What's In a Word?

by Antony Marcano

Languages are evolving entities. Sometimes that evolution comes through common misuse. A prime example is the word *enormity.*

In everyday language enormity is often used as a synonym for *enormousness*. Although the two words are etymologically related, they are not synonyms. Both words have similar origins in the Latin ex norma, meaning "outside the norm," but that's about as far as the connection goes.

According to the Cambridge English Dictionary, enormity originally meant "an extremely evil act or the quality of being extremely evil." So, an appropriate usage would be to say "the enormity of the crime was horrifying." Over the years, people have increasingly used enormity to mean size. For example, someone might say "the enormity of the crowd was overwhelming" when what he really means is "the enormousness of the crowd ..."

This common misuse has earned the word enormity the additional entry in various dictionaries: "very great size or importance." This definition is more commonly used than the original, so much so that few people even know the original meaning of the word.

Evolution through common misuse is a reality of human communication. This is acceptable as long as we don't lose the ability to express ourselves in the process.

Software development has at least one phrase that has suffered from the same syndrome of common misuse, and in the process, we have lost something important.

For as long as I can remember, I've heard people use the phrase *quality assurance* as a synonym for testing. This misuse seems to have become increasingly common, to the point where I can only find a handful of people who know its original meaning.

The origins of the term quality assurance are in manufacturing, building, medicine, and other (arguably) more mature industries. In those industries, however, there is another term that encompasses the activities that *we* often refer to as quality assurance—that term is *quality control.*

Many years ago, I met the newly appointed QA manager for a well-known software company. He had come from many years as a QA manager in the clothing industry and was very concerned that what this software company called QA was far from what he was used to. He was horrified to discover that the scope of his job had been reduced to what he had previously considered quality control. Quality control is when a product is *tested* or *inspected* to see if it is "fit for purpose." In the clothing industry, this results in "rejects" finding their way into discount stores and market stalls. In our industry, this results in these defects either being accepted or fixed. So, for us, you might say that the purpose of quality control is to answer the question "Does our product work?"

> By using the phrase quality assurance to refer to what is more properly called quality control (i.e., testing), we may have put a tick in a box that has long since lost its true meaning.

If what we call quality assurance is actually quality control, then what does quality assurance *really* mean?

Where quality control is more concerned with defect-*detection*, quality assurance (outside the software industry) is focused on defect-*prevention*. Quality assurance encompasses the activities you perform to assure that your process builds quality into your product. Or, more realistically, it encompasses the things you do to increase the probability that you are able to build a product of appropriate quality. Thus, quality assurance is more concerned with the process than the product. Or, we might say that the purpose of quality assurance is to answer the question "does our *process* work?"

So, if what we call QA is actually QC, then who is doing the *real* QA? Who, indeed, is answering the question "Does our process work?"

Some might argue that testers should take advantage of the misplaced letters "QA" in their job titles and take ownership of both QA and their usual QC responsibilities. This is a naïve assertion, since real QA is almost impossible unless those charged with that responsibility is empowered to affect the entire lifecycle.

Has the responsibility of real QA been allocated to organizational *process improvement* initiatives? What of the important role that real QA should play at the project level? On a project, the QA buck actually stops with the project manager. Yet new teams form, and the first time the team realizes that its *process* doesn't work (sometimes months into the project) is when the testers find ways the *product* doesn't work. Is this why testing has become known as QA?

By using the phrase quality assurance to refer to what is more properly called quality control (i.e., testing), we may have put a tick in a box that has long since lost its true meaning. Is this a problem? Yes, I think it is. It's a problem because of what we've lost or, more importantly, what we've hidden from ourselves as software professionals. We've hidden the fact that there's an important question that we are no longer asking. And, in the process, software development teams may have lost an important faculty that would enable them to be more efficient at building—and faster at delivering—better software. {end}