Tejas Software Consulting
Fort Worth, Texas
http://www.tejasconsulting.com

office 817-294-3998
cell 817-360-1597
fax 817-294-0572
faught@tejasconsulting.com

Position Statement
"Technology and Your Business"
Fort Worth Chamber of Commerce Small Business Council Breakfast
February 13, 2002

My specialty is software quality. I'll address three different angles on how software quality relates to the technology that you use in your business. (Hint: you'll probably relate most to item #3.)

1. Software Product Development

This is where I do most of my work, helping companies manage the quality of software products that they create for sale. Odds are, though, you're not in the software business. Still, many of the skills I use in this area are applicable to the categories below.

2. IT Development

This is a fancy term that I'll use to refer to the software that you or an IT outsourcer creates for internal use in your company. If someone in your company creates or customizes software for internal use, this is basically the same process as item number one – you're creating software that will impact your business. The same engineering principles that can lead to better software apply here, though even seasoned IT engineers often don't apply them.

3. Software Consumers

You probably use off-the-shelf software products in some aspect of your business. The difference between your business and the software product developers is that your entire business doesn't depend on having high quality software. Or does it? If you can't email your customers, print out an advertising flyer, browse your supplier's web site, or pay your taxes, what is the impact on your business?

You have probably encountered bad software before. And you probably blamed yourself. "I'm no computer expert, I must be doing something wrong." And you may be too embarrassed to ask for help, or, heaven forbid, try to return the software to the stores, all of which have big signs saying that you can't return any opened software purchases.

(over)

This gets us into two areas –

A) Problem isolation

When you have a problem with software, you can benefit by doing some experimentation to learn more about the problem. You want to take a problem like "My word processor crashes every time I print" and isolate it down to something more like "My processor crashes when I print a document with a graphic larger than 200K in size." Then you can use this information to work around the problem, and you can also have a much more useful problem report to send to your software vendor.

Use a disciplined approach to isolate problems. For the word processor example, you can copy the file to a scratch file, then start removing parts of the file to see if the problem is still reproducible. Maybe you delete a large graphic from the file and then print again – POOF, the problem goes away.

B) Consumer advocacy

If you spend good money on software that's so full of bugs that it's just not usable, what do you do? Get your money back! Despite the dire warnings to the contrary that you'll find in the fine print of the software license, the Uniform Commercial Code gives you basic protections against defective products. You have the right in many cases to return software to either the retailer or the manufacturer.

However, one thing to watch for is a new law called UCITA that has been introduced in Texas and many other states. This law would make it much easier for software companies to sell you junky software and leave you with no recourse.

References

- Software consumer advocacy - *Bad Software*, by Cem Kaner, and also www.badsoftware.com
- UCITA information – "Consumer protection efforts threatened by UCITA" and "My observations on UCITA in Texas" linked on my web site at www.tejasconsulting.com
- A basic introduction to software testing – *Software Testing* by Ron Patton (but still a difficult read if you're not a software engineer)