



## Leveraging Support Data to Drive Product Improvements

By Françoise Tourniaire, Founder and Owner, FT Works

You may remember that old cartoon of the assembly line in which a hammer methodically smashes all the pots going through, only to have them painstakingly reassembled a few feet away by a poor soul with a lab coat labeled “Support”. Do you feel you’re wearing that coat? If so, it’s time to take action—this article shows you how to prevent support cases (and customers’ broken hopes) rather than solving problems after the fact.

### Get involved in pre-release activities

I regularly hear from my clients that they simply don’t have the bandwidth to get involved in product planning since they are so busy working with customers to fix issues with the current products. I sympathize. At the same time, the Support team can exert a wonderful influence and truly serve as the Voice of the Customer by getting involved in product planning, especially in areas that are close to the customer’s experience.

Help define minimum quality levels for product release *before* it gets to crunch time and there’s all kind of pressure to ship today, no matter what. While there’s no such thing as a bug-free product, many vendors have strict rules against releasing any product that contains a severe known defect, for instance.

Get involved in defining the tools customers require to be successful with the products, including documentation, implementation assistance, training, etc. The idea here is not to be the eternal stumbling block (in which case you will find yourself evicted from future efforts) but to remind everyone that customers need to be successful if the product itself is to be successful.

### Mine the support treasure

After the product is released you can analyze incoming support cases to detect trends in customer requests and to devise remedies to address the issues before they create yet more support requests (stop that

#### in this issue

- » [The First 100 Days: Observations from SSPA’s New Executive Director](#)
- » [Leveraging the 360° View of the Customer](#)
- » [What Your Knowledge Does Not Know](#)
- » [Support Data to Drive Product Improvements](#)
- » [Cost-Saving, Customer-Satisfying Web Self-Service](#)
- » [Creating Leadership in Every Cubicle](#)

hammer!). The remedies could include changes in the product, in the documentation, or in services other than support, as well as the always-popular additions to the knowledge base.

## 1. Properly log all cases

It may seem obvious, but if support staff routinely fails to record certain requests then the issues will be essentially invisible and it will be very hard to prevent them. This is common with “easy” requests, which are the most likely to be ignored at logging time. While easy requests are easy to resolve, they do consume staff time as well as create a burden on customers. No issue is too trivial to fix if it affects lots of customers (and look at it from the bright side: if it’s an easy request, it probably has an easy solution!)

The main reason why issues are not logged is that it takes too much effort to log them, especially for easy requests with quick answers. Scrutinize and streamline the logging process to ensure all issues get logged.

## 2. Adopt or improve resolution categories

While a proper root cause analysis requires skilled support staff (and we’ll come back to this point), it all starts with a solid set of resolution categories. We’re talking here about *post facto* categories, often called resolution or root cause categories, not about the routing categories that are used to classify requests when they are made. So where routing categories typically include the product name and the type of question being posed (say, ProductX/installation or ProductX/tuning) the root cause or resolution categories target the reason for the request, and typically cannot be determined until the case has been worked on for a while.

Resolution categories should, at a minimum, distinguish between defects (something’s wrong with the product) and how-to questions. A more inclusive list would read something like this:

- Product defect
- Documentation defect
- Enhancement request
- Configuration
- Third party product
- Customer education or some other euphemism for user error

Of course, you can add more but don’t go overboard: experience shows that support staffers who have to pick among more than 10 choices or more than 2 levels of categories routinely pick the top choice because they can’t be bothered to read through all the choices. More is not better.

So categories of the type Hardware failure/hard drive are probably fine a three-level category scheme such as Hardware failure/Hard disk/XYZ type is probably too much, especially if there are 15 different kinds of hard drives.

### 3. Leverage links for finer categorization

Especially in large support centers the combination of the resolution categories and the routing categories may not be enough to determine the cause of cases. If you have 3000 cases caused by bugs, it would be nice to be able to tell which specific bugs affect large numbers of customers. Most support organizations have a way to link cases and bugs so that should not be too hard. (If you don't link cases and bugs today, start now: simply log the bug number with the case record for a quick and easy workaround to a full-blown link.)

But what if you have 3000 cases that relate to Customer Education on Product X/tuning? Clearly it's a signal that tuning is important, difficult, or both but you probably don't want to wade through 3000 cases to find out. However, if you have knowledge base documents that discuss various areas of the tuning process and if cases are linked to the documents used to resolve them, you can simply follow the links and quickly determine what's going on. If you find 500 links to the *Basic Tuning Manual* from the documentation set, perhaps customers are just lazy (or cannot find the manual). If you find 500 links to *Secret Tuning Tips*, a document that's not for customer's consumption, perhaps you should make the document available to customers in self-service.

### 4. Perform regular root cause analyses

Here comes human ingenuity. You can run all the reports from categories and links, but it often takes someone knowledgeable with the products to make the determination of what's really important. Many support organizations simply survey the senior staff on what's important this week. This is a reasonable strategy for smaller organizations – as well as a good cop-out for larger organizations that don't have great tracking systems, great tracking discipline, or both.

Root cause analysis ferrets out the most important causes of cases as well as emerging trends. It answers questions such as:

- What are the 10 bugs causing us the most headaches?
- What are the top installation issues?
- Where do customers get lost in the documentation?
- What causes the cases that require level 2 intervention?

Although it would be wonderful to conduct full root cause analyses of all cases on a regular basis, few organizations have the capacity to do that.

Instead, they target a few specific areas that seem to be most troublesome. By all means perform regular sanity checks for general root cause analysis (such as whether the percentage of cases related to bugs is stable) but focus detailed analyses on products that are found to exceed the simple limits through the sanity check.

## 5. Create ROIs for top issues

No need to get fancy there. If your average case costs \$50 to resolve and you got 1000 cases last week on a particular defect, that defect is costing you \$50,000 a week in support costs alone, not counting customer aggravation and threatened lawsuits. On a less dramatic scale, if you get 10 cases (and spend \$500) answering questions about how to install the product on a Linux box, perhaps that's worth a new knowledge base document.

It's fine to use more detailed cost information if you have it. For instance, if cases escalated to level 2 cost \$100 rather than \$50 and all bugs require level 2 intervention, use the \$100 figure to cost bug-related cases.

## 6. Work with Engineering or Marketing to implement solutions

Many support organizations have standing meetings with the Engineering team to review bugs. The meetings are often dominated by the crisis of the moment (severe defects causing severe customer escalations) but there's no reason why you can't sneak in a few less severe issues that cause smaller-scale problems to large numbers of customers. The cost estimates you create should help you create eloquent justifications for why it's a good idea to attack less severe issues.

## 6. Keep at it

Like many support activities, root cause analyses must take place regularly to be truly effective. Don't get lulled into comfort if the case load drops – or panic if it spikes. Keep watching out for that hammer!

## About Francoise Tourniaire.....

Francoise Tourniaire is the founder and owner of FT Works, a consultancy firm that helps technology companies create and improve their support operations. She has helped many organizations improve their proactive support strategies. You can find more information at [www.ftworks.com](http://www.ftworks.com) or call 650 559 9826.

Comments? Suggestions? We would like to hear from you. Please email the editor at [sspanews@thesspa.com](mailto:sspanews@thesspa.com).

**Download PDF**



[next article>>>](#)

Distributed by SSPA - 11031 Via Frontera - Suite A - San Diego CA - 92127  
©2007 SSPA