

nf-core/

bytesize

**Nov 14
2023**

The Art of a Minimal Example

Phil Ewels, Seqera

The MRE



WIKIPEDIA
The Free Encyclopedia

≡ Minimal reproducible example

🌐 Add languages ▼

Article **Talk**

Read **Edit** View history Tools ▼

From Wikipedia, the free encyclopedia

In [computing](#), a **minimal reproducible example** (abbreviated **MRE**) is a collection of [source code](#) and other data files which allow a [bug](#) or problem to be demonstrated and reproduced. The important feature of a minimal reproducible example is that it is as small and as simple as possible, such that it is just sufficient to demonstrate the problem, but without any additional complexity or dependencies which will make resolution harder.

A minimal reproducible example may also be referred to as a **reprex**, a **minimal working example (MWE)**, a **minimal complete verifiable example (MCVE)**, or a **short self-contained correct example (SSCCE)**.

The MRE



WIKIPEDIA
The Free Encyclopedia

≡ Minimal reproducible example

🌐 Add languages ▼

Article **Talk**

Read **Edit** View history Tools ▼

From Wikipedia, the free encyclopedia

In **computing**, a **minimal reproducible example** (abbreviated **MRE**) is a collection of **source code** and other data files which allow a **bug** or problem to be demonstrated and reproduced. **The important feature of a minimal reproducible example is that it is as small and as simple as possible, such that it is just sufficient to demonstrate the problem, but without any additional complexity or dependencies which will make resolution harder.**

A minimal reproducible example may also be referred to as a **reprex**, a **minimal working example (MWE)**, a **minimal complete verifiable example (MCVE)**, or a **short self-contained correct example (SSCCE)**.

y tho?



y tho?



WIKIPEDIA
The Free Encyclopedia

≡ Rubber duck debugging

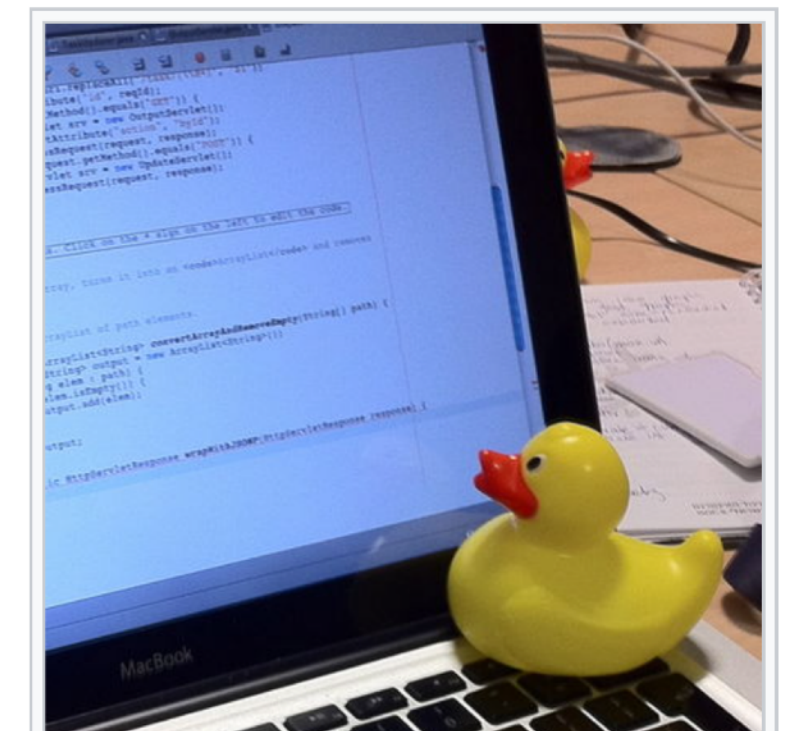
🌐 21 languages ▾

Article [Talk](#)

Read [Edit](#) [View history](#) [Tools](#) ▾

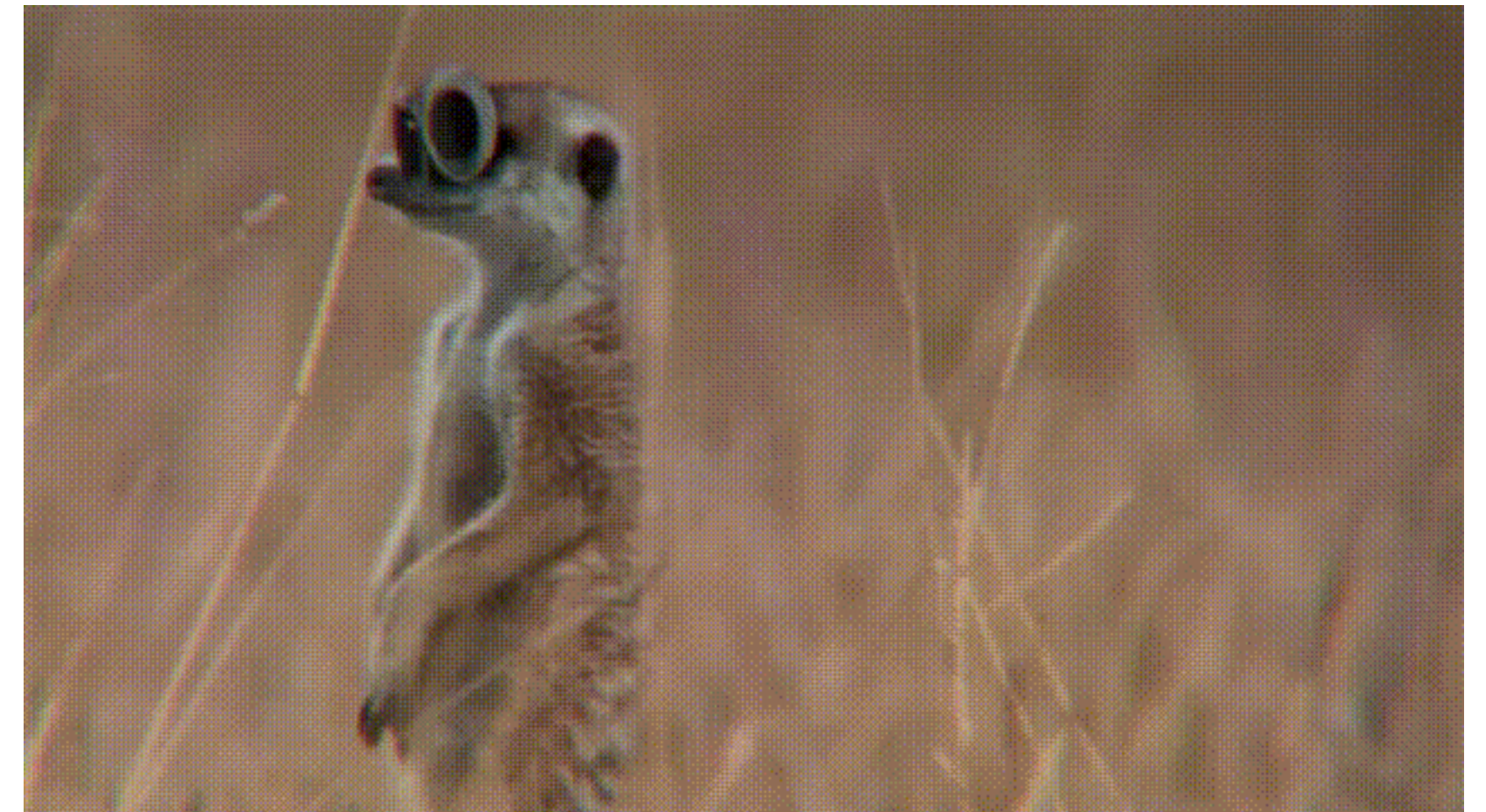
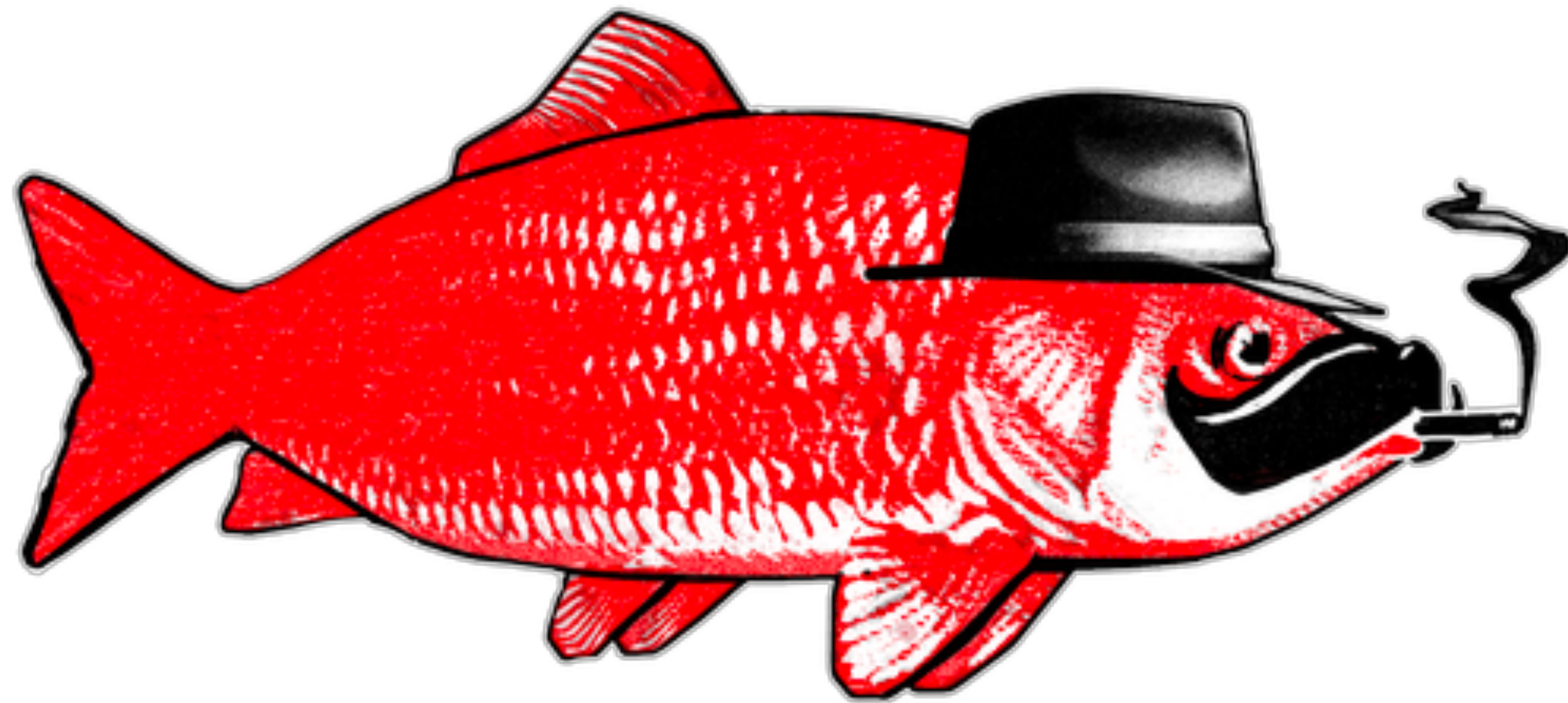
From Wikipedia, the free encyclopedia

In [software engineering](#), **rubber duck debugging** (or **rubberducking**) is a method of [debugging](#) code by articulating a problem in spoken or written [natural language](#). The name is a reference to a story in the book *[The Pragmatic Programmer](#)* in which a programmer would carry around a [rubber duck](#) and debug their code by forcing themselves to explain it, line by line, to the duck.^[1] Many other terms exist for this technique, often involving different (usually) inanimate objects, or pets such as a dog or a cat. [Teddy bears](#) are also widely used.^[2]

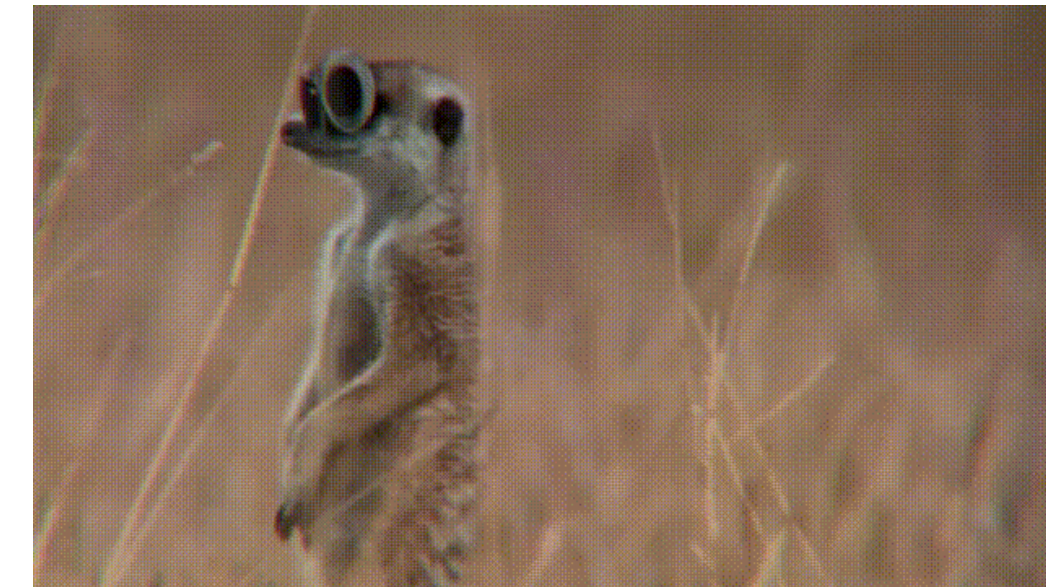
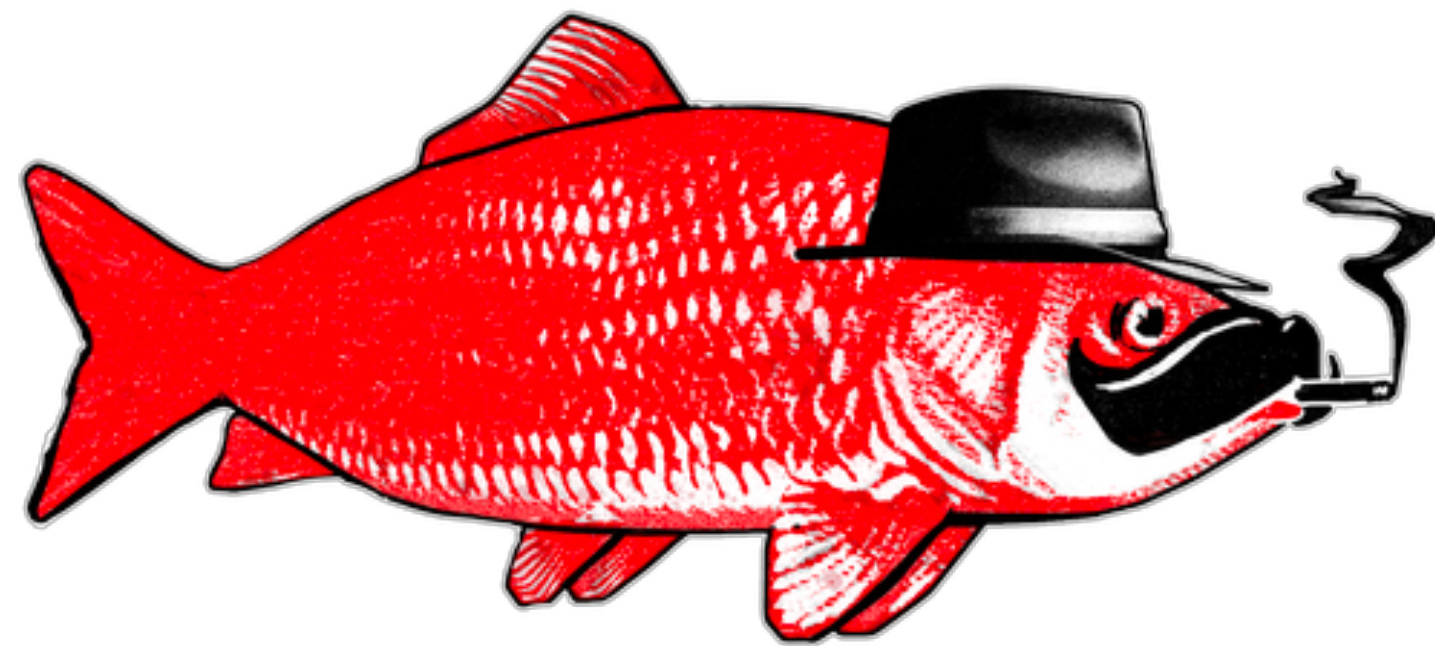


A [rubber duck](#) in use by a developer to aid [code review](#)

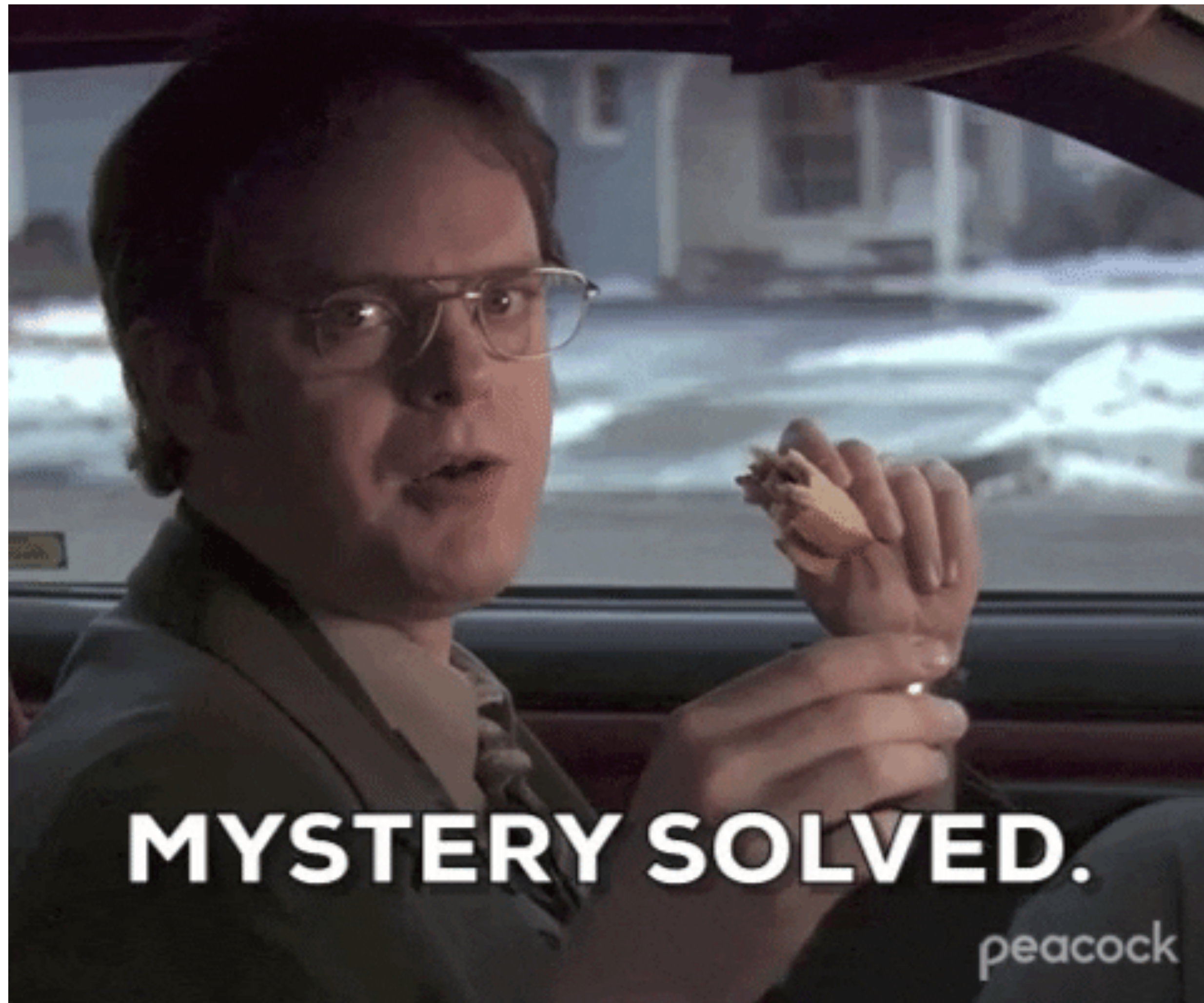
y tho?



y tho?



y tho?



y tho?



y tho?



how tho?

Minimal Reproducible Example

how tho?

Minimal Reproducible Example

how tho?

Minimal Reproducible Example



how tho?

Minimal Reproducible Example



Rike Hanssen
moreinfo



Slackbot

To help we'd love a few more details please:

1. The Nextflow command you used to launch the pipeline
2. Any custom Nextflow config files
3. The full error message
4. The corresponding `.nextflow.log` file, where you are running the analysis.

Remember to use triple backslashes on their own lines for code blocks.

Also take a look at <https://nf-co.re/usage/troubleshooting> for more information on debugging failed pipelines. Thanks!

how tho?

Minimal Reproducible Example

how tho?

Minimal Example



how tho?



how tho?



how tho?



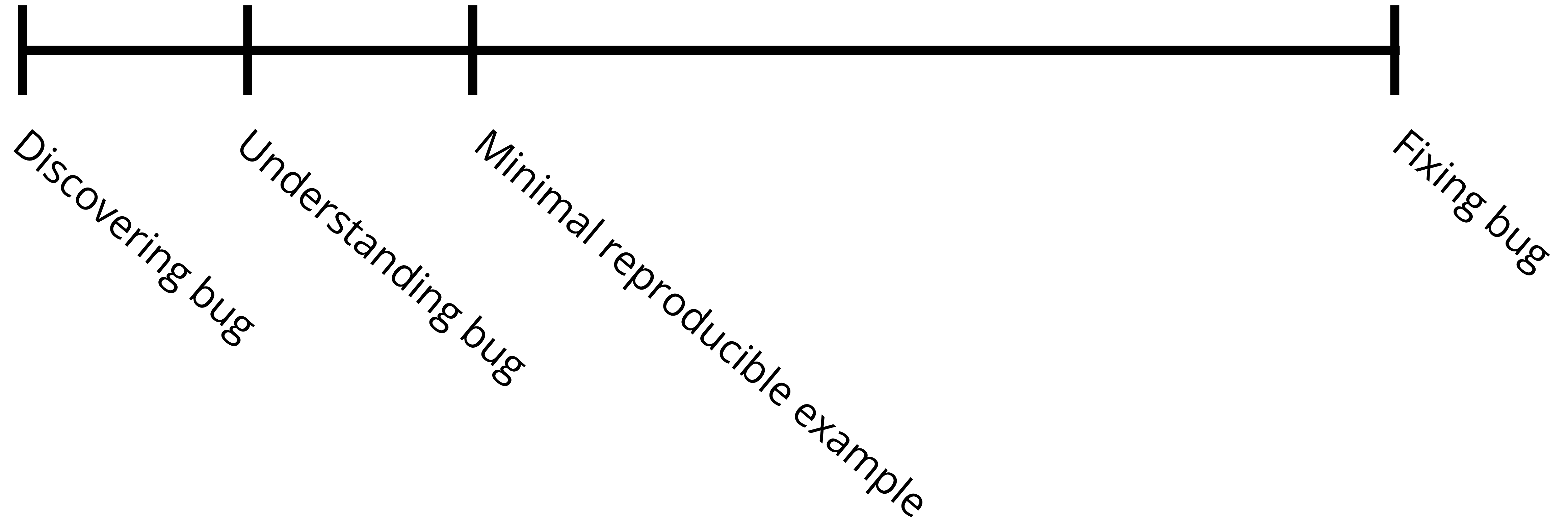
how tho?



how tho?



Recap



Control all the variables you can

Success!

