The Right Questions, A Universal Troubleshooting Guide (v8) by Jason Maxham (http://artoftroubleshooting.com/)

**Troubleshooting Basics:**
- Have ALL the prerequisites for operation been satisfied? For example, is it plugged in?
- Is the problem clearly defined?
- Can the problem be reproduced?
- What makes the problem worse? Better?
- What's changed or new?
- Have I done a full inspection (eg, a walk-around)?
- Is the system operating beyond its known limits?
- Can I perform routine (or neglected) maintenance?
- Can I reduce complexity by: 1) restoring defaults, 2) restarting/power cycling, or 3) turning off unneeded features or subsystems?
- Has someone else already solved this problem?
- Do I have the right tools?
- Should I document my work? Notes, pictures, etc.
- How is it supposed to work? What is normal operation?
- Does the machine know what's wrong? Are there error messages, diagnostics, or logs I can examine?
- Is troubleshooting the best use of my resources?
- Is there a workaround that's better? Can I swap or replace?

**Before I Make a Repair I Ask:**
- Will this repair cause downtime? Who is affected and needs to be notified?
- How long will this repair take? What happens if it's not finished on time (or ever)?
- What are the risks of this repair? Can it be reversed and what are the steps to get back to where I started?

**More Strategy Questions:**
- Have I kept presuppositions about causes out of the problem description? What are the facts?
- Can I change the order of the startup or workflow?
- Is everyone who might know the answer aware of the issue?
- Should I clarify or add detail to problem reports?
- Can I come back to this later, or work on a different aspect of the problem?
- Can I follow the flow, from beginning to end, to find the problem?
- Is the system a Black Box? Can it be opened up so I can examine its inner workings?
- What other types of failures could produce these same symptoms?
- Are environmental conditions (noise, temperature, weather, etc.) impeding my ability to work?
- Can I copy one that works?
- Have I made a logical leap that isn't justified? Have I chosen the simplest explanation possible?
- Can I deploy dedicated resources, limiters, or governors to lessen negative interactions between components and bring usage in line with capacity?
- What's the extent of the problem? Are symptoms repeated across systems? Conversely, what's NOT affected?
- How can I narrow down the problem space? Can I use half-splitting (aka, binary search)?
- Can I get another perspective on the problem? Can I troubleshoot with a partner? A team?
- Is there a bottleneck? If so, where?
- Is this my problem to solve? Is a co-worker, business partner, manufacturer, or vendor avoiding responsibility?
- Am I in over my head? Should I call in someone more experienced, like a professional, to help?

**Cleaning Up:**
- How do I know that I've fixed the problem?
- Should I add redundancy or capacity?
- Can I collect data to better understand the problem and detect it in the future?
- Would a routine maintenance program prevent recurrence? If already in place, can I perform maintenance more often (or better)?
- Is this problem the "tip of the iceberg?" Does it foreshadow something worse?
- Can I analyze the problem using Root Cause Analysis (like 5 Whys)?
- Will I use this situation to make changes that would have been difficult before?
- Was the failure intentional (sabotage, fraud, etc.)?
- Is it possible to devise a detector that automatically alerts me to this type of failure?
- Could the problem be avoided with stress testing or a break-in period?
- When will I communicate what was learned so that others can benefit? Can I create documentation like an incident report, service bulletin, or troubleshooting tree?
- Can I prevent recurrence with a checklist?