Reminiscences

Michael Wing
<wing@swcp.com>

My favorite column in ACM publications is the "Reminiscences on Influential Papers" in the SIGMOD Record. I don't know of anything else like it, anywhere. Each issue contains a handful of short essays (usually 2 or 3 long paragraphs), explaining why one research paper or another made an impression. Usually, the cited paper influenced the author's Ph.D. thesis or subsequent research.

Sometimes the essay explains why a seemingly obscure idea matters. In September 2005, Lucian Popa explains that "tuple-generating dependencies" and "equality-generating dependencies" (though superceded by more recent ideas) are still relevant in schema mapping in data exchange systems.

Other times, the essay bridges database to another field. In September 2005, Panagiotis Ipeirotis explains why a classic paper on statistics helped him characterize a web database. It also helped him realize that "research from other fields can be easily applied to traditional database problems."

In this column, the authors state what matters that they find in someone else's work. The authors find many different sources of inspiration in many different places. This column shows that in database, many different people do useful and important work.

SEN

I have thought that if software engineers had a similar column, we would realize that lots of important and successful work is going on.

There are many people who have positively, productive affected the work I do. I could easily write essays on David Parnas for modular composition, Barry Boehm for the Spiral model, Grady Booch for the first Object-Oriented Design, Watts Humphrey for Managing the Software Development Process, and Fred Brooks for the Mythical Man-Month. But, everyone knows about these works. Instead, I will give two examples from the world of practice.

Amalio Escobar

I worked for Amalio (my manager) in Tucson, Arizona for a year. He cared for people in general and his people in particular, more than any other manager I have ever worked for. When we went out to lunch as a group, he would give a buck or two to every homeless person we passed on the street. He sort of adopted his engineers into his family. He always urged his engineers to have fun.

Amalio was not perfect. He was a bit of a rabble-rouser and when he fought with his boss, and the conflict touched everyone.

Yet, ever since I stopped working for Amalio, I keep his example in mind. I believe that when engineers feel supported they do better work. Since then, I have tried to emulate his example.

Linus Torvalds

Linus Torvalds is the head of the Linux operating system. He creates new value in the world, things that did not exist previously. He is a nice guy that many people enjoy working with.

On the Wikipedia software engineering page, I listed Torvalds' name as an exemplary practitioner. Someone else removed his name a year ago claiming, "Linus Torvalds is just a hacker."

I want to point out that Linux is a living and breathing OS that is continuing to evolve for new uses and new platforms, while OS/360 has languished. If Linux were the result random hacking, it would have ossified long ago, like OS/360 has. Note that a core point of credibility for both Brooks and Humphrey as software engineers was their leadership of OS/360. Based on the empirical evidence, Linus Torvalds architected a much more adaptable system than Brooks or Humphrey ever did.

I want to point out that Linus Torvalds spent more time leading Linux than either Fred Brooks or Watts Humphrey ever spent leading the OS/360 project. Torvalds has combined the efforts of many thousands of software engineers from around the world, who mostly work for free. Both Brooks and Humphrey paid their software engineers to contribute to OS/360. Torvalds mustered and motivated a larger and more diverse workforce, for a longer time, than Brooks or Humphrey ever did.

Linus Torvalds is open to developing software using new approaches, which are now called agile and open-source, but which were not taken seriously when he and others developed them. Linux developed new tools and processes enabling a new generation of people to collaborate in new ways. Contrast this with the idolization of the past that was codified by the CMM. Torvalds tries new things and fits his time, in the same way that Brooks and Humphreys tried new things and fit their times when working on OS/360.

I do not mean to claim that any one of them is better than any other, because there are so many intangibles. Linux was built using experience gained on OS/360 and other projects. Now, we use high-level languages as a matter of course. And, the Internet and fast PCs make sharing code easy. Yet, I do claim that Linus Torvalds is a tremendous software engineering practitioner by any comparison to either Brooks or Humphrey, and that he deserves recognition as such.

Conclusion

I believe that there are many examples successful contributions in software engineering, some are famous, others not. Also based on these two examples, nice guys do get ahead in software engineering.