Why Can't We All Play Nice?

Linda Rising

This article is based on my research in stereotyping and collaboration—the two opposing forces that work to prevent and support the building of great teams. It was only late in my long career that I realized how important the "people" side is. Tools, programming languages, environments, and all the other technical stuff are important, but that "softer" side can be really, really hard. *–Linda Rising, Associate Editor*

I DON'T KNOW about you, but even though the US election is months away, I'm already worn out. All the political ads, each party trying to outdo the other and dig deep into the American psyche for some touchstone, some area of resonance, as it tries to grab our hearts and minds.

It's frightening what happens during a political campaign. We take sides. In a way, I guess we all do this all day, every day, on just about anything: North or South, black or white, Christian or Muslim, Red State or Blue State, Republican or Democrat, tall or short, fat or skinny, old or young.

I know the game. People want to convince you that they're on the "same side" so that you'll vote for their party or buy their product or work for their company. But I keep asking why we seem to want to grab an identity and use it to label others as "us" or "them"?

Evolutionary Psychology

I recently reread an article from *Harvard Business Review* written by Nigel Nicholson quite a while ago ("How Hardwired Is Human Behavior?" July/Aug. 1998). It was my first introduction to evolutionary psychology.

To uncover the answer to my question, we have to go back a few years—200,000 of them, to be exact—when humans first appeared on Africa's Savannah. Our Stone Age hunter-gatherer ancestors were focused on survival, which translates to food, shelter, and a mate with which to reproduce. A lot has happened since then, but scientists who study this area say that we're still carrying pretty much the same brain that we had back then. The drivers for evolutionary change haven't brought our mental equipment up to date. As Nicholson noted, "You can take the person out of the Stone Age, but you can't take the Stone Age out of the person!"

Over time, the human brain evolved. The process favored brains that were better at solving day-to-day problems: choosing shelter, hunting animals, gathering plants, negotiating with friends, defending against aggressors, finding mates, raising children, and so on. Those with better brains left more children and ultimately produced us.

Things changed dramatically with the appearance of agriculture 10,000 years ago. Humans could move beyond living hand-to-mouth, and a series of rapid developments brought us to the present day. Unfortunately, three factors contributed to our brains' remaining pretty much the same as they were in the Stone Age:

- By around 50,000 years ago, human populations had dispersed so that genetic mental mutations couldn't possibly spread.
- We haven't been subject to any environmental pressures that would cause evolutionary change.
- Ten thousand years seems like a long time, but it's not enough for significant genetic modifications to become established throughout humankind. The environment has changed, but we have not.

Stone Age hunter-gatherers constantly faced new puzzles. Which plants can be eaten without becoming sick or dying?

Where can we hunt? How can we tell whether someone can be trusted?

We're Great at Labeling

To provide structure to an uncertain world, our ancestors developed an impressive talent for sorting and classifying information. Researchers have found some existing illiterate tribes with a well-defined taxonomic knowledge of the animals and plants in their locality. We humans are great at creating categories for everything in our lives.

In the Stone Age, of course, these capabilities weren't limited to the flora and fauna. To do well in their communities, humans had to make the correct alliances. They had to know with whom to share food. They had to know what untrustworthy individuals generally looked like, because it would be unwise to deal with them. As Nicholson notes, "Human beings became hardwired to stereotype people based on very small pieces of evidence, mainly, their appearance and a few readily apparent behaviors."

People naturally sort others into in-groups and out-groups, just by their appearances and actions.

The power of classification remains with us today. People naturally sort others into in-groups and out-groups, just by their appearances and actions. We continuously subconsciously (and sometimes consciously) label other people: "She's a born leader," or "He's a bozo." Research has shown that managers decide as early as three weeks after hiring someone whether he or she will be successful or not.

But people are more complicated than our simple labels imply, and it's disturbing to know that we aren't wired *not* to see them as such. This might explain why some groups in organizations have trouble getting along. The battle between business and development has been with us from the beginning. The techies in IT departments have difficulty getting along with the groups they're supposed to support, and vice versa. We're all too busy labeling others as "them" and disregarding their contributions.

We're Happy in Small Groups

The world is increasingly complex, and we're incurring enormous costs by exercising our Stone Age decision-making process in complex information-based environments. This is a huge distraction in our quest to think globally and solve large-scale communal problems.

Fortunately, a counterbalancing inherited trait might save us. In addition to classifying those around us, we also have a hardwired tendency to tightly bond with those we work with in small groups as we head toward a common goal. In these small groups, everyone on the team is linked to everyone else so that you can't succeed unless the others do (and vice versa) or that you must coordinate your efforts with the efforts of others to complete a task. What results is respect for others' abilities and contributions.

It's pretty clear that we like being trusted and respected. It might be that on the secret list of wants and needs that we all have in our back pocket you'll find trust and respect at or very near the top.

Psychologists call this *social interdependence*. Even though we in software development don't always do a good job of measuring the impact of our practices, psychologists are all about measurement. They've observed in groups that exhibit social interdependence as

- an increased effort to achieve, both from the individuals and the group;
- more positive relationships, more giving and receiving of social support; and
- improved psychological health, including increased self-esteem and decreased anxiety and depression.

To bring this home, I thought I would share some examples from retrospectives I've facilitated at companies all over the world (just so you won't think this is a problem specific to the US). In one company, an enormous amount of time was spent (both on the project and in the retrospective) addressing the concerns of one subteam whose members didn't get a t-shirt, while all other teams working on the project did. The have-not subteam initially raised the issue at the retrospective. We captured it on a card and moved on, but the have-not subteam kept coming back to it. Each new issue was somehow tied to the t-shirts. You might be wondering, "How did this team lose out?" It was an honest mistake: a new project manager intended to buy t-shirts for members of all the teams, but somehow overlooked the have-not subteam, probably because it wasn't co-located.

Research has shown that clothing is a strong identifier for groups. Why do you think the military provides uniforms? When a group adopts a name, slogan, symbol, or t-shirt, there's a clear call to the hardwired "us against them" behavior. Managers, leaders, team members, please be aware of the power of identity. Yes, it helps "gel" a team, and yes, it can give lagging team spirits a boost, but it also causes a division between the team and the rest of the organization. All teams today need to work and play well with others: support groups, external testers, and, of course, clients, users, and customers. If you must have names, slogans, symbols, or t-shirts, disseminate them widely to involve as many others as possible and tone down any hint of exclusivity. Make it all about everyone, not just a special group.

Here's another story. A project team grew rapidly in a short time (lots of interesting material there!). The software folks ended up sitting in two rows of cubicles on either side of the hardware guys. During the retrospective, we learned that both groups of software developers got along just fine with the hardware guys but not with each other. Out of this "us against them" created by the separation in cubicle space, two architecture leaders emerged, each with a different vision for the product. There was so little understanding between the groups that when the time came, the software pieces wouldn't integrate. The result was a lot of blaming, finger-pointing, and ultimately an effort that cost the company an enormous amount but produced nothing.

In product development, as in real estate, what counts is location, location, location. Meaningful communication decreases as distance increases. When team members sit close together, communication goes up. Those who aren't close are automatically in the out-group. Without meaningful communication, misunderstandings arise, motives are assigned, and assumptions shoot up. It doesn't take long for wars to break out. To overcome this strong tendency to see others who are separated by distance as outsiders, continually look for ways of gathering together all people who are on a project. A Friday afternoon pizza party will work, for example. Get everyone in the same room with the freedom to walk around and share ideas. This problem has no easy solution in our development world of distributed teams. A good friend who has written patterns for this environment suggests that time be set aside in the schedule for each team (not just the managers) to visit the other sites. This, of course, adds to the cost, but the benefits can be considerable. I also recommend sharing pictures, recipes, and stories. The open source movement has its own solution to the distance problem.

On another large project, we noticed that the developers got along really well with the system testers. This was long before agile development proposed that the wall between code and test be razed and that programmers be involved in testing. In interviews with the test manager, we discovered what was at that time a radical approach to the intersection between the two phases in waterfall: code and test. The test manager said that on his projects, a system tester was always involved early with the development team. The tester could provide input on whether the requirements were testable and work side by side with developers instead of waiting for the code to be "thrown over the wall." It was remarkably successful and in retrospect seems so obvious, but it was radical thinking at the time. Instead of the worn-out labels, instead of an "us against them" view of the world, this team made great progress, addressed problems in a timely manner, and delivered quality code on time. It can be done.

Finally, a nonsoftware story. During World War I, enormous numbers of troops were required to spend years in hundreds of miles of trenches. What made trench warfare so different from most other combat was that often the same small units faced each other in immobile sectors for extended periods of time. After awhile, troops on both sides knew about events on the other side: when meals were served, when the wounded and dead were carried away, when reinforcements arrived. This closeness, across a few hundred yards of "no man's land" resulted in a strange phenomenon—pockets of peace broke

IEEE Software

HOW TO REACH US

WRITERS

For detailed information on submitting articles, write for our Editorial Guidelines (software@computer.org) or access www.computer.org/software/author.htm.

LETTERS TO THE EDITOR

Send letters to

Editor, *IEEE Software*
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
software@computer.org

Please provide an email address or daytime phone number with your letter.

ON THE WEB

www.computer.org/software

SUBSCRIBE

www.computer.org/software/subscribe

SUBSCRIPTION CHANGE OF ADDRESS

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify *IEEE Software.*

MEMBERSHIP CHANGE OF ADDRESS

Send change-of-address requests for IEEE and Computer Society membership to member.services@ieee.org.

MISSING OR DAMAGED COPIES

If you are missing an issue or you received a damaged copy, contact help@computer.org.

REPRINTS OF ARTICLES

For price information or to order reprints, send email to software@computer.org or fax +1 714 821 4010.

REPRINT PERMISSION

To obtain permission to reprint an article, contact the Intellectual Property Rights Office at copyrights@ieee.org.

out. No formal documents were signed. No meetings were held. Units on both sides just noticed that during certain times of the day, there was no shelling, no shooting. A mini-truce was held.

We hearken back to our definition from the psychologists and wonder, "What common goal did these different sides share?" After all, this was war. Of course, it only takes a second to realize that the common goal was the most important goal for all of us—survival. Probably those on both sides were saying, "It wasn't my idea to be here. I just want to go home." That's a lot to hold in common.

As I was preparing this article, I picked up the latest issue of *ACM Communications* and read about the ACM A.M. Turing Award winner, Judea Pearl, at the University of California, Los Angeles. Along with Akbar Ahmed, Chair of Islamic Studies at American University, Pearl is involved with the Daniel Pearl Dialogue for Muslim-Jewish Understanding. Daniel Pearl was Judea Pearl's son, a *Wall Street Journal* reporter who was kidnapped and killed in February 2002 by terrorists in Pakistan. Judea Pearl has passionately devoted himself to the topic of this article—the "us against them" problem and brings in his expertise as a scientist. I found the results quite compelling.

The bottom line is, yes, we're hardwired to stereotype. No, we can't do anything about that. Yes, being aware will help us be proactive and do what we can to counteract these inborn traits. Looking for commonality, the things we all care about, will help bring about the best alternative for that "us against them" hardwiring—by engaging our other hardwired tendency, which is to work collaboratively toward a common goal in an environment of trust and respect.

So at the national level, instead of saying, "I'm red, and you're blue" (fill in your favorite color), we should start looking for a common vocabulary. Words that reflect shared concerns will prove that we're on the same side. Let's have conversations about jobs, healthcare, education, and, especially in our troubled world, peace.

The Iroquois had a masterful way of negotiating conflict. Each person had to struggle to get to the point where he could articulate the other's position. This enhances collaboration instead of escalating the tendency of teammates who disagree to think of themselves as being on opposite sides. It changes the dynamic from the hardwired "us against them" to see ourselves as working to resolve an issue together. That's what the system testers and developers did in my third story. They realized that we're all in this together, that it's the job of all of us to deliver. It's about "our" bugs and "our" release. SW

Acknowledgments

Thanks to the excellent reviewing and commenting from my friends, colleagues, and reviewers: Ayse Bener, Robert Glass, Dave Thomas, and, especially, Rebecca Wirfs-Brock, who was my shepherd for this article. Thanks especially to Bob for pointing out the book *How Google Tests Software*, which describes the company's process of having developers also test and testers also develop.

LINDA RISING is an internationally known presenter on topics related to patterns, retrospectives, agile development approaches, the change process, and the connection between the latest neuroscience and software development. She has a PhD in the field of object-based design metrics from Arizona State University. Contact her at linda@lindarising.org.