Topic 3A: Apply Troubleshooting Methodology:

Having troubleshooting skills is very important for an IT technician. To troubleshoot effectively, you need to know how a system is supposed to work and what common problems might occur. However, complex systems don't provide straightforward answers, so that we must develop general troubleshooting skills to handle unexpected issues confidently.

Key Points of Troubleshooting:

When there is some problem happened on a computer or any other things, there are mainly three things behind it:

a. **Cause:** The reason why something went wrong (e.g. a **hard drive** is faulty)
b. **Symptoms:** The things that are happening because of that cause (e.g. because of **faulty hard drive** your computer shows a "blue screen").
c. **Consequence or Effect:** It is the result of the problem. (e.g. user cannot work)

In the business environment, it is necessary to solve the **Consequence or result** rather than solving the **root cause** at first. For example, if a hard drive is faulty, you should give the user a new computer to continue working while fixing the issues later because in business user's work has more priority and severity right. If you don't offer user a new computer, he should stay silent without doing any work which can impact the business. So, **an IT technician** should work in terms of **priority and severity**.

For example, If there is two problem in the company like **power failure** and **hard drive fault** in the company. Then in this case, obviously the power failure is urgent or critical right because it can affect the entire computer in the company. So, based on the urgency level, **power failure** has more **priority** than the **hard drive failure**. So an IT technician should have knowledge to prioritize tasks.

And another thing is like if the same problem is happening again and again, it might point to a larger issue. For example,

- Why is the hard drive faulty?
- Is it a one-time failure, or is there a problem with the environment, supply chain, or equipment quality?

In simple terms, troubleshooting is not just about fixing what's broken. It's about understanding the problem, managing its impact, and preventing it from happening again.

CompTIA A+ Troubleshoot Model:

Identify the Problem

- **Gather Information**: Ask the user about the issue, recent changes, and other relevant details.
- **Check for Changes**: Look for changes in the environment or system (e.g., software updates or hardware changes).
- **Backups**: If necessary, back up important data before making changes.

---

2. Establish a Theory of Probable Cause

- Start with the obvious reasons for the issue.
- Research externally (e.g., online resources) or internally (e.g., system logs) based on the symptoms.

---

3. Test the Theory to Determine the Cause

- **Confirm the Theory**: If the theory explains the issue, decide the next steps to fix it.
- **If Not Confirmed**: Develop a new theory or escalate the issue to someone more experienced.

---

4. Establish a Plan of Action and Implement the Solution

- Create a clear plan to resolve the problem.
- Follow vendor instructions or guidelines when applying the fix.

---

5. Verify Full Functionality and Implement Preventive Measures

- Ensure the system works correctly after the fix.
- If possible, take steps to prevent the issue from happening again.

---

6. Document Everything

- Record details about the problem, the solution, and the results.
- Proper documentation helps in solving similar issues in the future.

Further Discussing Every Steps:

Identify the Problem:

The troubleshooting process begins with identifying the problem. Identify the problem means finding the **root cause** of the problem, establishing the possible symptoms that could happen due to that root cause, and determining possible impact that can arise from this situation. When you determine the **consequences,** it helps you to prioritize the tasks as well.

a. **Gather Information from the User:**

   It is very important to gather each and every details form the user regarding the problem because they are the most important source of information for you to handle that issues. So, you can ask users many questions regarding this like:

   - When was the problem exactly happened?
   - What are the exact error messages appearing on the screen when it happened?
   - Is anyone else facing the same problem?
   - How long has the problem been occurring?
   - What was the recent changes you made with the system?
   - Has anyone tried to solve the problem before?

b. **Perform Backups:**

   After gathering the information, assess whether the data stored in the computer is important or not. And do **perform backups** if it seems necessary.

Establish and Test the Theory:

When troubleshooting a problem, it is essential to ask accurate and relevant questions. These questions helps you to determine the **severity of the problem** like how many systems or individuals are affected. They can also provide the clues that where we need to focus to solve this problem like software or hardware, OS, or other components. Hence, to troubleshoot the issues, we need to diagnose the problem which involves identifying **symptoms** and **possible causes.** For example, if a television fails to display anything when turned on, you might consider whether the television is faulty, if the fuse has blown, or if the broadcasting station is experiencing issues. Testing these possibilities systematically helps narrow down the actual problem.

- **Conduct Research**

  You cannot always rely on users to describe the problem accurately. That's why you need to do research as well to clarify symptoms and possible causes. One of the most useful troubleshooting skills is being able to perform research to find information quickly. Research involves anything that can be useful to solve the problem like you could consult with someone who has solved the same problem before as well.

- **Question the Obvious**
  This means when you find several causes that can be the reason of the problem, don't **overlook** any of them. Means don't ignore any of those causes. And try to isolate from those causes. If you can isolate your investigation to a particular subsystem by eliminating "non-causes", you can troubleshoot the problem more quickly.
    - Narrow down the problem is related to hardware or software.
    - Decide which hardware or software's component is faulty.
    - Test your theory.

Establish a new theory and Escalate:

If your initial theory does not work, establish a new theory. If the new theory is lengthy, then asses the business needs before. And if it seems that this lengthy process can affect the business process, look for shorthand way if possible. And if you cannot solve the problem just **escalate it**. Means, you can give the issue to like more experienced staffs, senior technical staffs, subject matter experts, warranty suppliers or manufacturers to repair it, etc.

If you want to escalate that problem with other senior staffs, consider before that they also have a lot problems to deal with. So before giving it to the senior staff, make sure you document each and every detail that you performed so that the senior staff does not need to repeat the same process again, which will make his job easier.

Implement a Plan of Action:

When you have like close theory that this could have happened because of this issue, then determine possible steps to solve the problem. When implementing your solution, you need to require effective decision-making so that it can be solved more efficiently. Sometimes, there may be several solutions that you think is correct to solve the problem. In this case, you need to have efficient decision-making skills to solve that process.

There are three ways or approaches to solve an IT problem:

- **Repair:** You need to make sure that the repair cost how much and is this the best option to solve?
- **Replace:** Replacing is more expensive and may be time-consuming if the parts is not available.
- **Workaround:** If neither repair nor replacement is cost-effective, then it may be best to find a workaround or just document the issue and move on.

If a part or system is under warranty, you can return the broken part for a replacement. To do this, you normally need to obtain a Return Materials Authorization (RMA) tickets from the vendor.

- **Establish a plan of Action:**
  When you determine the best solution, you must assess the cost, resources, and time it takes to solve it. And also, you should assess is there any side-effects of that solution or not. For example, if you install a software patch to solve the issue, that software patch may affect other applications. Whatever you do, it should be under the **corporate policies and procedures.**

- **Implement the Solution:**
  If you don't have authority to implement your solution, you should **escalate** it to senior management. If applying that solution affects the entire network of the company, you should choose the most suitable time to perform that. And when you implement a solution, **test** the changes at every part. If the change does not fix the problem, then reverse it and try something else. If you don't document your every changes, then it might take you to harder situations.

- **Refer to Vendor Instructions:**
  If you are completing troubleshooting steps under instruction from another technician—the vendor's support service, for instance—make sure you properly understand the steps you are being asked to take, especially if it requires disassembly of a component or reconfiguration of software that you are not familiar with.

Verify and Document:

After solving the problem, test and verify it with the user as well. And after solving the problem, restate or document what the problem was and how it was resolved, and then confirm with the customer that the incident log can be closed.

- **Implement Preventive Measures:**
  To fully solve the problem, you need to prevent it from happening again as well. For example, if the power cable on a PC blows a fuse, you should not only replace the fuse, but also check to see if there are any power problems in the building that may have caused the fuse to blow in the first place. If a computer is infected with a virus, ensure that the antivirus software is updating itself regularly and users are trained to avoid malware risks.

- **Document Findings, Actions, and Outcomes:**
  Most troubleshooting takes place within the context of a ticket system. This shows who is responsible for any particular problem and what its status is. This is very useful for future troubleshooting, as the same problem can occur in the future as well. Also, logs may be presented to customers as proof of troubleshooting activity. Write clearly and concisely, checking for spelling and grammar errors.

Some Questions and Answers:

What should you do after identifying the probable cause of a problem?
Test the theory to confirm if it is the actual cause; if confirmed, resolve the issue, or develop a new theory or escalate if it is not.

What should you check before opening the system case to troubleshoot a computer?
Ensure all important data on the computer is backed up to prevent any potential data loss during troubleshooting.

What should you do if you cannot determine the cause of a problem?
Consult a colleague, refer to documentation, search online, or escalate the issue to senior support staff for further investigation.

What should you do if the solution involves installing a software patch?
Identify potential risks, plan the implementation to minimize disruption, and prepare a rollback plan in case the patch fails.

What should you do after applying a troubleshooting repair, replacement, or upgrade?
Test the system to confirm the fix works and implement preventative measures to avoid the issue in the future.