2006 Salary Survey
Of Embedded Systems Engineers

Jack G. Ganssle
jack@ganssle.com

The Ganssle Group
PO Box 38346
Baltimore, MD 21231

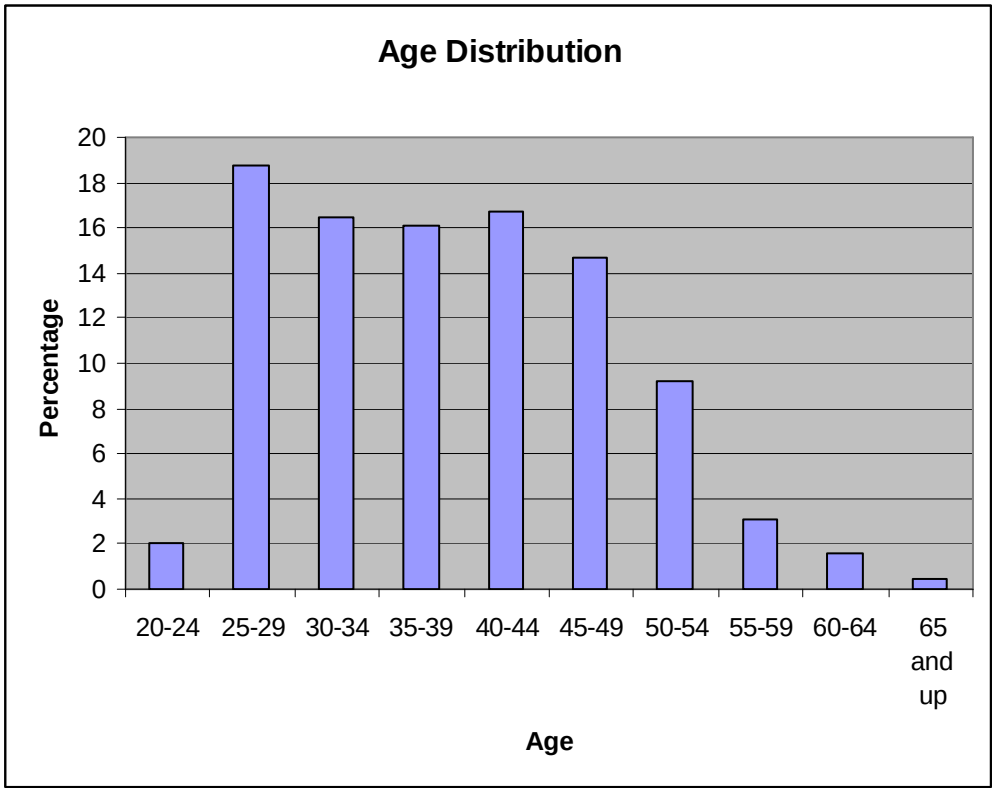
Every year Software Development Magazine runs a salary survey for programmers. Unfortunately that genre includes embedded developers, IT programmers, PC developers and even web designers. No one studies the embedded systems population, which is quite distinct from other dialects of developers.

On December 11, 2006 I sent a request to participate in an embedded survey to the approximately 18,000 readers of The Embedded Muse. 491 went to the web page and submitted data.

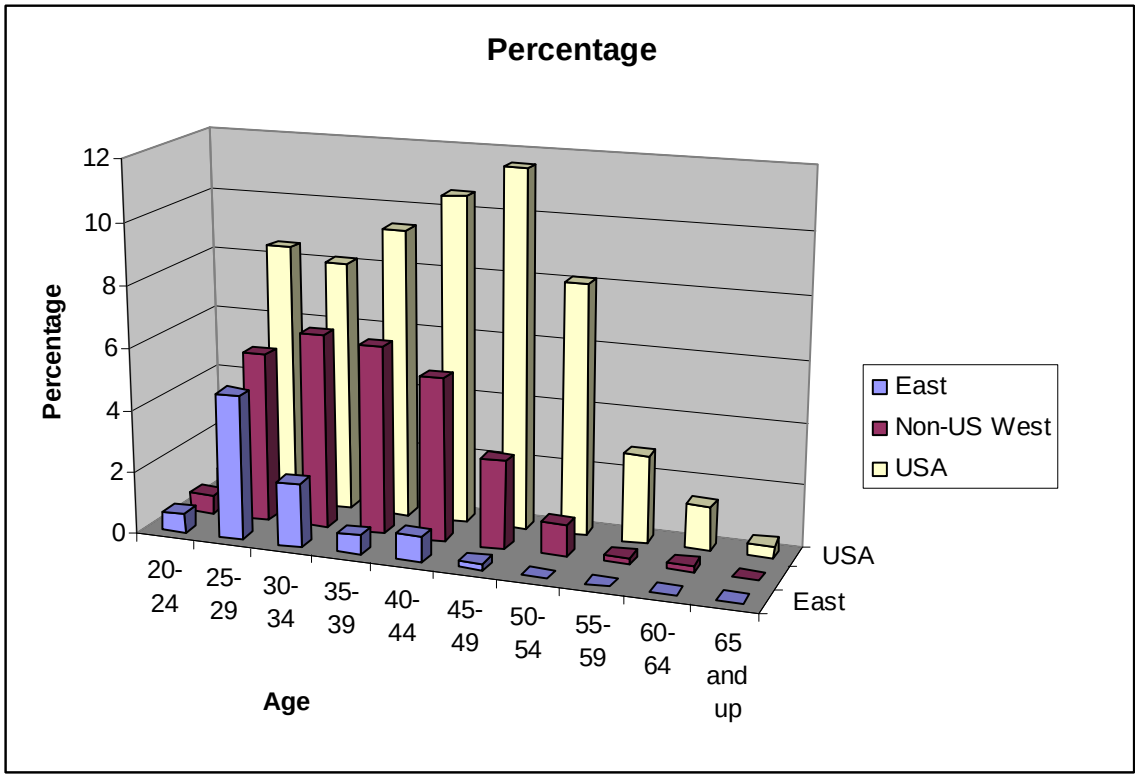
This is hardly a scientific survey; no tests were performed to insure the accuracy of the data. It's likely that more than a few errors were made, especially since overseas respondents were asked to convert their wages into US Dollars. But the data is interesting and suggestive.

Age

Most embedded developers are young to early middle age, with the biggest spike in the late 20s. By age 50 few remain in this field. One wag listed his age as "50 (oh God!)". The average age is 38.9 years.

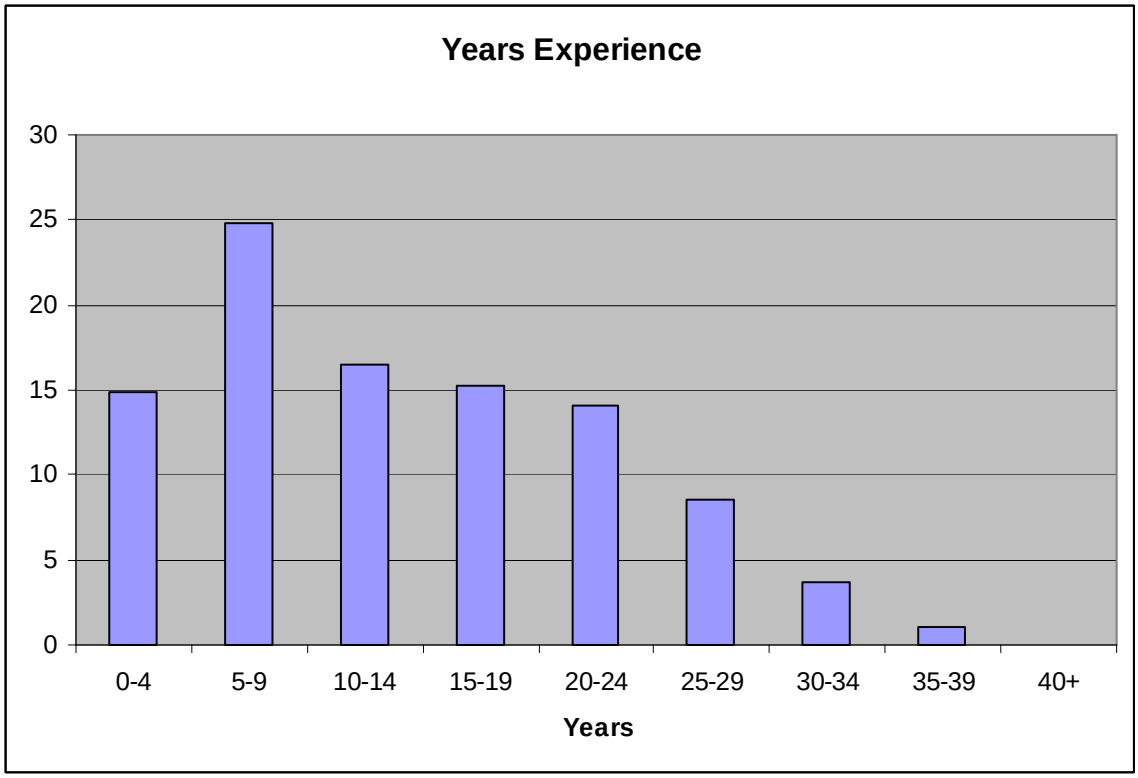


In my recent travels I've noticed that engineers in US and other "western" countries (Australia, Europe, etc) seem to be older than those in places like India and Asia. To try and get some insight I divided the world into three camps: the "east," "western non-US," and the USA. Apologies to all for being so western-centric in making such arbitrary divisions, but the numbers are indeed interesting. We in the United States are graying compared to the rest of the world. Eastern countries have by far a younger group of engineers.



Experience

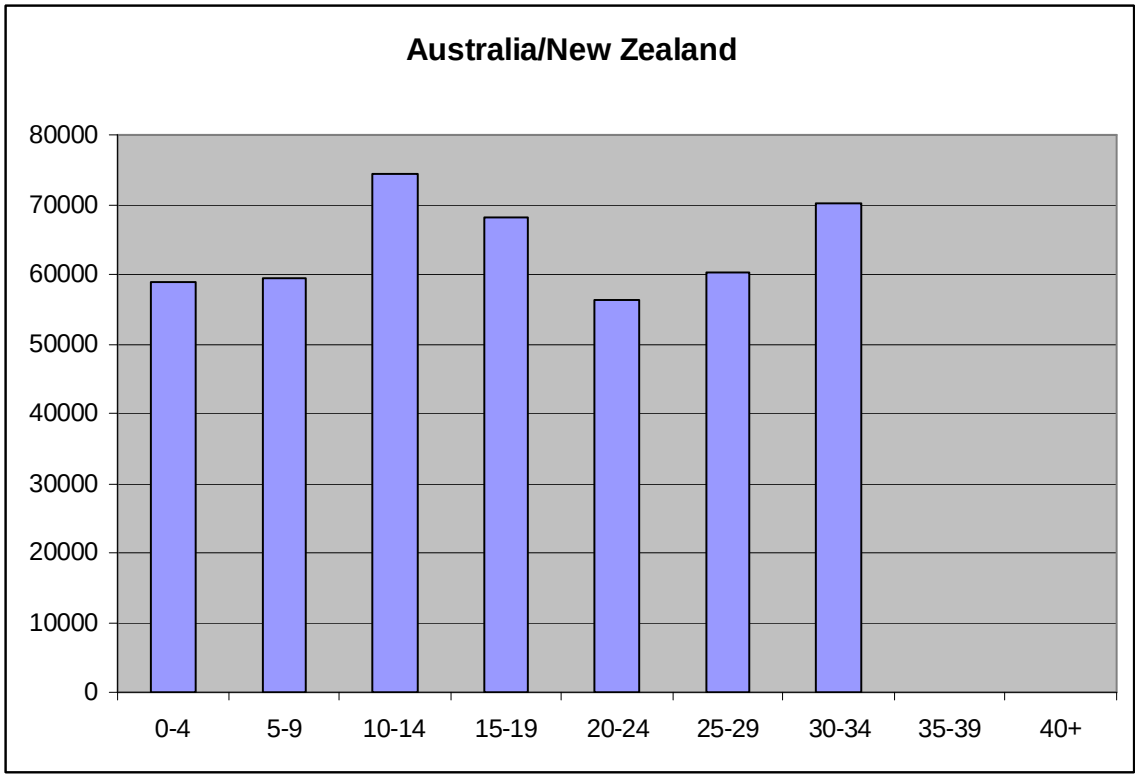
Not surprisingly, experience levels tracked ages closely, suggesting relatively few developers enter the field later in their careers. The average developer has 13.4 years of experience.

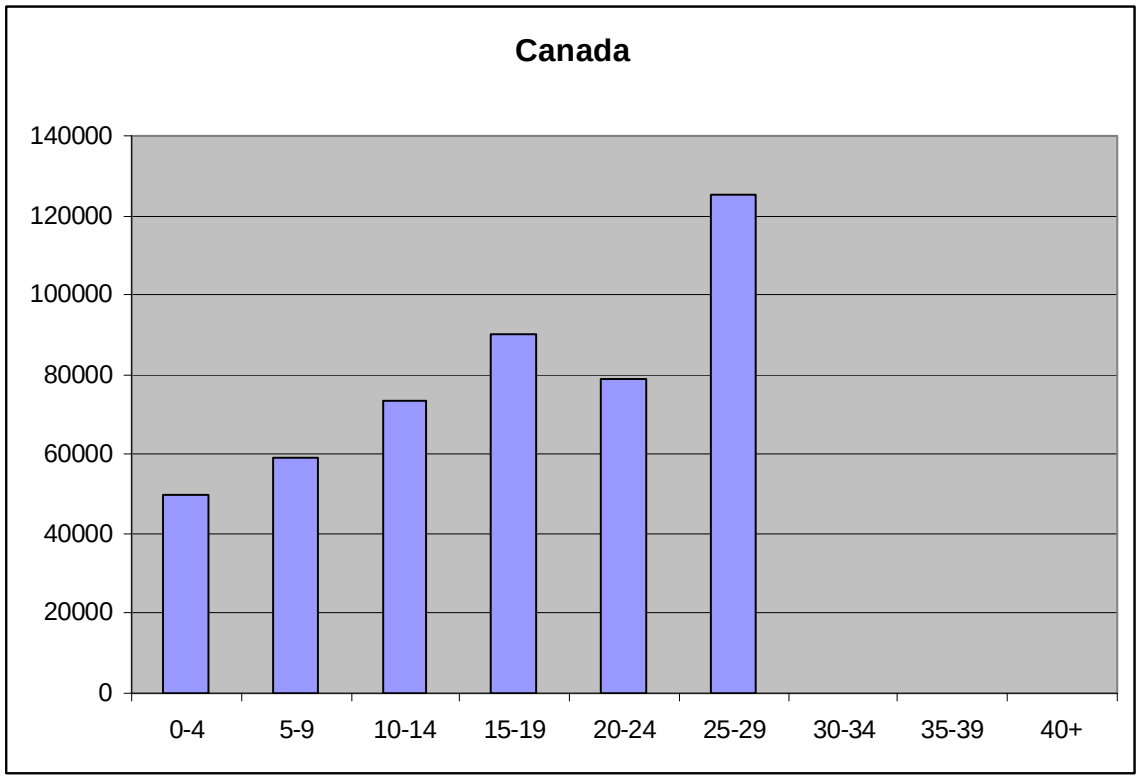


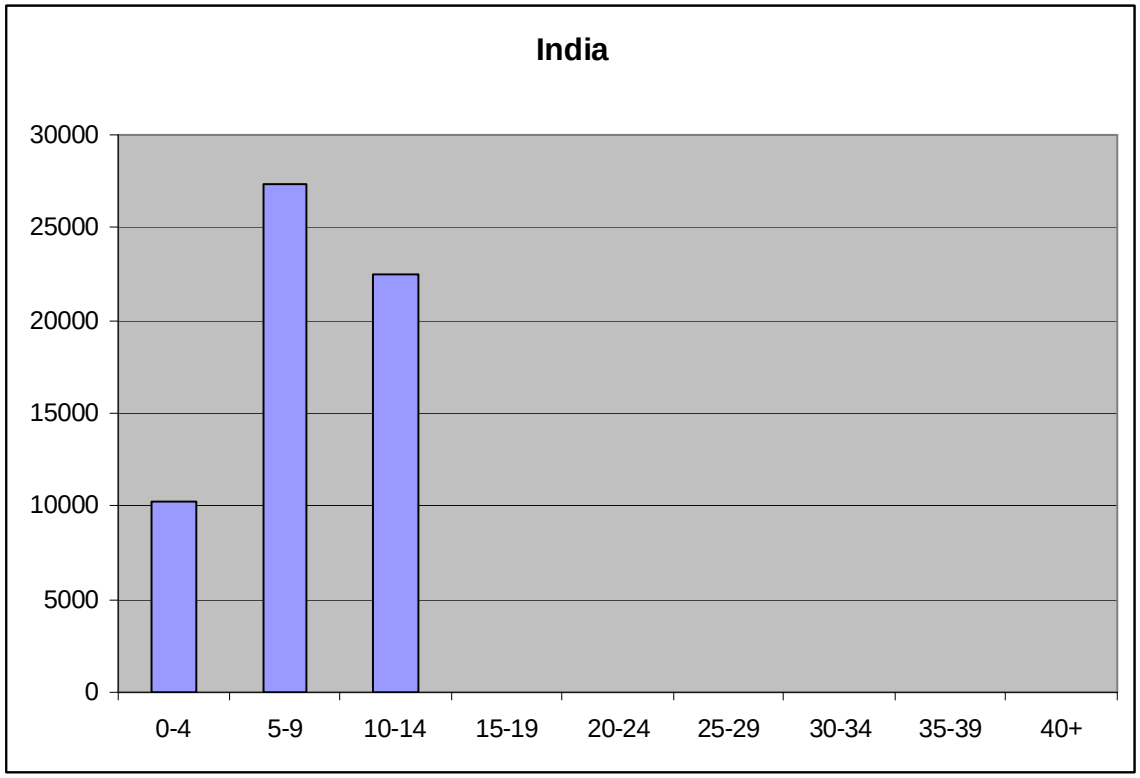
Salary

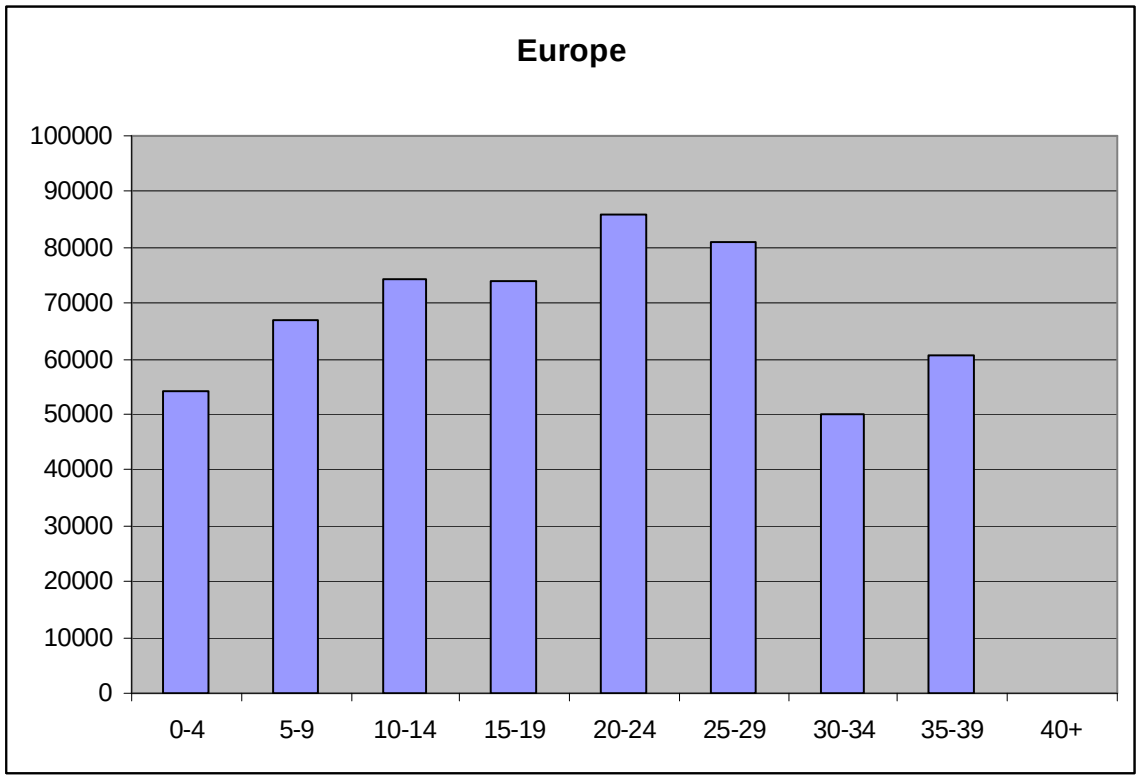
Salaries vary widely around the world, and by experience. The following graphs show experience in years versus salary in US dollars for a number of regions. Other areas had too few data points to be meaningful. I tossed out a few datapoints that failed the common sense test (like a $100k+ wage in India… though possible for some ex-pat, it skewed the data).

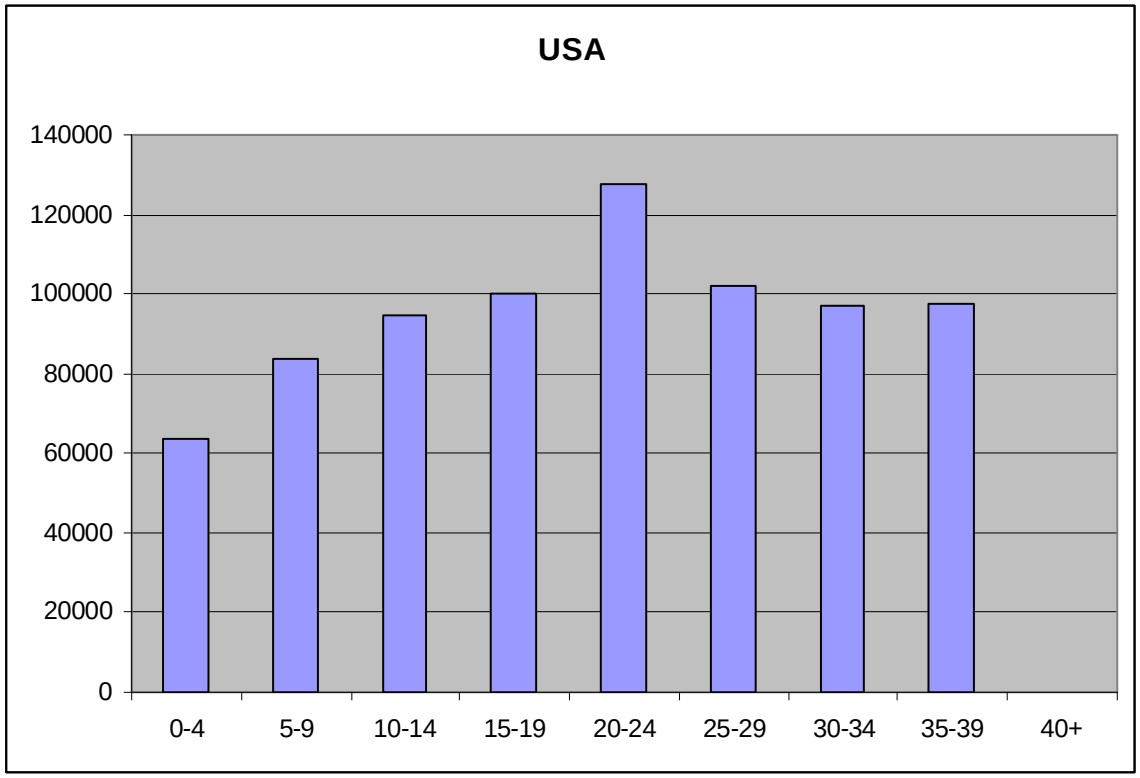
A couple of people complained that the data should be adjusted by region in the USA. That's a good point… but I didn't collect such data.







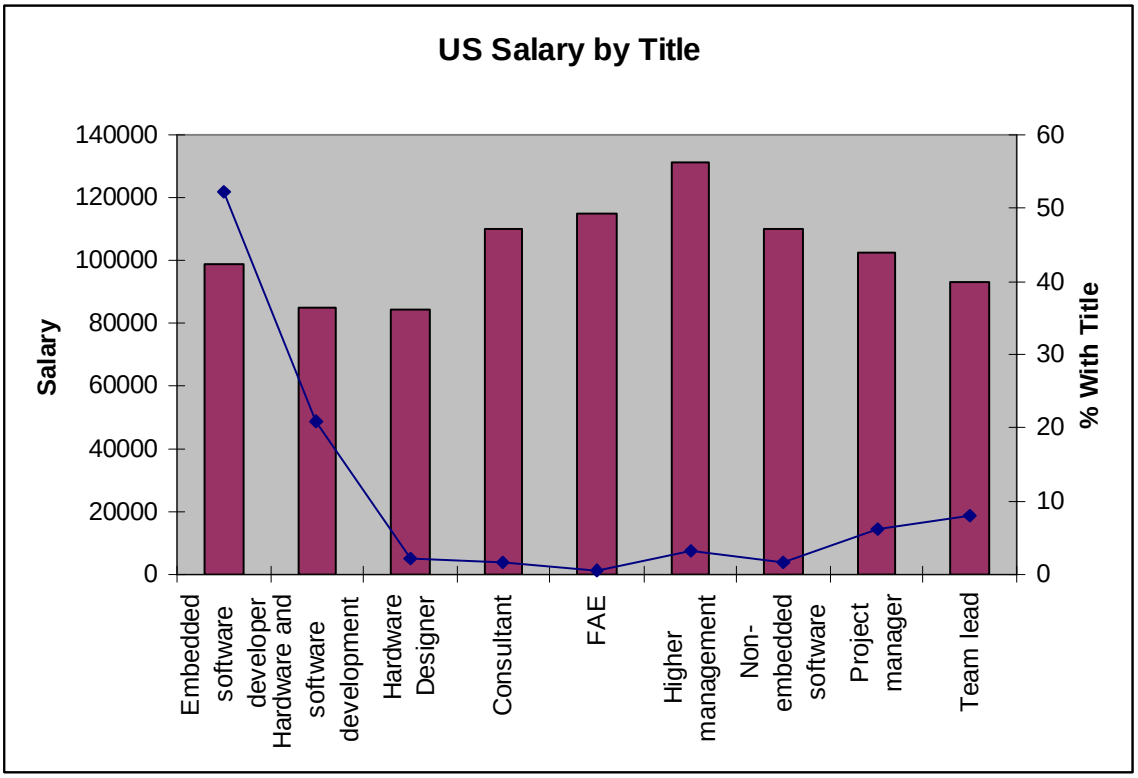




Salary by Title

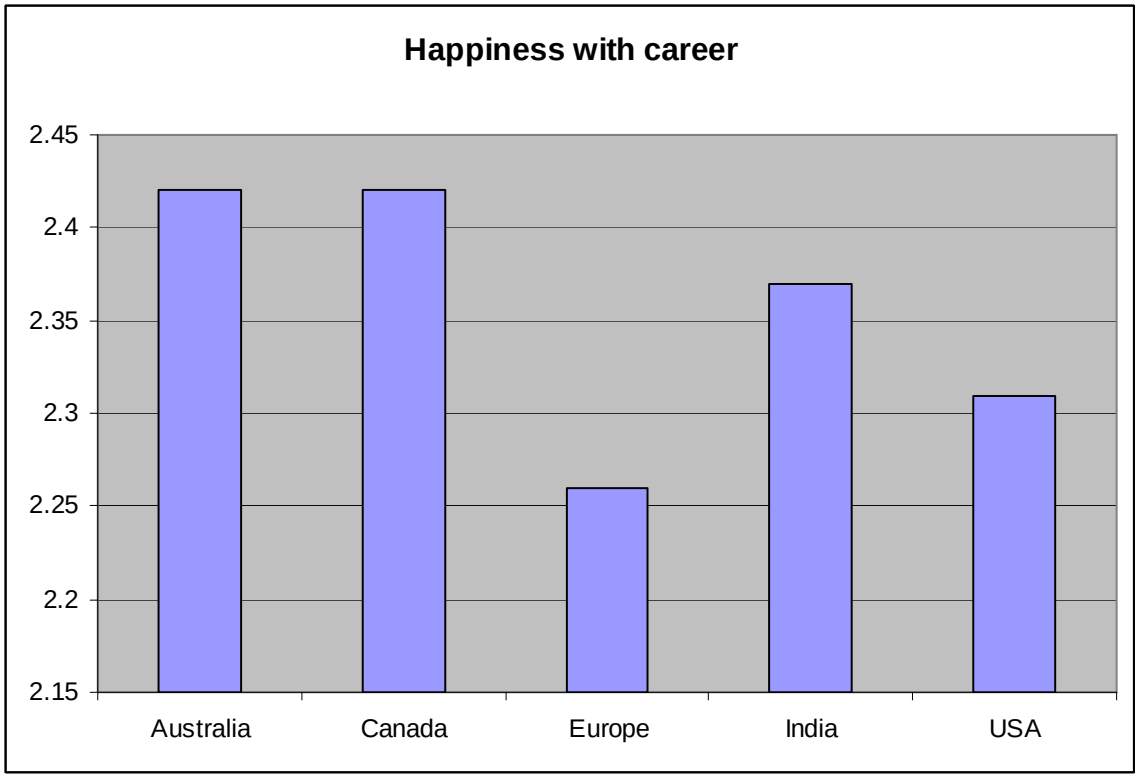
Interestingly, in Canada, Europe, Australia/New Zealand, and India most respondents develop embedded software. A few do hardware and practically none claimed to be in any sort of management role. There wasn't enough disparity in jobs to get a meaningful correlation between salaries and title. The USA had more useful data though, as shown in the following graph.

52% of respondents are embedded software developers; 21% do both hardware and software. Discouragingly, non-embedded programmers make more than firmware folks!



Happiness

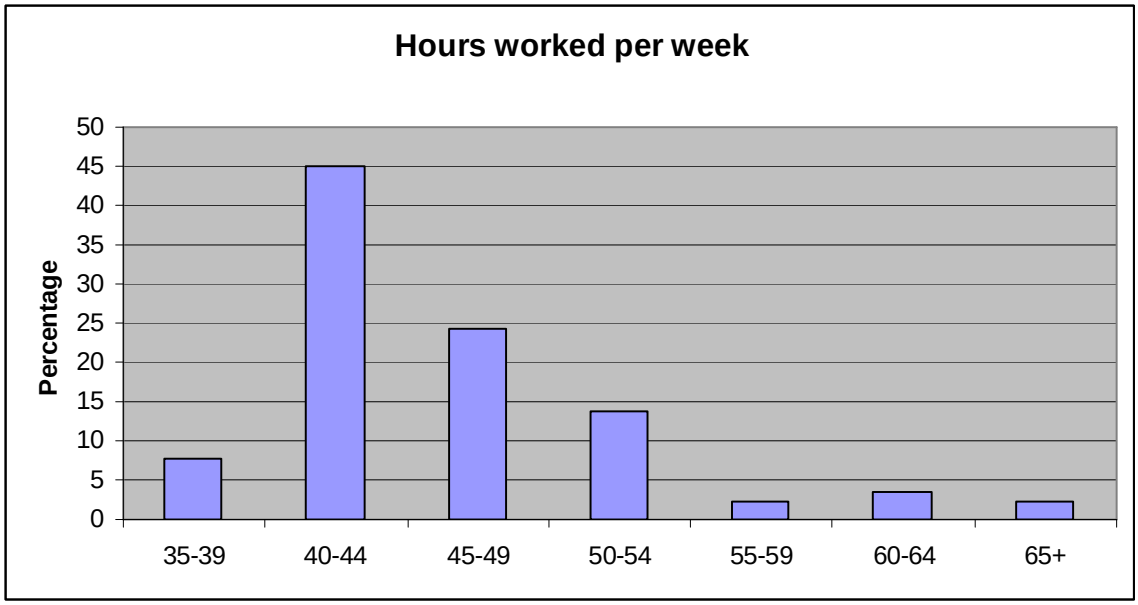
Perhaps attempting to measure happiness is a fool's quest, but as a middle-aged gent who has seen too many colleagues burn out from despair and overwork I'm convinced we *must* pursue happiness first and salary second. In the survey respondents rated their happiness with their career on an enumerated scale of [love it, reasonably happy, somewhat unhappy, hate it]. I rated the factors from 3 (love it) to 0 (hate it). Those results were normalized to the number of responses in each category.



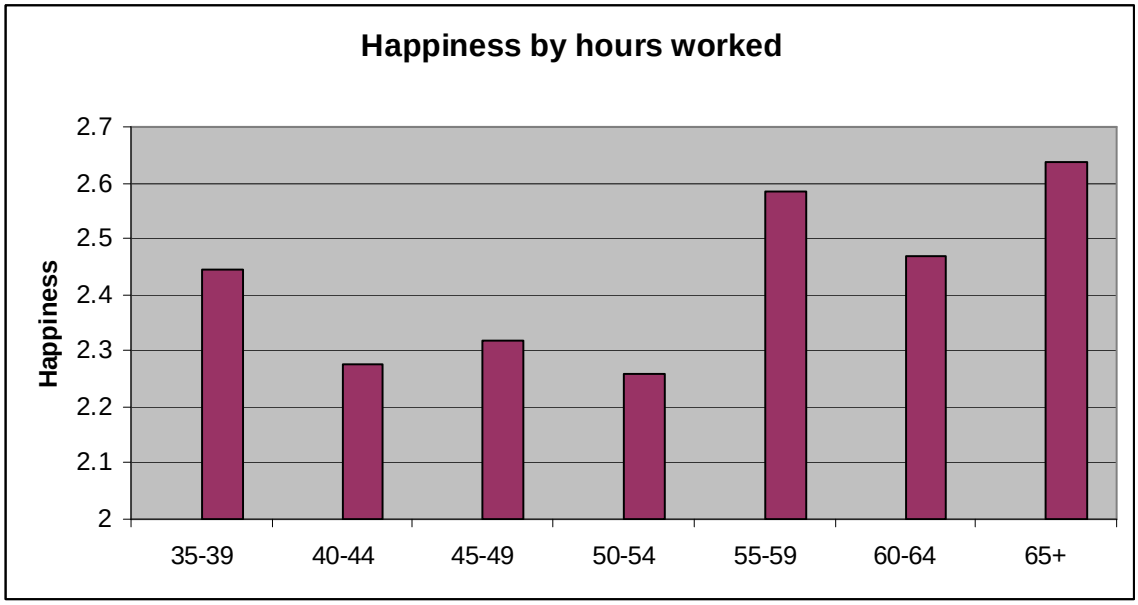
In terms of the current job, 9.8% of us are looking for another job; 12.2% report being unhappy in it, and a whopping 77.5% are happy.

Hours Worked

We grumble a lot about overtime, but 80% of us work less than 50 hours per week.



Perversely, the happiest engineers work the most hours!



The Future

I asked how people felt about the future of engineering, giving four possible responses:

- Expect a strong demand for engineers
- About the same
- Demand likely to diminish
- It's likely to be offshored
- I'm hoping it's offshored to us!

The answers tabulate as follows:

	USA	Canada	Australia/New Zealand	India	Europe
Expect a strong demand for engineers	31%	37%	58%	76%	61%
About the same	33%	26%	21%	3%	44%
Demand likely to diminish	21%	26%	21%	3%	15%
It's likely to be offshored	14%	0%	0%	5%	5%
I'm hoping it's offshored to us!	1%	0%	0%	14%	3%

Comments

Many respondents made comments, some of which are listed here:

Although I am burned out, it is a great area and bound to expand as more and more devices embed computers.

As I get closer and closer to 50, I feel less secure in my job and what my options would be if I lost it.

Avoid it, go to medical school

Based on what I have seen in the last 20 years, I probably would not recommend engineering to my son.

Been laid off twice in four years, my salary has been stagnant over that time.

Engineering has a poor social status in most English speaking countries, however in non-English speaking countries e.g.: Germany, they have a very high status similar to that of Doctors.

H1B workers and offshoring are the greatest threat to engineering in the USA. I have told my children to NOT go into engineering for this reason.

Hanging on until my job is outsourced.

High demand for EEs and Computer Scientists in British Columbia, Canada.

I am a consultant -- After marketing, admin, travel, bookeeping, etc, I am able to bill about 25 hours per week -- less hours at beginning of year, more hours at end of year.

I am from New Zealand and have been living in the United Kingdom for 2 years. I think electronic engineers here are really feeling the offshoring pains (seems to be later here than other places in the world) but they have not realised its not all going and that there is some work that should never go and will come back. There seems to be a pessimistic outlook in England, but I think it will pass once everything settles down (global thinking). I do exactly the same job in London as I did in NZ, but get paid about 2.5 times as much... how can these companies remain competitive? Every engineer I have met from any country is basically as competent and any other... but there are differences and that's where your job security is.

I love being an engineer. However, I couldn't recommend engineering as a career. Companies can't seem to move engineering functions out of the country fast enough. There is absolutely no job security any longer.

I moved from medical to automotive and I am amazed at the amount of auto code that is being generated

I think our colleges and universities need to do a better job training students for real careers in engineering. Certainly there's a place for the advanced theory and certainly it's important to learn how to think. But practical, real-world, hands-on engineering experience is important too. Our schools should require some type of co-op experience or other hands-on training. If I had it to do over again, I would take more courses with labs and would co-op.

I think that I'm happy because of the people I work with, and because it's a small company. I'd gladly give up some $$$/yr to be in a smaller company where I can become 'the expert' on just about anything. That's why I love my job.

I think that one has to be careful when comparing salaries across different countries. I have done a lot of work with Swedish engineers and they were earning roughly 3-4 times what we were in SA but their standard of living was very comparable mainly due to the high cost of living in Stockholm.

I dearly love the field of embedded design, but employment has been rough and I wouldn't recommend this line of work to anyone unless they're a) passionate about the work, and b) very, very talented.

I understand that companies must be profitable or they cease to exist. US companies can hire engineers from India, China and Eastern Europe for less than they would pay a US engineer. As one of my friends said, 'We are in a race to the bottom.' How long will it be before there ARE no more US engineers? I do not have the answers. I am concerned about the future.

'd like to see data from Japanese engineers. From what I understand they've got an entirely different business model where engineers are truly empowered...

I'm a Senior Software Engineer in a very large aerospace company. I also have team leadership responsibility. Most engineers here watch more and more of our work moving offshore and we are just guessing at how long we can keep some technical work in-house. The Commercial work is already gone and my people have a temporary protection due to our US Military work but that is eroding quickly as even US Military work is moving out under 'export licenses.' I love embedded systems but I'm going to have to do something else soon. It's clear that in a few years that work won't exist internally at my company.

It seems that the trend towards moving development off-shore at any cost is letting up. In my opinion this is due to the limited success these projects have had (read failures). Off-shoring will continue, but hopefully will be tempered with some realism for a change.

'Offshoring' is a natural 'supply and demand' economic trend. However, sometimes organizations overlook the 'cost of low-cost' in terms of learning curve, (potential) reduced quality due to lack of experience or miscommunication, and employee turn-over rate in 'low-cost' countries. For those of us 'On shore' the challenge remains to keep ourselves valuable. That means we probably should not plan on 'coding' long term, and should migrate toward system specification, architecture and design, quality assurance ('testing', etc.), and customer-focused activities.

Since engineers in India cost 1/4 or less of a US employee, can we assume this holds true for other jobs in the technology industry? If so, companies can save a lot more money by outsourcing the CxO's and their HUGE fat-cat bonuses and magically-timed stock options.

There is no engineering/software developer shortage and never has been. Just corporate and university propaganda to boost supply and 'customers', respectively. The only time the shortage shouting diminishes is when the unemployment rate during recessions rises so blatantly high that such comments become patently ludicrous even to casual observers. (Sometimes the terminology then switches from finding 'qualified' to 'highly qualified' employees. The worse the recession, the more difficulty companies claim to have finding 'good people'.) If there is such a terrible techie shortage, why aren't salaries rising? Why has the average time needed to secure another job risen to the highest levels since WWII?

Very concerned about outsourcing in the short term. I'm reasonably certain that in the long run, companies will find it doesn't work. But until then, we're in for a rocky ride. Education is a problem at all levels from K-12 right through the number of PhDs. Engineers need to organize! At least politically.

Wage stagnation and even deflation, along with the associated offshoring of jobs from the US are a huge concern for me. Unsure of how to reverse the trend. Considering subsistence farming as a second career.

Better Firmware... *Faster!*

A One-Day Seminar

Presented at

Your Company

Does your schedule prevent you from traveling?

This doesn't mean you have to pass this great opportunity by.

Presented by Jack Ganssle, technical editor of *Embedded Systems Programming Magazine*, **author** of *6 books and over 600 articles*

More information at www.ganssle.com

The Ganssle Group
PO Box 38346
Baltimore, MD 21231
(410) 504-6660
fax: (647) 439-1454
info@ganssle.com
www.ganssle.com

For Engineers and Programmers

This seminar will teach you new ways to build higher quality products in half the time.

80% of all embedded systems are delivered late…
Sure, you can put in more hours. Be a hero. But *working harder is not a sustainable way to meet schedules.* We'll show you how to plug productivity leaks. How to manage creeping featurism. And ways to balance the conflicting forces of schedules, quality and functionality.

… yet it's not hard to double development productivity
Firmware is the most expensive thing in the universe, yet we do little to control its costs. Most teams deliver late, take the heat for missing the deadline, and start the next project having learned nothing from the last. Strangely, *experience* is not correlated with *fast*. But *knowledge* is, and we'll give you the information you need to build code more efficiently, gleaned from hundreds of embedded projects around the world.

Bugs are the #1 cause of late projects…
New code generally has *50 to 100 bugs* per thousand lines. Traditional debugging is the *slowest* way to find bugs. We'll teach you better techniques proven to be up to 20 times more efficient. And show simple tools that find the nightmarish real-time problems unique to embedded systems.

… followed by poor scheduling
Though capricious schedules assigned without regard for the workload are common, even developers who make an honest effort usually fail. We'll show you how to decompose a product into schedulable units, and how to use killer techniques like Wideband Delphi to create more accurate estimates.

Learn From The Industry's Guru

Spend a day with Jack Ganssle, well-known author of the most popular books on embedded systems, technical editor and columnist for Embedded Systems Programming, and designer of over 100 embedded products. You'll learn new ways to produce projects *fast* without sacrificing quality. This seminar is the only non-vendor training event that shows you *practical* solutions that you can implement *immediately*. We'll cover technical issues – like how to write embedded drivers and isolate performance problems – as well as practical process ideas, including how to manage your people and projects. *Contact us to learn how we can award each of the attendees 0.7 Continuing Education Units.!*

Seminar Leader

Jack Ganssle has written over 600 articles in Embedded Systems Programming, EDN, and other magazines. His five books, **The Art of Programming Embedded Systems**, **The Art of Developing Embedded Systems, The Embedded Systems Dictionary, The Firmware Handbook,** and **Embedded Systems, World Class Designs** are the industry's standard reference works

Jack lectures internationally at conferences and to businesses, and was this year's keynote speaker at the Embedded Systems Conference. He founded three companies, including one of the largest embedded tool providers. His extensive product development experience forged his unique approach to building better firmware faster.

Jack has helped over 600 companies and thousands of developers improve their firmware and consistently deliver better products on-time and on-budget.

Course Outline

Languages

- C, C++ or Java?
- Code reuse – a myth? How can you benefit?
- Controlling stacks and heaps.

Structuring Embedded Systems

- *Manage* features… or miss the schedule!
- Using multiple CPUs.
- Five design schemes for faster development.

Overcoming Deadline Madness

- Negotiate realistic deadlines… or deliver late.
- Scheduling – the science versus the art.
- Overcoming the biggest productivity busters.

Stamp Out Bugs!

- Unhappy truths of ICEs, BDMs, and debuggers.
- *Managing* bugs to get good code fast.
- *Quick* code inspections that keep the schedule on-track.
- Cool ways to find hardware/software glitches.

Managing Real-Time Code

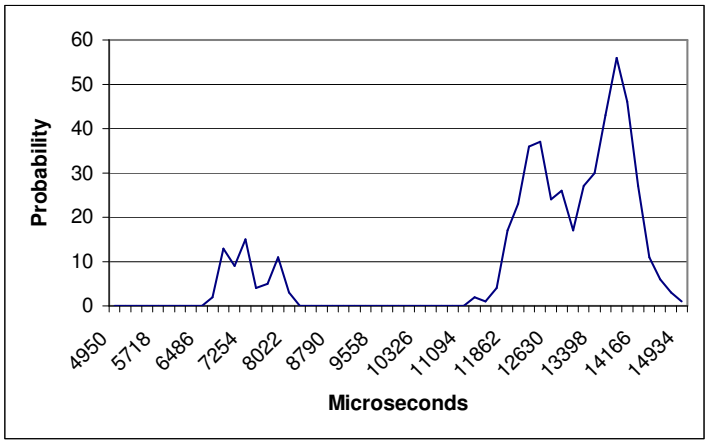
- Design *predictable* real-time code.
- Managing reentrancy.
- Troubleshooting and eliminating *erratic crashes.*
- Build better interrupt handlers.

Interfacing to Hardware

- Understanding high-speed signal problems.
- Building peripheral drivers faster.
- Inexpensive performance analyzers.

How to Learn from Failures… and Successes

- Embedded disasters, and *what we must learn*.
- Using postmortems to accelerate the product delivery.
- Seven step plan to firmware success.



Do those C/C++ runtime routines execute in a usec or a week? This trig function is all over the map, from 6 to 15 msec. You'll learn to rewrite real-time code proactively, anticipation timing issues before debugging.

Why Take This Course?

Frustrated with schedule slippages? Bugs driving you batty? Product quality sub-par? **Can you afford *not* to take this class?**

We'll teach you how to get your products to market faster with fewer defects. Our recommendations are *practical*, *useful today*, and *tightly focused* on embedded system development. Don't expect to hear another clever but ultimately discarded software methodology. You'll also take home a 150-page handbook with algorithms, ideas and solutions to common embedded problems.

If you can't take the time to travel, we can present this seminar at your facility. We will train all of your developers and focus on the challenges unique to your products and team.

Here is what some of our attendees have said:

Thanks for the terrific seminar here at ALSTROM yesterday!
It got rave reviews from a pretty tough crowd.
Cheryl Saks, ALSTROM

Thanks for a valuable, pragmatic, and informative lesson in embedded systems design.
All the attendees thought it was well worth their time.
Craig DeFilippo, Pitney Bowes

I just wanted to thank you again for the great class last week. With no exceptions, all of the feedback from the participants was extremely positive. We look forward to incorporating many of the suggestions and observations into making our work here more efficient and higher quality.
Carol Bateman, INDesign LLC

Here are just a few of the companies where Jack has presented this seminar:
Sony-Ericsson, Northup Grumman, Dell, Western Digital, Bayer, Seagate, Whirlpool, Cutler Hammer, Symbol, Visteon, Honeywell, Kodak and Western Digital.

Did you know that…

***… doubling the size of the code results in much more than twice the work*?** In this seminar you'll learn ways unique to embedded systems to partition your firmware to keep schedules from skyrocketing out of control.

… you can reduce bugs by an order of magnitude before starting debugging? Most firmware starts off with a 5-10% error rate – 500 or more bugs in a little 10k LOC program. Imagine the impact finding all those has on the schedule! Learn simple solutions that don't require revolutionizing the engineering department.

***… you can create a predictable real-time design*?** This class will show you how to measure the system's performance, manage reentrancy, and implement ISRs with the least amount of pain. You'll even study real timing data for common C constructs on various CPUs.

… a 20% reduction in processor loading slashes development time? Learn to keep loading low while simplifying overall system design.

… reuse is usually a waste of time? Most companies fail miserably at it. Though promoted as the solution to the software crisis, real reuse is much tougher than advertised. You'll learn the ingredients of successful reuse.

What are you doing to upgrade your skills? What are you doing to help your engineers succeed? Do you consistently produce quality firmware on schedule? *If not . . . what are you doing about it?*

Contact us for info on how we can bring this seminar to your company.
e-mail: info@ganssle.com or call us at 410-504-6660.