Doccumentation

Ticketing Systems and Documenting Your Work

Have you ever worked hard on something that had a lot of steps and took a long time only to have to do it again th
IT world. That's why it's important to document the work you do. Documentation might seem like a time suck but it
documenting an issue. Bugs are issues with the system that weren't caused by an external source. Imagine if ever
at all. The IT industry utilizes systems just to keep track of this for you. Some examples are Bugzilla, JIRA and Re
and provide updates. A great way to use the system for documentation is to update the ticket with what the issue i
for two reasons. The first is that it keeps the user in the loop. The second is that it helps you audit your steps in ca
create a documentation trail. You have a lot of options of where you want to write and store your documentation. Y
lots of other mediums. Just make sure it's accessible to everyone else in your company. If you have a monthly rec
refer back to them when it's needed. Documentation isn't a set it and forget it situation. Systems and processes ar
that you aren't reading something that's old. One last thing I want to call out about writing documentation is that yo
technical document. You want to be as concise as possible so that when someone reads your document, they car
(Required)
en
?

Process Documentation

Let's take a look at examples of good and not-so-good documentation. Here's the deal; you encounter a strange issue when helping a user out. This issue happens so often that you and your colleagues have encountered it. No documentation is the worst documentation. Imagine if it took you hours to figure out an issue to a problem and you didn't write it down. Your colleague encounters the same issue and takes hours to figure it out, then he also doesn't write it down. This can go on and on. It only takes a little bit of effort to create documentation and it can save you so much of your time, your company's time, and your users time.

This isn't the best example of documentation. The problem that IT support specialist stated isn't specific and it leaves you with more questions than answers, and while it tells you what we'll fix an issue, it doesn't tell you how. Documentation should be straight and clear cut. Your reader shouldn't have any questions when following the instructions you listed. Now, this is a good example of documentation.

It starts off with a very specific and clear problem. It gives you background information on what the issue is. It even gives you the exact instructions on how to fix the issue, including which settings to navigate to and where. Remember, always write documentation that makes it easy for your reader to follow.

Documenting in Ticketing Systems

Now that we've talked a little bit about documenting processes, let's talk about how you write documentation in ticketing or bug systems. You don't have to leave a full example of process documentation for every ticket you handle. If you encounter the same issue, just write the documentation once, then refer back to it. One of the more important aspects of writing documentation in a ticket or bug, is that you leave an audit trail to see what worked and what didn't. Let's take a look at some examples of awesome documentation and not so awesome documentation in ticketing and bug systems.

This isn't helpful at all since we don't know what the issue was or what the IT support specialist did to fix it. If someone stumbled upon this ticket with the same issue, it would be pretty useless.

This is an example of a great ticket documentation.

The tech described what the issue is, what caused the issue, and the specific steps they took to resolve it.