

Shifting the Burden—Whose Monkey Is It?

by Don Gray

“Repeatedly curing a system that can cure itself will eventually create a system that can't.”

—Marvin's Second Great Secret, Jerry Weinberg

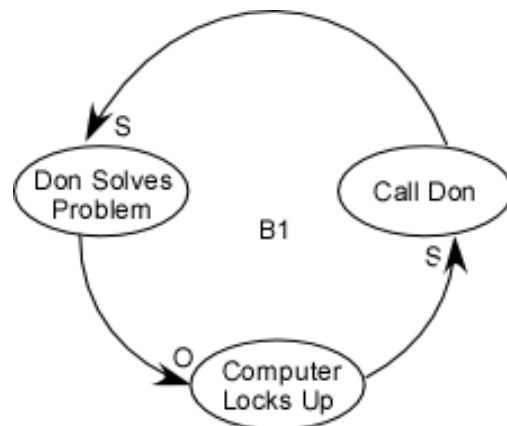
“Don, the software's locked up again! Can you come up here tomorrow and fix it?” George was on the other end of the conversation. George and I had started working together when his employer moved a production line from Florida to Virginia. This move created all sorts of problems¹.

The daily struggles getting the hardware, software, and process playing nicely together had become a weekly check-in with an occasional on-site visit. This made the request seem a little odd.

Taking a deep breath allowed me to work through some quick thoughts². The software was running on 15 different computers. There wasn't any way they could all lock up at the same time. And what does “lock up” mean anyway? What's going to happen different between now and tomorrow, or now and next Monday? And if I drop what I'm currently working on to rush up there tomorrow, am I really helping George? What long-term consequences result from this?

Solution #1 - The Quick Fix

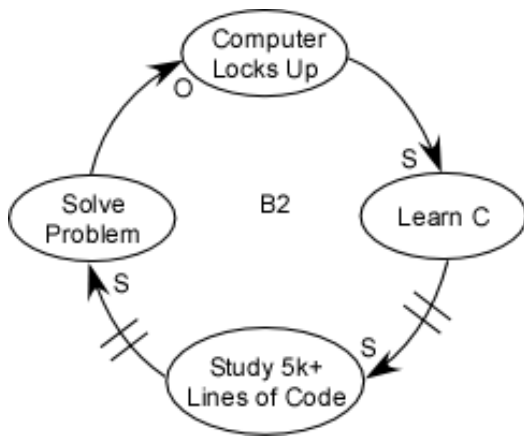
Accepting that a problem is an undesirable difference between what we want and what we have, George had a problem. Since I had helped George with several other problems, it seemed natural to him that I should help with this problem also. Using a causal loop diagram, the situation looks like this (see my article “The Diagram of Effects” for more on causal loop diagrams):



The more the computer locks up, the more Don gets called. The more Don gets called, the more problems he solves. The more problems Don solves, the less the computer locks up. This is a balancing loop that brings stability to George's process. It's also a symptomatic solution, a quick fix to make the pain go away. But following the steps in diagram B1 only solves the immediate problem; it does not solve the underlying fundamental problem. Think of it as treating the symptom, not the disease.

Solution #2 - George Learns to Solve the Problem

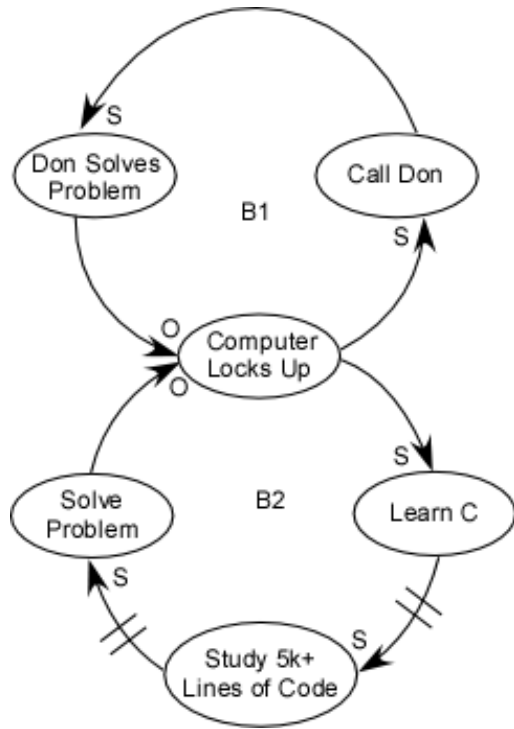
There's another possible answer. What does it look like if George solves the problem without calling Don? First George needs to learn "C" much better than he currently knows it. That takes time. This time delay gets represented by the "||" mark on the line in B2 below. Then he needs to figure out where in the 5000+ lines of code the problem resides. That also takes time. Finally, George can solve the problem, but by now something else is broken worse somewhere else, everything gets put on hold, and when George finally gets back, he's lost his train of thought, and has to start solving the problem again. Diagrammatically we represent it this way:



This is also a balancing loop bringing stability to the system. B2's advantage over B1 is B2 represents a systemic solution. The more the loop executes, the better the system gets at solving the problems. This reduces the system's dependence on outside interventions. The disadvantage of B2 is the time delays involved in learning "C" and understanding the application code. The time delays do get less the more the loop executes.

Solution #3 - Combining the Symptomatic and Systemic Solutions

Since B1 and B2 both contain "Computer Locks Up" we can combine the two causal loop diagrams into a more complete problem-solving picture:



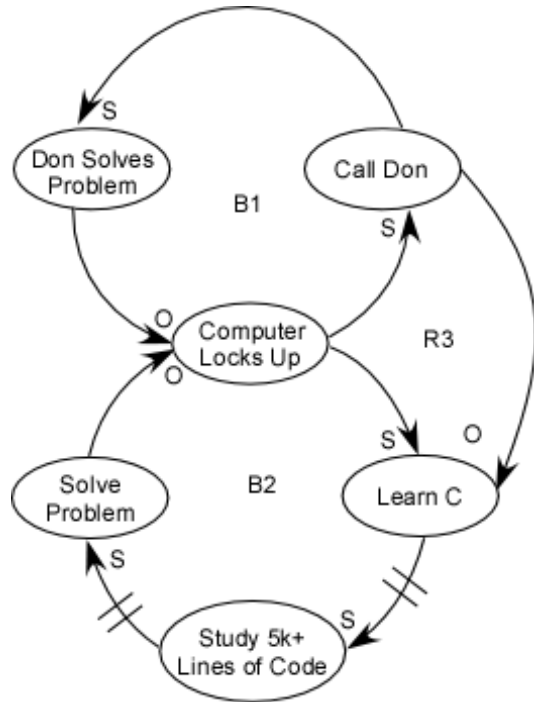
This diagram shows the how the event “Computer Locks Up” can trigger one of two responses: a symptomatic response to make the problem go away quickly (B1), or a systemic response (B2) where the system becomes more capable of solving problems without external influence. This pattern occurs often enough that systems people have given it a name: “Shifting the Burden.” I'd been aware of this archetype, but preparing for my “Déjà vu” session³ with Diane Gibson for the AYE Conference sharpened my awareness of what could go wrong.

An Unintended Consequence - Too Much Help

There happens to be a long term reinforcing loop that can show up with this archetype. This occurs when the symptomatic solution (B1) gets all the action. Don becomes better at both “C” and knowing the code base. This makes it easier and quicker for Don to solve the problems, and Don becomes indispensable.

This means the system's ability to “cure itself” becomes atrophied. George loses his ability to persuade management to give him training. Production becomes accustomed to “great customer service” from Don. The ability to tolerate “pain” goes away, and the quick fix becomes the only fix.

Invoking the symptomatic solution not only starts B1, it initiates R3. Calling Don reduces the need to learn “C,” thereby weakening the system's problem solving ability. Carried to the extreme, this becomes an addictive response, and cripples the system.



And The Final Answer Is

Either balancing loop can be chosen to a given problem occurrence. Handled properly B1 can be used to alleviate acute problems, while invoking B2 to solve chronic problems. This pattern of selectively, consciously choosing how to deal with each problem has resulted in George's ability to solve more code problems and start making enhancements to the code base.

I appreciate Stuart Scott and Jerry Weinberg for their suggestions about this article.

* * *

References

¹<http://www.ayeconference.com/Articles/Howdidthishappen.html>

²<http://www.ayeconference.com/Articles/Dontdosomething.html>

³<http://www.ayeconference.com/wiki/scribble.cgi?read=SessionFour022>

and

<http://www.ayeconference.com/wiki/scribble.cgi?read=SessionFive005>

For more information about the causal loop diagrams used in this article, see the developer.* article by Don Gray called "The Diagram of Effects" at this address:

http://www.developerdotstar.com/mag/articles/gray_diagram_of_effects.html

Read more articles in the developer.* Systems and Software series at this address:

http://www.developerdotstar.com/mag/categories/systems_software_series.html

About the Author

For a quarter century, Don Gray has worked in applied cybernetics, starting in machine and process automation and migrating to organizational systems and change. His major areas of interest are cognition, modeling, and understanding system change. His work focuses on integrating people, projects, and processes. He is a host for the Amplifying Your Effectiveness (AYE) conference. Visit his website at www.donaldegray.com.