electronic design
The Authority on Emerging Technologies for Design Solutions

What's The Difference Between Ad Hoc And Formal Version Control?

Electronic Design
Marc Depret
Marc Depret, Altium
Fri, 2013-05-10 09:07

All industries struggle with time-to-revenue. Nowhere is this more evident than in the electronics industry with its constantly evolving technologies, shrinking engineering budgets, and continued pressure to deliver lower-cost products to market faster. Such trends challenge today's engineers with finding ways to decrease time-to-market while lowering both production and process cost.

One simple solution involves improving collaboration across the hardware development team and other associated engineering disciplines involved in the overall development process. Doing so ensures the development team can more effectively orchestrate and track its designs across multi-site teams.

Collaboration improvement requires engineers to implement more effective data management, and that necessitates version control. Essentially, version control can be approached in two ways: using an ad hoc solution or one that is more formal. The engineer's choice has a significant impact on the benefit to the development process that can be realized.

Ad Hoc Version Control

Version control is not a new concept. In fact, if you ask engineers today if they use version control, the answer likely will be a resounding yes. Dig a little deeper, however, and you find that what they define as version control may really just be an ad hoc solution that uses a "Save As" capability and then changes the file name to, say, version 2.

Using ad hoc version control to manage an entire design process provides the engineer with a history of final name changes. However, it fails to provide little else. It doesn't help the engineer understand what changed, when it changed, who made the change, or even why a particular change was made in the first place. Moreover, using ad hoc version control often opens the door to other potentially painful problems, such as:

- Continuing to iterate a design without telling anyone once it has been sent to the next person in the tool chain
- Struggling to keep various elements of a design in sync with one another
- Missing a release date due to being bogged down with data integrity issues rather than focusing on designing

Finding an alternate option to this ad hoc form of version control has become paramount to engineers who find themselves trying to balance the desire to meet customer expectations with the need to improve product development efficiency.

Formal Version Control

Given that time is a key pressure in product development, affecting profitability and inhibiting the ability for companies to recoup development costs, wasting time working on the wrong version of a file or dealing with an endless maze of file folders and e-mail chains has become unacceptable. Formal version control offers the ideal solution.

Related Articles

- Formal Verification Reaps Benefits Of Distributed Processing
- Board-Level Design Suite Takes On Engineering-Data Managemen
- Test Engineering Platform Enhances Data Management

With a formalized approach, version control is integrated directly into the design interface, spreading the capability across the entire design process. Doing so allows engineers to make faster, better designs to reduce time-to-revenue and eliminates missed or repeated processes and the need to outsource repetitive tasks and designs, all while reducing errors in design, purchasing, and manufacturing.

Formal version control often is integrated with other design tools such as printed-circuit board (PCB), FPGA, and embedded software design. It also may be integrated with release management automation, making the development process faster and much more efficient. A powerful differencing engine within the tool can help highlight even the smallest graphical, parametric, and connectivity changes between different versions of any design files.

Because of such functionality, engineers now can fully understand and refer back to every design change that's ever been made to any element of a design. They also can merge or combine any of the design changes in any version of a project, both logically and graphically. Moreover, engineers can implement a true version control capability across the entire design process and do so natively, directly in their engineering desktop.

Formal version control also ensures multi-site and on-site collaboration that's better coordinated. That collaboration is aided by centralization of all design elements into a repository to make sure engineers don't waste their time looking for or working on the wrong version of a design. Files then can be seamlessly shared across any medium, infrastructure, or development system. It also streamlines access and traceability of all data while maintaining absolute synchronization.

Formal version control goes a long way in removing the risks and issues associated with handling versions of data during design. However, more and more design teams also require a solution for managing release data and the design data commonly used by manufacturers, procurement and logistics, and other interested parties.

Version control can be a powerful tool for any modern electronics engineer. Look no further than the results of a 2011 study performed by the Aberdeen Group. According to the study, 61% of best-in-class companies or those that perform in the top 20% of the market use formal version control for each data element on the PCB—two and a half times more than their competitors.

Faced with such statistics, perhaps the real issue these days isn't whether or not engineers are using version

control, but why they aren't using formal version control. For those engineers looking to put their focus back on design and on trying to have a real impact on the market, it's time to do away with the status quo. Making the move to formal version control is the logical first step.

Marc Depret *is a senior product manager for smart data management technologies at Altium in Shanghai. During his more than 16 years at Altium, he has held a number of key posts in its R&D department including director of platform technologies and director of product development.*

Source URL: http://electronicdesign.com/embedded/what-s-difference-between-ad-hoc-and-formal-version-control