



An e-newsletter published by
Software Quality Consulting, Inc.

Spring 2012
Vol. 9 No. 1

Welcome to **Food for Thought™**, an e-newsletter from **Software Quality Consulting**. I've created free subscriptions for my valued business contacts. If you find this newsletter informative, I encourage you to continue reading. To receive this newsletter via e-mail, visit www.swqual.com and enter your e-mail address for a free subscription..

Your continued feedback on this newsletter is most welcome. Please send your comments to steve@swqual.com

This Month's Topic

Becoming Best-in-Class

Today's cars have a frightening amount of software. As I reported in my newsletters in **February** and **March** of 2010, today's automobiles have upwards of **100 million** lines of code. By comparison, about **6.5 million** lines of code are required for the avionics and on-board support systems for the new Boeing 787 Dreamliner. What does all of this automotive software do? As reported in IEEE Spectrum [4], software is used in the following areas:

Air-bag system	Anti-lock Brakes	Automatic Transmission
Alarm system	Climate control	Collision avoidance system
Cruise control	Communication system	Dashboard instrumentation
Electronic stability control	Engine ignition	Engine control
Electronic seat control	Entertainment system	Navigation system
Power Steering	Tire pressure monitoring	Windshield wiper control

Ford recently introduced a touch screen control for several systems to its vehicles. Safety issues related to using a touch screen while driving aside, the software-intensive MyFord Touch system has been nothing but trouble for customers.



MyFord Touch screen

"One customer reported the navigation system often malfunctioned, the rearview camera frequently stayed on while the vehicle was moving forward and the system randomly rebooted. The voice controls typically do not work until the vehicle has been on for five to 10 minutes, meaning short trips require dialing phone calls by hand, only to have the call cut off when the system finally starts up..." [5]

Frustrated Ford owners paid about \$395 for the MyFord Touch option only to find it exceedingly difficult to use and, in some cases, not usable at all. While Ford has been somewhat successful in trying to improve customers' perception of quality, this recent software glitch is a significant setback.

The problem Ford is having with its MyFord Touch software [1] indicates how poor software can dramatically affect customer satisfaction. Ford dropped from 5th to 10th in the latest Consumer Reports annual ranking of auto manufacturers and dropped from 10th to 20th in the Consumer Reports annual reliability ranking. J.D. Power & Associates ranked Ford 23rd in its new car quality survey – down from 5th place.

Software problems can and do affect customer satisfaction which in turn have a dramatic impact on sales. How then can companies like Ford become best-in-class when it comes to software?

What does best-in-class mean?

Best-in-class companies develop products that have fewer customer-reported defects and, as a result, provide more value to customers. From a software perspective, best-in-class companies have very few customer-reported problems. Most best-in-class software companies use the **Defect Removal Efficiency** [2] metric to measure their performance along with frequent customer satisfaction surveys.

Defect Removal Efficiency measures how effective your tests are at removing defects your customers are likely to find based on actual customer use over a defined period of time. The metric is calculated as follows:

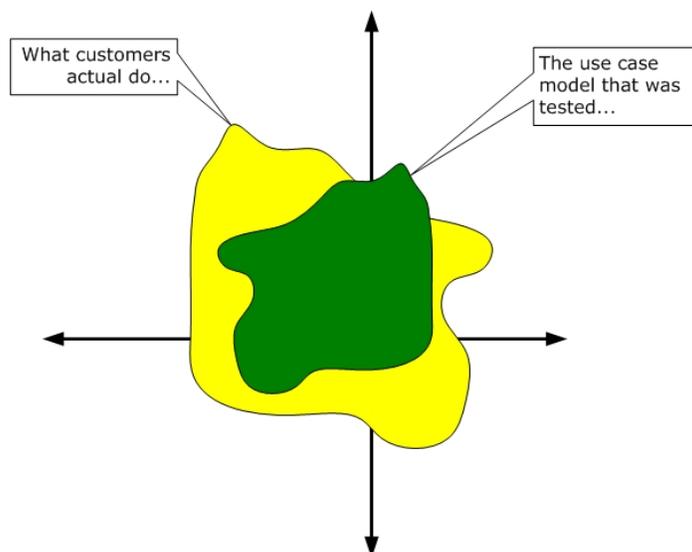
$$\frac{\text{Total defects we find prior to release}}{\text{Total we find} + \text{customer-reported defects}}$$

Best-in-class software companies have defect removal efficiencies of 99.5% or higher. The higher the defect removal efficiency metric, the fewer problems your customers experience in using your software. Note that very high defect removal efficiency doesn't mean that there aren't any defects in the software – just that customers aren't finding them. The defect removal efficiency for mediocre software is between 80-85%.

A survey performed by Capers Jones [3] of over 1,500 projects showed the following:

Defect Removal Efficiency (%)	Number of Projects	Percentage of Projects
> 99	6	0.4%
95-99	104	6.9%
90-95	263	17.5%
85-90	559	37.3%
80-85	408	27.2%
< 80	161	10.7%

Low defect removal efficiency can result from a lack of understanding of how your customers are actually using your software. Watts Humphrey gave us the notion of the "testing footprint" (illustrated below) which exposes differences in how you test your software and how your customers are actually using your software. Best-in-class companies truly understand how customers are using their software and have focused testing efforts based on this understanding.



How to become best-in-class

As you might expect, becoming best-in-class requires a strong commitment from management. Here then is my **Top Ten List** for becoming a best-in-class software company:

1. **Balance defect prevention and defect detection activities.**

Management should provide incentives and rewards for preventing problems before they occur. Management needs to recognize the significant cost savings that can be realized by focusing on preventing problems rather than fixing problems. Management can do this by providing incentives and recognition for those who proactively prevent defects on project teams.

See my **Nov 2011 newsletter** for more details on this topic...

2. **Improve requirements writing skills.**

Poorly written requirements are the norm on many projects. Clearly written, testable requirements are the lynchpin for improving defect removal efficiency and becoming best-in-class. Project teams need to learn simple techniques for expressing requirements clearly and unambiguously to prevent problems associated with poorly written requirements.

3. **Involve SQA early to assess requirements testability.**

SQA engineers need to become involved early on to assess requirements and identify testability issues early when these requirements are less costly to change.

4. **Identify Root Cause of every customer-reported defect.**

Every customer-reported defect represents an aspect of customer use that is not well understood. Each customer-reported defect should be considered for root cause analysis in order to determine why this defect escaped and how did we miss it?

5. **Perform Pareto Analysis and take corrective action.**

Once root causes are identified, a pareto analysis should be performed to determine what aspects of your process are causing the largest number of defects – this is the proverbial 80/20 rule. Fix that part of your process that is causing these defects in order to improve defect removal efficiency.

6. **Use an effective Peer Review process.**

Many companies claim to perform peer reviews but few do them well. Peer reviews have been proven time and time again to be extremely effective in finding problems that would be far more difficult and costly to find in test. Work on improving the effectiveness of your peer review process by requiring participants to come to review meetings prepared!

7. **Everyone must take responsibility for his or her work.**

Becoming best-in-class requires a strong personal commitment from software developers and SQA engineers. Software developers must be willing to say:

“I believe my code is as good as it can be and correctly implements defined requirements. I challenge anyone to prove me wrong.”

Similarly, SQA should be willing to say:

“I believe my tests are as good as they can be and accurately reflect defined requirements and how our customers use our software. I challenge anyone to

prove me wrong.”

8. Use a combination of testing perspectives.

From experience, we know that using a combination of several testing perspectives or stages increases the ability to identify defects. Examples of testing perspectives or stages include unit testing, integration testing, user testing, system testing, etc.

As reported by Capers Jones [3]

Number of Testing Stages	Percent of Effort Devoted to Testing	Cumulative Defect Removal Efficiency
1 testing stage	10%	50%
2 testing stages	15%	60%
3 testing stages	20%	70%
4 testing stages	25%	75%
5 testing stages	30%	80%
6 testing stages*	33%*	85%
7 testing stages	36%	87%
8 testing stages	39%	90%
9 testing stages	42%	92%

* Six test stages, 33% cost and 85% removal efficiency are U.S. averages.

9. Increase domain knowledge of testers.

Testers need domain knowledge in order to know where they should focus their testing efforts. The testing footprint mentioned earlier can be an effective tool for focusing the testing effort – but it requires that testers really understand what real users do. How can testers acquire domain knowledge? They need to spend time observing what real customers do when they use your software. They also need to spend time in Customer Support listening to the kinds of problems customers encounter on a day-to-day basis.

10. Measure progress using customer satisfaction surveys and defect removal efficiency.

Measuring customer satisfaction on a regular basis and responding to feedback is a sure sign of a best-in-class company. When customer problems arise, best-in-class companies use the opportunity to win back customer trust by going above and beyond what customers would typically expect. For example, instead of simply pushing out hot fixes, find a way to delight customers by doing things like extending their support contract or providing free support for a limited time.

Another sign of a best-in-class company is measuring and using the defect removal efficiency metric to continually improve. Customer satisfaction and defect removal efficiency are good examples of measurements that can be tied to overall corporate quality improvement goals.

How to win over unhappy customers – or not

Okay, so you own one of the Ford cars with the defective MyFord Touch software. What happens now?

Ford recently announced they are sending updated software on thumb drives directly to about 300,000 customers so they can install new software themselves. Alternatively, they can bring their care in to a dealer who will do the installation for them.



If you want to do the upgrade yourself, it will take about an hour of your time and the **car must be running the whole time.** This will waste about a gallon of expensive gas. Let's see, I need to spend an hour of my time and waste a gallon of my gas to fix a defective product that I paid a lot of money for... hmmm...

Am I missing something here or are all those unhappy MyFord Touch customers still unhappy?

'till next time...

Monthly Morsels

Every month in this space, you'll find additional information related to this month's topic.

1. Priddle, A., "Ford sending fixes for MyFord Touch systems", Detroit Free Press, Mar 6, 2012
2. Jones, C., "Software Defect Removal Efficiency", IEEE Computer April 1996.
3. Jones, C., SOFTWARE DEFECT REMOVAL: THE STATE OF THE ART IN 2010.
4. Charette, R., "This Car Runs on Code", IEEE Spectrum, February 2009.
5. Bunkley, N., "After Ratings Drop, Ford Reworks Touch Screens", NY Times, March 5, 2012.

About SQC

Software Quality Consulting provides a full-range of software engineering services for safety-critical industries and mission-critical projects. Our goal is to help create safety-critical and mission-critical software that meets our client's needs, complies with all applicable standards and regulations, with the highest level of quality possible, and in the most cost-effective and timely manner possible.

To learn more about how we can help your organization, [visit my web site](#) or [send me an email](#).

Food for Thought, Predictable Software Development, Act Like a Customer, and ALAC are trademarks of Software Quality Consulting, Inc.

Copyright © 2012, Software Quality Consulting, Inc. All rights reserved.

Graphic design by [Sarah Cole Design](#)