

# Debugging Data and Inputs



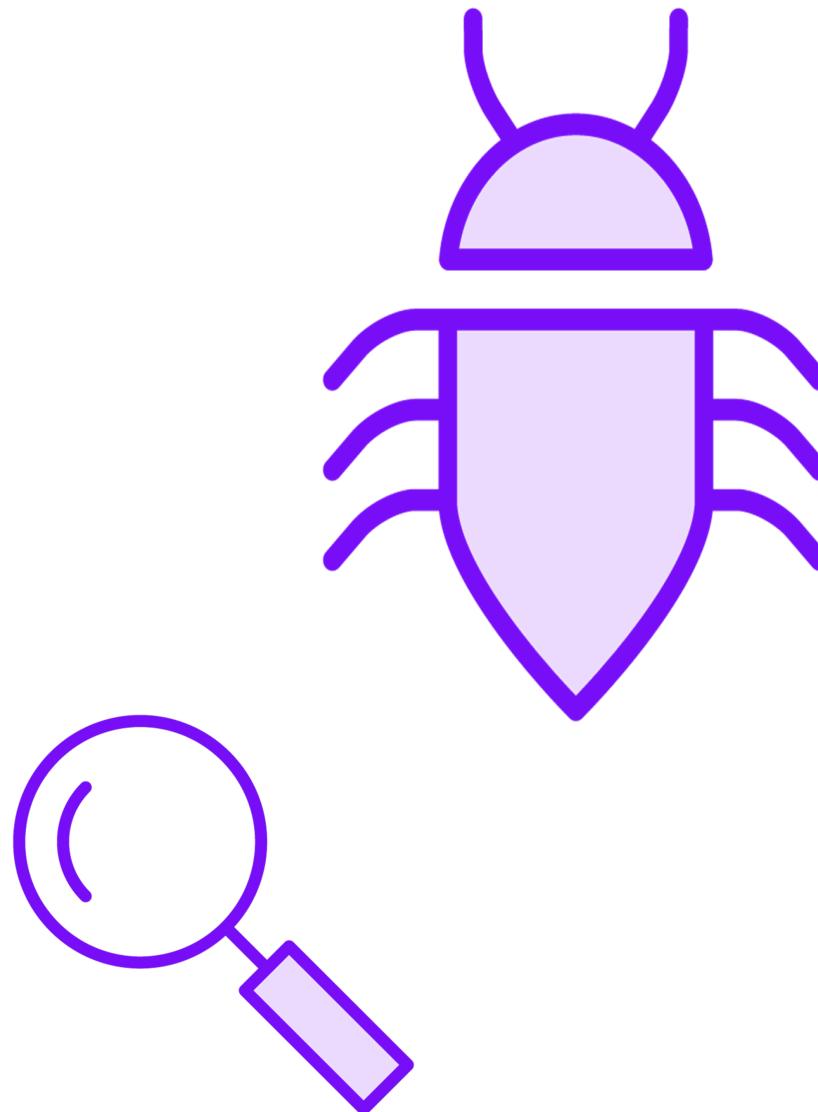
**Neil Morrissey**

Solutions Architect

[linkedin.com/in/neilmorrissey](https://linkedin.com/in/neilmorrissey) | [www.neilmorrissey.net](http://www.neilmorrissey.net)



# Common Debugging Traps

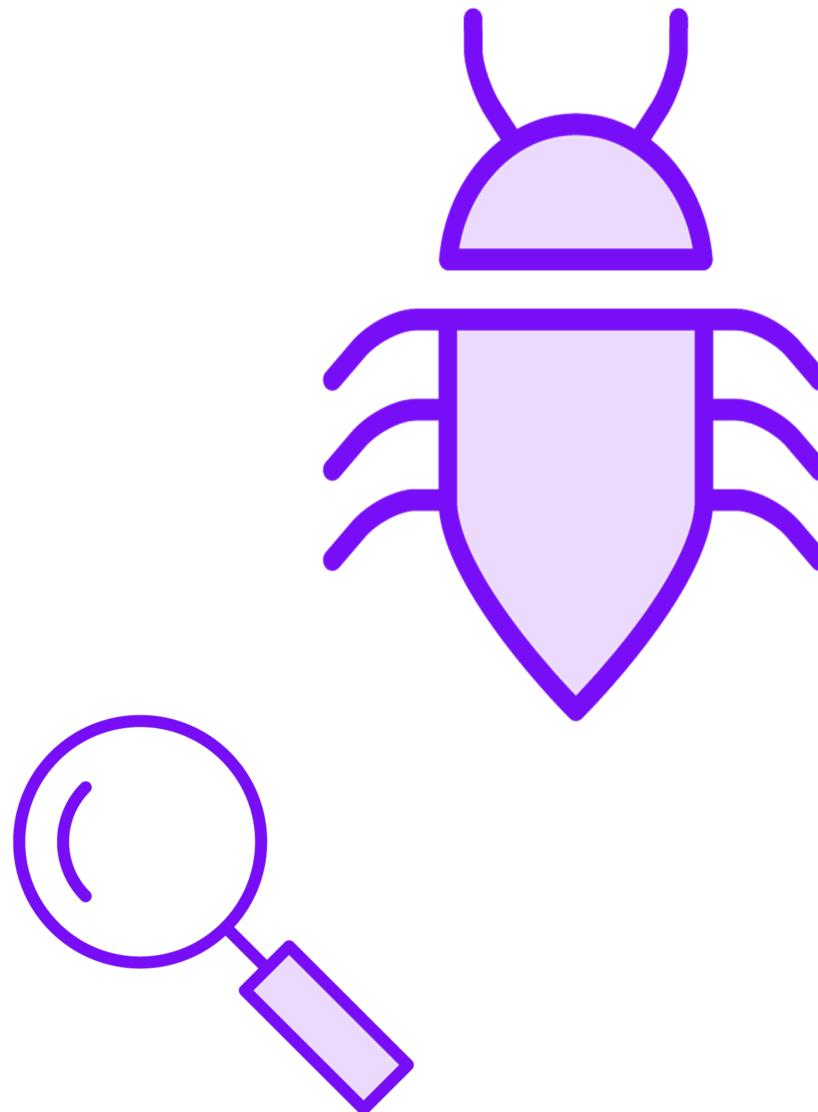


- Don't guess or try random things in code**
- Don't assume the problem is trivial**
- Don't code for 'special cases'**
- Don't debug by superstition**





# Common Debugging Traps



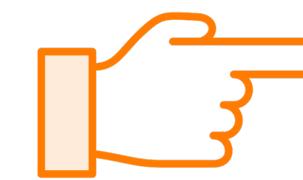
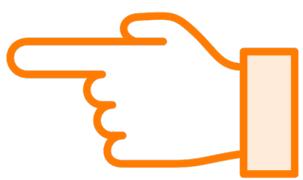
- Don't guess or try random things in code**
- Don't assume the problem is trivial**
- Don't code for 'special cases'**
- Don't debug by superstition**





Poor Kenny









# Module Overview



- Common traps in debugging**
- An effective approach to debugging**
- Testing and stabilizing the error**
- Inspecting data with the debugger**
- Modifying data in the debugger**
- Fixing the defect**
- Visual Studio Code data debugging features**





# An Effective Approach to Debugging



# An Effective Approach to Debugging

## The Scientific Method

Gather data through repeatable experiments

Form a hypothesis that accounts for the relevant data

Design an experiment to prove or disprove the hypothesis

Execute the experiment

Analyze, refine and repeat as needed

## Debugging Code

Stabilize the error

Using the scientific method

Locate the source of the error

Using the scientific method

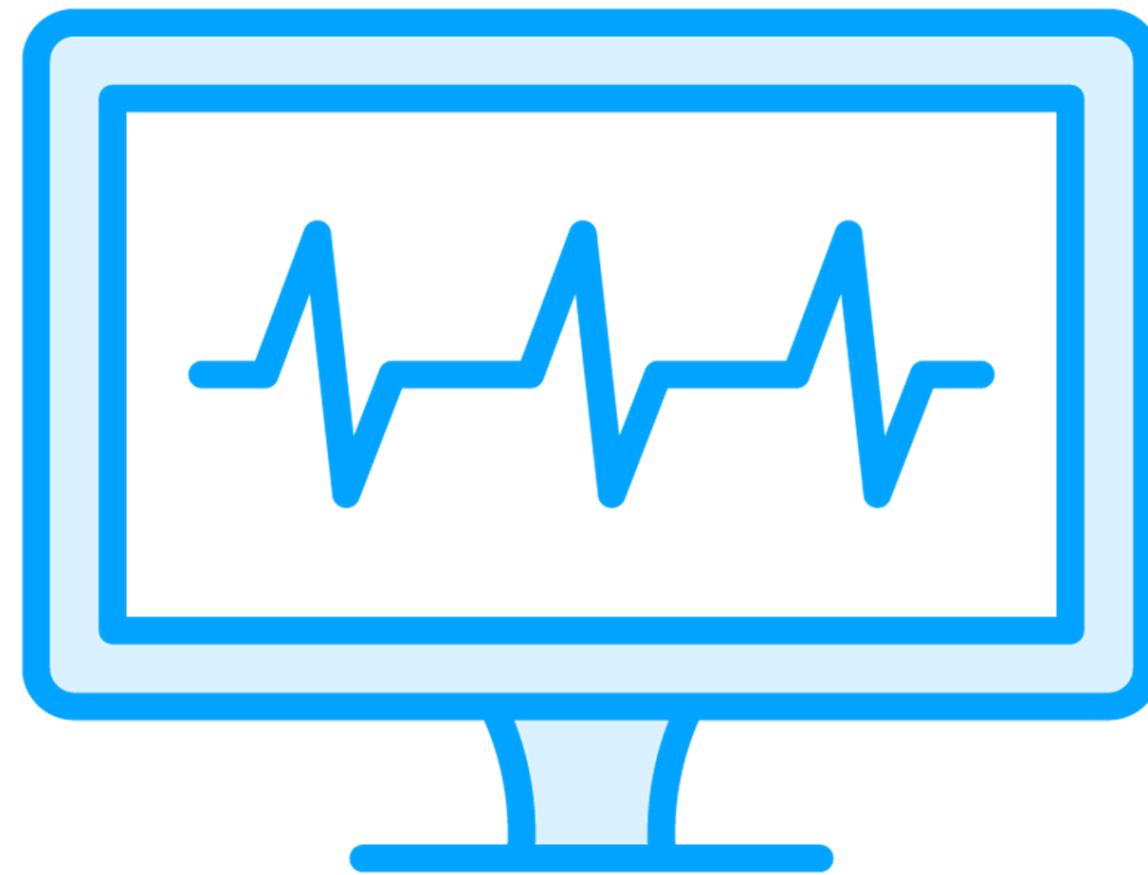
Fix the code

Test the fix

Look for similar mistakes elsewhere in the code



# Stabilizing the Error



**Repeatability**

**Narrowing the test case**

- Hypothesis, test, analysis, repeat if needed
- Which factors aren't relevant?
- Change factors and re-run tests
- Eliminate factors that don't cause the error



# Observations from the field



**Experiment and keep notes**



**Reproduce the problem locally, if possible**



**Use logs as input for production bugs, if possible**



**Interview users and watch them use the system**



**Be careful copying production data – you don't own it.**





# **Intermittent Errors?**

## **Enterprise Patterns: Concurrency in Business Applications**

**Neil Morrissey**





## **Stabilize the error**

- Testing using the user interface
- Using queries against the database



# Observations



**Which page has the problem?**

- Reports page

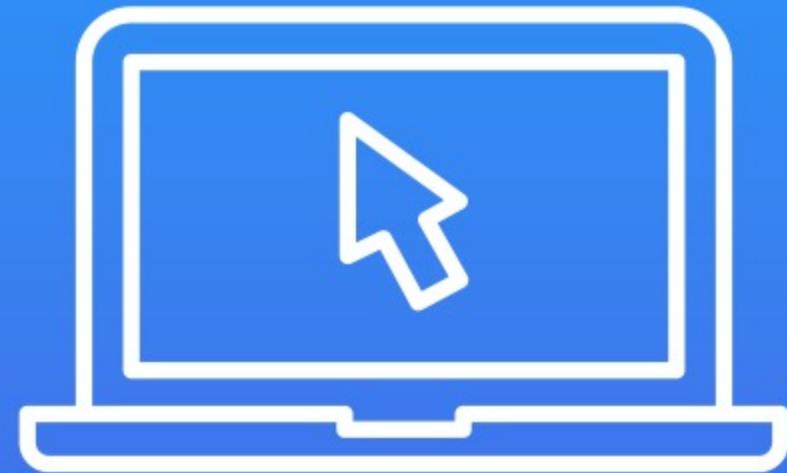
**Problem happens for multiple months**

- But not all months

**When totals are different, Reports page is always LESS than the Expenses Search page**

- Is reports page making calculations based on fewer records?





## Visual Studio Code

**Inspect data in the variables window**

**Add variables to the Watch window**

**Persist variables that go out of scope**

**Modify inputs in the debugger**



# Module Summary



**Common debugging traps**

**An effective approach to debugging**

**Stabilizing a calculation error in the UI**

**Inspecting data in the debugger**

**Modifying data in the debugger**

**Fixing the defect**

**Data debugging features of VS Code**



# References



**Code Complete, Second Edition**  
**Author: Steve McConnell**  
**Publisher: Microsoft Press, 2004**



**Up Next:**

**Up Next:**

**Advanced Breakpoints and Tracing**

---

