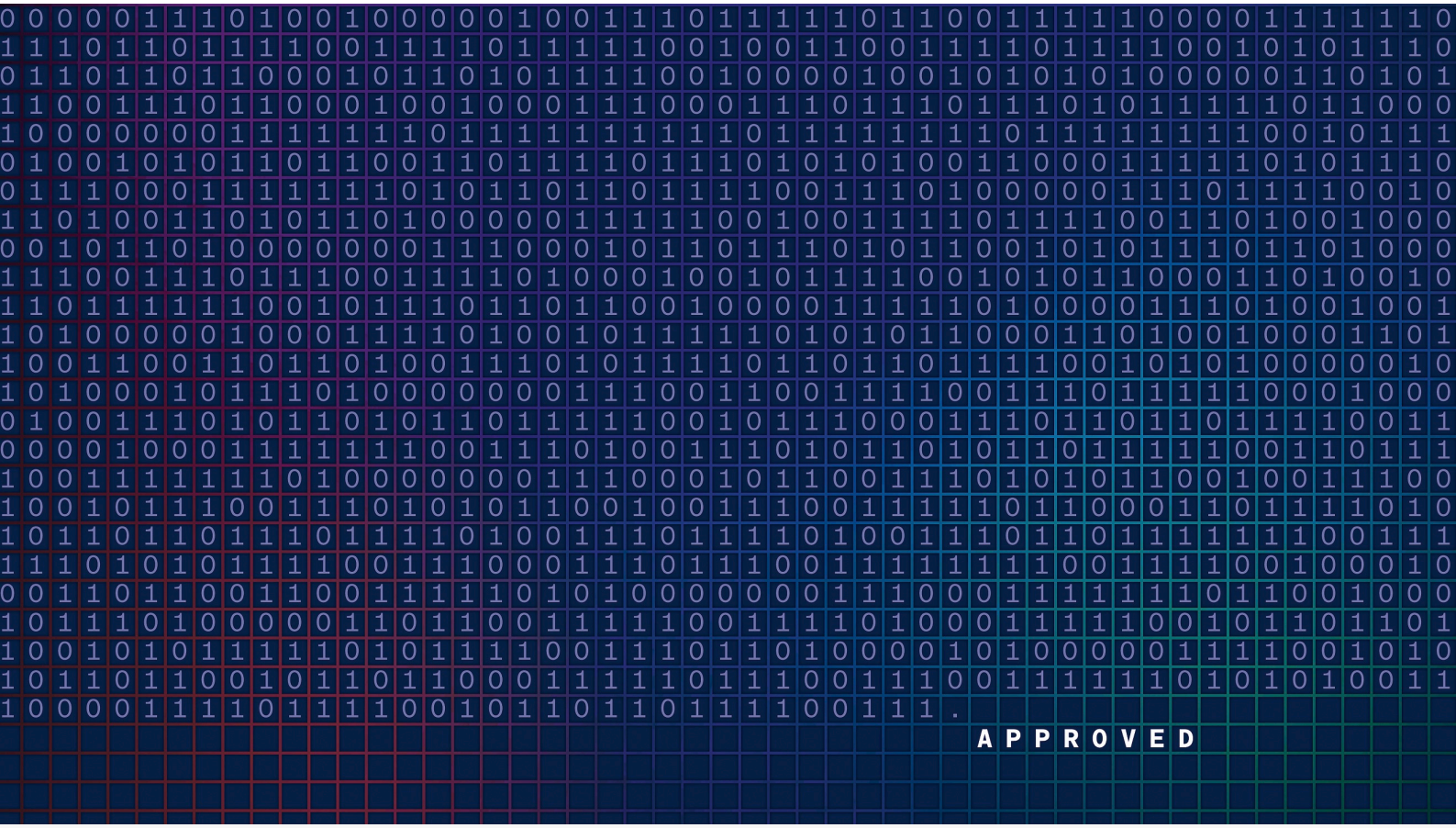

Secure Software: Putting code to the test

10/09/2024 Software plays a key role in modern life, but also in vehicles. For this reason, OEMs and suppliers use proven methods and tools to identify errors in programs as early as possible. Researchers are already working on new approaches, including ones based on artificial intelligence.

Today, vehicles are mobile computers with a network of up to 100 electronic control units that control the engine and battery functions, monitor the air conditioning system, and control the infotainment system. And more and more intelligent features such as adaptive cruise control and automated driving functions are being added. All this is only possible with the help of complex software.

As the complexity of the software increases, so too does the work required by OEMs and suppliers to avoid errors and thus ensure high software quality. On the one hand, they use the industry's standard processes for orientation—particularly Automotive SPICE (ASPICE) and ISO 26262—as well as in-house coding and quality guidelines that, for example, prohibit the use of error-prone functions in certain programming languages. With its development processes, Porsche Engineering reliably and reproducibly achieves ASPICE Level 2. This not only corresponds to the current state of the art and

constitutes the essential basis for approval in the vehicle, but also assures customers that errors and non-conformities can be detected and remedied at an early stage.

In software development, Porsche Engineering applies the V model (see illustration): On the left, from top to bottom, you will find the steps System requirements, System architecture, Software requirements and Software architecture. At the base of the V is the software design, which is followed on the right-hand side from bottom to top by the steps of unit tests, integration tests and qualification tests as well as acceptance test and use. “Every step on the left matches a test step on the right,” explains Stefan Rathgeber, Director Software High-Voltage Systemsat Porsche Engineering. “In unit tests, for example, we test the smallest unit at the functional level, component tests then follow one level above that.” For all levels, there are test catalogs that factor in all variants.

A dedicated team of quality managers at Porsche Engineering checks whether all process steps are adhered to and documented during software development. They constantly conduct reviews to find problems as early as possible, because the effort required for troubleshooting and remediation rises sharply in later development phases. The impossibility of testing every possible constellation, for example, means that certain unfortunate combinations can cause problems.

The numerous vehicle variants, equipment versions and software updates after delivery of the vehicles pose a special challenge. “Vehicle variants are a big topic at the moment,” says Thomas Machauer, Lead Engineer at Porsche Engineering. “You try to test only the differences between the variants and thereby find the best compromise between effort and quality.”

Professor Ina Schaefer of the Karlsruhe Institute of Technology (KIT) studies the subject. She concentrates on the intelligent generation and prioritization of test cases, particularly for software variants. “The combinatorics of the individual system are further potentiated by the combinatorics of the variants,” she explains. “We therefore ask ourselves the following questions: What do you need to test in order to cover the variant space well? In what order should the tests be carried out? If we succeed in only testing the differences, we will achieve a more efficient test process.”

Intelligent Test Selection

A current doctoral dissertation in Schaefer’s department deals with the intelligent testing of different vehicle variants when an update is transmitted over-the-air. “Updates make everything more complicated because the vehicles in the field have very different software versions,” says Schaefer. “In the past, the software was only tested at the start of series production. In the future, you will have to re-test with every update, which will greatly increase the work involved.” This also raises completely new questions: What do you need to test on an individual vehicle to make sure everything is safe? What effect does the update have on the vehicle’s components? To answer these questions, Schaefer’s team has developed a prototype tool that examines the variant space and provides a list of configurations to be tested. Other tools suggest the best test sequence.

But even the most intelligent test cannot cover all mathematically possible combinations of input and output values in a given software. That's why other scientists are working on the verification of code. "What computer scientists mean by that is proof," explains Professor Ralf Reussner of KIT. "They want to show mathematically that a program does exactly what the specifications prescribe. This approach is already being used in particularly safety-critical areas of aviation, and individual car manufacturers have already studied it as part of research projects."

What initially sounds promising, however, often reaches theoretical limits. There are mathematical problems, for example, whose truth or falsehood cannot be determined automatically—for example, true statements for which no proof exists. What that means for software: Certain properties cannot be tested for all computer programs with an algorithm—for example, whether a program stops or gets stuck in an endless loop. "In general, we will therefore never be able to build a verification tool that can take any given program code and fully automatically prove that it is error-free," says Reussner. However, the scientists are often lucky and do indeed automatically find proof of correctness. And if that is not possible, interactive tools are used in which the human intervenes in the process when necessary. "In practice, however, these tools do an astonishing amount automatically," says Reussner.

Specification by way of Specifications

Another problem in software verification: If the specifications themselves are incorrect, the proof is also worthless. To prevent this, computer scientists have developed their own specification languages that resemble programming languages. "They can be used to describe logical relationships, but also temporal propositions," says Reussner's KIT colleague Professor Bernhard Beckert. "This involves a great deal of effort, however: The precise specification of software today takes about the same time as writing the code. However, the simple description that the software does not contain typical errors such as a division by zero is less complex." The easiest thing would be if the precise specification could be automatically derived from the software specifications, which, in Beckert's opinion, could be possible in the future with the help of artificial intelligence: "Large language models such as ChatGPT might be able to do that."

However, artificial intelligence could also independently search for program errors—for example, as a complementary step to formal verification. This approach, dubbed 'neural bug detection', is being studied by the working group of Professor Heike Wehrheim at the University of Oldenburg. "You can train an AI to find errors with the help of correct and incorrect programs," says Wehrheim. "It's still early days for the subject, but it's booming at the moment and is being studied by many researchers." So far, however, the approach is only suitable for small programs or simple, individual functions. "Better software is definitely possible, even if it will never be 100 percent error-free," says KIT expert Schaefer, summing up the current situation. "But with good processes and innovative procedures, we can consistently improve the quality. I'll never run out of things to study in this field."

Info

Text first published in Porsche Engineering Magazine, issue 1/2024.

Text: Christian Buck

MEDIA ENQUIRIES

Sandro Kälin

Head of Communications Porsche Schweiz AG
+41 41 487 91 16
sandro.kaelin@porsche.ch

Link Collection

Link to this article
https://newsroom.porsche.com/fr_CH/2024/innovation/porsche-engineering-software-error-detection-37262.html

Media Package
https://pmdb.porsche.de/newsroomzips/4ebd7106-1a84-438c-9844-4f06f0f78420.zip