

PROFESSIONAL TESTER

SUBSCRIBE
It's FREE
for testers

Essential for software testers

August 2011 | £4 / €5 | v2.0 | number 10

THIS ISSUE OF
PROFESSIONAL TESTER
IS SPONSORED BY



HOW TESTERS COMMUNICATE AND HOW THEY SHOULD

Including articles by:

Bogdan Stefan Bereza
Victo

Brad Higgs
HP

Ian Molyneux
Intechnica

René Tuinhout
Richard Janssen
Logica

George Wilkinson
Aquidium

Wrong message received loud and clear

by Bogdan Stefan Bereza

Words fail us



Bogdan Stefan Bereza explores the limitations of test communication improvement

Discussion about how testers communicate has a tendency to rely on rather naive psychology. Obvious generalizations (“testers demand a lot of detail and are pedantic”, “the best basketball players are tall” etc) are of no use and less obvious ones are too often untrue, making them dangerous.

That is not to say psychological and social issues are not important to the success of projects. I believe it is obvious that they are, and the huge amount of material, especially conference sessions, that aims to define ideal tester traits and find foolproof ways to motivate testers can't all be wrong to do so. In fact the importance of personal behaviour is hugely underestimated by managers fed on PRINCE2, ITIL and PMI. I recommend Tom De Marco's novel *The Deadline* (Dorset House, ISBN 9780932633392) for some entertaining insight into the

effects of that. But knowing something is important does not make it easy to manage or, necessarily, make trying to manage it a good idea.

How communication happens is closely linked to what is being communicated and that, of course, can be anything. I think this tends to limit the usefulness of theories about how testers communicate. For example I do not believe that:

- IT projects require different communication skills from any other project types such as building or wilderness expeditions
- the desirable communication or behaviour patterns of testers, requirement managers, designers and programmers are different
- communication skills and patterns can sensibly be discussed in isolation from other psychological and social aspects.

So I will restrict myself in this article to the practical, and ask you the reader, if you choose to use it, to do so in practice only, not to support crazy ideas such as that testers are different from other people or “the mindset to be used while testing and reviewing is different from that used while developing software” (*ISTQB Certified Tester Foundation Syllabus Version 2011*).

Permutation of misunderstanding

Words can break your bones. Imagine a conversation between the driver of a car and his passenger, who is also his wife. Passenger: “The traffic light is green”. Driver (shouting angrily): “I know!”

Why did he shout? In his series of three books *Miteinander Reden* (“To talk with each other”, rororo,

How testers communicate and how they should

ISBN 9783499627170) Friedman Schulz von Thun suggests how to understand why. According to him here are four “channels” in every human conversation. Here they are expressed as parameters of a function definition in pseudocode:

```
listen(message facts; message appeal; message self revelation;
message relationship)
//analyse message here
if upset then return false //stop listening
else
//process message here
return true //continue listening
endif
```

Here are the parameter values the passenger sent when invoking the function:

```
driver.listen("We're equal","The traffic light is green",
"Keep going","I feel safe with you and love to help you");
```

Here are the values the driver received:

```
driver.listen("I got you!",NULL,"I am ashamed of my husband
when he drives","I am superior to you and must help you");
```

This means that the driver thinks that making him angry was actually his wife's goal, whether consciously or subconsciously. So even what appears to be a perfectly innocent and factual statement can be misunderstood as personal attack.

Let us assume that these are in fact the only four important parameters, and that each is binary, ie can take on only two meanings, both absolutely defined. Even with these gigantic oversimplifications, every message can be interpreted in 16 ways of which one is correct.

In communication, the simple is complicated, even more so when we consider how easy and how tempting it is to disguise an actual, intentional personal attack as an apparently factual

statement in, for example, an incident report. Doing that makes the factual information communicated meaningless: instead of reading it literally, recipients try to decide for themselves what it is “trying to say”. That's even more damaging than the direct, honest expression of opinion we are so often warned against, eg “this is a really bad mistake and should not have been made”.

The point is there should be no opportunity for hidden messages. Then messages which are not there will not be imagined either. In an article on the EuroSTAR blog (<http://eurostarconferences.com/blog/2011/4/28/21-great-myths-of-popular-testlore.aspx>) I wrote about what I consider a myth, that incident reporting requires good interpersonal skills: “If you believe it does, you have a desperately inadequate incident reporting process. If your incident reporting process is good and appropriately automated, it minimizes the impact of interpersonal issues. The process, and the tool, automatically forces even most socially incompetent people to provide all needed information in a socially acceptable way. It is the process, not its users, which must have good interpersonal skills.”

The same principle applies to all our work, not just incident reporting. Any confusion caused by lack of clear-cut responsibilities, obligations, decision structure and communication channels creates food for the growth of cancerous anger and chronic conflict. Proper process and automation can work wonders in placating people, achieving far more than therapists or, worse, amateur psychological dabbling can. The burden of routine and boring but error-prone chores makes one irritable. A creative, organized environment makes one happy. Provided you understand how a tool works it is easier to accept criticism from it than from a person.

Furthermore, communication can be improved and personal conflicts

prevented by test automation in general, not just by automating communication. It is much easier to refrain from criticism and accept other people's fallibility when you feel secure that automatic controls will prevent small mistakes from becoming full-blown disasters. I remember a Monday when a static analyser generated many more incidents and warnings than was usual, a fact displayed vividly on the application manager's dashboard. It turned out that several developers had attended a wedding the previous weekend and perhaps celebrated a little too well. But the developers had previously accepted static analysis as a valid check of their work. They had not tried to abuse it as a safety net, but had done their best as always and on this occasion been less successful than usual due to a human situation (tiredness, stress, distraction, circumstances etc – we need draw no moral distinction). Instead of complaining or getting angry, the manager was able to recognize a temporary situation that would cause no serious harm and take no action.

Importance of understanding how people are different

Although I said above that testers are not different from anyone else in any general way, I do believe that differences in individual personality affect work and that important factors can be measured and matched with activities, bringing benefit. As the *ISTQB Certified Tester Advanced Level Syllabus Version 2007* states, "The goal is to make each individual successful as an individual while contributing to the overall success of the team. This is largely done by matching personality types to team roles..."

There are techniques to measure certain personality traits reliably in less than ten minutes, and task or role requirements (less reliably) in a few hours. Many are well known but here's one that might be new to some readers. The psychometric assessments available at <http://www.thomasinternational.net>

measure on four basic scales: dominance, influence, steadiness and compliance. I discussed them in my presentation to the Test Management Summit in 2007 (slides and notes are available at <http://uktmf.com/index.php?q=node/55>). For example, Alexander the Great was high on dominance and low on steadiness. The eponymous protagonist of Scott Adams' cartoon strip *Dilbert* – to many, an archetypal software engineer – is low on influence and high on compliance. He believes that good engineering practices are to be followed rigidly regardless of other factors.

If, like the editor of this magazine, you do not believe in stuff like this, think it is rubbish and feel threatened by the very idea of someone measuring your personality at all, too bad, because I assert that these methods work in practice. They are not to be misused for labelling people, but rather as a fast and effective, though of course not perfect, early warning system. They can enable leaders to say things such as: "Steve, I know you are high on steadiness and would rather work with well-defined requirements, but please suffer me to

ask you for some agile-exploratory stuff for a while, there is a good chap" and "Susan, although you score high on dominance and influence and would love to lobby for better test process, please will you accept comparing these two very large result files for a week or two?".
[If I were Steve or Susan I would tell this leader to get lost –Ed]

How testers get information and how they don't

Focusing too much on improving communication may take attention away from more dangerous issues. For example modern tools make it easy for a PM to receive the accurate information that a certain feature carries an initial estimate failure risk of 50% and that 80% decision coverage of relevant components has been achieved with no outstanding defects. But it does not tell him or her if or by how much the risk has diminished as a result. It cannot, because almost nothing is known about how to translate test results into accurate, quantitative risk information. Perhaps more testers should communicate less and think about how to solve that problem more ■

Bogdan Stefan Bereza, previously known as Bogdan Bereza-Jarociński, is a testing consultant, speaker and trainer and a long-time contributor to Professional Tester. He is the proprietor of VictO (<http://victo.eu>)

