



## **If software is your business, Coverity is your friend**

In the spring of 2007, when EE Times still came out in print, I stumbled upon an article announcing the latest release of a static code analysis product. “Static code analysis can find bugs in software by analyzing source code, without any need to execute the program” the article claimed. (The article is still available at [http://www.eetimes.com/document.asp?doc\\_id=1304520](http://www.eetimes.com/document.asp?doc_id=1304520))

Intrigued, I checked it out and quickly realized its potential. After a short courtship by our R&D staff, Verific became a proud licensee of what was then known as Coverity Prevent Express, which I guess nowadays goes by the name Coverity Quality Adviser.

As in most quality assurance products, your biggest bang for the buck is in your initial deployment, as they show you all your booboos from the previous years. Once we had worked our way through those, and we were pretty much “Coverity clean”, it actually took us little effort over time to stay that way.

As a software IP company, and especially as one that ships source code to its licensees, this is very valuable to Verific. After all, our mistakes get multiplied many times due to our 60+ licensees and their combined 100,000 end-users. Hence we happily run Coverity and clean up a few code mistakes each month before we ship.

Now, it is not all sunshine with Coverity. Our coding rules require judicious null-pointer checks, and in many cases these are superfluous. Detecting this check is sufficient for Quality Advisor to assume the pointer could potentially be null, and it will flag subsequent dereferences as defects. These false positives are time consuming to analyze and classify, and we’ve not been able to find a good way to preempt them. But we take dealing with the false positives anytime over not finding the real ones.

Over time, some of our licensees have also become Coverity converts. We often notice because they then send us a list with all those pesky false positives mentioned above. Other than those, it’s unlikely they find any other issues.

Ah, it feels good to be ahead of the curve.

*first published on EDA Cafe, Aug 5, 2013*

