



Debugging debugging

Why we must approach
debugging differently

Dr. Elinor Swery

 @elinorswery
@rookoutlabs





Who Am I

- Director of Solutions Architecture at Rookout
- Obsessed with the intersection of people and technology
- Enjoy working with many different companies

Developers do a lot

Developers



What my friends think I do



What my mom thinks I do



What society thinks I do



What my boss thinks I do



What I think I do

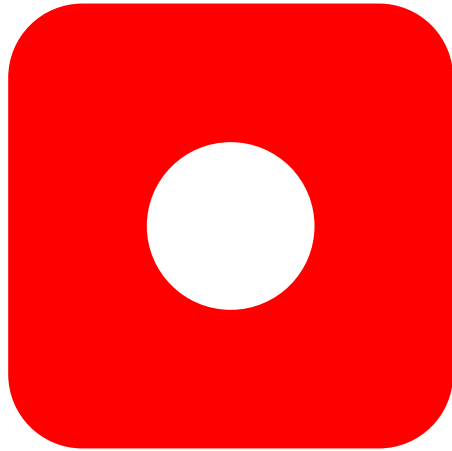


What I actually do

“75% of developers’ time is spent on solving bugs”

Coralogix

What would your team rather work on?



Bug

OR

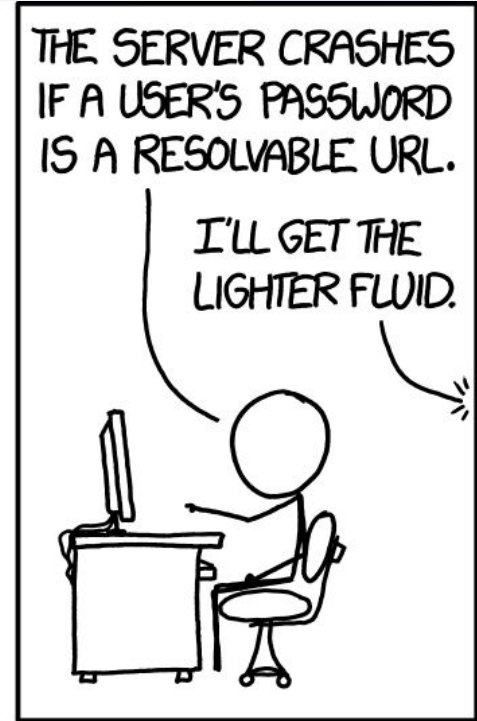


Feature

Problem #1

Fixing old things is
simply not as fun as
creating something
new





Problem #2

Too much pressure
causes stress and
anxiety



- Run the program with the debugger turned on. Set a breakpoint at the top of the function.
- Confirm that the input to the function is what you think it should be. If it's supposed to be "Banana," make sure that that is indeed what you're working with. If not, you've found your error. The question then is "why is it that my function isn't getting the input I thought it was going to get?," and you can work backwards to see how that happened.
- Step over the sort function. What output do you get from sort(input)? What output did you expect to get from sort(input)? If these disagree, you've found what's going on - the real culprit is that sort(input) is broken. Now you can start looking at sort to see what's wrong.
- Step over the countRunsIn function. What output do you get from it? What output did you expect to get from it? If they disagree, you've found the error - you're not properly counting runs, and that means you should look at countRunsIn.
- Look at the overall return value. If you correctly sorted the input and correctly counted runs and yet you *still* aren't getting the right answer, that means that the error is possibly a logic error. Maybe sorting the characters and counting runs just isn't a good way to solve this problem.

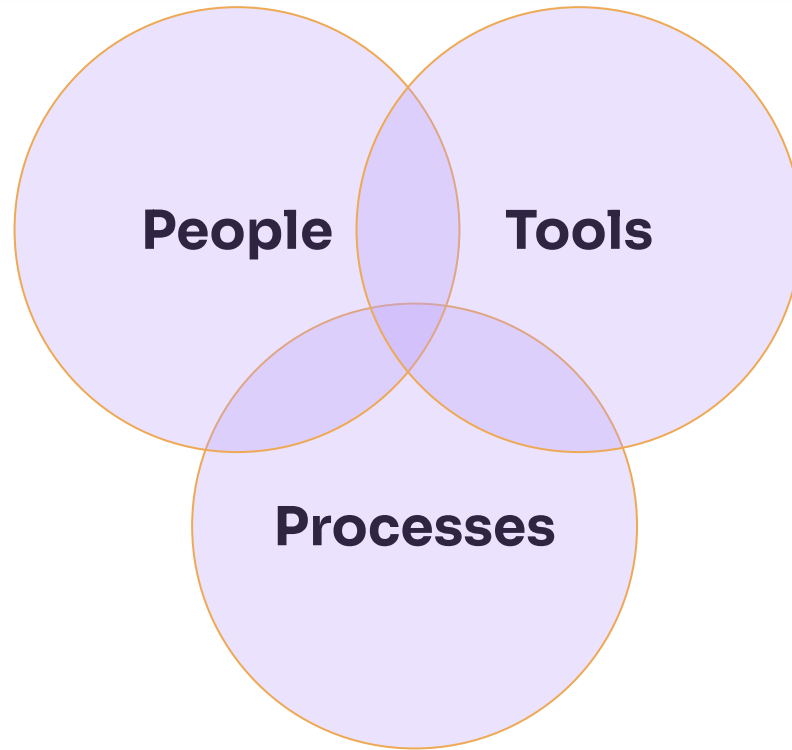


Problem #3

The tools we give our teams are simply not right for the job at hand



As managers, we should focus on:



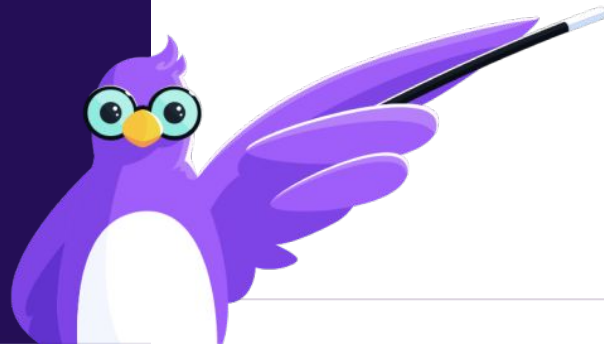
People

- > Better market the debugging task
- > Create a team based ownership mentality
- > Establish trust with the various stakeholders



Processes

- > Adjust sprint commitment / backlog if bugs come in
- > Establish a prioritisation method for new bugs
- > Have the groundwork already done for the developers





Tools

- > Create a cloud native toolbox
- > Ensure these are integrated in your team's daily workflow
- > Data is gold, but too much will weigh you down



Thank You!

Let's continue the
conversation!



@elinorswery
@rookoutlabs