

A Human Experience

I had the pleasure of attending the Agile Development Conference in August in Salt Lake City. Conference chair Alistair Cockburn promised that it would be “not your average conference.” It was a promise kept. The quality and variety of speakers, topics, and venues was effective and impressive. Two conference highlights lead to this month’s message.

Famed visionary Gerald Weinberg, who has been in the industry since before FORTRAN (!), gave the keynote address, imparting “key notes” to ensure success for programmers and their managers. Development agility is difficult to observe and does not lend itself to useless but tenacious measures such as lines of code produced per unit of time. Agile development promotes team ownership, and the key measure of success is a working product in the hands of happy users. I found it painfully easy to identify with programmers who work for managers who, as Jerry described them from his experience, would rather fail knowing whom to blame than to succeed without knowing whom to credit. He then characterized a Great Programmer as one who:

- Has a “service” posture (We’re in the software business to meet the needs of others. Too many hackers really do erroneously believe that users just get in the way.)
- Learns continually, no matter how skilled he or she is today
- Openly shares his code, insights, and expertise with others
- Gives credit where credit is due, especially to his “supporting cast” (colleagues, vendors, good management, office staff, family, etc.)
- Does not sacrifice quality or integrity for “job security” (whatever *that* is)
- Takes care of his whole self (gets plenty of rest, just “hangs out”, plays with kids, etc.)

Most of our time as developers is spent interacting with other people, hence the pre-eminence of so-called “people skills”. (Alistair’s DBA is aptly named, “Humans and Technology”). In his typical, simple, wisdom-rich style, Jerry made his points by drawing from everyday life-experience, this time observing seeing-eye dogs in action. I came away sensing a critical industry-wide need for more Seeing Eye Programmers.

The Agile Alliance is a confederation of influential individuals who have united around a small set of key principles¹, not the least of which is that they value “individuals and interactions over processes and tools.” Much (but never enough!) has been said about the value of developers and the teams to which they belong. Bjarne Stroustrup wrote, “Too often, we forget the human aspects of system building and consider the software development process as simply ‘a series of well-defined steps, each performing specific actions on inputs according to predefined rules to produce the desired outputs.’ The very language used conceals the human involvement! Design and programming are human activities; forget that and all is lost...Managers often forget that organizations consist of individuals. A popular notion is that programmers are equal and interchangeable. This is

¹ See the Agile Manifesto at <http://agilemanifesto.org>.

a fallacy that can destroy an organization by driving out many of the most effective individuals and condemning the remaining people to work at levels well below their potential.”² Programmers are more than cells in the spreadsheet of corporate assets.

But end users are people too, and developers often overlook or ignore that important fact. Last month I wrote about my uncomfortable Linux end-user experience. I am well aware of the tremendous value of open-source software development and support it at every opportunity, but its very nature results in only the software that open-source developers are willing to contribute. Open-source developers do what they want when they want (how can it be otherwise?); there is no one to act as an authoritative advocate for end users. Fortunately, the Agile Development Conference did not leave this stone unturned, as I attended one of the best presentations I’ve ever witnessed on the subject. Larry Constantine’s tutorial on usage-centered design opened this programmer’s eyes to ways of meeting end-user needs with directness, accuracy, and simplicity. I ran right out and bought the book³ (sorry if I got the last good used copy :-).

It’s all about people. Notwithstanding the steady increase in automation, software development is and shall remain very much a human activity.

² Stroustrup, *The C++ Programming Language*, Special Edition, Addison-Wesley, pp. 693, 716.

³ Constantine & Lockwood, *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, Addison-Wesley, 1999.