Volvo Cars: More Drive for Volvo Automotive Software

Volvo Cars Corporation (VCC) is one of the most well-known and respected car brands in the world. The company has earned quite a reputation for providing the latest in technology and safety to global customers. In the VCC development center on the outskirts of Gothenburg, Sweden, a staff of thousands of engineers create technology for next-generation passenger cars. The focus of Volvo Cars R&D activities, regardless of whether they apply to electrical, combustion or hybrid systems, is increasing the performance of powertrain components (engine, transmission, gearbox). Hence, the control software for powertrain applications plays a vital role.

Increasing the Magnitude of Volvo Cars in-house Software Development

Volvo Cars Torslanda Plant 2004
Fig. 1: Volvo Cars Torslanda Plant 2004

Volvo Cars Corporation (VCC) introduced model-based in-house software development mainly for engine and drivetrain control approximately 20 years ago. The benefit of model-based development is the fact that domain expertise of field engineers can be combined with automated high-quality software creation. The functional models of the future control software are created by so called functional developers who are experts in their respective fields. The functional developers focus on the mechatronic system and its functional behavior. The software generation toolchain backs up the functional designers to produce easily readable, testable and maintainable software models, including the controller code. The VCC propulsion and powertrain division started off in 2002 with a rather modest team of roughly 10 software developers and a straightforward basic process. Manual Bat files were used for code generation and the software build process at that time.

Over the past few years, more and more functionality has been added to the controller software due to growing demands from a manifold of different vehicle projects such as different drivetrain variants and extended combustion functionality. The complexity and sheer volume of required functionality also increased exponentially. This resulted in more and more validation and verification steps needing to be inserted into the development chain. The team of developers has recently grown to more than 100 domain experts and functional developers. With this increase in team size, the main challenge has been to integrate the different software components into a working system and to tackle the number of faults that had to be tracked in the software repository. Every single release of a new software product became more and more challenging. At the time, the Apache SVN was used as the backbone for the management of the different software versions under development.

Volvo Cars Shifts to Continuous Integration

In 2014, the upper technical management of Volvo Cars decided to introduce best practices, tools and approaches that originated in the advanced software industry. In the present-day software industry, Continuous Integration (CI) and Nightly Builds are used to ensure that different components of a software system can be integrated into a full-blown software system easily, and that existing legacy functionality will not be corrupted by new features. In addition, individual developers receive swift feedback on their contribution to the software system, and resource-intense test processes are carried out on a larger server overnight. By using regression tests, Nightly Builds and Continuous Integration, the “integration nightmare” that usually unfolds when people wait for release day to merge their parts into the release branch can be successfully avoided. These vital principles of CI were introduced to VCC’s entire line of electronic control modules.

An additional development was introduced in 2018, namely a new CI system based on ZUUL from OpenStack. ZUUL's motto is: 'Don't merge broken code.' Therefore, only tested and analyzed code is allowed to merge on master. ZUUL has many benefits, but one of the most significant is the speculative merge step, which allows for large parallelization. ZUUL is based on the change management system, GIT, a fast, scalable, distributed revision control system. Moreover, the GIT add-on, Gerrit, is used to provide for collaborative code reviews. Hence, ZUUL is the brain and listens to events in Gerrit and sends jobs to Jenkins. The newly introduced CI-based development chain is referred to as the Volvo Cars ECM CI chain. From implementation of the software onwards, every single new commit automatically triggers a set of integration and regression tests. The changes committed by the developers are validated by running automated tests against the build. This new development approach fully adheres to the concept of frontloading quality assurance processes. In addition, the key objective to reduce the number of post integration issues to the absolute minimum was achieved fairly quickly.

MES Tools Deployment at Volvo Cars

Fig. 2: Current & future VOLVO CI toolchain for ECM software creation

So far, the focus at VCC has been on backbone technology for version management, code generation, and the quality of the final code. However, in model-based design, the quality of the final code depends on the set up of the software models that provide the basis for automated code generation. The functional model represents the most central artifact and thorough validation of the software model is key for the final code quality results. For this reason, Volvo Cars Powertrain went in search of advanced, practical model quality metrics and quality support tools in order to close the gap.

In 2017, VCC evaluated a set of model quality tools designed by Model Engineering Solutions. VCC decided to gradually introduce the guideline checker, MES Model Examiner® (MXAM) and the model structure checker, MES M-XRAY® (MXRAY) (which is a fully integrated part of MXAM since 2020), into the existing CI toolchain. The MES tools compute quality metrics and analyze guideline compliance for each and every model passing through the CI toolchain. The collected metrics are used as quality gates in the process and a successful commit of a software model is only possible for the developer if the quality threshold has not been exceeded. The results of the analysis are reported back to the individual functional designer, who is fully accountable for their module change. Figures 2 and 3 show the aggregated results of a check gate and the way they are reported back. By clicking on the link, more detailed results pop up and efficiently support the developer in fixing the model.

SW METRICS Generated From Reports

With the support of MES experts, the VCC propulsion and powertrain division surveyed and selected fitting model quality guidelines for their projects. As a result, modeling guideline sets for regular development projects and for highly safety-critical projects at Volvo Cars propulsion were compiled and tested based on the specific modeling style required in this area.

Fig. 3: Feedback table on model complexity

In the ECM CI chain, each and every commit triggers an automated set of tests, including the complexity analysis and the modeling guideline compliance check by MXAM. The results, such as modeling guideline compliance, complexity, unit test results, and code review results, are provided to the developer in the form of summarized feedback with links to detailed reports. The overall objective is to set up a learning cycle that stimulates early and self-accountable error prevention in the process.

The ECM CI is designed to encompass all the process steps from the first model commit to the review of the autogenerated code. It is also able to incorporate configuration settings that influence the checks, the test configuration, and the workflow in the toolchain.

Achieved Results with the CI Driven Development Chain

Fig. 4: Andreas Wikerstål, Ph.D.

Although the introduction of the new CI driven toolchain took some time and effort, it has already shown numerous benefits. First of all, it significantly changed collaboration across software teams. The main objective of frontloading of validation and quality assurance was achieved, productivity improved and the load of tedious manual work decreased significantly. However, the biggest improvement came from the MES tools, since reduced complexity helps to understand functionality better, which in turn means faster and more robust tuning of Powertrain parameters (calibration). Complexity reduction also results in the refactoring of models and that unit tests are easier to compose. According to Andreas Wikerstål, PhD, ECM SW architect at Volvo Cars propulsion, the ECM CI chain “provides us with the necessary toolset to master the ever-increasing volume of software-based functionality and to maintain the software robustness and quality at the same time. On a functional model level, the MES tools allow us to measure and increase quality at an early stage of the process. This saves us a lot of time and money at the critical final stages of code creation and integration."

The Way Forward to a Fully Integrated CI chain

Fig. 5: Johannes Foufas

In the years to come, Volvo Cars aims to gradually expand its model-based design process to include all development projects. More and more users will use the toolchain in which MXAM plays an integral role. Volvo Cars is fully committed to further expanding the ECM CI toolchain and to fully automate the remaining semi-automatic steps. Johannes Foufas, Product Owner CI/CD at Volvo Cars propulsion asserts that: “We are currently investigating to move towards ZUUL v3, since this product will give us in-repo configuration, live configuration changes, native support for multi-node jobs and Ansible job content. We will also work hard to remove all manual steps and only keep peer review, product owner approval and release based on GIT tags. All other steps should be automated if possible.” In the future, the Software CI toolchain set-up will automatically trigger the next step in the chain without any user interference. More elaborate modeling guidelines will be introduced step by step to frontload even further and to gradually improve model guideline compliance.