Automating Repetition

To be maximally efficient and productive, we all need to "program" all our repetitive tasks

By Marc Prensky

Published in *Educational Technology*, Sept-Oct 2014

[1081 words]

I work with many people who do excellent, future-oriented work with technology, including making videos, websites, blogs, and using social media. Many of them are, like myself, part of the (Last) Pre-Internet Generation. One observation I frequently make is that these people (me included) are often working far less efficiently than we could be. A big reason for this inefficiency is that too many of us are not adept at using even the simplest of automation tools, and have not integrated these into our other work.

To be efficient (and far more effective) in this day and age, it is important that we automate *all* our repetitive technology tasks — *everything* we do more than once or twice. Those tasks can be as simple as responding to emails or texts in standard ways or signing them with an identical "signature." Or they can be as complex as putting out daily or monthly aggregations or reports. As soon as we have to do anything identical (or almost identical) more than once with technology, it is far more efficient to write a "program" to do it and, in the future, push only one button to execute that program. And we should all be able to do this.

Relatively simple tools to do this have existed for much of our professional lives — menu choices and macros in Microsoft Office, for example — but they often go unused. Much of the reason, I believe, is that we weren't taught to look for and take advantage of such efficiencies. But we could — and should — teach ourselves and our students to do this. In fact these days, our students can often teach us.

Automating our repetitive tasks is often much easier than we think, but too often we don't even try. There are many tasks I do that I know I could further automate, but I often think that I am "productive enough." Yet whenever someone is billing *me* by the hour, it becomes a priority for me that they be automated as much as possible. Don't we all really owe our students — i.e. our clients — the same?

Automation capabilities are being built into more and more of the programs we use, and we should try to leverage these built-in solutions. And when they don't fit our needs, rather than give up, we should create the precise solutions we need. Recently, I asked for a new, ongoing report from some consultants, and was told that, since it was not in the menus of the programs they used, my customized request would require "lots of manual labor" every month.

It shouldn't. Whenever the same solution is needed repetitively (or continuously in real time) that work can be and should be automated — because if we write a macro or program once, the machine will do it forever, without our having to intervene at all. In order to do this, of course, we all need to learn to become, at some basic level, programmers. Programming is now a basic skill for any intellectual work, in the same way as reading and writing skills are. Not being able to do this ourselves is like once having to go to a scribe every time we needed to write anything down. And if we can't (yet) do it ourselves, we should turn to our kids and our students for help.

A person who is a model for this kind of efficiency — an absolute master whom I am in awe of and strive to emulate — is the technology writer David Pogue. He is by far the most productive person I know, pumping out multiple books, columns, scripts, and email responses with a rapidity and efficiency that is breathtaking. He is often asked how he achieves such efficiency and productivity, and in a *New York Times* column entitled "Pogue's Productivity Secrets Revealed" (June 20, 2009) he described his methods, and some of the personalized automation he uses. For example:

- He has set up all his computers so that he needs to type only the first couple letters of many common words, and the software expands the rest. He's created a list of the phrases he personally uses most — including "ty" for "thank you" and "hth" for "Hope this helps! – David." (He does this customization with external "typing-expansion" software, rather than with the tools in Microsoft Office, so he can use the same shortcuts in every single one of his programs.)

- For all his "longer" writing, Pogue dictates into Dragon NaturallySpeaking voice-to-text software. Today's voice-to-text is far better than it used to be, and adapts automatically to your vocabulary and way of speaking with little or no "training" required — plus it learns and gets better as you use it. I personally use this and I highly recommend it. If you ever tried and abandoned it, give a another chance.

- Pogue makes extensive use of macros, making up his own keyboard shortcuts. "I use Shift-Control-W to open Microsoft Word, Shift-Control-C for my calendar program, and so on," he writes. "The less I have to use the mouse, the faster I fly."

Many of us, of course, currently do some or all of these things. But why don't we all do all of them all the time? What this requires is *always thinking in terms of automating anything and everything that is repetitive*. All students and educators have huge amounts of repetitive tasks to do. Those of us (e.g. tech coordinators or coaches) who already know how to do it should be helping teachers and students automate all their tasks to the greatest extent possible, blowing past any objections to its being "better done manually," for whatever reasons. If our students and teachers don't learn to automate, they won't be as efficient as they need to be. Period.

As the tasks we automate become more complex, "automation" crosses into "coding," and "real programming." I and others have long argued that programming — making our machines do what we want them to do — is a key literacy for the 21st century and beyond. More and more schools are now offering programming classes. Getting all your students (starting in the early years) to use these classes to automate their own repetitive tasks — and to always think in terms of automating every repetitive part of what they do — as well as training all students to help their teachers do this — is a great way to expand, or get started with, such programming classes, and, at the same time, to make this crucial skill "real."

Marc Prensky is an internationally acclaimed thought leader, speaker, writer, and consultant in the field of education. He is the author of five books: From Digital Natives to Digital Wisdom *(Corwin, 2012)* Teaching Digital Natives: Partnering for Real Learning *(Corwin, 2010),* Don't Bother Me, Mom, I'm Learning *(Paragon House, 2006),* Digital Game-Based Learning *(McGraw Hill, 2001) and* Brain Gain: Technology and the Quest for Digital Wisdom *(Palgrave MacMillan 2012).* **Reimagining K-12 Education for the Future: a New Core for a New Age** *is his upcoming sixth book (from Corwin), and he is also writing a book about a new curriculum for the world.*

Marc is the founder of ***The Global Future Education Foundation and Institute****, a not-for-profit organization devoted to promoting Future-oriented and Accomplishment-Based Education, and a better worldwide curriculum. Marc is also the the founder and CEO of Games2train, whose clients include IBM, Bank of America, Pfizer, the U.S. Department of Defense and the L.A. and Florida Virtual Schools.*

Marc holds an MBA from Harvard and a Masters in Teaching from Yale. His writings can be found at www.marcprensky.com *. Contact Marc at* marcprensky@gmail.com *.*