**Finding Report #2**

*The issues in the language of the web;*

JavaScript is supposedly a functional programming language. (Actually, it’s multi-paradigm, mainly imperative with some functional capabilities and a smattering of object orientation, via object prototypes.)

Unfortunately, JavaScript is also a dysfunctional programming language, imbued with a myriad of “warts” and “gotchas.” It lacks the discipline to be a serious software engineering language, thanks to loose typing and freewheeling coercions, and their wildly inconsistent semantics.

JavaScript doesn’t even have a proper integer type! JavaScript also doesn’t have a proper array type like you find in most normal programming languages. Just to be clear, despite ECMA working hard to fix JavaScript, it has yet to address the abovementioned problems.

JavaScript apologists are constantly telling us that all programming languages have warts. What they don’t tell us is that most languages have far fewer warts than JavaScript has. How big is the problem for this language?

*Are there similarly long lists for Java or Python?*

These same apologists also tell us that by applying good practices, including linters, you can avoid the bad parts, and you’re left with programming joy.

Isn’t that what Douglas Crockford’s book is all about? This is a fairy tale, of course. Good practices are not a perfect solution. Sooner or later, JavaScript’s loose typing and inconsistent semantics will bite you in the ass. Sooner or later, the lack of integers or proper array type will prove annoying.

These things are like land mines, lying in wait with infinite patience to blow off your legs. Good practices mitigate but do not eliminate the problems.

The need to be constantly aware of the bad parts most definitely places an added and unnecessary cognitive burden on the programmer.

So why do JavaScript proponents continue to promote its use?

The answer is:

JavaScript’s large user community and ecosystem of tools and libraries, not to mention the large number of jobs available for JavaScript developers. This answer also explains the longevity of all the mainstream languages, irrespective of their faults, including: PHP, C++, Perl, and Visual Basic (languages everybody loves to hate).

The community and ecosystem argument is certainly quite compelling. There is no doubt that these things confer substantial benefits for both businesses and programmers. The benefits, however, must be balanced against the downsides, such as: