

developer.***A Web Magazine for Software Developers**

The Purpose-Driven Programmer

by Donna L. Davis

Somewhere between algorithms and altruism lies Programmer's Purgatory. Sure, career burnout is occasionally attributable to work volume, but more often it results from inner conflict: opposing forces colliding in a reality rip tide. The money's pretty good, but why am I here? Am I to fritter away my youth hunched under the stark rays of a computer monitor, fingers agile and athletic, while the rest of my body atrophies into a mass of useless flesh, like the center of a giant squid with glassy eyes staring blankly ahead?

Software developers choose the field for a number of reasons, but the top two are the technical challenge and compensation. Yet, career satisfaction surveys repeatedly highlight that financial remuneration is not enough to keep an employee happy and motivated. While the technology field is ripe with challenge, a healthy percentage of perceived drudgery comes with the package.

For every line of code written, a page of associated documentation is birthed from the womb of regulation and best practices. For every stunning code snippet written, today's systems analyst spends exponentially more time wearing the consultant hat, building relationships, and learning the underbelly operations of business.

Then when the opportunity to actually write code comes along, what does it involve? More often than not, cookie-cutter forms the color of corpses with emotionless command buttons, comforting clients in their predictability. Add too much creativity and voila! You've just violated the standard. "Classically trained" in switching theory and assembly language to produce software symphonies, developers often find themselves churning out Java jingles or dull development dirges. Visual Basic: the name conjures images of a preschooler's picture book. Where's the elitism in that?

Insulated by technology, hidden in the business basement, huddled behind monitors and keyboards, developers often become narrowly focused on problem punch lists. Manning help lines and responding to service requests, 100 complaints clog the information portals for every wow-you're-incredible. Even when dusted off and suited up for a consulting gig, analysts operate under a compartmentalized mentality, feeling only distantly responsible for success and heavily to blame for failure.

Then there's the deployment obstacle course. Better develop close friendships in the Validation department and it wouldn't hurt to have a member of the family employed as the network administrator. Otherwise, well, that's what contingency is for. Float is commonly believed to be a project management term, but in reality it's a loose reference to the sort of buoyancy device needed to keep the developer from drowning in a sea of details and high tech red tape.

So, if money only matters to a point, and challenges are sandwiched between thick slices of dry, wheat toil, what keeps the developer showing up day after day?

The answer is the same for developers as it is for social workers, doctors, teachers, stay-at-home moms, and Wal-Mart greeters: *raison d'être*. Developers must make the tangible connection between the code under their fingertips and humanity's bottom line. That includes applications driving 911 dispatching, the launch of a life-saving new drug, or any number of seemingly mundane activities that keep the food chain connected and the world of commerce spinning.

The unpleasant sound of skeptic scoffing is audible through mind mapping technology, spanning paper and time continuums. What is that? Developers have no such base, humanitarian needs? They're a primal lot, focused on code, coffee, and possibly a weekend sci-fi flick or their latest wireless gadget that sends email and flushes the toilet?

Don't buy the stereotype. Programmers are sensitive, caring people too. If you cut them with the sharp edge of a technical document, will they not bleed? (Or is that just the red ink of a manager's edit?) Unfortunately, developers have been pigeon-holed as mechanical code-spewing drones for so long, they may not even recognize the curious flutter of sensitivity sprouting like a code kernel from within. It's the need that compensation, a laptop with docking station, nor a Dilbert calendar can fill.

To scale the precarious tip of Maslow's programming pyramid, developers must start with their project at hand and trace the code straight to its final resting place among the steaming masses.

Assignment: Design an application to manage a drainage district. Yawn and yuck. Images of sewers and post nasal drip dance in the mental bubble over the developer's head. He stretches the waistband of procrastination to the point it unravels and curls below the bulging belly of boredom. After shuffling the bitter assignment from the quadrants of urgent, important, forget about it, and the project black hole, shame and the imminent threat of termination rears its ugly head. District maintenance form. Parcel maintenance form. Admin form. Login form. *Sigh* After long days and weeks, the coding colonoscopy through the bowels of drainage data is complete.

A ditch is a funny thing. Nobody cares about it when the sun is shining. In fact, the next door neighbor fills it in with transplanted sod to extend the beautiful, green slope of his manicured lawn. When the weatherman starts predicting the Storm of the Century, however, it's another story. Suddenly canals and ditches are the throbbing, clogged arteries threatening the heart of the county.

In the eerie aftermath, television crews are never sent to corporate offices to interview the programmer who was personally responsible for technology that spared the lives and homes of 64 potential victims.

Focused on the latest buzzwords or corporate downsizing, the developer may never even make the direct connection himself. And that would be a shame. Perhaps the Programming Panacea is like Dorothy's ruby slippers: the stuff that was there, under our substantial noses, all the time. Purpose-driven programming means connecting the code dots until the face of a desperate mother, a brave fireman, or a hungry child is visible.

A developer is a lot of things: psychologist, mathematician, designer, and journalist. He is also a programming philanthropist, and that realization just might bridge the gaping hole of discontent that neither money nor a mere challenge can.

###

Donna Davis is a database administrator, project manager, programmer, supervisor, and professional author living and working in the North Carolina, US. She can be reached through the editorial staff of [developer.*](#).