

More Joel on Software: Further Thoughts on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity

Copyright © 2008 by Joel Spolsky

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (paperback): 978-1-4302-0987-4

ISBN-13 (electronic): 978-1-4302-0988-1

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Jeffrey Pepper

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Associate Publisher | Project Manager: Grace Wong

Senior Copy Editor: Ami Knox

Associate Production Director: Kari Brooks-Copony

Production Manager: Kelly Winkquist

Compositor: Dina Quan

Proofreader: April Eddy

Indexer: Broccoli Information Management

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

part one

Managing People

one

MY FIRST BILLG REVIEW

FRIDAY, JUNE 16, 2006

In the olden days, Excel had a very awkward programming language without a name. “Excel Macros,” we called it. It was a severely dysfunctional programming language without variables (you had to store values in cells on a worksheet), without locals, without subroutine calls: in short, it was almost completely unmaintainable. It had advanced features like “Goto,” but the labels were actually physically invisible.

The only thing that made it appear reasonable was that it looked great compared to Lotus macros, which were nothing more than a sequence of keystrokes entered as a long string into a worksheet cell.

On June 17, 1991, I started working for Microsoft on the Excel team. My title was Program Manager. I was supposed to come up with a solution to this problem. The implication was that the solution would have something to do with the Basic programming language.

Basic? Yech!

I spent some time negotiating with various development groups. Visual Basic 1.0 had just come out, and it was pretty friggin’ cool. There was a misguided effort underway with the code name MacroMan, and another effort to make Object-Oriented Basic code-named Silver. The Silver team was told that they had one client for their product: Excel. The marketing manager for Silver, Bob Wyman, yes, that Bob Wyman, had only one person he had to sell his technology to: me.

MacroMan was, as I said, misguided, and it took some persuading, but it was eventually shut down. The Excel team convinced the Basic team that what we really needed was some kind of Visual Basic for Excel. I managed to get four pet features added to Basic. I got them to add Variants, a union data type that could hold any other type, because

otherwise you couldn't store the contents of a spreadsheet cell in a variable without a switch statement. I got them to add late binding, which became known as IDispatch, a.k.a. COM Automation, because the original design for Silver required a deep understanding of type systems that the kinds of people who program macros don't care about. And I got two pet syntactic features into the language: `For Each`, stolen from `csh`, and `With`, stolen from Pascal.

Then I sat down to write the Excel Basic spec, a huge document that grew to hundreds of pages. I think it was 500 pages by the time it was done. ("Waterfall," you snicker. Yeah, yeah, shut up.)

In those days, we used to have these things called BillG reviews. Basically, every major important feature got reviewed by Bill Gates. I was told to send a copy of my spec to his office in preparation for the review. It was basically one ream of laser-printed paper.

I rushed to get the spec printed and sent it over to his office.

Later that day, I had some time, so I started working on figuring out if Basic had enough date and time functions to do all the things you could do in Excel.

In most modern programming environments, dates are stored as real numbers. The integer part of the number is the number of days since some agreed-upon date in the past, called the epoch. In Excel, today's date, June 16, 2006, is stored as 38884, counting days where January 1, 1900, is 1.

I started working through the various date and time functions in Basic and the date and time functions in Excel, trying things out, when I noticed something strange in the Visual Basic documentation: Basic uses December 31, 1899, as the epoch instead of January 1, 1900, but for some reason, today's date was the same in Excel as it was in Basic.

Huh?

I went to find an Excel developer who was old enough to remember why. Ed Fries seemed to know the answer.

"Oh," he told me. "Check out February 28, 1900."

"It's 59," I said.

"Now try March 1."

"It's 61!"

"What happened to 60?" Ed asked.

"February 29, 1900, was a leap year! It's divisible by 4!"

“Good guess, but no cigar,” Ed said, and left me wondering for a while.

Oops. I did some research. Years that are divisible by 100 are *not* leap years, unless they’re also divisible by 400.

1900 wasn’t a leap year.

“It’s a bug in Excel!” I exclaimed.

“Well, not *really*,” said Ed. “We had to do it that way because we need to be able to import Lotus 123 worksheets.”

“So, it’s a bug in Lotus 123?”

“Yeah, but probably an intentional one. Lotus had to fit in 640K. That’s not a lot of memory. If you ignore 1900, you can figure out if a given year is a leap year just by looking to see if the rightmost two bits are zero. That’s really fast and easy. The Lotus guys probably figured it didn’t matter to be wrong for those two months way in the past. It looks like the Basic guys wanted to be anal about those two months, so they moved the epoch one day back.”

“Aargh!” I said, and went off to study why there was a check box in the options dialog called “1904 Date System.”

The next day was the big BillG review.

June 30, 1992.

In those days, Microsoft was a lot less bureaucratic. Instead of the eleven or twelve layers of management the company has today, I reported to Mike Conte, who reported to Chris Graham, who reported to Pete Higgins, who reported to Mike Maples, who reported to Bill. About six layers from top to bottom. We made fun of companies like General Motors with their eight layers of management or whatever it was.

In my BillG review meeting, the whole reporting hierarchy was there, along with their cousins, sisters, and aunts, and a person who came along from my team whose whole job during the meeting was to keep an accurate count of how many times Bill said the F word. The lower the f*** count, the better.

Bill came in.

I thought about how strange it was that he had two legs, two arms, one head, etc., almost exactly like a regular human being.

He had my spec in his hand.

He had my spec in his hand!

He sat down and exchanged witty banter with an executive I did not know that made no sense to me. A few people laughed.

Bill turned to me.

I noticed that there were comments in the margins of my spec. He had read the first page!

He had read the first page of my spec and written little notes in the margin!

Considering that we only got him the spec about 24 hours earlier, he must have read it the night before.

He was asking questions. I was answering them. They were pretty easy, but I can't for the life of me remember what they were, because I couldn't stop noticing that he was flipping through the spec . . .

He was flipping through the spec! [Calm down, what are you a little girl?]

. . . and THERE WERE NOTES IN ALL THE MARGINS. ON EVERY PAGE OF THE SPEC. HE HAD READ THE WHOLE GOD-DAMNED THING AND WRITTEN NOTES IN THE MARGINS.

He Read The Whole Thing! [OMG SQUEEE!]

The questions got harder and more detailed.

They seemed a little bit random. By now I was used to thinking of Bill as my buddy. He's a nice guy! He read my spec! He probably just wants to ask me a few questions about the comments in the margins! I'll open a bug in the bug tracker for each of his comments and make sure it gets addressed, *pronto!*

Finally, the killer question.

"I don't know, you guys," Bill said, "Is anyone *really* looking into all the details of how to do this? Like, all those date and time functions. Excel has so many date and time functions. Is Basic going to have the same functions? Will they all work the same way?"

"Yes," I said, "except for January and February, 1900."

Silence.

The f*** counter and my boss exchanged astonished glances. *How did I know that? January and February WHAT?*

"OK. Well, good work," said Bill. He took his marked-up copy of the spec.

. . . *wait! I wanted that . . .*

and left.

“Four,” announced the f*** counter, and everyone said, “Wow, that’s the lowest I can remember. Bill is getting mellow in his old age.” He was, you know, 36.

Later I had it explained to me. “Bill doesn’t really want to review your spec, he just wants to make sure you’ve got it under control. His standard M.O. is to ask harder and harder questions until you admit that you don’t know, and then he can yell at you for being unprepared. Nobody was really sure what happens if you answer the hardest question he can come up with because it’s never happened before.”

“Can you imagine if Jim Manzi had been in that meeting?” someone asked. “‘What’s a date function?’ Manzi would have asked.”

Jim Manzi was the MBA-type running Lotus into the ground.

It was a good point. Bill Gates was amazingly technical. He understood Variants, and COM objects, and IDispatch, and why Automation is different from vtables, and why this might lead to dual interfaces. He worried about date functions. He didn’t meddle in software if he trusted the people who were working on it, but you couldn’t bullshit him for a minute because he was a programmer. A real, actual, programmer.

Watching nonprogrammers trying to run software companies is like watching someone who doesn’t know how to surf trying to surf.

“It’s OK! I have great advisors standing on the shore telling me what to do!” they say, and then fall off the board, again and again. The standard cry of the MBA who believes that management is a generic function. Is Steve Ballmer going to be another John Sculley, who nearly drove Apple into extinction because the board of directors thought that selling Pepsi was good preparation for running a computer company? The cult of the MBA likes to believe that you can run organizations that do things that you don’t understand.

Over the years, Microsoft got big, Bill got overextended, and some shady ethical decisions made it necessary to devote way too much management attention to fighting the US government. Steve took over the CEO role on the theory that this would allow Bill to spend more time doing what he does best, running the software development organization, but that didn’t seem to fix endemic problems caused by those eleven layers of management, a culture of perpetual, permanent meetings, a stubborn insistence on creating every possible product no matter what (how many billions of dollars has Microsoft lost, in R&D, legal fees, and damage to reputation, because they decided that not only do

they have to make a web browser, but they have to give it away free?), and a couple of decades of sloppy, rapid hiring has ensured that the brainpower of the median Microsoft employee has gone way down (Douglas Coupland, in *Microserfs*: “They hired 3,100 people in 1992 alone, and you know not all of them were gems.”)

Oh well. The party has moved elsewhere. Excel Basic became Microsoft Visual Basic for Applications for Microsoft Excel, with so many TMs and ®s I don’t know where to put them all. I left the company in 1994, assuming Bill had completely forgotten me, until I noticed a short interview with Bill Gates in the *Wall Street Journal*, in which he mentioned, almost in passing, something along the lines of how hard it was to recruit, say, a good program manager for Excel. They don’t just grow on trees, or something.

Could he have been talking about me? Naw, it was probably someone else.

Still.