

In the world of agile development, context is key.

Agile processes are not a technology, not a science, not a product. They constitute a space somewhat hard to define. Agile methods, or more precisely *agile software development methods or processes*, are a family of approaches and practices for developing software systems. Any attempt to define them runs into egos and marketing posturing. For our purposes here, we can define this space in two ways:

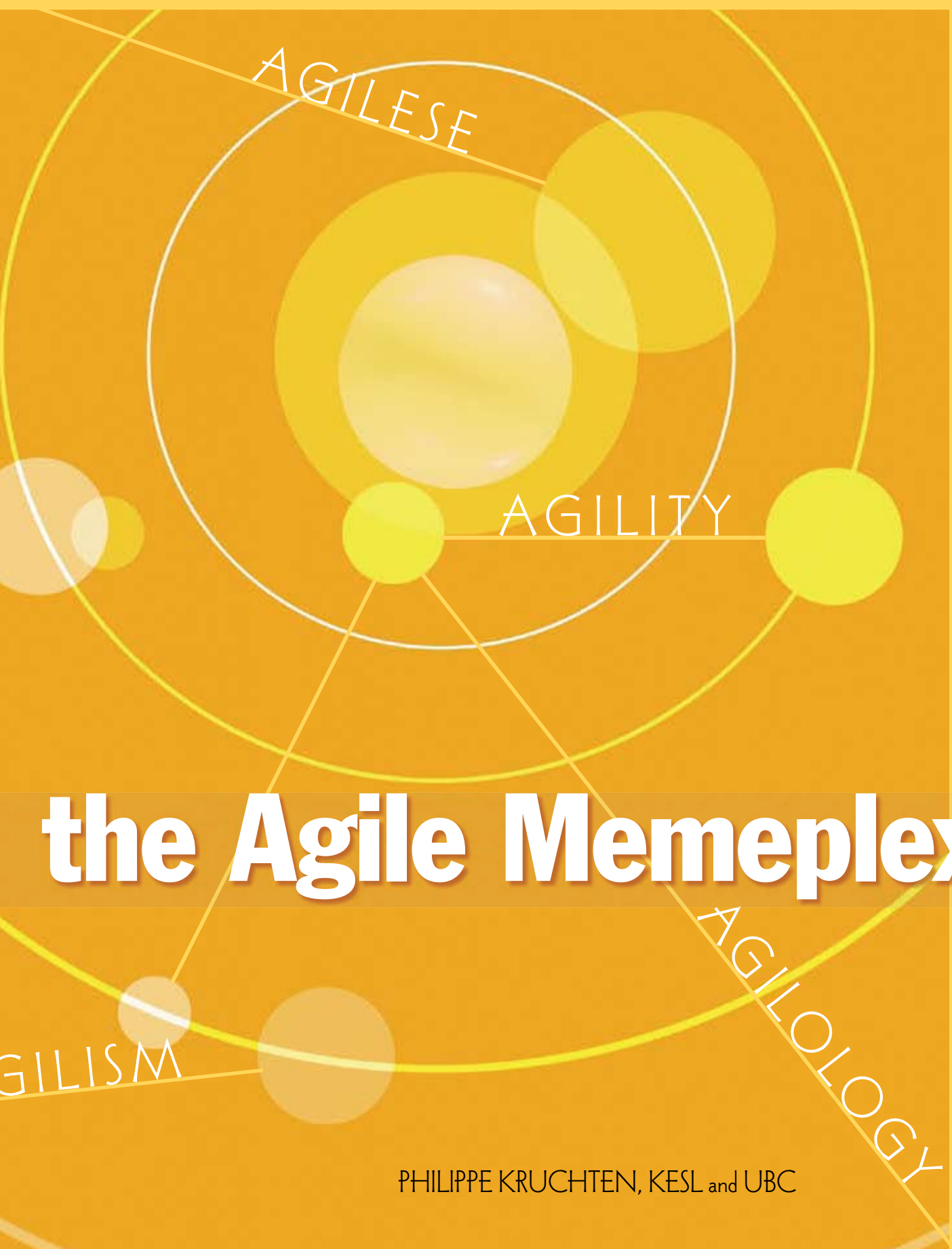
By enumeration. Pointing to recognizable members of the set: XP (extreme programming), scrum, lean development, DSDM (Dynamic Systems Development Method), Crystal, FDD (feature-driven development), Agile RUP (Rational Unified Process) or OpenUP, etc. (though some would claim RUP to be anti-agile).

By some predicate. Assessing the degree of compliance to the “agile manifesto,”¹ but with a somewhat more

Voyage in

AGILITIS

AC



PHILIPPE KRUCHTEN, KESL and UBC

Voyage in the Agile Memeplex

subjective approach. Or even more vague, by complying with Jim Highsmith's criteria for an agile ecosystem: "chaordic" (i.e., at the same time chaotic and ordered), collaborative, and streamlined.

There have been attempts to define agile methods *in contrast* with other methods. In their 2003 book *Balancing Agility and Discipline*, Barry Boehm and Richard Turner compared them with disciplined methods,² but agile methods can demand high levels of discipline, too. Many latecomers seem intent on articulating a contrast with some rather mythical and very bad "waterfall process," carrying all evils in software development practices.

I refer to this space as *agility*: the set of concepts, processes, and practices that are part of these agile methods, even though they are not all specific to them—iteration, for example. I refer to the social context of people, teams, and organizations that are interested parties, users, critics, or researchers of agility as the *agile world*. I use the term *agilistas* to speak about the key advocates—spokespersons of agility who make themselves visible by either having a loud voice or making a significant impact.

AGILITY AS A CULTURE

My first claim is that agility is not a technology, science, or product but a culture. I realize that this presents an even more difficult task: defining *culture*. Dutch sociologist Geert Hofstede defines culture as "the collective *programming of the mind* which distinguishes the member of one group or category of people from another."³ Linguist Helen Spencer-Oatey defines culture as "a fuzzy set of attitudes, beliefs, behavioral norms, and basic assumptions and values that are shared by a group of people, and that influence each member's behavior and his/her interpretations of the 'meaning' of other people's behavior."⁴

One of the implications of this definition has to do with the function of culture. More specifically, besides being an influencing factor on people's behaviors, culture also affects how people interpret others' behavior. This interpretive function of culture is especially important in intercultural situations, as the behavioral norms of one culture may be misunderstood or regarded as incongruous by other cultural groups.

We can observe some of this in the agile world, where people say or write things such as "this is very waterfall" when often they have never seen a waterfall project. For example, a waterfall project may have customers on site and provide continuous integration, unlike what agilistas may claim. Often they are not aiming at inherent properties of the waterfall, but at characteristics of bad projects. So "waterfall" becomes synonymous with "the other culture"—a scapegoat, a strawman, a piñata.

A culture can be rich and varied, and as expatriates can attest, learning another culture is difficult, because many of the values it relies on are embedded in our brains during a long education process, beginning at a very young age. The same applies to agility; since there is not a clear underlying science or a well-packaged body of knowledge, much of it is really transmitted by oral tradition and by imitation (i.e., learning to do something by seeing it being done).

AGILITY AS A MEMEPLEX

In his book *The Selfish Gene*,⁵ and as an analogy for the concept of *gene*, the British biologist Richard Dawkins reintroduced the concept of *meme* (short for *mimeme*, from the Greek μίμησις, imitation) as a replicating "chunk" of culture—a unit of information residing in a brain, a unit of cultural transmission, a basic building block in cultural evolution,⁶ in the sense that this is getting passed from one person to another as an integral part of a culture.

One of the key features of memes is that they replicate easily; the host is a person's brain, and the medium is natural language. "If a meme can get itself successfully copied, it will."⁷ The memes that are successful at this replication are those that challenge their hosts, that make their hosts feel good about themselves, those that are easily described or remembered, that are not boring, that stand out. "Effective memes will be those that cause high fidelity, long lasting memory," and not necessarily the ones that are "important or useful."⁸ Memes are a bit like computer viruses. "Memes now spread around the world at the speed of light, and replicate at rates that make even fruit flies and yeast cells look glacial in comparison.

They leap promiscuously from vehicle to vehicle, and from medium to medium, and are proving to be virtually unquarantinable.”⁹

Looking at a software project, agile or not, how can we apply memetics to get a deeper or different understanding of the phenomena we observe? In a 2005 article about project management, Jonathan Whitty of the University of Queensland invites us “to consider that most of what we call a project and what it is to manage one is an illusion: a human construct about a collection of feelings, expectations, and sensations, cleverly conjured up, fashioned, and conveniently labeled by the human brain.”¹⁰ Peter Senge of MIT also suggests that these “tacit mental models exist below the level of awareness as images, assumptions, and stories which we carry in our minds of ourselves, other people, institutions, and every aspect of the world.”¹¹

Most of what we know and practice in managing agile software projects has been accumulating in our brains in the form of memes. As they got there through very different paths, and because they do not replicate perfectly (unlike genes), we end up with different sets; there is no real common, shared understanding, only an illusion of shared understanding, which is itself creating some issues inside and outside the agile world.

The clever use of positive adjectives is part of the success of the agile memes: who wants to be *not agile*? Since there is no absolute scale of agility, we can all claim we are agile, more agile than you, than them, or than last month. Thousands of people “discovered” that they were “doing agile” for years. (As an aside, the use of the word *extreme* in XP proved to work both ways, facilitating replication but also slowing adoption because of its negative connotation in some software development cultures.)

Agility is not just *one* single self-replicating idea, you may object, but a complex set of ideas, practices, experience, etc. This is where we introduce *memplex*, a contraction of *memes complex*, a term used by Hans-Cees Speel for sets of memes that replicate better as a group.¹² The 50 or so memes that constitute the core of agility are interconnected and support each other, something that the creators of XP had noted and emphasized for their small subset of the agile memplex. Agile teams (and agile user groups and consultancies) have proven to be an effective means of replicating, evolving, and spreading agility memes in the agile memplex.

DECONTEXTUALIZATION

Decontextualization happens when something being studied is removed from its usual environment. When

we cut out the relationships with the natural or original environments, then things start to be different or behave differently.

For agility, decontextualization is carried out through large numbers of very short books, short presentations, blog entries, slides, e-mail threads in telegraphic style, etc. Ideas (the memes) are replicated, but in shortened forms where only one small aspect is being carried out, and reiterated—pounded out for memorization—to make the meme stick. Most often, what is left out is the exact context where the idea, concept, or process applies, or the context in which it was created and found useful. These short forms of the meme gradually become caricatures of themselves, little icons, or mere slogans.

Sometimes we see an explicit dismissal of the context as not relevant, and the vehicle makes claims—explicitly or implicitly—of some form of universality of the idea. For example, a short meme such as “iteration must be three weeks long” is meaningless without an explicit context (what if my project is only three weeks long and has three people?). Other favorites are “no BUFD” (big upfront design) or “YAGNI” (you aren’t going to need it). They all had some value in a given context, but far too often have lost this context on their way through multiple replications. I have seen this attitude lead to massive agile project failures when some focus of software architecture, not producing anything immediately of value for end users, was systematically dismissed by the local agilistas until they hit a wall (i.e., a point beyond which they could not progress without some rather massive refactoring that would not fit inside a sprint or two).

Unlike genes, which are almost digitally coded, memes are often replicated with low fidelity as soon as they become a tad complex. Agile memes are not immune to this rapid drift.

AGILESE

Jargon is terminology that is specific to a group or profession. It’s like slang or argot, which cyberpunk science fiction writer Bruce Sterling defines as “the deliberately hermetic language of a small knowledge clique...a super-specialized geek cult language that has no traction in the real world.”

One step in the decontextualization process is to separate key elements from their normal everyday “vanilla” roots and give them more exotic names, even when the concept is the same or very close (see table 1 for examples). This adds to the replicability of the meme, which becomes “cuter” with a new name and easier to remember. The host’s feeling of being part of a special

Voyage in the Agile Memeplex

smaller “in” group adds further to the replicability: “I am special in my organization, and I bind with a small group of other special people, an elite group.” The members of this group recognize each other by speaking a special argot called *agilese*.

Add to this a series of more or less inspiring metaphors, involving golf or self-cleaning ovens. The effect will stabilize rapidly: because of the widespread replication of the memes, the group of people using the jargon is greater and the “selectiveness” appeal decreases. This means the terms begin to lose their specific meanings. Thus, we end up with a large number of people using terms without any context.

AGILISM

“When a population of imitators start imitating one another, the emergent result is culture,” writes Cambridge anthropologist Robert Aunger.¹³ This is how most of the agile community created the agile culture, or agile memeplex, with little underlying theory or systematic organization of knowledge. After a somewhat chaotic start, the school of agilism is now seen as “crossing the chasm.” Each year, off the coast of British Columbia, in the inside passage between Vancouver Island and the mainland, the various pods (or matrilineal families) of orcas that normally roam the coast independently meet for a great weeklong celebration; this is thought of as increasing the gene pool. The yearly grand gathering of all the gurus and talking heads at agile and XP conferences, and their supporters, seems to serve the function of increasing the meme pool.

Outside of the core group that boldly continues to develop and share agilism, however, we witness a significant degrada-

tion of some of the key messages about agile practices. Not that we do not know more about their usage, effectiveness, etc., but much of what is known and transmitted of that culture has become decontextualized.

Although we probably do not have time to study, balance, and trade off every single decision we make in a software project, much of the agility appears to be dogma: time-boxing, delivering early and often, keeping the end user in the loop, etc. As part of the loss of context, we have also lost the rationale for certain practices; new practitioners are left crossing their fingers and just assuming that since the same memes are in so many brains, they must be good.

AGILITIS

In a 1991 essay, Dawkins describes certain kinds of memes as “viruses of the mind.”¹⁴ He pushes the analogy a bit and speaks of the sufferers as exhibiting a range of “symptoms.” Here are three descriptions by Dawkins of symptoms that are present in the agile world.

1. *“The patient typically finds himself impelled by some deep, inner conviction that something is true, or right, or virtuous: a conviction that doesn’t seem to owe anything to evidence or reason, but which, nevertheless, he feels as totally compelling and convincing. We doctors refer to such a belief as ‘faith.’”*

TABLE 1

Vanilla	Agilese
Need, requirement, use case, scenario	User story, saga, epic
Phase, iteration	Sprint
Unscheduled work items	Backlog
Rework, redesign, modification, scrap	Refactoring
Postmortem, debriefing, end-of-project review	Retrospective
Progress meeting	Scrum, standup meeting, huddle
Productivity	Velocity
Risk, issue, constraint	Impediment
Team	Pod, cell

Agilistas often lose track that our topic is not natural science, that they are not getting closer to the knowledge of the one true way of developing software, and that all of this is just collections of opinions, with varying degrees of consensuality.

2. *"Patients typically make a positive virtue of faith's being strong and unshakable, in spite of not being based upon evidence. Indeed, they may feel that the less evidence there is, the more virtuous the belief."*

This attitude is typically not seen so much among the gurus of the agile faith, but among some of the later adopters.

3. *"The sufferer may find himself behaving intolerantly towards vectors of rival faiths, in extreme cases even killing them or advocating their deaths. He may be similarly violent in his disposition towards apostates (people who once held the faith but have renounced it); or towards heretics (people who espouse a different—often, perhaps significantly, only very slightly different—version of the faith). He may also feel hostile towards other modes of thought that are potentially inimical to his faith, such as the method of scientific reason which may function rather like a piece of anti-viral software."*

I have now seen many postings on blogs where one ounce of doubt triggers numerous rants—and sometimes nasty attacks.

AGILOLOGY

That last quote from Dawkins on the method of scientific reason will cause true scientists to react this way: "Oh, no, we are here to put the science into it." We now approach the domain of agiology, where scientific methods, preferably empirical quantitative methods, are applied to various bits and pieces of agility in an attempt to prove or disprove its superiority to some reference, which is often a very caricatural waterfall or disciplined approach. The research questions then often follow the pattern: *Look how good we could be relative to how bad we could have been.*

Unfortunately, because of the cost and difficulty of running really meaningful full-scale experiments (within the time frame of a single master's or doctoral thesis), small practice elements (individual agile memes) are tried out with groups of undergraduates, making generalization questionable (and often not questioned at all).

I reiterate here my assertion: *this is not natural science.* Unlike physicists who may get incrementally closer to an understanding of how atoms work, there is no ultimate "one and true" way of software development that we need to get at incrementally by digging harder.

If a scientific paper concludes that, say, pair programming is more efficient by 27.2 percent, with or without

explicitly stating "as observed with 32 undergraduate students writing a 50- to 80-line piece of Java code for a simple desktop program," decontextualization rapidly transforms this message into the following: "It was scientifically proven that pair programming is 27 percent more effective than the waterfall way;" then into this: "By adopting pair programming you'll cut your time to delivery by a quarter." The context is lost, sometimes intentionally, sometimes naively. Pair programming is just an example of an agile meme; replace it with your favorite one.

CONTEXT, CONTEXT, CONTEXT

Paraphrasing the motto of real-estate folks—location, location, location—I would like to propose one simple practice applicable to all the agilistas, agilese-talkers, militant practitioners of agilism, and researchers in agiology: "Context, context, context."

Explain the context of the project(s) where the message originates from (domain size, duration, etc.). If relevant, indicate your own history and background. I know much more about large technical software (command and control, telephone switches, realtime embedded stuff, compilers) than I know about management information systems, databases, or e-commerce. When I sat down with Kent Beck of XP fame for a long lunch in Montréal and we described to each other our respective backgrounds, it became clear that we had acquired different software development cultures, which had led us to different approaches, each successful in its own context, though apparently in contradiction at the superficial level when decontextualized (mostly by third parties).

The first principle of the context-driven testing school is: "The value of any practice depends on its context."¹⁵ Define your terms. They are part of the context. Until agility becomes a commodity taught in high school, relying on agilese is not enough. My user story is not yours. There is no ISO standard on sprint backlog.

Second, do not generalize your message to other contexts if you do not know them. A lot of software in the world is not facing a human user, cannot be delivered incrementally, has only one feature, and is algorithmically complex. Much of the agile memplex does not literally apply, though many agilistas love to extrapolate, "If it worked for me in my context, I do not see why it would not work in yours." This puts the burden of figuring out whether or not the message applies on the imitator.

The third principle of the context-driven testing school is: people, working together, are the most important part of any project's context. Ask for *context* before

Voyage in the Agile Memeplex

you shoot at others who may be operating in a very different context. Many of the mistakes we make result from assuming too much of a common mental model. Peter Senge suggests to “combine inquiry and advocacy... make your own reasoning explicit... actively inquire into other’s views that differ from your own... how did you arrive there?”¹⁶ It is not because a practice is part of the agile memeplex that you must use it.

COOLER HEADS MUST PREVAIL

I have been distraught at the level of dogmatism, bigotry, contempt, or just plain ignorance that I witness in the agile world. I am not blaming the topnotch agilistas, though they sometimes, and just for effect in writings and presentations, reduce their messages to their essential bones, to the slogan level, and they omit the context—both source and applicability.

As agility is crossing the chasm, however—as you can see if you attend any big software synod such as SD East or West or OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications)—many more people say (or repeat) rather uninformed messages with a strong conviction and little background, scoffing at anybody who dares to question their claims, even if it’s just a clarification about scope or context.

For writing these words, I’ll be shot dead as a traitor to the agilism cause, a defender of the waterfall church, a dinosaur, the über-curmudgeon, though I do value agility or agile practices in the proper context, and with the tainted glasses of my own 33-plus years of experience. But I would like my friends and colleagues to keep cooler heads, to question assumptions, not assume too much of a common, shared mental model, and contextualize what they hear, read, say, or write. ☺

REFERENCES

1. Agile Alliance. 2001 Manifesto for Agile Software Development. <http://agilemanifesto.org/>.
2. Boehm, B. W., Turner, R. 2003. *Balancing Agility and Discipline—A Guide for the Perplexed*. Boston, MA: Addison-Wesley.
3. Hofstede, G. 1997. *Culture and Organizations—Software of the Mind*. New York: McGraw-Hill.
4. Spencer-Oatey, H. 2000. *Culturally Speaking: Managing Rapport through Talk across Cultures*. New York: Cassel.
5. Dawkins, R. 1976. *The Selfish Gene*. Oxford, UK: Oxford University Press.
6. Blackmore, S. 1999. *The Meme Machine*. Oxford, UK: Oxford University Press.
7. Ibid.
8. Ibid.
9. Dennett, D. C. 1990. Memes and the exploitation of imagination. *The Journal of Aesthetics and Art Criticism* 48: 127-135.
10. Whitty, S. J. 2005. A memetic paradigm of project management. *International Journal of Project Management* 23 (8): 575-583.
11. Senge, P. M. 1990. *The Fifth Discipline*. New York: Currency Doubleday.
12. Speel, H.-C. 1996. Memetics: On a conceptual framework for cultural evolution. In *The Evolution of Complexity*, ed. F. Heylighen and D. Aerts. Dordrecht, Holland: Kluwer.
13. Aunger, R. 2002. *The Electric Meme*. New York: The Free Press.
14. Dawkins, R. 1995. Viruses of the mind. In *Dennett and His Critics: Demystifying Mind*, ed. B. Dalhblom, 13-27. Malden, Massachusetts: Blackwell Publishers.
15. Contextualists. 2007. The seven basic principles of the context-driven school. <http://www.context-driven-testing.com/>.
16. See reference 11.

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

PHILIPPE KRUCHTEN, Ph.D., P.Eng., is the principal of Kruchten Engineering Services Ltd. He is also cofounder and president of Agile Vancouver and a professor of software engineering at the University of British Columbia. He led the development of the Rational Unified Process and is author of the book *The Rational Unified Process: An Introduction* (Addison-Wesley Professional, 2003).

© 2007 ACM 1542-7730/07/0700 \$5.00