No, That’s Not It

I spend a significant amount of my face time with our clients helping them understand the business benefits of IID (iterative and incremental development). The biggest hurdle I have to overcome is helping them realize they aren’t already doing what I’m suggesting. Hakan Erdogmus presented this very clearly in his column “Novelty in Sameness” (May/June 2007). I have already referred several people on my team, as well as several clients, to review the article. Keep up the good fight.

Clay Nelson
Technical Business Unit Executive—East Region,
IBM Rational Software
caynelson@us.ibm.com

Services on demand: Do they deliver?

In his cogent article in the July/Aug. issue (“On-Demand Enterprise Services: Where’s the Catch?”), Hakan Erdogmus discussed several benefits of on-demand enterprise services as touted by an IT executive. There’s at least one more significant factor to consider, of a technical nature. Having the desktop closely coupled to the provision of computing power has strong advantages:

■ It provides better control of response times.
■ Bandwidth inevitably decreases as the user and the computing source become farther apart (compare, for example, the speed of accessing a local hard disk, a local area network, a wide area network, and the Internet).
■ In some applications, the browser paradigm pales by comparison to a “traditional” thick-client solution, such as most rich-media ones, including a lot of games.

The providers of on-demand enterprise services may well try to dismiss these arguments by saying that they can be overcome with higher bandwidth networks, mixed solutions where some of the computation takes place at the local client, and so on. However, although of course WAN speeds will increase, I very much doubt that the ratio of the speed of local versus remote access will change substantially. The PC revolution occurred for a reason, and I personally doubt that we will see a move back to the “mainframe” style of computing.

What will happen in practice is that the spectrum of solutions will increasingly blur, as is indeed happening today—when my local word processor downloads clip art from the Internet, is this a pure thick-client application? What about a multiplayer graphics-intensive game? When an on-demand customer relationship management application generates and exports an address list in a format for printing envelope labels, is this a pure on-demand application?

There’s another factor, more emotional perhaps. People and organizations want to differentiate themselves. For good and obvious reasons, on-demand services are less con-
figurable in practice than thick-client ones. Think, for example, of traditional spreadsheets and those available on demand on the Internet. The on-demand services are at a disadvantage. Hakan turned the argument on its head by claiming less complexity as an advantage for the on-demand scenario—I believe that, quite contrary to the apparent reasonableness of this argument, it simply doesn’t hold water. Software is always becoming more complex. The IT executive Hakan discussed in the article even tacitly admitted this when mentioning “handsome payoffs for customers when they instantly inherit any improvements made.” This ability to configure, even to customize, applications gives the traditional scenario an advantage. For instance, in my area of expertise (software for production and supply-chain optimization), companies configure and even customize applications because they believe this will lead to a competitive advantage. I have little doubt that, properly conceived and executed, this traditional thick-client strategy does indeed deliver the required results.

Constantine Goulimis
Director, Greycon Ltd.
cng@greycon.com

Help, not hype

I just read Panagiotis Louridas’s article (“Declarative GUI Programming in Microsoft Windows,” July/Aug. 2007); well done, to both author and editor. This type of column is important and deserves more space than the few pages it was given. I tend to pick up ACM Queue to read when I have a break because it relates more to the working practitioner. With so much hype in the industry, the new Software Technology column should help people identify potentially useful nascent technologies and show others for what they are: a lot of hype without much value.

Gregg Irwin
Pointillistic Software
gregg@pointillistic.com

Idioms vs. design patterns

I’m not sure exactly what Grady Booch had in mind by “this context” when he said “Idioms, a term Jim Coplien first used in this context ...” (“The Well-Tempered Architecture,” July/Aug. 2007). But the concept of idioms in programming languages goes way back, at least in the APL community. The FinnAPL group has been publishing idiom lists for decades; Alan J. Perlis and Spencer Rugaber’s Yale University technical report The APL Idiom List was published in 1977.

In a recent comment on an ACM Queue article, Curtis A. Jones brought up the fact that that term was one of Ken Iverson’s pet peeves: “Ken Iver-
son disliked the term ‘idiom’ for short APL expressions as in Alan Perlis’s and Spencer Rugaber’s APL Idiom List or the FinnAPL Idiom List because the word ‘idiom’ implies ‘... not clear from the usual meaning of the words in the expression ...’ (American Heritage Dict. ca. 1989).” I recall that Ken preferred terms such as “phrase,” even though it doesn’t have the right connotations (to me, anyway).

Robert Bernecky
CEO, Snake Island Research
bernecky@acm.org

Grady Booch responds:
I appreciate this historical perspective. James Coplien is the first one I’d encountered who put idioms in the context of the patterns vocabulary, offering the distinction between idioms and design patterns, which is the genesis of my comment. You are correct in noting that the concept of idiom is far older. I’ll be sure to make that clear in all my future writing.

Information democratization
As a regular Loyal Opposition reader, I’m accepting Robert Glass’s invitation to share my viewpoint about blogs. I see them as a postmodern phenomenon where information from official authorities is no longer the only source of evidence on which people base their beliefs.

This greater variety of sources, of course, has pros and cons. Perhaps among the pros, there’s a bit more stimulus toward the proliferation of critical thinking. Readers weigh the value of what they read (whoever the writer might be), and writers practice logical thinking while articulating their thoughts in writing.

Marco Antonio Dorantes Martínez
Senior consultant on software design & architecture,
Microsoft Consulting Services
marcodorantes@hotmail.com

I loved Robert Glass’s article regarding blogs (“What’s with This Blog Thing?” Sept./Oct. 2007). I agree wholeheartedly that the information presented in blogs (and other “Web 2.0” services) is often flawed. I, too, battle with students wanting to cite Wikipedia. On more than one occasion, I’ve had to deduct points when my students tried to cite their personal experience. When I asked one such student what her credentials were, she proudly stated, “But I have a bachelor’s degree in project management!” I recently wrote an article for IT Professional where I claimed that many Web 2.0 services represent little more than a new face on the old “Web 1.0” services. After all, isn’t a blog simply a glorified BBS of yesteryear?

Ilya V. Yakovlev
Director, Center for Information Systems, St. Cloud State University
iv yakovlev@stcloudstate.edu

Working at the frontiers of knowledge
The University of Groningen provides high quality research and education, is internationally oriented, respects differences in ambition and talent, works actively with business, the government and the public, and ranks among the best universities in Europe.

The University of Groningen wishes to realise three goals in the 2003-2010 period. Investment in the highest quality, recognition of differences in talent and ambition and the stimulation of cross-border research and education.

A full description about this vacancy is available on the university web site:

(www.rug.nl)

The university has a special careers advisory service for the partners of new staff who move to the area.

The university is an equal opportunities employer. Because women are still underrepresented in a number of fields, they are particularly encouraged to apply.

Faculty of Mathematics and Natural Sciences

(207145)

Highly qualified candidates are invited to apply for an academic position at the Faculty of Mathematics and Natural Sciences of the University of Groningen.

Assistant, Associate or Full Professor in Software Engineering

The University offers a salary depending on the level of appointment, qualifications and work experience as assistant professor € 3,060,- up to a maximum of € 4,761,- as associate professor € 4,242,- up to a maximum of € 5,670,- as full professor level 2
€ 4,697,- up to a maximum of € 6,841,- gross per month for a full-time job (38 hours per week). Employment basis: tenure-track for assistant professor and permanent (tenure) for associate or full professor.

Further information can be obtained by prof.dr.ir. Marco Aiello, phone +31 50 363 39 48 e-mail: aiellom@cs.rug.nl
http://www.rug.nl/informatica/onderzoek/programmas/SoftwareEngineering

Rijksuniversiteit Groningen