

The Embedded Muse 173

Copyright 2009 TGG

January 19, 2009

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com. Subscribe and unsubscribe info is at the end of this email.

EDITOR: Jack Ganssle (jack@ganssle.com)

CONTENTS:

- Editor's Notes
- Quotes and Thoughts
- Funny Datasheets
- Computer Science Education
- Response to Last Issue's Quotes and Thoughts
- Jobs!
- Joke for the Week
- About The Embedded Muse

Editor's Notes

Did you know it IS possible to create accurate schedules? Or that most projects consume 50% of the development time in debug and test, and that it's not hard to slash that number drastically? Or that we know how to manage the quantitative relationship between complexity and bugs? Learn this and far more at my Better Firmware Faster class, presented at YOUR facility. See <http://www.ganssle.com/classes.htm>

Brazil! The country of samba, feijoada and caipirinha. And, of course, embedded systems. I'll present a two-day version of my Better Firmware Faster seminar in Sao Paulo March 25th and 26th. See <http://workshop.embarcados.com.br> for more information.

How about a two-fer? I'm presenting a public version of the Better Firmware Faster class April 3 in San Jose, CA. Come to the Embedded Systems Conference that week AND join me for a day to learn my approach to crafting world-class systems in less time. See <http://www.ganssle.com/classes.htm> for more info.

Philip Freidin and many others sent links to 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (<http://cwe.mitre.org/top25>). These are largely about security

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

concerns. Most we all know about, yet far too much new code still commits the same sins. It's good to review potential trouble spots from time to time.

Quotes and Thoughts

When Leo Tolstoy wrote Anna Karenina he could have been thinking about cubicles: "There are no conditions of life to which a man cannot get accustomed, especially if he sees them accepted by everyone around him."

This is especially timely considering a recent article (described here: <http://www.news.com.au/business/story/0,27753,24906913-5017672,00.html>) where researchers found open offices lead to lower productivity and higher stress.

Funny Datasheets

Bob Paddock sent this link: <http://www.national.com/an/AN/AN-761.pdf> . It's a paper about electronic fuzing of bombs and other weapons, but ends with the disclaimer "National's products are not authorized for use as critical components in life support devices..."

Makes you think!

Computer Science Education

A number of recent articles in the computer science press complain about the poor state of CS education. Companies and pundits decry university's reliance on Java for teaching students how to be effective programmers. I agree. While Java is a useful language indeed, in the embedded space it has little market share. And the strength of Java is also its weakness (at least for embedded work): It does a great job of insulating the developer from the hardware.

Yet half or more of the firmware developers I meet have EE degrees. It's odd that these experts at transistor theory, electromagnetics, and IC design spend their career cranking C and C++ code.

Perhaps the reason for this is the simplification of hardware and explosion of firmware. I suspect there are more EEs around than needed. Companies find them desirable for software development due to their deep knowledge of how things work.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

But as the firmware's complexity increases it also becomes more like application code. Implementing drivers and ISRs will always require hardware insight, but that might be 1% of a large project. GUIs, net connectivity, and data reduction looks pretty much the same whether it's on a PC or an embedded system.

Some contend that EEs are better problem-solvers than CS folks, due to their analytical nature that spouses find so infuriating. Perhaps. I suspect most of those skills are developed on the job, and not as a result of academia.

EEs do learn an awful lot of software in college. But only in the worst possible way. Few schools stress software engineering; most teach people to write programs. "Here's the fundamentals of the language, now write a program to do this." Process is almost unheard of. Nearly none know about inspections, PSP, CMM, XP, or any other disciplined development strategy.

CS majors get more of a sense of these techniques, though for them, too, coding is generally the only thing that counts. They're graded on producing projects, not on how they build a project.

I think coding should be abolished in college, at least until the Junior year. Any idiot can crank code. Stress design, process and problem-solving early. These folks will mostly be working as part of teams, not as solo heroes. Give them the skills they'll need to be effective team members. Teach them to build big projects reliably. Make them abhor cowboy development.

It matters little which methodology people learn. Teach them to think through implications of their design decisions, to model when necessary, to review constantly, and to expect coding to be an almost trivial endeavor.

My dream is for the new grad of the future, EE, CS or CE, to come out of school expecting businesses to employ some sort of disciplined development process. When they find that their company uses heroics and overtime I hope they are horrified... and quit. Let's hope they demand real design, and they avoid plunging into coding too early. For though they may initially tilt at the windmills of inertia, in time these people will be the team leads and then management.

Response to Last Issue's Quotes and Thoughts

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

Responding to: “Of course, we don't have time for this. Time to market is king. The company who actually did this might get scooped every time and go out of business, or at least that's the prevailing thought. The coder who does this will be slower (according to prevailing thought) and will get fired,” Vlad Zeylikman wrote:

“This is true but it is also a matter of educating the management and presenting the issue in the proper light. if you have to educate an engineering manager, all is lost right there. but if the management in question is not engineering, then the right way to do this is in terms of the cost of maintenance. this is by far the biggest cost of any software: redesign, redesign on the fly, and bug fixes. Delivery time and features can be discussed in this context and a balance (a compromise) can be found.

“This is not theory: I was responsible for the software in a start-up and had to do exactly that with the upper management. I was fortunate to be able to convince them to trust my expertise and let me do it my way. They were very nervous when in the middle of the project there were no engineers in the lab at mid night. But when we ran integration, things came together in a matter of 2-3 weeks - simulation tools are a great help. We did burn some midnight oil when we brought up the boards, but that was the exception. Management was very much surprised and relieved when the system was running ahead of schedule.

“But I would agree that this experience is the exception.”

In response to “Or use a different language that spots these sorts of bugs early (e.g. Ada - as another of your posters noted)...” Martin Dowie had this to say:

“Another thing I've noticed is that quite a number of people are switching from C to C++...only to not use the language features in C++ that can actually help avoid C-style errors (e.g. don't use exceptions, still use C-style arrays, don't use references but still uses pointers everywhere, don't use the default container libraries, don't use namespaces, still use macros instead of consts etc).

“In fact, from C++ they tend to use only the keyword 'class' and continue to do everything else as they had in C... I call this 'C +1'.”

Jason Cohen wrote: “In my experience his list of bugs is NOT where most bugs happen if you're dealing with experienced developers.

“Even in embedded C projects I've almost never come across those types of bugs. Simple defensive code practices, use of -Wall (which isn't that much effort if done from the start), basic unit testing (not top-to-bottom) and code review prevents essentially all

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

these issues.

“At ITWatchdogs the bugs were often of an integration nature -- browser-specific bugs, talking with hardware components, dealing with error correction, handling Internet attackers, talking to various SMTP and SNMP servers, etc..

“Integration bugs can't be found by any of the methods he lists. Integration tests are often possible but MUCH more difficult than usual unit tests.

“Then of course there's project-level problems like not having the features you need, building the wrong thing, or building something too confusing. (At least 1/2 our tech support at ITWatchdogs was not about a bug in the "spec" sense but in something confusing to the user.)

“Furthermore if you use Java -- which more and more embedded systems are -- even fewer of his problems (or solutions) are applicable.

“If you're defining "negligence" as root-level bugs that simple, almost trivial and well-understood processes can almost eliminate, and you're just not using those processes, then I agree with him. But if you make it any more broad than that (e.g. failure to test the web interface against IE 6), then his list of bug sources and solutions don't match what I've seen.”

Jobs!

Let me know if you're hiring firmware or embedded designers. No recruiters please, and I reserve the right to edit ads to fit the format and intents of this newsletter. Please keep it to 100 words.

Wow – none this week. How is the recession effecting your company?

Joke for the Week

Not a joke, but funny in its own way. Don't shout at your hard disks:
<http://uk.youtube.com/watch?v=tDacjrSCeq4> .

About The Embedded Muse

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

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

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.

Copyright 2003 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at info@ganssle.com for more information.

The Ganssle Group, www.ganssle.com