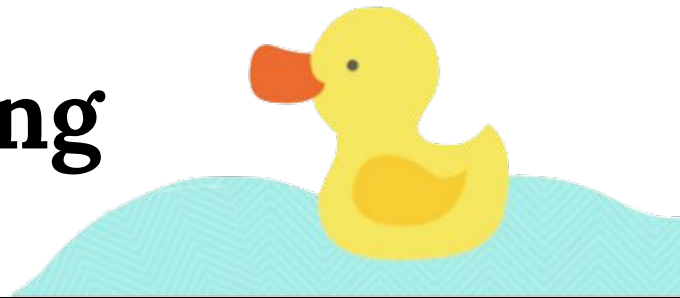


Let's Learn About **Rubber Duck** Debugging





Instructions for use

About

This deck is supposed to serve the purpose of educating M&T Employees about Rubber Duck Debugging, a preliminary debugging method that every programmer should have in their toolkit.

Questions?

If you have any questions about this specific presentation, please feel free to reach out to mchowdhury@mtb.com





Hello!

*I am **Rubber Duck***

I am here because I love to solve programmer's problems.

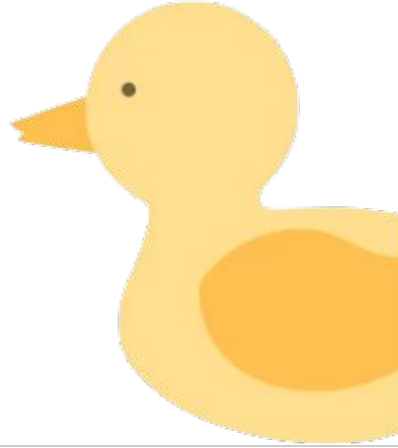
You can find me at **@M&T Bank**

1

Acquire Rubber Duck

This is step one

Beg, borrow, steal, buy, fabricate or otherwise obtain a rubber duck (bathtub variety).

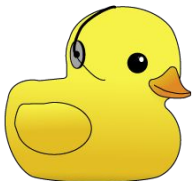


2

Ask Duck

Consent culture is important (!)

Place rubber duck on desk and inform it that you are just going to go over some code with it, if that's all right.



3

Explain code

Make sure to go line by line

Explain to the duck what your code is supposed to do, and then go into detail and explain your code line by line.





4

Problems solved

Silly mistake located and terminated

At some point you will tell the duck what you are doing next and then realise that that is not in fact what you are actually doing. The duck will sit there serenely, happy in the knowledge that it has helped you on your way.

Note: In a pinch a coworker might be able to substitute for the duck, however, it is often preferred to confide mistakes to the duck instead of your coworker.

“

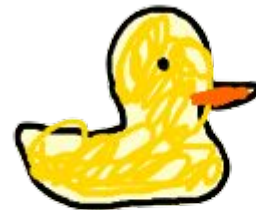


5

Bigger problem Than we thought

Doesn't always happen but...

If the duck starts talking back to you.
Please seek help
Take a break from the code
Let's get some water or something



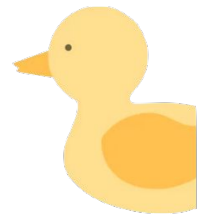


Here are some **resources**

Here's a **quick start** guide

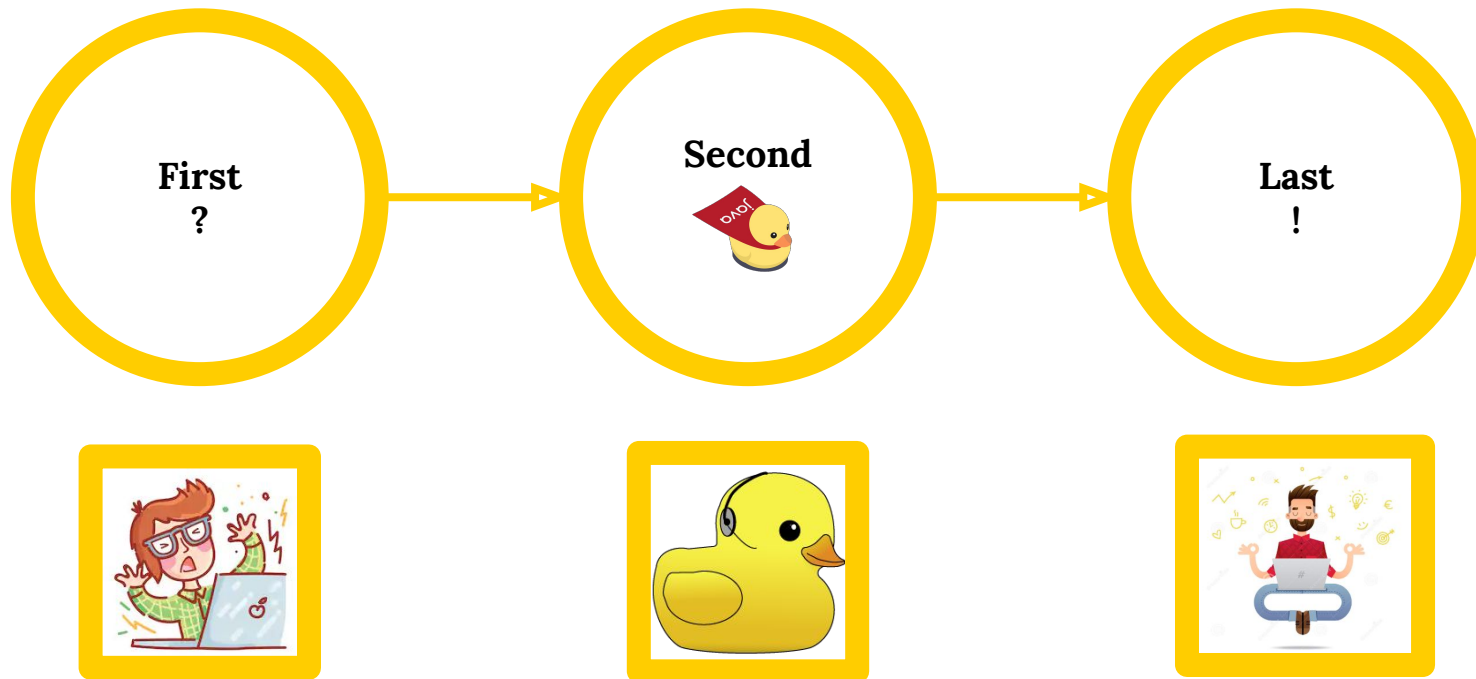
- blog.codinghorror.com/rubber-duck-problem-solving/
- selftaughtcoders.com/rubber-duck-debugging/
- swag.devrant.com/products/devducks-java-rubber-duck

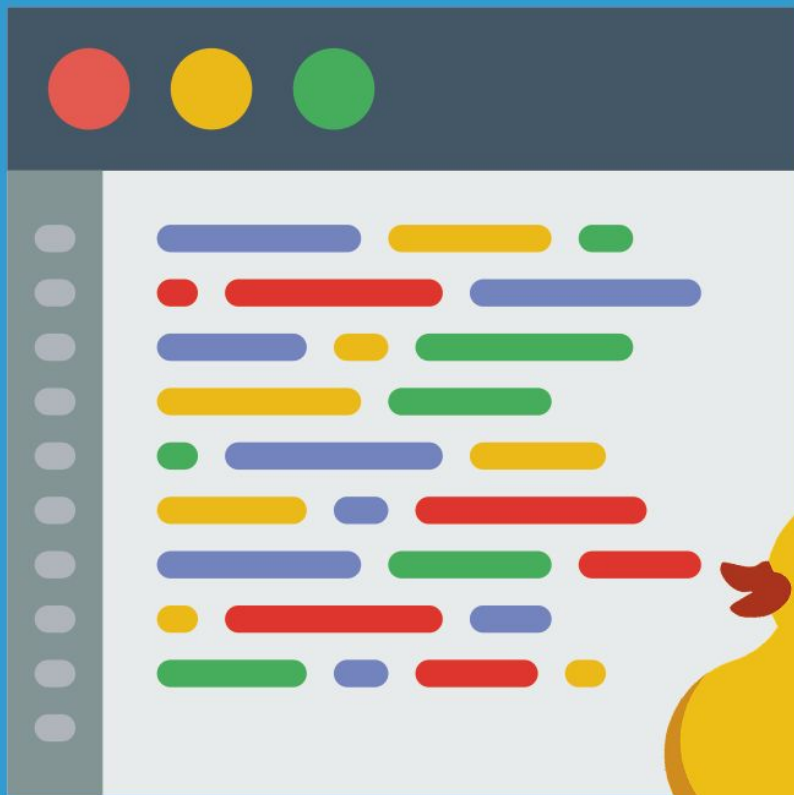
Original slide info credit goes to rubberduckdebugging.com*





Recap - Our process is easy





That's the

RUBBER DUCK DEBUG

method



Thanks!

Any **questions** ?

You can find me at

- mchowdhury@mtb.com
- Lafayette Court – Enterprise Architecture and Governance

