CASE STUDY

SuperTest

Helping TrustInSoft guarantee its customers 100% bug-free source code

SolidSands

Introduction

Software development tool company TrustInSoft, which serves the international aeronautics, telecommunications, industrial IoT, and automotive industries via its offices in Paris (France) and San Francisco (CA, USA), is not a typical customer for Solid Sands' SuperTest compiler test and validation suite. The company's core product – TrustInSoft Analyzer – is an advanced source code analyzer that verifies the correctness of a C or C++ program before it gets anywhere near a compiler. So for a company that's not particularly interested in compiler development or validation, what is TrustInSoft using SuperTest for? The answer lies in the need for ISO 26262 functional safety certification from one end of the toolchain to the other – from source code to binary.

TrustInSoft Analyzer

As far as compilation is concerned, TrustInSoft Analyzer only requires minimal information about the compiler, such as whether it's a GCC-like target and whether it's compiling for a 32- or 64-bit architecture.

TrustInSoft Analyzer

"For the analysis, we never compile anything. So while some compiler-related information is needed to parameterize the analyzer, we assume that when the compiler is eventually invoked, it will work as expected. So, it is up to the customer to select a compiler that implements the C or C++ standard correctly," says Monate.

Trustinsoft Analyzer

"Of course, they should validate that their compiler works correctly using a test suite such as SuperTest, because if the compiler incorrectly translates their source code, any property that we prove mathematically on the source code is not guaranteed to transfer to the executable. Note that even if a compiler is perfectly conformant to standards, compiling code that contains undefined behavior will generate an executable that makes no sense and may contain serious

and unpredictable safety and security issues. That is why you need to use TrustInSoft Analyzer and SuperTest."

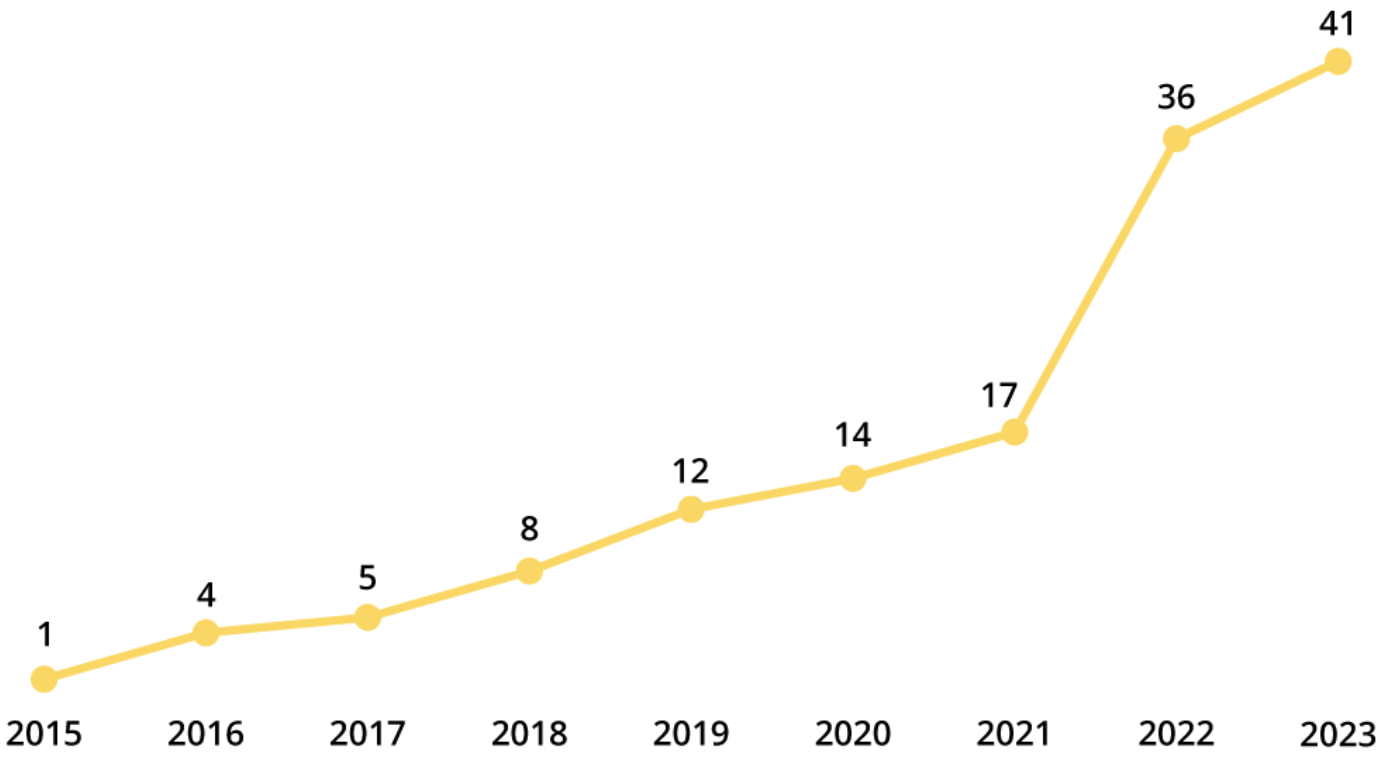
- To guarantee that the source codeexecutable have the same semantics.
- The most important thing is that the analyzer and the compiler interpret the C or C++ language precisely as defined in the relevant standard.
- For TrustInSoft, SuperTest is one of the tools it uses to check TrustInSoft Analyzer's compliance.

"Using SuperTest we identified a few subtle parts of the standard that were not yet fully supported by TrustInSoft Analyzer and are now implemented and qualified. We were also able to identify some minor discrepancies in the way Solid Sands and our own engineers interpreted the C and C++ standards, which made it a highly rewarding collaboration, and one that we hope to continue," says Monate. "What really matters to us is that SuperTest helped us to extract a coverage matrix of the standard and to present that as evidence of the quality of our tooling – that we never say something incorrect about the code – based on a completely independent test suite that is carefully designed to check all possible corners."

1. To guarantee that the source codeexecutable have the same semantics.
2. The most important thing is that the analyzer and the compiler interpret the C or C++ language precisely as defined in the relevant standard.
3. For TrustInSoft, SuperTest is one of the tools it uses to check TrustInSoft Analyzer's compliance.

For software developers of safety-critical systems who need end-to-end ISO 26262 qualification of their toolchain, using a combination of TrustInSoft Analyzer and SuperTest to validate their source code and ensure correct compilation is a powerful solution that fits seamlessly into continuous integration environments.

Figure 1: Active eBPF Landscape Projects by Year



Welcome to the Open Era of Computing!

Join us and see how open technical collaboration along with the support of many RISC-V programs can accelerate your business.

riscv.org /membership/ | info@riscv.org