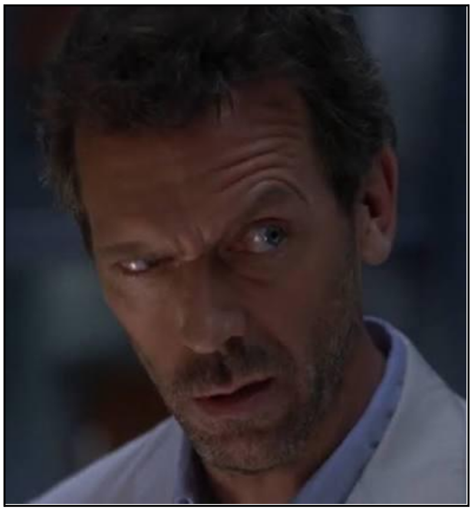
June 23, 2025

Debugging Software Via Differential Diagnosis (“Differential Debugging!”)

“Differential- go!” -Greg House, MD

If you’ve never seen an episode of *House, MD*, you’re missing out on a dramatized and fictionalized (yet also entertaining) display of a hospital setting. The series follows the fictional Dr. Gregory House (played by the fantastic actor Hugh Laurie), a world-leading doctor in the field of Diagnostic Medicine.

He is depicted with a fellowship that spends each episode diagnosing and treating patients suffering from vague and anomalous medical conditions. Invariably, each episode shows multiple sessions where House and his team attempt to identify their patient’s ailments through a process referred to as “differential diagnosis”. It is here where our “story” begins.



What is Differential Diagnosis?

The practice of differential diagnosis goes back decades. I have seen a reference online stating that it formally began “in the late nineteenth century”[1]. It is a method of diagnosing an illness by brainstorming ideas of medical issues that display the same symptoms as those that the patient is presenting. Doctors will then rule-out the issues by identifying what symptoms the patient is not presenting with- and use what is left to narrow down a stable diagnosis.

[1] PubMed | Herbert French (1875-1951) and his differential diagnosis a "work of reference unique in medical literature"

The key term here is the "differential", which shows how ailments are different from each other (in terms of symptoms). Once a thorough differential is performed, physicians can then use the diagnosis achieved from the differential as a basis for administering treatment.

Differential diagnoses can use many sources of information as input for making decisions. These can include documented medical histories, patient interviews, medical intake and discharge reports, test results, and other bio-level data values. *If you are already a programmer by training, you might see where this article is headed and how it relates to the concept of debugging code.*

"Strong Medicine", a physician YouTube channel, has a 3-part video series of how differential diagnoses are performed.

What is Debugging?

Grace Hopper (circa 1940s) - Helped originate the term "debugging"

Debugging is the practice of finding issues (bugs) in a program or codebase and determining a solution to remove the issue from that code. The goal is to achieve an efficient solution to the acute problem. This also usually involves some level of *preventative maintenance* to keep such a bug from recurring in the program. In practice, however, several mitigating factors can sometimes lead programmers to implement a less-than-efficient solution to the problem. Such factors include system downtime, increased pressure due to user and management complaints, technical complexity of the issue, and feasibility of deploying the suggested fix.

Sometimes debugging is a straightforward process, and developers can achieve an efficient solution quickly- sometimes not. It is due to these mitigating factors that debugging tends to be more of an art than a science. In my professional opinion, debugging is where the creativity and acumen of a programmer really shines. The ones that don't build their debugging muscles are usually overshadowed by the ones that do. Don’t be *that* coworker - the one that never internalizes when an issue happens and does not research anything before “reaching for the answer” from their teammates. Being that person is inadvisable.

As it turns out- people tend to like their doctors to follow a similar pattern! “I am hurting ‘doc’, and I don’t know why.” The physicians that people want to see (and pay) can efficiently fix such a problem.

Problems in software can be found, realistically, at any point in the software development lifecycle (SDLC). They can “crop up” during compilations or builds, during runtime, or during deployments. They can be as simple as a user reporting an issue to a support team (“the website isn’t letting me log in”) or as complex as a team of engineers uncovering a logic issue that is affecting many areas of a program (“this SQL error impacts every line item of the billing system since the patch deployment made yesterday on 6/1/2024”). *As you can imagine, learning to debug effectively is a professional superpower.*

Putting the Two Processes Together!

It occurred to me, during rewatching an episode of House, that debugging can be viewed as a technological analog to differential diagnosis in the practice of medicine. What if one took the procedures outlined in diagnostic medicine and applied them to a software problem? *Let’s take a glimpse at what that would look like…*

Gathering Data

I mentioned earlier that a differential diagnosis utilizes data for making informed diagnosis decisions from a variety of decisions. Debugging a system is no different. Oftentimes, the system log is one of the first places to check when you are trying to track down an issue- both during production and development. Application logs are chock-full of helpful information, particularly the timestamp and log message fields that are present in virtually every type of application log.

Your users are another good source of data. If a support team fails to catch an issue in time, the users will often submit an issue report that alerts support teams of an issue. This kind of report sheds light on the acute problem.

It’s Differential-ing Time

“A customer facing web application is ‘just spinning’” - suppose that this is a note that you get, in a support ticket, from a user. On the surface, this is unhelpful and can be somewhat misleading. It’s great if the users can give you more information in their initial report. Some questions that you might want to ask here:

- Is this when you’re trying to reach the site (or when you’re trying to log in)?
- Is the customer using one of the approved/known web browsers?
- Is this an isolated report or have you received multiple tickets from different users?

Even these basic questions can make a big difference in your differential. They can help you identify if it’s a system-side or client-side issue from the beginning. Think of these questions as akin to your doctor asking you: “when did you first start feeling like this?”

For all that is holy in the world, please pay attention to this next paragraph

As you track down the answers to these questions, open up a document and take notes. These notes are incredibly useful for “crossing off” what you have already ruled out vs. what you have yet to try. They are also important for describing what events happened; if a customer requires a root cause analysis (“RCA” - a document that identifies why a problem occurred, what was done to fix it, and what steps have been taken to help prevent the problem in the future) any notes that you take during troubleshooting sessions will quickly morph into your RCA report.

Once you have interrogated the users and the initial trouble tickets, it is prudent to check your “observability infrastructure.” This includes your application log files, your system heartbeats, your security event management logs (Splunk, anyone?), and any real-time dashboards that you may have to manage the application in question. In the context of a clinician- this is the same as your patient telemetry (blood pressure, blood oxygen, heart rate, etc.). These systems give you a high-level, general, idea of where to look for your problem.

This paper is just intended to give you inspiration. Go forth and solve a problem in your system. Sometimes the issue is masked by other things. Differential diagnosis is a tool that might just give you the answer :)

–Eric C. Grasby–