

## 2.6.2 Troubleshooting Process Facts

Good troubleshooting is a process that combines knowledge, experience, and intuition. As you practice service and support in a work environment, you will add to your experience and develop intuition that will help you to quickly solve a variety of problems. This lesson covers the following topics:

- Troubleshooting steps
- Troubleshooting tips

### Troubleshooting Steps

Regardless of your current troubleshooting abilities, you will benefit from following a systematic approach to problem solving. The following process has proven effective in a variety of situations:

Step	Description
1. Identify the problem	<p>When identifying the problem, resist the urge to start fixing things at this point. To identify the problem:</p> <ul style="list-style-type: none"><li>▪ Ask the user to describe the problem, check for error messages, or recreate the problem.</li><li>▪ Establish what has changed. Most often, problems are caused by new hardware, software, or changes to the configuration. If necessary, carefully ask users to discover what might have changed that could have caused the problem. Remember to inquire about environmental and infrastructure changes that could be relevant.</li><li>▪ Review system and application logs.</li><li>▪ Check knowledge base articles, network topology diagrams, and any other documents that may provide helpful information.</li></ul>
2. Back up the system	<p>Before making changes to the system, back up user and system data (or make sure a recent backup exists). While some changes can be made without affecting user data, you should back up data to protect against unintentional data loss caused by making changes.</p>
3. Identify possible causes and identify a theory of probable cause	<p>Check for simple, obvious, and common problems first. For example, check power cords, connectors, and common user errors.</p>
4. Test your theory	<p>Test your theory to verify the cause of the problem.</p> <ul style="list-style-type: none"><li>▪ If your theory is not correct, examine other possible causes (return to the previous step).</li><li>▪ At this point, if the problem is caused by simple things like an unplugged system, you can safely take actions to resolve the problem.</li><li>▪ If the cause is not a simple one, identify the necessary steps to correct the problem.</li><li>▪ If you cannot identify the cause of the problem, or if the problem is beyond your ability or responsibility to fix, escalate the problem.</li></ul> <p>Escalation means turning the problem over to someone more capable of handling the problem. When escalating the problem, be sure to detail the actions you took and the information you have discovered up to this point.</p>

5. Create an action plan	<p>To create an action plan, address the most likely problem and account for side effects of the proposed plan. For example,</p> <ul style="list-style-type: none"> <li>Will the fix result in significant system downtime?</li> <li>Is the resolution best left for other times of the day?</li> <li>Is there a temporary solution that should be implemented immediately?</li> </ul> <p>When side effects have been weighed against the fix and all concerns have been addressed, fix the problem.</p>
6. Test the solution	<p>When you are testing your solution, do the following:</p> <ul style="list-style-type: none"> <li>Ensure that the problem is fully resolved and that implementation did not cause any new problems.</li> <li>If necessary, take additional actions to prevent the problem from happening again.</li> </ul>
7. Ensure satisfaction	<p>After the problem is fixed, ensure the customer's satisfaction and explain what you did to fix the problem. If possible, have the user perform the task to make sure that they understand and accept that the problem has been resolved.</p>
8. Document the solution and process	<p>Documenting the solution and process provides you with a record of what the problem was and what you did to solve the problem. In the future, you can check your documentation to see what has changed or to help you remember the solution to common problems. It will save you time and money when troubleshooting problems.</p>

Remember that troubleshooting is a process of both deduction and induction. Experience will tell you when deviating from this process can save both time and effort.

## Troubleshooting Tips

Keep in mind the following tips when troubleshooting systems:

- Often the hardest part of troubleshooting is to reproduce the problem. You might need to ask the user questions to identify exactly how the problem occurred, or you might need to watch them perform the task again to reproduce the problem.
- If a hardware device or a software program causes a specific error, check the manufacturer's website for additional help in troubleshooting the error.
- To help diagnose issues, you can run special software tools supplied by the hardware manufacturer.
- In addition to a basic toolkit, you can keep a few spare parts on hand that you know to be in working order. If you suspect that a component has failed, replace it with the known good spare. If that solves the problem, replace the faulty component.
- Intermittent problems are particularly difficult to troubleshoot. Check for environmental conditions such as kinked cables or overheated components.
- If you have problems identifying a hardware error, you can simplify the system by removing all but necessary components (processor, memory, and hard disk). Add devices one at a time and restart the system. If an error occurs, remove the newly added device and troubleshoot that device. Another strategy would be to remove a single device and restart the system, seeing if removing that device corrects the problem.

- Some problems might be caused by software errors, not hardware failures. You might need to begin by updating the drivers or unloading software.
- Before you make changes, always consider corporate policies and procedures and the changes' impact on other people and components.

---

**Copyright © 2021 TestOut Corporation All rights reserved.**