

## Re: Future of programming

**Source:** <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2004-01/0765.html>

---

**From:** RH ([nospam\\_at\\_hotmail.com](mailto:nospam_at_hotmail.com))

**Date:** 01/30/04

Date: Fri, 30 Jan 2004 03:08:41 GMT

I thought I'd just add a dose of non-theoretical reality to this thread.

–Last client spent \$2.5 million over 2 years trying to roll out an Oracle HRIS system.

–Current client bought a "timekeeping package" for \$300K and is now paying \$200/hr for changes that any college graduate could do.

–Current client is considering a \$2 million package for A CRITICAL BUSINESS FUNCTION! Can you imagine the stupidity? And the maintenance fees?

I hate to be obvious but there are only three types of people in the programming world:

–Application Support: these are the people that will generally refuse to learn anything new and promote "package buying". They show up for work with lots of beepers and "all is well".

–Employed developers: these are the people working in a company that are ALWAYS fighting management and Application Support because they know a scam when they see it.

–ISVs — who have employed the developers above to deliver \$200/hr solutions to the lower food chain because their client says "oh! it's too difficult!".

The OO comment is just irrelevant. Clients don't care (or even know) about OO solutions. They just see solutions in terms of their business needs.

"Clark F. Morris, Jr." <[cfmtech@istar.ca](mailto:cfmtech@istar.ca)> wrote in message news:[bv1q10\\$pm\\$2@news.eusc.inter.net](mailto:bv1q10$pm$2@news.eusc.inter.net)...

> *There will be a lot less custom programming and far greater use of packages in the future. That is in part because even relatively inexpensive computers have enough memory, CPU power and disk space to compensate for the inefficiencies of generalized programs that execute code unneeded by the individual organization. An even more important factor is the documentation and help capability provided by the packages. A common complaint about today's systems is that few people really know what they do. In-house systems are done in an environment that either programmers or end users are expected to document how the system works. Most programmers (myself included) don't like to write and the end users don't have the time. In addition there are increasing demands on systems to handle such things as security and privacy.*

## comp.lang.cobol: Re: Future of programming

- > *Setting up decent security that protects the organization and provides*
- > *audit trail without tying it in knots is not easy. Privacy*
- > *considerations are also difficult to understand let alone implement.*
- >
- > *The previous paragraph states why I doubt the need for IT professionals*
- > *will be eliminated. Packages which provide tools for customization*
- > *(report writers, tables such as tax tables, exit points for inserting*
- > *custom code, etc.) are still going to require people who have the*
- > *ability to understand how the application affects the organization.*
- > *There is procedural coding within the components and the gluing together*
- > *of batch processes is still a procedural process. Yes, online and*
- > *interactive operation is becoming increasingly important and having*
- > *processes that communicate through defined and enforced interfaces is a*
- > *noticeable and necessary advance (that was the opinion of the*
- > *SHARE/Guide Language Futures Task Force back in 1984). The industry*
- > *also should have learned that the tools that are used to customize*
- > *and/or upgrade an application have to have editing, audit trails and*
- > *consistency checks. However, installing, testing and tuning are not*
- > *skills that come easily to the average user. It is one thing to put*
- > *Quicken on your home computer and another to install a major ERP (SAP,*
- > *Peoplesoft, etc.) where the organization may deal in partial units*
- > *(heating oil anyone) and have complex tax considerations as well as*
- > *response time requirements.*
- >
- > *The waterfall method took so long because we didn't have some of the*
- > *tools we have today. Even the advent of word processing has changed*
- > *things drastically. Documentation became maintainable. Now we even*
- > *have single sourcing where manuals and the on-line help use a common set*
- > *of specifications. Some organizations have even learned that it isn't*
- > *only IT that has a peculiar jargon and that everyone in the organization*
- > *has to use a term in the same way. I believe that RAD works today in*
- > *part because the base has been built painfully using the waterfall and*
- > *sometimes just trial and error. We have systems that more or less*
- > *function. Organizations may need to spend a large amount of resource*
- > *finding out how they really work so that they don't blow up the*
- > *organization in the process of upgrading or replacing the systems. We*
- > *may be using UML (Universal Modeling Language) instead of COBOL and*
- > *various other tools but there will still be people implementing the*
- > *complex systems. The tools will help the various parts of the*
- > *organization work together to accomplish goals. What the tools won't do*
- > *is give the organization the willingness to work collectively. The*
- > *tools won't make people share and talk to each other. Walmart is*
- > *reported to be able to enforce a common standard on their store*
- > *applications for at least two reasons: the need for someone to be able*
- > *to go into any store and have the system work the way they expect it to,*
- > *and the IT department works with those who actually interface with the*
- > *systems to see what problems there are and how they can be solved. I*
- > *understand it is a part of the culture that it is assumed that things*
- > *are not perfect and that they can be made better. The founder, Sam*
- > *Walton was reputed to always ask what went wrong in the previous period*
- > *because he was looking to improve the organization rather than to fix*

- > *blame.*
- >
- > *OO may be part of the answer but I frankly am unimpressed with the idea*
- > *of components from multiple sources. If I am going to spend money, it*
- > *is going to be for a package. I won't (and on my PC don't) care whether*
- > *it is OO. I do care about whether it has adequate help so that I can*
- > *use it. I do care about whether I can customize it.*
- >
- >
- >