

# The Final Triumph of the Junior Programmer

Conrad Weisert, Information Disciplines, Inc., Chicago  
December 12, 2001

## ***A radical terminology shift***

Growing evidence indicates that there are no longer any *junior programmers*. No, they aren't extinct; they're just now called "senior programmers"!

I've gotten a half dozen calls in the past few weeks from recruiters and brokers, asking if I knew anyone who would be available (or would be available myself) for a *senior programmer* role in an exciting project. In answer to my questions the caller revealed the following about the project and the proposed role:

- The high-level design (*architecture*) of the application is complete.
- The project has no mechanism for establishing and disseminating project *standards*.
- No detailed project plan is available to the project team.

Furthermore, the senior programmer is not expected to contribute to any of those areas. One caller even admitted that any attempt to do so would be decidedly unwelcome. The senior programmer is just expected to carry out detailed coding and testing tasks assigned by the project manager.

Isn't that role exactly what we used to call "*junior programmer*"?<sup>1</sup>

Terminology evolves, of course, but such a drastic change needs some justification, along with an explanation of what on earth *non-senior* programmers are doing to differentiate their role from their senior counterparts.

## ***Focus on minutiae***

There's one way to find out. If you proceed to the next stage in the recruiter/broker's agenda, you'll encounter either:

- An "in-depth technical interview" (often by telephone<sup>2</sup>), or
- A certification (or other online) examination.

---

<sup>1</sup> Another term was "coder".

<sup>2</sup> Usually with a caller who is himself obviously *junior* in terms of his grasp of both programming issues and the state of his own project.

Both focus almost exclusively on the lowest level of detail in the use of various current tools. They pay no attention at all to:

- issues of program quality and maintainability,
- the forms in which specifications are expressed,
- data definition, class structure, and data representation choices, not even those that are central to the new application<sup>3</sup>,

Some of the worst<sup>4</sup> programmers I've encountered could easily pass these interviews and examinations.

## ***The new working definition***

We begin to suspect that the recruiter/broker's definition is something like this:

<p><b>senior programmer:</b> a programmer of any level of experience and skill, who has committed to memory more low-level details about one or more current tools than most other programmers.</p>
---

If our suspicion is valid, this new kind of seniority is entirely tool-specific. Any hack coder can now attain "senior programmer" status by working for a few months in a project team role in which he or she memorizes, say, Visual Basic's property lists and event-firing-sequence. Even if he or she churns out the most atrocious and unmaintainable program code and has no conception of high-level design issues, the recruiter/broker (and presumably someone in the client organization) will consider him or her "senior" for purposes of project staffing.

---

<sup>3</sup> In the course of these discussions I've discovered (a) a huge "object-oriented" financial trading system that had no class definition (or even a preferred standard) for representing amounts of money and (b) an accounts payable system, also said to be object oriented, that had no class for representing information about a vendor! The caller couldn't understand why such a thing would be needed.

<sup>4</sup> As measured both by their productivity and by the quality of what they produce.