Concise Guide to Debugging Anything (1)

Your System:

1. Understand the System (Read the code, documentation or ask)

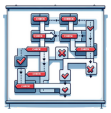
- Thoroughly read the system's documentation and consult with experts to understand its intended functionality.
- Carefully step through the code, understanding what each part is supposed to do. When in doubt, don't guess – refer back to the documentation or ask for clarification.

2. Replicate the Failure (Observe and record the bug occurring again)

- Replicate the failure under the exact conditions it occurred, not just similar ones.
- For intermittent failures, vary the conditions until the issue can be can consistently replicated. Document every detail, including any seemingly irrelevant ones.

3. Search and Identify the Bug (Don’t theorize without evidence)

- Engage in a thorough search to identify the exact cause of the bug. Rely on debug logging to make the bug and its cause visible.
- Avoid theorizing about potential causes without evidence - do a practical search.
- Remember the debug process might inadvertently modify the conditions and hide the failure.



4. Narrow the Search (Divide & Conquer or Successive Approximation)

- Apply the divide & conquer algorithm to narrow down the search area. Understand the range of the search and determine if the bug is upstream or downstream from the current point.
- Ideally, start at a known problematic point and work back up through the system, checking each branch until finding the source of the bug.

5. Change One Thing at a Time (Control all other variables)

- When attempting to replicate the failure or identify the bug, change only one variable at a time, changing back any variables to the original condition before the next test.
- Keep a forensic mindset, analyzing what has changed since the last time the system worked correctly.

Based on Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems by David Agans; Examples and guide by andypi.co.uk

Concise Guide to Debugging Anything (2)

6. Keep an Audit Trail (Write down details of the debugging process)

- Keep a detailed audit trail of all actions taken, the order in which they were done, and the results of each action.
- This record is invaluable in ensuring that all areas have been checked, in providing a clear account of your debugging process to others, and serving as a reference for future issues

7. Check Obvious Assumptions (that are fast to verify)

- Always verify the most basic assumptions first, such as whether the system is powered on, the service is running or expired data is cached.
- Before narrowing your search, confirm that the entire scope of the system is being checked, including all tools, dependencies and platforms you're working with.

8. Ask for Help (from online resources and then experts)

- When you've hit a wall, a fresh perspective from others can shed new light on the problem.
- Start with async online resources like StackOverflow, ChatGPT before asking experts (e.g. Github issues) which introduces delay
- Always provide a bug description, logs, errors and what has been checked so far, etc.

9. Confirm the Fix Works (By testing with it applied and removed)

- Rigorously test the fix to confirm it addresses the issue. Then, remove the fix and retest to ensure that the issue reoccurs, confirming that your fix is directly resolving the problem.
- For particularly elusive bugs, add logging to capture details of the failure so it can be traced if it occurs again in production.

10. Fix the Underlying Process (Find the design or systemic problem)

- Reflect on the debugging process and identify any systemic improvements that could prevent similar issues in the future.
- For example, standardizing error and debug logs, enhancing test coverage, automating testing and deployment to minimize human error, and thoroughly understanding dependencies during the design phase.

Based on Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems by David Agans; Examples and guide by andypi.co.uk