

# The Embedded Muse 137

Editor: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

November 21, 2006

You may redistribute this newsletter for noncommercial purposes. For commercial use contact [info@ganssle.com](mailto:info@ganssle.com).

EDITOR: Jack Ganssle, [jack@ganssle.com](mailto:jack@ganssle.com)

## CONTENTS:

- Editor's Notes
- Book Reviews
- Does Expensive = Good Quality?
- An Alternative Environment
- Beautiful C++
- Tools
- 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?

I'll cover all this and much more December 1 in Santa Clara, California, in my "Better Firmware Faster" seminar. Check out <http://www.ganssle.com/classes.htm> for details. The site is just a few miles from San Jose airport, which is inexpensively served by Southwest Airlines and others. Come for the seminar on Friday and explore San Francisco over the weekend! Better yet, check out the Computer History Museum in Mountain View.

Do you have a dozen or more developers who could benefit from this firmware training? I can present my Better Firmware Faster class at your facility. See <http://www.ganssle.com/classes.htm>.

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

I wrote an article about introspection this week (<http://embedded.com/showArticle.jhtml?articleID=194500077> ), both for driving and embedded development. For US readers, have a happy Thanksgiving... and do drive safely during this mad travel weekend.

## **Book Reviews**

Ye gads, my input stack of technical books is getting dangerously high. I try to read two a month but sometimes get sidetracked by other, less trying, volumes. Two favorites from the last couple of weeks include Trevor Corson's "The Secret Life of Lobsters" and Jacquelyn Mitchard's beautifully-written "The Deep End of the Ocean."

In the technical arena I greatly enjoyed "Agile Estimating and Planning," by Mike Cohn. Mike knows how to write in an engaging matter on subjects that are inherently rather dry.

This book is of value for anyone building software, whether working in an agile environment or a more traditional plan-driven shop. Possibly the main agile thrust of the book is Mike's insistence on re-estimating frequently, whenever there's new information. But wise managers should stay on top of schedules which always evolve, no matter what approach is used to build the system. Re-estimation might not be common, but bad estimates are.

He does use a lot of agile terminology like "story points," and he makes good arguments for decomposing a system into features or components of features. If your outfit is more plan-driven everything he says still applies.

Though Mike never uses the word "Delphi" his approach is awfully close to the method the Rand Corporation invented, and that has been very successful on software projects (for more, see <http://www.processimpact.com/articles/delphi.html>).

A key concept is that of "velocity," or keeping track of the rate of software development. That gets fed back into the evolving schedule to converge on a realistic date relatively early. Of course, velocity fails when the team accepts scope changes without doing a re-estimate. Agile developers use short iterations and frequent deliveries to elicit these changes early; many projects just can't do that and so should employ a change control procedure that tells the stakeholders the impact of a revised requirement. There's nothing on change control in the book, which is probably appropriate but a shame.

The book has no real-world data. There are plenty of wise words telling the reader how to make the right measurements in his project, but sample velocity and other sorts of metrics from real projects would be helpful to establish a range of benchmarks.

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

The chapter on making decisions based on financial impact to the company assumes one can quantitatively predict inherently hard-to-determine factors, like “add feature X and we’ll get 50 new customers.”

Mike recommends adding a buffer, and tries to distinguish this from schedule padding. Yet the boss may find it hard to make the distinction and will likely remove it. Buffers/padding/2x factors are all wonderful but are a poor substitute for a proper estimate.

Like many technical books this paints a picture that’s incomplete, but it contains so much wisdom, and is such an easy read that I recommend it highly. You may or may not be able to apply the techniques... but you’ll come away thinking deeply about better ways to create and manage a realistic schedule.

Then there’s the “CMMI Survival Guide,” by Suzanne Garcia and Richard Turner. It’s not a step-by-step guide to the Capability Maturity Model Integration at all. In fact, there’s virtually no description of the CMMI, and the authors assume the reader has pretty deep knowledge of the subject, which is described in detail in a 700 page doc (!) from the Software Engineering Institute. The book falters as a result; CMMI-novices wishing to “survive” making a transition to the CMMI will struggle to understand the authors’ advice.

The authors provide no compelling reasons to embrace the CMMI. Though heaps of praise for the process are given there are no actual or even anecdotal measures of how CMMI can help. (See <https://bscw.sei.cmu.edu/pub/bscw.cgi/0/395854> for some good data). You have to be pre-proselytized, so if you’re not already a CMMI convert you’ll quickly lose interest. For the book is not particularly engaging, and suffers from recursive book syndrome: you have to read the whole thing to get a sense of any part of it, a flaw the authors admit early on.

Each section starts with quotes that purport to be related to process improvement. Interestingly, these quotes are quite similar to the ones in Mike Cohn’s Agile Estimating book.

The book isn’t worth reading unless you’re going to start the CMMI. And even then read it only after you get some CMMI training, because the lingo will confuse novices. It has very good references to useful resources, and it gives a moderately useful, though tough to understand, roadmap to CMMI success.

This is not really like a self-help book; you’ll need a therapist (a process improvement consultant) as well.

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

If I were starting an outfit down the CMMI road I'd buy this book, and use it a little. Otherwise, give it a pass.

## Does Expensive = Good Quality?

Eric Shufro wrote: "I thought I might share an interesting story with you detailing my latest road trip in a \$180, 000 Bentley Continental; this is the second time actually.

"It seems that this particular car has a few serious software problems. In fact, some bugs are so critical that they are literally show stoppers.

"Last year, I was approximately 8 hours into the drive between New York and Florida when all of the dash board controls and indicators went nuts and locked up. Fortunately, the engine was still running, but I had no choice but to pull over and power cycle the ignition along with all accessories in order to continue safe operation of the vehicle.

"Last week however was a different story. I began the journey not wanting to use additional cell phone power since it had not been charged the day prior. Rather than pair my phone over blue tooth to the Bentleys in car cell phone, I chose to move my SIM card into the Bentleys SIM card slot. At first, this seemed like a brilliant and well thought through alternative in supplying a SIM card socket inside of the car. I inserted my SIM card into the socket and was greeted with a dash board message indicating that the cell phone was not currently available. I simply figured I was in a no service area. Soon thereafter, service became available and I called a few of my friends. It was not long however before service became unavailable once again. This seemed strange because I was not in a remote area and there were cell phone towers visible from the highway every few miles. Meanwhile, my mother was one car ahead and had coverage with the same cell phone provider. It became apparent that the cell phone application had crashed! I spent an hour trying to remove the SIM card while driving, but the SIM card socket appeared to be damaged and would not gracefully eject the SIM card. I stopped at a gas station to borrow a pair of pliers to remove the SIM card so that I could insert it back in to my cell phone. Service conveniently resumed. This was the only SIM card ever inserted into the socket and it clicked into place with ease. It felt as though the ejection spring was broken on the first insertion.

"Additionally, the Bentley uses a radio transponder as an alternative to a physical key to identify the owner of the car. All you need to do is be within close proximity of the vehicle with the key and you can unlock the doors, activate the accessories or start the vehicle. Sadly, recognition software is quite buggy and it would not recognize the transmitter while placed within the vehicle. Needless to say, I could not use this method

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

to start the car. The transmitter had plenty of battery power and was able to unlock and lock the doors from 25 feet away. So for laughs, I closed the doors, locked them with the transponder, unlocked them, opened the door, closed it and BINGO. The car was happy to recognize the presence of the transponder. This occurred 3 times over the length of the trip and was quite frustrating.

“Lastly, I won’t go into detail about how USELESS the navigation system is because it can not be programmed by an ordinary human. Not to mention the screen is a nice 7 or so inch display with a 2 by 2 navigation window in the center. I guess it’s more important to look at the ATM style buttons than the map anyway. What a waste.

“So after this long explanation, I feel that it is necessary for developers to take a step back and ask the question: Did we build it well? Sometimes building the right product, correctly, doesn’t mean that it is ready to be shipped.”

## **An Alternative Environment**

PC programmers might have product lifecycles measured in microseconds, but some embedded products stay viable. Yet during that time everything changes. Twenty years ago the PC of choice used a 286 processor running DOS. A decade later we were still stuck on early Pentiums (with the division bug just cleaned up) under the often unstable Windows 95.

Though I like Windows XP it’s awfully hard to checkpoint a development environment. How much of the project settings are buried in the registry? Will Vista, or some future descendant of Vista, run today’s tools?

Chris Pitcher wrote recently about a different approach to preserving a development environment, using the puppy distribution of Linux from Barry Kauler in Western Australia. “It is minimal in terms of size (~75M base) but pretty powerful in what it can do, and easily under-estimated.

“It has a couple of neat features. One is that it can be easily extended: copy any executables and libraries into the current directory, run the re-master script from the menu and you have a system with the additions you just made.

“It also allows you to save your settings and workfiles when you shut down. These are added as an extra track on the CD or DVD which gets pasted in next time you boot.

“So what we are doing at work is along these lines: To start ( e.g. an AVR project) take the master puppy-avr CD and boot.

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

“Use a memory stick for source/object/binary files while debugging.

“When the project is working to everyone's satisfaction, save the work files back onto the CD, file it away, and wait for the day when the system next needs to be upgraded/modified/got working again.

“Then you take the CD out and pick up where you left of.....

“Credit goes to Barry Kauler. It's worth reading his developer blog if you can find it... the whole site is pretty convoluted. He even took a PC on holidays, in a shack in the bush somewhere, to keep plugging away. (there's a picture of it.)

“Then Guido from tuxgraphics ( [www.tuxgraphics.org](http://www.tuxgraphics.org) ) who sells a knoppix CD with the AVR toolchain installed. AFAIK he was the first to do this, and he was my inspiration for modifying puppy; he also has a first rate collection of articles on his site, covering all sorts of practical details of working with AVR chips.”

That's pretty cool.

## Beautiful C++

Ralf Holly had this to say about nice C++ code: “If you really want to see amazing C++ code, have a look at the Loki library (<http://loki-lib.sourceforge.net/>) or read Andrei Alexandrescu's 'Modern C++ Design'. But be warned! It's not meant for C++ beginners...

“Unfortunately, really good C++ code is often perceived as cryptic or even 'disgusting' by beginners. It's a little bit like Single Malt Scotch Whisky: The first couple of times, you might not like it at all, but as you gain experience, you discover it's beauty and elegance and appreciate the subtle details and nuances of color, smell and taste.”

## Tools

Juergen Kemmler on tools: “The Editor I preferred is expensive, but ... Take a look on it. <http://www.RistanCASE.com>

“Polyspace is used to clean up our safety critical code, it works well. We perform "Module Tests" and project analysis.”

From a reader who wishes to remain anonymous: “In response to your question about static analysis tools, I'd just add that I've been using PolySpace for about 18 months now.

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

The results for C and Ada95 are better than C++ (this is possibly due to the language definition itself not lending itself to analysis).

“We can run 'whole program' analysis in both C and Ada95. Hints:

“1) add loads of memory (4GB)

“2) run on GNU/Linux, not Windows (40% speed improvement). The tool uses Cygwin on Windows and this seems to slow things down.

“But your question was really ‘does it work?’ The short-ish answer is ‘Yes, with some effort and no tool is perfect.’

“The tools understand some constructs better than others, so this is obviously easier to get right on 'blue sky' projects rather than 'legacy'.

“E.g. "Const correctness" is an absolute `_must_` in any library routines the unit under test calls. Without this PolySpace must assume the worst, e.g.

```
"int myLibRoutine(int* i); // no const correctness
// assume that 'i' can be changed to point to anything and
// that '*i' can be change to any value.
```

"Instead:

```
"int myLibRoutine(const int* const i); // with const correctness
// the value 'i' can not change and the value '*i' can not change
```

“Little things like this go a huge way to improving both the speed and quality of results. NB: it is also just 'best practice' coding style!

“Possibly one of the most telling stats is that we find loads more bugs in the C code than the Ada code. I can think of 1 bug that PolySpace uncovered in Ada code in all that time. It arguable that we'd have been a lot cheaper to not bother with C or PolySpace and just write everything in Ada!!”

Jeff Brentlinger is using the static analyzer from Klocwork: “We are early in the use of Klocwork ([www.klocwork.com](http://www.klocwork.com)), so I did not reply to your 1st request for people using Static Code Analysis Tools (I was hoping some of the other users would chime in).

“One of the problems we have had is resisting the urge to use this new tool on the current project that is in trouble. "Project X is late, so let's run this tool over the code...Great now we have 200 critical code issues on top of the 200 bugs reported by the test group (we won't even think about the 70,000 other issues the tool found).”

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

“Like any quality improvement tool/method/process, static analysis tools need to be use early in the project (built in quality).

“Having address that small part of the cultural issues, what about the tool itself? We have yet to have Klocwork flag an issue that was not at least "poor coding" (if that code is executed there will be a problem, but is that path ever executed?).

“One advantage of this type of tool (over lint or just turning on all the compiler warnings) is that the results are "public". Anyone on our network can log in to the system and see that Project X has 165 Critical Code Issues, far better than the situation where the compiler warnings (or lint report) just stay on the build machine seen by no one.”

## **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.

Applied Data Systems, Columbia MD, seeks Hardware/Electrical Engineers, WinCE Software Engineers and Embedded Linux Software Engineers. See <http://www.applieddata.net/careers.asp>

ADS is the leading supplier of ARM and Intel(r) XScale, single board embedded system computers for customers who need an application-ready Single Board Computer (SBC) running Windows CE, Embedded Linux or the operating system of choice.

To apply, please refer to <http://www.applieddata.net/careers.asp> and contact Deborah Grenier, ADS Human Resources Director, at [dgrenier@applieddata.net](mailto:dgrenier@applieddata.net)

Nautronix plc is recruiting Embedded Software Engineers, to develop its range of innovative sub-sea acoustic positioning and communication technologies. Positions are offered in Aberdeen, Scotland

This is an excellent opportunity for engineers with signal processing experience to work on leading edge underwater acoustic technology

We need software & firmware developers with experience ranging from microcontrollers to Windows & Linux-based embedded PCs, programming in C/C++, VB, .net, perl, VHDL

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

For more details, and a short presentation on the technology, see <http://tinyurl.com/ygko74> .

Principal Quality Assurance Engineer, Emerson Process Management - Asset Optimization, Eden Prairie, MN.

Support the enhancement of engineering development processes, work practices, and procedures with a goal of achieving a high level of product integrity while supporting a time-to-market strategy by:

- Working with Emerson Center Of Excellence Quality Center
- Drive continuous improvement activities
- Perform project and quality audits
- Facilitate actions to address customer reported product issues
- Coordinator for customer and outside agency audits
- Promote and ensure conformance to the Quality system

Requirements:

- 3 plus years experience of SEI, CMM and CMMI
- Practical experience in doing Software Development, Test and PM
- Practical experience in assessment and auditing
- Practical experience in defining and implementing metrics programs
- Familiarity with quality improvement Tools
- Certified in Software Quality Assurance (CSQE) desired
- 4 year degree in Engineering or Computer Science or equivalent and six years related experience.

Send your resume to [Deanne.Weber2@EmersonProcess.com](mailto:Deanne.Weber2@EmersonProcess.com)

## **Joke for the Week**

From Catherine French:

Kauffman's Paradox of the Corporation - The less important you are to the corporation, the more your tardiness or absence is noticed.

The Salary Axiom -

The pay raise is just large enough to increase your taxes and just small enough to have no effect on your take-home pay.

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

Weiner's Law of Libraries -

There are no answers, only cross-references.

Kenny's Law of Auto Repair -

The part requiring the most consistent repair or replacement will be housed in the most inaccessible location.

Second Law of Business Meetings -

If there are two possible ways to spell a person's name, you will pick the wrong one.

-Corollary - If there is only one way to spell a name, you will spell it wrong anyway.

Lampner's Law of Employment -

When leaving work late, you will go unnoticed. When you leave work early, you will meet the boss in the parking lot.

Quile's Consultation Law -

The job that pays the most will be offered when there is no time to deliver the services.

Loftus's Law -

Some people manage by the book, even though they don't know who wrote the book or even what book.

Van Gogh's Law -

Whatever plan one makes, there is a hidden difficulty somewhere.

Troutman's Fifth Programming Postulate -

If the input editor has been designed to reject all bad input, an ingenious idiot will discover a method to get bad data past it.

Lerman's Law of Technology -

Any technical problem can be overcome given enough time and money. - Corollary -

You are never given enough time or money.

Lovka's Advice -

Never rely on a person who uses "party" as a verb.

## **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](mailto:jack@ganssle.com).

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

To subscribe, send a message to [majordomo@ganssle.com](mailto: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*”. ". BUT - please use YOUR email address in place of “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](mailto: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](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**