PARASOFT

CASE STUDY, A PARTNER SHOWCASE

Boosting Performance in Modern Development Workflows With IAR Build Tools for Linux

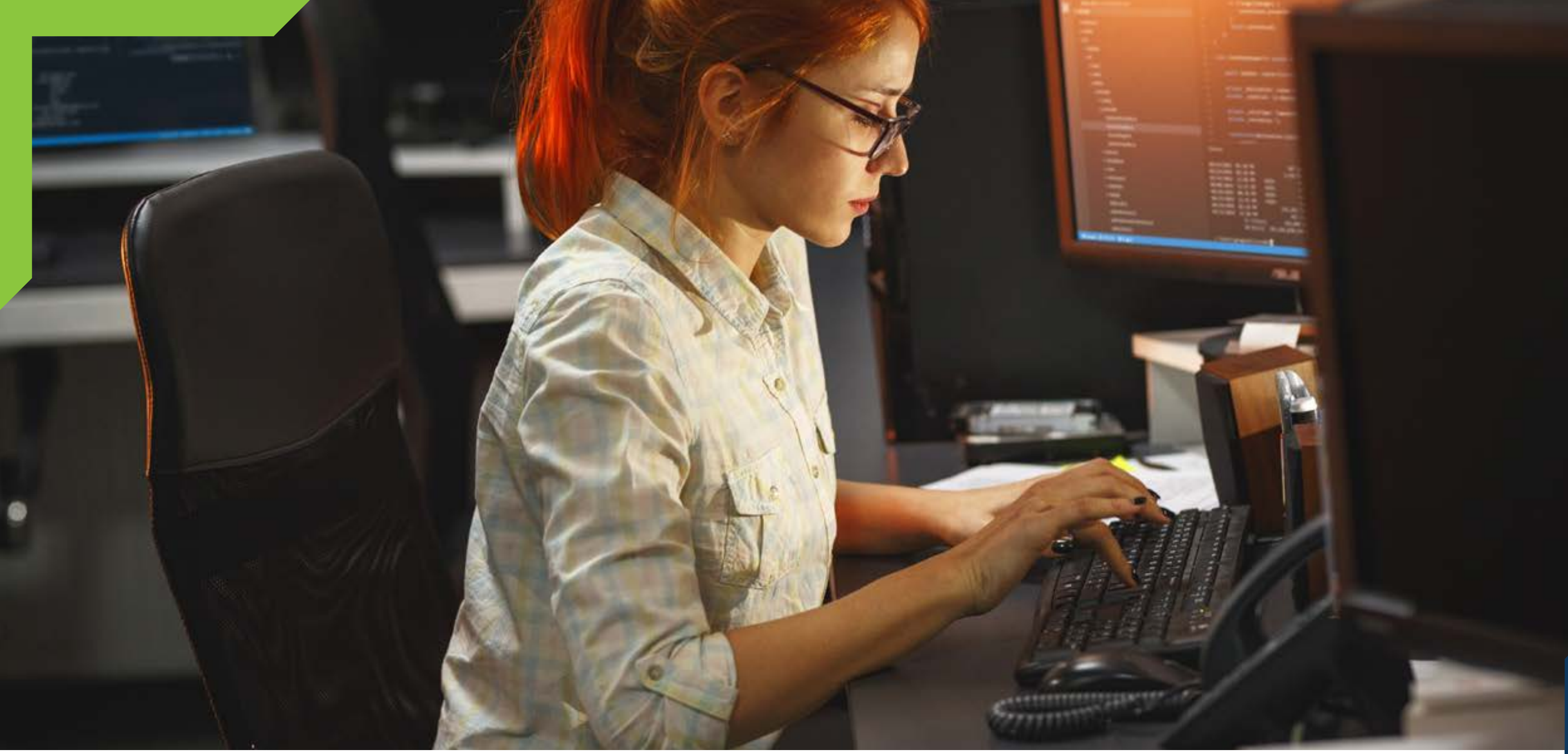
IAR Systems, founded in 1983 in Uppsala, Sweden, has been supplying best-in-class development tools for embedded systems for over 30 years. This leading software and services provider for embedded development specializes in C and C++ compilers, debuggers, and other tools for developing and debugging firmware for 8, 16, 32-bit, and modern 64-bit processors. Their tools are prevalent in embedded, consumer, and industrial marketplaces where development engineers likely use one of their products more than 30 times each day.

Traditional development workflows are prone to bottlenecks. IAR Systems takes a DevOps approach to aid teams in removing bottlenecks and accelerate modern development workflows. IAR Systems also supplies Linux-based build tools to facilitate a wide variety of build environments and continuous integration and delivery (CI/CD) pipelines.

INSTANT
Code Feedback

IMPROVED
Code Quality

FASTER
Time-to-Market



CHALLENGES

While the overall approach is to move embedded software development teams to DevOps and embrace continuous integration, testing, and delivery, it does come with challenges. There are the three primary difficulties that embedded developers face when adopting a modern development environment and implementing CI pipelines within the limitations of their current workflow.

1. Getting feedback on code.
2. Getting feedback on code in a timely manner—often and early.
3. Poor source quality slowing downstream reviews and testing.

To increase development efficiency and product quality, companies are looking for solutions to automate the entire CI/CD pipeline. Moving to a CI/CD workflow is one thing. Automating it is another.

APPROACH

By automating the CI/CD workflow, developers get almost instant feedback on code. To seamlessly blend automation into their modern workflows, developers need build tools and automated software testing technologies.

IAR Systems' tools integrate into practically any CI build system as part of a DevOps pipeline. Along with CI/CD and automation, Linux development platforms, Git-based workflows with a dynamic approach to branching, and dockerized toolchains are becoming critical components for teams to incorporate into their environments.

A collaborative set of tools is integral to a seamless CI/CD pipeline. Here's an example:

- Orchestration
- Version control
- Build, analysis, and test tools
- Containerized development environment

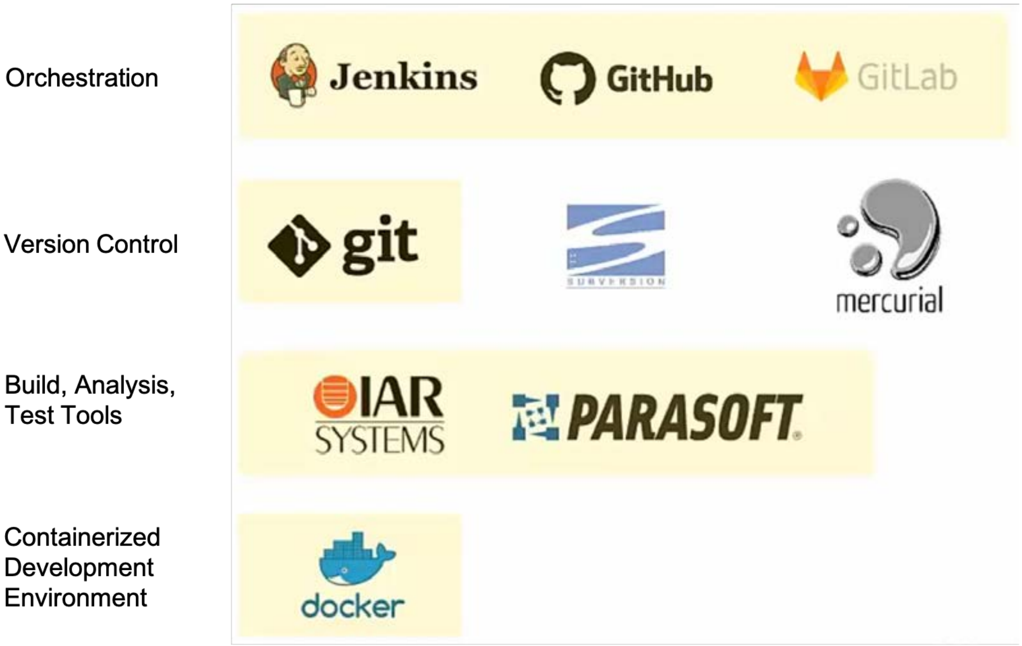


Figure 1: Tool components of a continuous integration environment. Source: IAR Systems

SOLUTION

Embedded development teams are looking for a solution that works with their IAR tools like compilers. The partnership between IAR Systems and Parasoft enables integration of modern development workflows, primarily automating the CI/CD pipeline.

> *“IAR tools for Linux integrate well with Parasoft. The combination speeds up development and raises the quality for your organization.”*
>
> —Felipe Torrezan, Field Applications Engineer, IAR Systems

ADOPTING DEVOPS FOR EMBEDDED SOFTWARE DEVELOPMENT

Following a DevOps methodology—or life cycle—ensures that teams aren't working in silos. With the integration of IAR Systems and Parasoft, development and operations can communicate during the entire SLDC, including while:

- Developing software.
- Finding and fixing bugs.
- Adding new features.

This back and forth communication ensures that teams can collaborate without anything falling through cracks.

While DevOps is suitable for embedded software development, it often has constraints that other types of development don't, such as limited processing power and memory. The target hardware may impact your toolchain choice and the types of testing that are possible, but the basic components of the CI/CD pipeline remain the same.

The best way to evolve a legacy build system to CI/CD is to start with simple objectives like:

- Integrate a build system with an orchestrator.
- Start analyzing code quality with static analysis.
- Adopt automated unit testing.

A critical aspect of continuous integration pipelines is toolchains that support a hands-off, command line-driven operation, most often on Linux server hosts. This is why IAR Systems built tools for Linux. They're important in helping embedded developers move to a CI/CD workflow approach.

Empowering Software Developers With Build Tools for Linux

Software developers who build safety-critical applications to work directly on their Linux host environment can eliminate toolchain version management with IAR Build Tools for Linux. These use the leading build tools from IAR Embedded Workbench and are equivalent alternatives to build tools on Windows. They can integrate with other Linux build tools and orchestrators like Jenkins and GitLab. IAR Systems also provide a build utility called IAR Build.

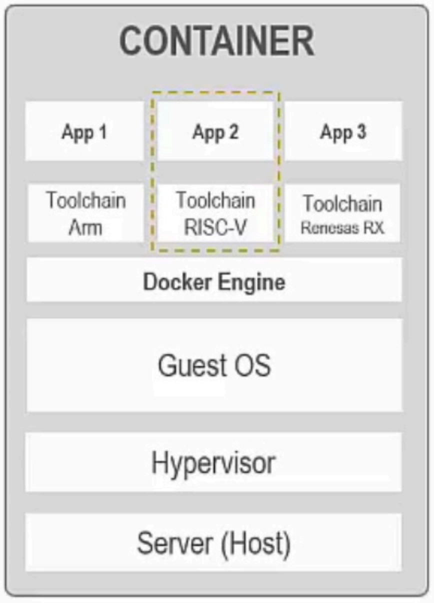


Figure 2:
Container architecture with IAR Systems tool chains. Source: IAR Systems

Boosting Productivity With Docker Container Development Environments

Containerized development environments are becoming popular in software organizations because they provide an easy-to-deploy, version controlled tool suite that's identical for each developer. Development teams can boost flexibility and productivity by enabling containerized deployments of the IAR Build Tools for Linux for use with Parasoft C/C++test.

IAR Systems and Parasoft C/C++test support containerized environments. IAR provides premade Docker images for customers to adopt and customize. Developers can reap the benefits of running these pre-packaged containers when performing static analysis and unit testing.

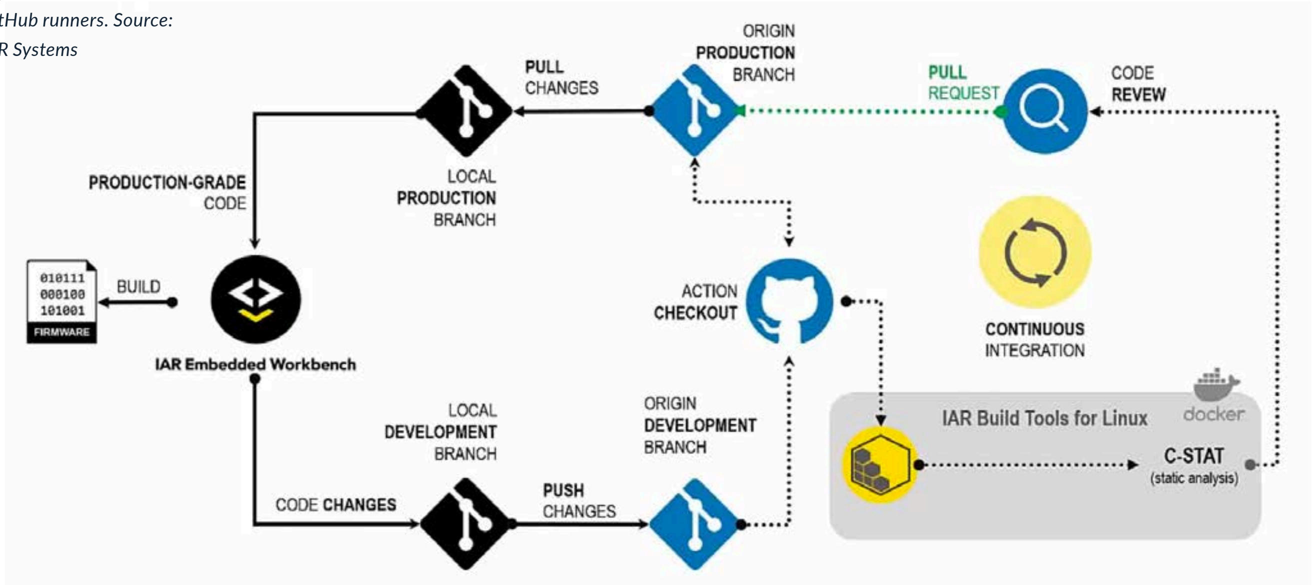
Integrating With Typical CI Workflows

IAR Systems' tools integrate completely with typical CI workflows as illustrated below using GitHub or Jenkins runners. The integration starts from the origin production branch where a developer usually pulls the change and the production-grade code into the IAR embedded workbench IDE.

When developers work on new features or bug fixes, they can test on a target device. Once satisfied with the results, they take the code change and commit to the local development branch, then push the change back to the origin, which triggers a GitHub action.

This action starts a Docker container to build the project and test it. The output results are completely automated and put to use in code inspections and reviews. If approved, the production branch gets updated, and the cycle repeats itself as many times as required.

Figure 3:
Typical CI workflow with GitHub runners. Source: IAR Systems



RESULTS

Transitioning to an automated CI/CD from a traditional workflow helps embedded developers improve development efficiency and product quality. IAR Systems' integration with CI/CD workflows gives embedded developers:

- Instant feedback on code
- Better overall code quality
- Focused manual testing efforts and code reviews on high-quality code
- Better products out the door faster
- Significantly better build and analysis processing times

Let's revisit those three primary challenges highlighted at the top of this case study. The combination of IAR Build Tools for Linux and Parasoft C/C++test automate the entire CI/CD pipeline and addresses those challenges in the following ways:

1. **Getting feedback on code.** With an automated workflow, development engineers write and test code, and when they're satisfied with it, commit the code into the software system. That code gets automatically built into the rest of the software system. It then gets compiled and executed for static analysis and unit code testing, resulting in feedback on the latest core software.

2. **Getting feedback on code in a timely manner—often and early.** When developers get feedback is as important as how they get it. Ideally, developers need to understand issues in their code as they create it. This requires a toolchain that reacts quickly to changes and provides feedback in a timely, informative, useful way. Modern continuous integration pipelines serve this very purpose.

3. **Poor source quality slowing downstream reviews and testing.** An advantage of detecting and fixing bugs in code early is the impact downstream. Fewer bugs mean a better product. Detecting and fixing bugs early also saves time and money during development.

By implementing a modern workflow and automating the entire CI/CD pipeline, teams reach a point where they no longer find flaws and all tests are passing. This entire testing and delivery process speeds up time to market with a clean, solid system.

TAKE THE NEXT STEP

Learn how your safety-critical software development team can simplify automated Linux builds and testing to accelerate its CI/CD pipeline for delivery of continuous compliance and quality. Watch the on-demand webinar.

ABOUT PARASOFT

Parasoft helps organizations continuously deliver quality software with its market-proven, integrated suite of automated software testing tools. Supporting the embedded, enterprise, and IoT markets, Parasoft's technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award winning reporting and analytics dashboard delivers a centralized view of quality enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives — security, safety-critical, Agile, DevOps, and continuous testing.