early 1,200 people attended the Agile 2007 conference in Washington, D.C. in August. The conference was sold out months in advance, so evidently the upward trend in demand is continuing. Meanwhile, the agile software development movement is displaying signs of maturity with sobering, reconciliatory, and reflective undertones. If these undertones become more pronounced, they will represent a noteworthy departure from the euphoric, dogmatic, and unquestioning attitudes that, according to critics, dominated the agile community in its early years.

**Indicators**

The traditional software engineering community has evolved in how it perceives the agile movement. Many respectable agilist folks nowadays admit that they’ve learned and evolved, too. While not turning their backs to the Agile Manifesto, some display a mildly critical view of their beloved paradigm, timidly acknowledging its limitations and disapproving attitudes that, according to critics, dominated the agile community in its early years.

Another subtle indicator is the growing recognition of early software pioneers who advocated similar concepts long before the Agile Manifesto existed. Such people are now more regularly featured in agile software development events.

While not widespread, an open attitude toward approaches that were once dismissed as unagile or “waterfallish” is also in the air. At the front end, requirements engineering, architecting, and interaction design—and at the back end, traditional quality assurance and testing approaches—are notable examples that attract interest. A minority can now envision cross-pollination among approaches that previously represented sharply opposing postures. This is progress, even if it doesn’t signal an imminent end to the agile mentality’s apparent sectarian isolation.

**Self-perceptions**

To gauge the agile software development community’s perception of how far it has come, I ran a casual survey in the conference hotel’s hallways in D.C. Armed with a tablet PC, I intercepted unsuspecting delegates, flashing the touch-sensitive screen that displayed a few binary questions.
“Would you take a 10-second survey on perceptions of agile? I swear, it won’t take more than 10 seconds. Just answer these five questions by hitting ‘Yes’ or ‘No’ with your fingertip. You can skip any question.”

“Isn’t there a ‘Maybe’ option?”

“Sorry, just ‘Yes’ or ‘No!’ (There go seven seconds.)

While a few walked away with suspicious, disapproving looks that made me feel like a sleazy telemarketer, more than enough gentle people actually humored me. I thank those who took the time to respond. After my 150th victim, I decided to stop and enjoy the conference.

However, before Forrest Shull, our Empirical Results editor, lectures me for my nonrandom sampling, the results can’t be generalized to mean-looking people. Finally, I didn’t have approval from some ethics board. There.

To compensate for the survey’s deficiencies, I present the results as a meaningfully colored bar chart in figure 1. Because a bar chart can’t be sufficiently informative without interpretation, I insist on elaborating.

**Everybody knows something about agile**

Only a minority of respondents (23 percent) identified themselves as newcomers to agile (meaning new to the concept, not the conference). This was important for me to know because I wondered whether newcomers would have different perceptions. It appears that the results are representative only of non-newcomers. Oh well.

Despite my nonrandom sampling, the low representation of newcomers perplexed me a bit because it doesn’t quite account for the tremendous growth the conference has been enjoying (unless most newbies are mean-looking).

**Maturity and penetration**

Of the respondents, 65 percent said that agile had come of age. Subjectively, many didn’t seem particularly confident about their answer, or were confused. When they asked for clarifi-
FROM THE EDITOR

New Column!

I’m pleased to announce the first edition of Jeff Patton’s new column, User Centric. Some of you might know Jeff as an expert on software usability and interaction design. Others might have heard of his good name in the context of agile software development: he’s a co-recipient of the Agile Alliance’s 2007 Gordon Pask Award. You might even have attended one of his famed hands-on workshops. Jeff is one of those rare connectors who straddle the normally separate worlds of developers and end users, and he does it well. His writings will convince you that users deserve a central voice in software development. I hope you’ll enjoy them.

Self-threats

So, why do its most ardent advocates think that “agile” has become a buzzword? Could it have something to do with its lingo’s all-too-often-diluted overuse? Or its principles’ indiscriminate overapplication in infinite regress in arbitrary and sometimes completely inappropriate contexts? Philippe Kruchten’s recent satire “Voyage in the Agile Memeplex” (ACM Queue, July/Aug. 2007, pp. 38–44) takes a stab at this kind of bizarre decontextualization. At times, the tendency to overuse and overapply borders on the ridiculous. A suicidal example that I recently came across is the suggestion to bootstrap the Agile Manifesto by continuously revising it.

For many critics, a second issue threatening the trend’s maturity is the

cation, I replied “I mean, has it matured?” and left it at that. If I could put

over my overeager-researcher hat, I’d ex-

plain, “The agile movement’s maturity is

now official!” But I’ll use my discretion

and hold back, wisely ignoring this

strong statistic.

A weak majority, 56 percent, said

that agile had become mainstream. Be-

cause the remaining 44 percent of the

agile folks think they might still be

lurking in the fringes, I’ll also refrain

from declaring victory of massive pen-

etration—unfortunately. I’m hearing,

“We’re not there yet.”

A bittersweet consensus

An overwhelming majority, 92 per-

cent, said that “agile” has become a

buzzword. An unverifiable nuance was

that this affirmation came with a sigh

or a head shake. An even greater ma-

jority, a whopping 95 percent, thought

that agile had had impact. The ones

who hit the Yes button did it with an

expression of great conviction—but

to be quite honest overeager-researcher me.

So, why do its most ardent advocates think that “agile” has become a buzzword?

Could it have something to do with its lingo’s all-too-often-diluted overuse? Or its principles’ indiscriminate overapplication in infinite regress in arbitrary and sometimes completely inappropriate contexts? Philippe Kruchten’s recent satire “Voyage in the Agile Memeplex” (ACM Queue, July/Aug. 2007, pp. 38–44) takes a stab at this kind of bizarre decontextualization. At times, the tendency to overuse and overapply borders on the ridiculous. A suicidal example that I recently came across is the suggestion to bootstrap the Agile Manifesto by continuously revising it.

For many critics, a second issue threatening the trend’s maturity is the

cation, I replied “I mean, has it ma-
tured?” and left it at that. If I could put

over my overeager-researcher hat, I’d ex-

plain, “The agile movement’s maturity is

now official!” But I’ll use my discretion

and hold back, wisely ignoring this

strong statistic.

A weak majority, 56 percent, said

that agile had become mainstream. Be-

cause the remaining 44 percent of the

agile folks think they might still be

lurking in the fringes, I’ll also refrain

from declaring victory of massive pen-

etration—unfortunately. I’m hearing,

“We’re not there yet.”

A bittersweet consensus

An overwhelming majority, 92 per-

cent, said that “agile” has become a

buzzword. An unverifiable nuance was

that this affirmation came with a sigh

or a head shake. An even greater ma-

jority, a whopping 95 percent, thought

that agile had had impact. The ones

who hit the Yes button did it with an

expression of great conviction—but

to be quite honest overeager-researcher me.

So, why do its most ardent advocates think that “agile” has become a buzzword?

Could it have something to do with its lingo’s all-too-often-diluted overuse? Or its principles’ indiscriminate overapplication in infinite regress in arbitrary and sometimes completely inappropriate contexts? Philippe Kruchten’s recent satire “Voyage in the Agile Memeplex” (ACM Queue, July/Aug. 2007, pp. 38–44) takes a stab at this kind of bizarre decontextualization. At times, the tendency to overuse and overapply borders on the ridiculous. A suicidal example that I recently came across is the suggestion to bootstrap the Agile Manifesto by continuously revising it.

For many critics, a second issue threatening the trend’s maturity is the

lack of genuine debate at the grassroots level. The openness seen in some niche circles doesn’t appear to percolate to that level. A reflective panel that I attended demonstrated this reluctance. The panel started on good footing with calls to open-mindedness, moderation, and inclusiveness, but gradually degenerated into the typical us-versus-them love-in, thus defeating its own purpose. If such attitudes don’t subside, the agile movement risks losing the cautious, qualified endorsement and interest it has begun receiving from the traditional software engineering and research communities. Whether this matters or not is of course another question.

**Evolution through devolution**

It’s counterproductive to equate abstract shades of agility with degrees of good and noble. The Agile Manifesto might be timeless, but the adjective “agile” as an umbrella term will ultimately outlive its usefulness, if it already hasn’t. The explosion in industrial adoption of agile processes, Scrum in particular, and individual practices such as test-driven development will keep the term afloat for the near future. But how should the community respond to its central vocabulary’s continuing dilution?

Once the main messages have reached the masses, it will be time to deconstruct agility by moving away from labels and refocusing on what ultimately matters: its essence. That essence consists of the articulated values and principles, as well as the concrete practices, techniques, and methods that the movement has spun off. These are the real intellectual tools that deserve permanence. Each such tool has varying degrees of merit in different contexts, depending on whether it is applied independently, synergistically, or in combination with complementary approaches.

What’s your take? Write me at hakan.erdogmus@computer.org.

---

**Calling for Articles**

Many recent advances in science have been dependent on software. Much scientific software is written by scientists themselves or by multidisciplinary teams of software engineers and scientists. In the former case, scientists face the challenges of knowing little SE beyond coding and of working in a culture that often devalues the skills required to develop software. In the latter case, teams must overcome different cultures and communication difficulties. This issue aims to explore these challenges and present possible solutions.

**TOPICS OF INTEREST:**

- Case studies of scientific software development
- Analyses, grounded in practice, of the particular characteristics of and problems facing scientists as professional end-user developers, scientific software development, and multidisciplinary development teams comprising software engineers and scientists
- Discussions of tools, techniques, and strategies designed to support scientific software development
- Discussion of how tools, techniques, and strategies software engineers commonly deploy might be usefully deployed, perhaps with some modification, by professional end-user developers
- Particular interest in submissions grounded in practice

**PUBLICATION:** July/August 2008

**SUBMISSION DEADLINE:** 9 January 2008

**GUEST EDITORS:**

- Judith Segal, The Open University, j.a.segal@open.ac.uk
- Chris Morris, Daresbury Lab, c.morris@dl.ac.uk