Enhance productivity with R&D

In the early years of Computer Aided Drafting (CAD), most systems were very generic but highly customizable. At that time, firms that invested in CAD not only paid a fortune for computer hardware and software, but also needed to pay technical support costs and hire computer support staff to run it. Many firms hired programmers to further customize the software with homegrown 'applets' - small application modules.

These generic drafting applications required so much customization that end-users met regularly to exchange productivity tips and program customizations. The subroutines these users brought back to the office made their own firm's use of CAD that much more productive - and the tweaks made to these applets by each office allowed the software to adapt a little differently in each office.

End-users also had frequent contact with the programmers that wrote the CAD program - mostly for help in creating these customizations, but also to report bugs and recommend enhancements to the basic program. We also bragged about the enhancements we made to their software with our own applets.

The CAD developers found some of these enhancements so compelling that they folded the best of those capabilities into the next version of their software - co-opting the ideas and work of the innovative end-users for the benefit of their entire user community.

Each CAD package grew in much the same way - but in different directions, depending upon the strengths of the base package and the particular needs of its users.

When you hire people with CAD skills, the most valuable will be the ones who have used more than one CAD program. These people will have worked with labor-saving tools in programs other than your own - and those tools can be incorporated into your applications.

When you review the current batch of CAD software packages, you'll find routines and 'ways of doing things' in each of them that would be beneficial to your office. Almost all modern CAD packages are extendable using various programming tools - some using built-in tools, and some using separate general-purpose programming tools.

The point is - you want to include the **best** productivity features your office could use - not just the ones that happen to come with your off-the-shelf CAD software. The way to do this is called **R&D** - Rob and Duplicate.

I have used this technique to write project startup scripts, sheet layouts programs, keynoting, inserting common office standard details, BOMA area calculations, extracting building data for Internet access by the client, switching between common presentation formats, easy revision standards, creating schedules from CAD data, inserting common symbology into CAD drawings, and more. Most of these efforts paid for themselves within months because they were used often by lots of people. Some, like Internet access to client data, made the firm more valuable to it's clients. Some failed.

Make sure the procedure you automate is one that's really a standard - not just something you

If you've seen a productivity feature in another application that is poorly (or not at all) supported in your own CAD application, go through some quick calculations. First, figure out how much time would be saved throughout your office if you had that capability (there may be benefits other than time saved to include in your calculations). Then find out what it would cost to add the feature to the software (using consultants or in-house developers) - then make your call.

The tough part of this assessment is estimating the cost of programming. A programming job includes both design (application, data and user interface design) and construction (writing code). Like selecting estimates for design and construction, you must be critical in assessing these estimates.

wish was standard. If everyone feels they have their own best way to do a materials schedule, automating **your** way will not change their methodology. Use other methods (political or otherwise) to establish office standards - and then use automation to make using them easier.

When assessing estimates by in-house staff treat it like any other in-house estimate. Most in-house programmers - like other staff - are notoriously bad at estimating labor and schedules. In your head you probably have your own multiplication factor used for in-house time estimates for project work. Use that technique for your in-house programmers as well. Also make sure they have the time to set aside to work on that project - too many important in-house development efforts are sidetracked by more urgent (but often less important) matters.

When assessing outside estimates, check them out as you would any other contractor. Check references. Were they on time and within budget with other projects? Would their past clients hire them again? You know the drill - it's how your own clients judge your firm!

What enhancements have you made to your office applications? - e-mail me at mhogan@id-8.com.

Michael Hogan, AIA - head chiphead at Ideate, provides custom web solutions and provides consulting services to the AEC industry in Chicago. He welcomes comments by e-mail at mhogan@id-8.com