The Embedded Muse 67

Editor: Jack Ganssle (jack@ganssle.com) October 16, 2001

A Dirty Word (Again)

In the last issue of the Muse (http://www.ganssle.com/tem-back.htm) I ranted about the evils of deferring "system integration" to a late stage of the project. A number of people responded, but I was especially taken with Mike Pelley's comments:

"Here is another thought related to 'integration' and 'early and often': At Rockwell Automation, we do some complex embedded systems. One concept that has helped with developing a new system of many complex parts is a 'thin thread'. This is defining the minimum feature set that includes at least one of each of the types of parts of the system that can demonstrate that the pieces can work together. We do automation products which have programming software, controller, communications modules and I/O modules. We pick a very small subset of the programming language, the minimum number of communications features (but using all the layers of the stack), and the simplest I/O module. The "thread" winds its way through all the products. The 'thin' focuses on a minimum subset that demonstrates that everything is talking correctly. It is not even enough to do any useful control. It is amazing how many issues this effort drives out. And it is a whole lot easier to debug a minimal system than one that is feature rich. Then you work on making the tread 'thicker' until you have something that is ready to ship. But those seem to go much better once the 'thin thread' is working. You've eliminated many of the misunderstandings of basic concepts and specifications. You've developed much better working relationships among the various members of the team."

(Mike warned me not to credit him with this innovation, as it was already in place when he came to that business.)

The Best Programming Book Ever

Software has two missions: to *do* something, and to *communicate* the programmer's intent to future maintainers, or to people wishing to reuse portions of the code. In my opinion, any bit of code that doesn't do *both* of these things really well is totally unacceptable.

But the code itself, the C, C++ or whatever, is not particularly easy to read. Well written comments are the basic structure of any program. The comments describe the intent, the

complexities, the tricks and the issues. Code just obfuscates these. Great comments are a basic ingredient of great code.

That means the comments have to be well written, in English (at least for those of us working in English-speaking countries), using the noun-verb form. Each sentence starts with an upper-case letter; the rest of the sentence uses appropriate cases. Comments that don't conform to the basic rules of grammar are flawed.

Donald Knuth talked about this at length. He felt that the literate programmer:

- Is concerned with the excellence of style
- Can be regarded as an essayist
- With thesaurus in hand, chooses variable names carefully
- Strives for a comprehensible program
- Introduces elements in a way that's best for human understanding, not computer compilation.

And so, I've long believed the best programming book ever is "The Elements of Style," by William Strunk and E. B. White (Allyn & Bacon; ISBN: 020530902X). In a mere 105 pages the authors tell us how to write well. Just five bucks from Amazon.com. Or, you can get it for free at http://www.bartleby.com/141/index.html.

Unfortunately, most developers are notoriously bad at writing prose. We NEED the help that this book contains.

One rule: "Use active voice". Doesn't that sound better than "the use of the active voice is required"? It's also shorter and thus easier to write! Another: "Omit needless words," a great improvement over "writers will examine each sentence and identify, characterize, and excise words in excess of those essential to conveying the author's intent."

An appendix lists words that are often misused. English is quite a quirky language; it's easy to make really stupid mistakes. One example: mixing up the verb 'effect', the noun 'effect' and 'affect'.

The book makes an interesting comment about signing letters: "Thanking you in advance." This sounds as if the writer meant, "It will not be worth my while to write to you again." Instead write, "Thanking you," and if the favor which you have requested is granted, write a letter of acknowledgment. I like that!

My biggest pet peeve about poor writing is mixing up "your" and "you're." I've seen billboards with these words confused – talk about advertising your ignorance!

The book gives 18 simple rules, far fewer than the nuns attempted to beat into my brain so long ago. Follow them and your comments, and thus the code, will improve.

Thought for the Week

Three engineers and three accountants are traveling by train to a conference. At the station, the three accountants each buy tickets and watch as the three engineers buy only a single ticket. "How are three people going to travel on only one ticket?" asks an accountant. "Watch and you'll see," answers an engineer. They all board the train.

The accountants take their respective seats but all three engineers cram into a restroom and close the door behind them. Shortly after the train has departed, the conductor comes around collecting tickets. He knocks on the restroom door and says, "Ticket, please." The door opens just a crack and a single arm emerges with a ticket in hand. The conductor takes it and moves on. The accountants saw this and agreed it was quite a clever idea.

So after the conference, the accountants decide to copy the engineers on the return trip and save some money (being clever with money, and all). When they get to the station they buy a single ticket for the return trip. To their astonishment, the engineers don't buy a ticket at all. "How are you going to travel without a ticket?" asked one perplexed accountant. "Watch and you'll see," answered an engineer. When they board the train the three accountants cram into a restroom and the three engineers cram into another one nearby. The train departs. Shortly afterward, one of the engineers leaves his restroom and walks over to the restroom where the accountants are hiding. He knocks on the door and says, "Ticket, please."

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words "subscribe embedded *your-email-address*" in the body. To unsubscribe, change the message to "unsubscribe embedded *your-email-address*".

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.