11-1-2007

Dreaming in Code

Jeffrey Barlow
Pacific University

Follow this and additional works at: http://commons.pacificu.edu/inter07

Recommended Citation

This Book/Site Review is brought to you for free and open access by the Interface: The Journal of Education, Community and Values at CommonKnowledge. It has been accepted for inclusion in Volume 7 (2007) by an authorized administrator of CommonKnowledge. For more information, please contact CommonKnowledge@pacificu.edu.
Dreaming in Code

**Description**

**Rights**
Terms of use for work posted in CommonKnowledge.

This book/site review is available at CommonKnowledge: [http://commons.pacificu.edu/inter07/48](http://commons.pacificu.edu/inter07/48)
Dreaming in Code

Envision computing and the people who do it as a sort of pyramid. On the bottom are the billions of us who can sit down with a computer and use it to some good purpose. That purpose has to be supported with adequate off-the-shelf software, probably some version of a Microsoft product, most likely an integrated suite like Microsoft Office.

Just above that bottom level is some much smaller number of people who can work in a variety of operating systems, such as both Windows, Mac OS, and perhaps even Unix. Even there, this group is likely to still take their software off the shelf. They may be virtuosos with their favorite program, the go-to person in the office who can help us customize the program to our own needs, or troubleshoot it for us, perhaps downloading updates or researching on the web to find incompatibilities that come between us and our work.

The next level above that group is a yet still smaller group of people who can work at the operating system level, making changes in the way the OS interacts with the software, or adapting our machines to do something that was not originally planned by the makers, or most importantly to us, troubleshooting the actual operating system as opposed to the programs we are using. Usually, this person works not in our office, but in our Information Systems department, or at the repair shop where we unpack our failed machine and shamefacedly ask them to make it all better.

A level above this group is a still smaller group, so small that we cannot begin to guess what percentage they may be of the user base at the bottom of the pyramid, but clearly they are many zeroes to the right of the decimal point that might express their numbers in ratio to us, the user base. This is the group of those who actually write the software that runs the system or our...
application. This book is about those people.

But to climb this tiresome metaphor one more time, at the very top stand a very few people—those who can understand what those on the level just below the summit are doing, and explain that to those of us at the very bottom. One of those at the top is Scott Rosenberg, the author of *Dreaming In Code*. Rosenberg is a cofounder of Salon.com and has written widely about computing.

Rosenberg would probably object most strenuously to the above typology, which seems to put him at the top of the hierarchy. But the work being reviewed is not about how to write software, it is about explaining the process in order that a general audience might understand and appreciate it, and at that, Rosenberg is a master. [1]

Because of his background, which included some programming, he was invited to participate in a project to create a new software application. The man behind that project was Mitch Kapor, one of the major figures in the industry. [2] This project seemed to all concerned to be a relatively simple undertaking. Kapor, because of previous successes, had virtually unlimited funding and an ability to draw in highly successful and creative participants.

The project itself, code named “Chandler”, seemed both timely and easily accomplished. It was to be a new PIM, a personal information manager, which would manage email, appointments, address books, to-do tasks and note taking. [3] These, of course, are the stuff of daily access for most of us. We might wonder why such a program was thought of as necessary, but the ultimate goal was to put it into one seamless application, which would run across platforms and operating systems.

More than three years after the project began, it was more than 1.5 million lines of code deep, had been revised repeated times, scaled back extensively in its intended purposes, cost millions of dollars as dozens of key figures came and went, and still did not work. This, we learn, is the fate of more than 90% of software projects.

Why, we might wonder, is a failed project worth an entire book, and quite a dense one at that? *Dreaming in Code* is “readable” only in the sense that the topic itself is so abstract that to produce a work that is comprehensible, let alone often very interesting, is itself an achievement.

But there are many reasons why the book deserves the wide audience that it ultimately achieved. The work teaches us how the programs we use daily are developed; it teaches us how the industry itself has developed; it explains the various schools of thought in how the work should be done, and where the industry seems to be going. We come to understand that proprietary vs. “Open Source” is more than just a quixotic choice for developers, but one reflecting deep social and economic divides.

And perhaps most importantly, *Dreaming in Code* tells us a great deal about the people who
choose to do this work. Many of them, of course, are famous names, even to those of us at the bottom of the pyramid, and meeting them as individuals is interesting. But even as yet-to-be-famous faceless programmers, they are a very distinctive group and many of us need to know more about them because we will interact with them at critical moments in our lives as employers, supervisors or just as consumers of the products of their labor.

For most of us, programming culture is a deep mystery that we will never penetrate. Rosenberg works here as a sort of cultural anthropologist, who not only explains the culture but tells us why we should care, the ultimate task of every good anthropologist. We learn, for example, that Asperger’s Syndrome [4], a sort of mild form of autism often found among high-functioning individuals, probably has a genetic component and occurs frequently in the families of programmers. [5]

And anyone who plans a project involving creative work with computers would do well to learn the meaning of terms such as “snakes”, “forcing functions”, “recursions” and ultimately, “dogfood”, to list only the non-technical terms. [6]

At the last, Chandler becomes one complex cautionary tale, one from which any reader might well learn. We can approach that often-delayed new application we install with a much greater appreciation of how it got to our desk, who developed it, and what it does once we install it.

Notes


[3] Rosenberg, 61—it is perhaps in index of the actual complexity of this seemingly easy project that we do not get a clear explanation of the project itself until more than sixty pages into the work!


[5] See page 133. Speculation as to the explanation for this fact vary widely…there may be a genetic element in the condition. Perhaps it skips generations or is lightly manifested in any particular individual, but when two programmers meet at work and procreate, their offspring may have a more serious case than either parent, who perhaps went undiagnosed.

[6] A snake is a problem that you cannot quite get your hands on; a forcing function is an operation that requires you to make choices before moving forward; a recursion is a problem that
takes you back to previous unsolved problems; dogfood is what you have at the last, when your project fails.

This entry was posted in Uncategorized by Editor. Bookmark the permalink [http://bcis.pacificu.edu/interface/?p=3421].

3 THOUGHTS ON “DREAMING IN CODE”

community
on February 4, 2014 at 10:21 AM said:

We are a group of volunteers and beginning a brand new scheme in our community. Your web site provided us with valuable facts to paintings on. You’ve performed an impressive career and our entire neighborhood can be thankful to you.

nigeria entertainment news
on February 4, 2014 at 10:32 AM said:

This is my very first time pay a visit at here and i am definitely impressed to read everthing at alone place.

nigeria entertainment news
on February 4, 2014 at 10:42 AM said:

you have any? Kindly permit me understand so that I may possibly just subscribe. Thanks. as well conceive so , perfectly written post!