso que otros científicos están trabajando en la verificación del código. "Lo que los científicos de la computación quieren decir con eso es prueba", explica el profesor Ralf Reussner del KIT. "Quieren demostrar matemáticamente que un programa hace exactamente lo que prescriben las especificaciones. Este enfoque ya se está utilizando en áreas particularmente críticas para la seguridad de la aviación, y fabricantes de automóviles individuales ya lo han estudiado como parte de proyectos de investigación".

Lo que inicialmente suena prometedor, sin embargo, a menudo alcanza límites teóricos. Hay problemas matemáticos, por ejemplo, cuya verdad o falsedad no se puede determinar automáticamente, por ejemplo, declaraciones verdaderas para las que no existe prueba. Lo que eso significa para el software: Ciertas propiedades no se pueden probar para todos los programas de computadora con un algoritmo, por ejemplo, si un programa se detiene o se queda atascado en un bucle infinito. "En general, por lo tanto, nunca podremos construir una herramienta de verificación que pueda tomar cualquier código de programa dado y probar automáticamente que está libre de errores", dice Reussner. Sin embargo, los científicos a menudo tienen suerte y de hecho encuentran automáticamente pruebas de corrección. Y si eso no es posible, se utilizan herramientas interactivas en las que el humano interviene en el proceso cuando es necesario. "En la práctica, sin embargo, estas herramientas hacen una cantidad asombrosa de trabajo automáticamente", dice Reussner.

Especificación mediante especificaciones

Otro problema en la verificación de software: Si las especificaciones mismas son incorrectas, la prueba también carece de valor. Para evitar esto, los informáticos han desarrollado sus propios lenguajes de especificación que se asemejan a los lenguajes de programación. "Se pueden usar para describir relaciones lógicas, pero también proposiciones temporales", dice el colega de Reussner en el KIT, el profesor Bernhard Beckert. "Sin embargo, esto implica un gran esfuerzo: la especificación precisa del software hoy en día toma aproximadamente el mismo tiempo que escribir el código. Sin embargo, la descripción simple de que el software no contiene errores típicos como una división por cero es menos compleja". Lo más fácil sería si la especificación precisa pudiera derivarse automáticamente de las especificaciones del software, lo que, en opinión de Beckert, podría ser posible en el futuro con la ayuda de la inteligencia artificial: "Los modelos de lenguaje grandes como ChatGPT podrían ser capaces de hacer eso".

Sin embargo, la inteligencia artificial también podría buscar independientemente errores de programa,

por ejemplo, como un paso complementario a la verificación formal. Este enfoque, denominado 'detección neural de errores', está siendo estudiado por el grupo de trabajo de la profesora Heike Wehrheim en la Universidad de Oldenburg. "Puedes entrenar una IA para encontrar errores con la ayuda de programas correctos e incorrectos", dice Wehrheim. "Todavía es temprano para el tema, pero está en auge en este momento y está siendo estudiado por muchos investigadores". Hasta ahora, sin embargo, el enfoque solo es adecuado para programas pequeños o funciones individuales simples.
"Definitivamente es posible un mejor software, incluso si nunca será 100 por ciento libre de errores", dice la experta del KIT Schaefer, resumiendo la situación actual. "Pero con buenos procesos y procedimientos innovadores, podemos mejorar constantemente la calidad. Nunca me quedaré sin cosas que estudiar en este campo".

Información

Texto publicado por primera vez en la revista Porsche Engineering, número 1/2024.

Texto: Christian Buck

MEDIA ENQUIRIES

Elizabeth Solís

Public Relations and Press
Porsche Latin America
+1 (770) 290 8305
elizabeth.solis@porschelatinamerica.com

Link Collection

Link to this article
https://newsroom.porsche.com/es/2024/tecnologia/pla-porsche-engineering-software-error-deteccion-37269.html

Media Package
https://pmdb.porsche.de/newsroomzips/c5d92740-cfa5-424e-a324-cc5ce1c5653f.zip