Pessimism, Entitlement, and other virtues of a successful debugging approach:
Story time

Act 1. Disappearing nodes!

Allen's client David, membership director for a charity service organization, emails to say that nodes are disappearing from his site. Two that were there yesterday are now gone, as is one that was definitely there last week. Alarmed, Allen check's the sites recent log entries and finds no node deletions recorded in the past 24 hours. He then compares the current node list to last week's backups, and finds lots of nodes that have been deleted -- but is unsure if any of them are relevant.

Allen asks David if he can remember which nodes were deleted; David names three nodes; each one was an entry for a local chapter of David's organization. Allen checks the list of nodes deleted in the last week -- and none of David's missing nodes are listed there. Allen then checks /admin/content and finds the nodes still exist. Why does David think they're deleted?

Since David's arrival at the organization 9 months ago, he's been using a views-generated page as a calling list to check in on local chapters. It's called "Newest Chapters" even though the organization doesn't often add new chapters. Recently some chapters have been dropping off the list.

Allen explains to David that this view shows only chapters that have been added in the last year, and since these three chapters were added just a little over a year ago, they've just now rolled off the list. Because this organization rarely adds new chapters, David had "gotten away" with using it as a comprehensive chapter list. Allen points him to the "All Chapters" page, and David is happy again.

Act 2. Can't save new content!

Allen's client Sarah launched her new e-commerce site last month; she calls Allen to say that something's wrong because she can't save new content. She's tried and tried and it looks like it should save when she clicks "Save," but then it's just not there. She's quite pressed for time and stressed out, so Allen says he'll look into it and email her when it's fixed.

Allen then tries to reproduce the issue by creating some pages on the site, which works just fine.

He email Sarah and asks for more details. Sarah reports that she's continued trying to save the new page, and is growing more frustrated because she's on a deadline to publish an article about the upcoming promotion. No matter what she tries, her page at first appears to be saving, but then is not to be found in the "Hot Deals" section listing.

Allen calls Sarah and walks through the node creation on the phone. At this point they realize that Sarah has been forgetting to assign the "Hot Deals" taxonomy term to the created node, so although the page is created, it's not listed where Sarah expects to see it. Sarah smacks her forehead and publishes the "Spring specials" article, and Allen heads over to /admin/content and deletes the 47 copies of it that Sarah had created.

Act 3. I can't masquerade

Allen's client Jim uses the Masquerade module when conducting user trainings and support at his company. One day Jim emails Allen to say he can't masquerade as anyone; he needs that functionality, but wants to be sure this fix doesn't derail other development tasks that Allen is already working on for him. Allen spends a couple of hours on the live server slogging through the Masquerade code using vim, to look for a solution, but can't make much headway and eventually turns to the other development tasks, because these still need some work and are due next week.

By the time those are completed, Jim has had to reschedule several training sessions, so he inquires again about the Masquerade fix.

Allen finally turns to Google, and then to the Masquerade issue queue, where he finds a regression bug introduced by a recent security update. Allen recalls that the security update was applied to the live system a couple of weeks ago, so this is likely the issue. Allen notices that the fix for this issue has already been committed and released in the next Masquerade version, which was released just two days after the security release that contains the bug.

Allen runs `drush up masquerade`, and the problem is fixed.

Act 4. Newsletter emails aren't going out!

Allen's client Karen works at a small non-profit organization and contacts Allen for help because her CiviCRM newsletter emails aren't being received by her constituents.

Thinking of all the things that could prevent email delivery, Allen gets to work. He checks the mail server reputation to see if it's blacklisted; he fiddles with the postfix configuration; he restarts postfix; he reboots the server; something has to work, he figures.

Each time he changes something, he tries sending a test mailing, which requires clicking through several configuration screens. It's time-consuming, but he's a hard worker.

Finally he realizes that the CiviCRM cron job is not being executed. Something must be wrong with the crontab entry, so he tries a couple of changes there, but still no luck.

Allen knows CiviCRM cron jobs make use of a specific Drupal user, and after examing crontab, he finds that the username being used for this job belongs to a user who no longer exists on the site -- ah, success!! Allen recreates and re-configures the user account so that the cron job will run properly.

He then emails Karen to tell her that the problem has been found and fixed. Karen is relieved and writes back a note of thanks.

Three days later, Karen writes back to say that the emails still haven't gone out. What's going on, she asks?

Allen checks the crontab entry and finds that in all his changes to the cron command, he had introduced a typo, so that the cron job had still not been running after all.

Allen corrects the typo, says a little prayer, and emails Karen that all is well.