

Perl Case Study: Mobile Platform Build System

This is a case study about a project [we did in Perl](#) for one of our customers, ST-Ericsson, a global wireless platform and semiconductor supplier to top mobile device manufacturers.



The challenge

The customer needed a system to handle building of executable units for mobile platform software products. This system had to fit to the software modules structure and to their interdependencies and also to have the ability to build a certain set of items on different toolchains (compilers). The system needed to interface compilers, "make" programs, configuration management (CM) systems and debuggers. Also the business model included the delivery of (some) source code files to the end customers.

"The build system is critical for our business. Without a proper solution our whole business model will fall apart."

Customer Tools Manager

They already had developed a build system in house and also outsourced to some third party contractors, however the system was not stable. It had many bugs and frequent re-releases were done due to the build system problems. Also the build times were huge.

ST-Ericsson didn't have enough expertise to handle this problem so they decided to reach out and hire an external contractor, a Perl outsourcing company, Evozon. They reached out to us through another third party, one of our customers, who recommended our company as an ideal candidate for this job.

The solution

We started our research by conducting studies to identify an appropriate solution for ST-Ericsson business demands and to address all the issues from the build system that became very unstable.

The existing build system had configuration files where the required information for the build process was stored and also had information for the files that needed to be packed and delivered to end customers. ST-Ericsson offered the build tools chain together with the software modules so that their end users had everything that is needed for a build.

The build tool then had to parse all the software modules configuration files for gathering all the data needed for the build and based on that generate automatically a makefile, then perform the actual build operation.

Evozon fixed the major problems of the build system and found solutions to optimize the parsing of the configuration files, introduced multithreaded builds and parallel builds, reducing this way the build times. Also we found a solution to unify the build and release tools, the delivery tool was integrated in the build tool increasing efficiency. A new tool was created for installing the software platform modules, as this was done previously by delivering a zip archive containing source code files to the customers.

The implementation

The main programming language that was used for the build system implementation was Perl. The tool contained its own Perl distribution, so after installing it, the tool could be used as standalone, without the need to install additional tools. Also C++ was used for creating executable files and libraries. NMake was chosen for building projects based on commands generated from description files. This build system was essentially a command line tool, but it was possible to integrate it with an IDE (such as Eclipse) that made use of these command line interfaces. Also the build times were optimized by either building with Jom or by using multithreaded build processes.

Technologies and tools:

Perl, LIBXML, Nmake, Jom, C++, multithreading

The result

Our improvements produced a positive result for the build system, it was stabilized and made reliable. The system did not have any major issues during the development and maintenance/support. The build times were reduced with 70-80% from the initial times. The efficiency was increased by the unification of the build and release tools, providing a convenient way to deliver source code files and patches to the end customers.

The software platform tool installer that was created had simplified the installation process for the ST-Ericsson end customers. Also various build tools utilities were created to optimize this process.

This improved build system helped ST-Ericsson to develop its business, as the build process was at the very core of the mobile platform development and customer delivery activities.

"I am happy to see that you keep high quality in all aspects of your work, this makes it possible for me to give the right promises.Your work is highly appreciated, very well done and keep up the good work!!!"

Customer Project Manager

Dumitru Negrea