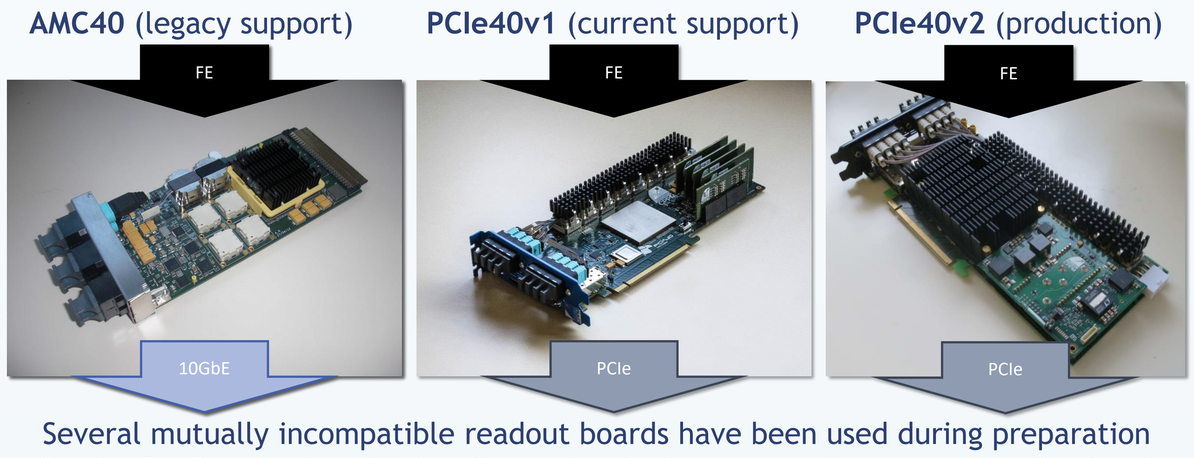
An automated pipeline for continuous integration of FPGA firmware and software for the LHCb Run3 upgrade

Paolo Durante[1], Luis Granado Cardoso[1], Joao Vitor Viana Barbosa[1], Federico Alessio[1], Guillaume Vouters[2]

[1]CERN (European Organization for Nuclear Research),
[2]LAPP (Laboratoire d'Annecy-le-Vieux de Physique des Particules)

ABSTRACT

The readout system for the current Run3 upgrade of the LHCb experiment is based on a common FPGA readout board called PCIe40. The LHCb Online group provides engineering support and development to the different LHCb sub-detectors, mainly in relation to readout board firmware, Linux software, and control systems based on the WinCC Open Architecture. Here we present the automation infrastructure that was implemented on top of GitLab CI in order to make our development cycle tighter, faster, more observable and reproducible.



Several mutually incompatible readout boards have been used during preparation for the Run3 upgrade. Additionally, each sub-detector implements specialized and mutually incompatible protocols for configuration and data acquisition. Automation becomes an indispensable tool to support such combinatorial complexity with limited resources.

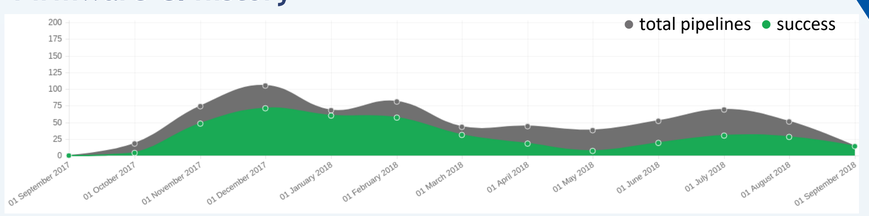
CONCLUSIONS

Over one year after introducing CI in our development cycle, we have found a well working solution fitting the very specific needs of the ongoing LHCb upgrade. As the new LHCb experiment comes together and the scale and complexity of the system increase, these tools will prove invaluable in helping the Online group fulfill its responsibilities towards the rest of the LHCb collaboration.

Software developers use traditional git commands.

Firmware developers also use git, but supplemented by ad-hoc python scripts to simplify git submodule management and submission of merge requests.

Firmware CI history



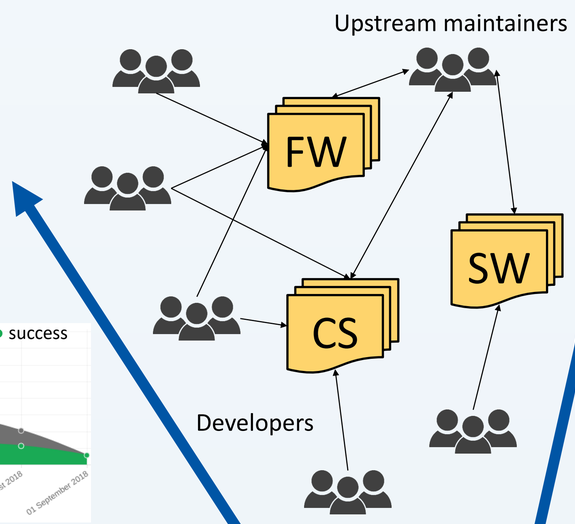
Tab.1 Firmware CI matrix

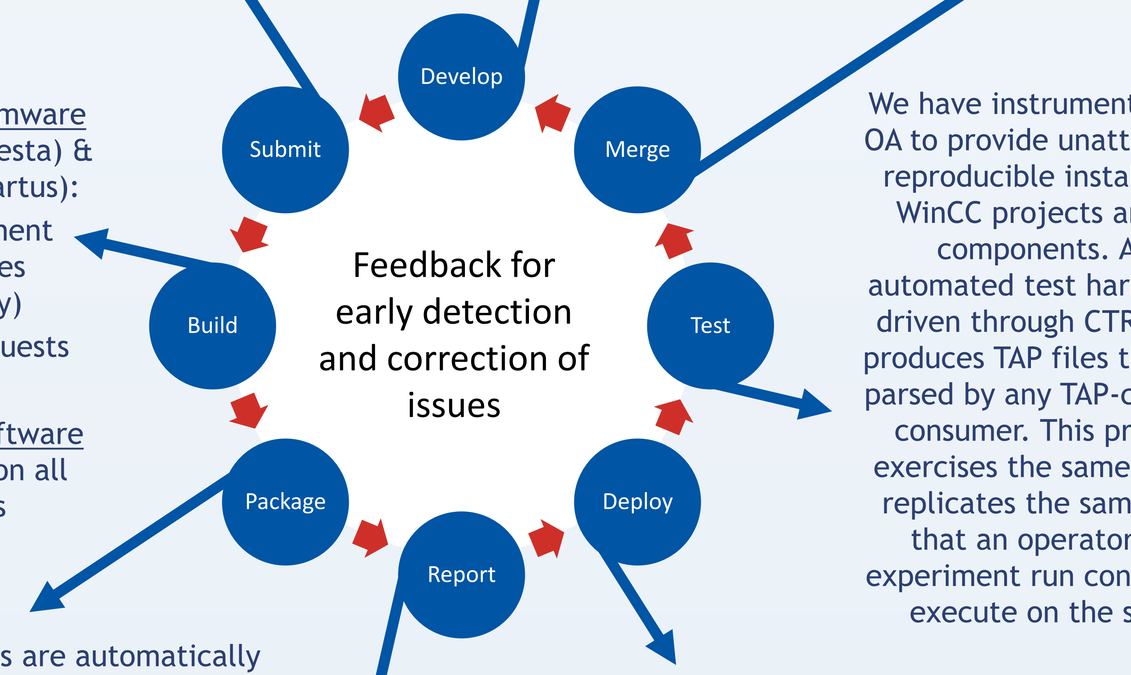
Config	AMC40	PCIe40v1	PCIe40v2
LLI	X	X	X
PRBS	X	X	
PCIe	N/A	X	X
10GbE	X	N/A	N/A
SOL40	X (MD)	X (MD)	
MD/GBT/int	X	X	
MD/GBT/ext	X	X	
MD/WB/int	X	X	
MD/WB/ext	X	X	
MD/VELO	X	X	
MD/RICH/int		X	
MD/RICH/ext		X	
MD/UT	X	X	
MD/SciFi		X	

Tab.2 Software CI matrix

OS	Config	AMC40	CCPC40	DAQ40	PCIe40
CC7	dim	X	N/A	X	X
	hwloc	X	N/A	X	X
	nodeps	X	N/A	X	X
	alldeps	X	N/A	X	X
	dkms	N/A	N/A	N/A	X
	gcc7	X	N/A	X	X
	wincc		N/A	X	X
	docs	X	N/A	X	X
SLC6 (legacy)	dim	X		X	X
	hwloc	X		X	X
	nodeps	X		X	X
	alldeps	X	X	X	X
	dkms	N/A		N/A	X
	wincc		N/A	N/A	N/A

Our FPGA synthesis flow implements an automatic caching mechanism with dependency tracking, which avoids unnecessary rebuilds and significantly reduces build times during day-to-day development (example below).





Automated firmware simulation (Questa) & synthesis (Quartus):

- development branches (nightly)
- merge requests

Automated software compilation on all branches

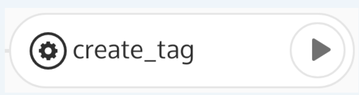
All artifacts are automatically packaged and published in RPM format, this includes:

- Software tools and libraries
- Device drivers (DKMS)
- Documentation
- JCOP components
- FPGA images (.sof & .pof)

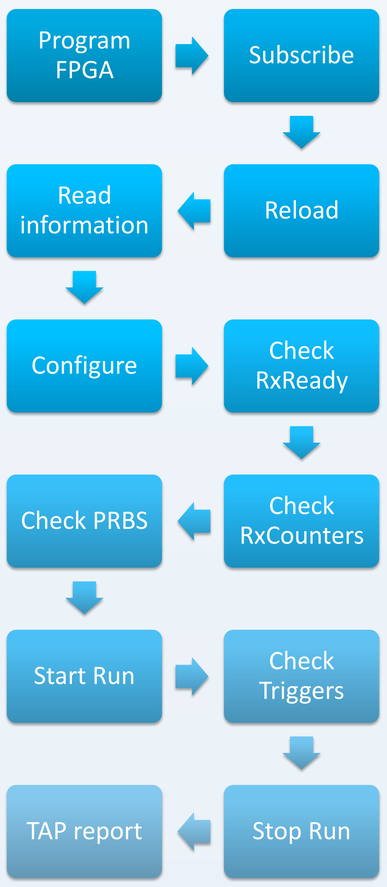
FPGA images embed configuration parameters, git tags and hashes for firmware traceability in the field.

Developers are distributed between CERN and several external institutes within the LHCb collaboration. Most repositories are either already enabled for Continuous Integration (CI) or are being migrated to it. Maintainers at CERN and LAPP (who are also developers) review code submitted for mainline inclusion and schedule releases.

Code merged and tagged in the master branch (at the maintainers' discretion) is considered production-ready and is automatically published in dedicated RPM repositories (distinct from the staging repositories).



We have instrumented WinCC OA to provide unattended and reproducible installation of WinCC projects and JCOP components. A new automated test harness, fully driven through CTRL scripts, produces TAP files that can be parsed by any TAP-compatible consumer. This procedure exercises the same code and replicates the same actions that an operator or the experiment run control would execute on the system.

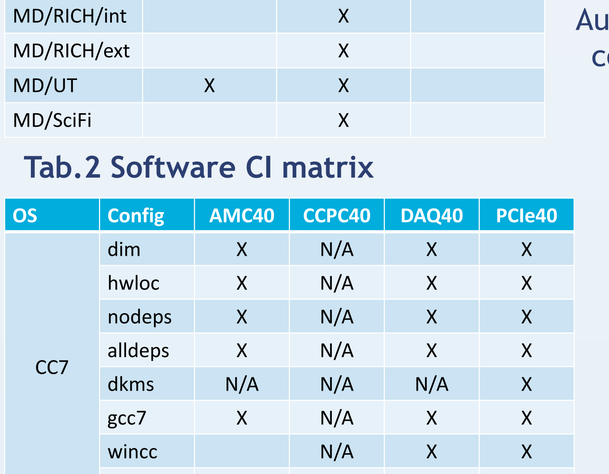


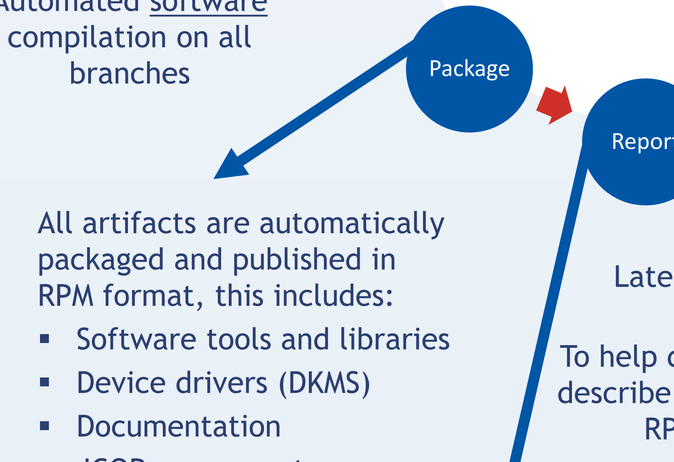
Latest developments can be automatically installed from staging repositories. Production repositories are also provided to automatically deliver tested packages.

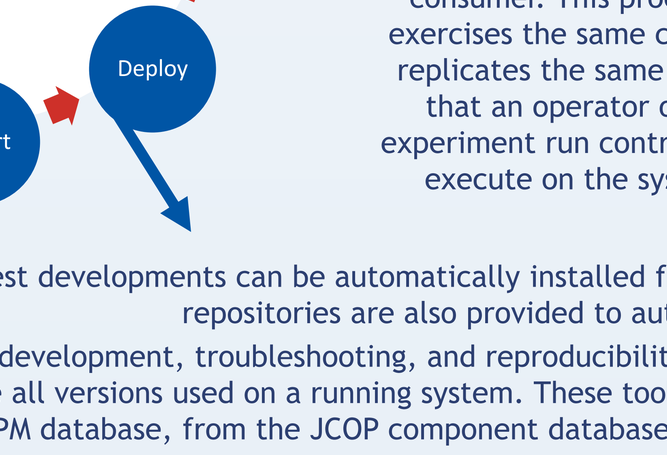
To help development, troubleshooting, and reproducibility, we provide tools to automatically describe all versions used on a running system. These tools collect information from the local RPM database, from the JCOP component database, and from the readout board itself.

A new deployment environment dedicated to larger scale integration tests is currently being commissioned at Point 8 at CERN.

In addition to creating FPGA images, after firmware compilation we generate Quality Of Results (QoR) reports in JSON format. Historical JSON data is automatically made available on the web for analysis and visualization (examples below).







Automatic JSON reports include firmware metrics such as:

- FPGA resource utilization (ALMs, BRAMs, DSPs, PLLs, HSSIs)
- System resource utilization (CPU time, wall time, memory)
- Critical path slack and TNS for all clocks and all corners

An interactive web dashboard to visualize QoR reports along different dimensions is currently under development.

paolo.durante@cern.ch