The Limits of Mikado Method

About Me

BACKGROUND

Senior developer at walley

Started out with Commodore 64 year 1986

Comp. Science programme CTH/GU 2005

A knack for sustainable code, refactoring, ...

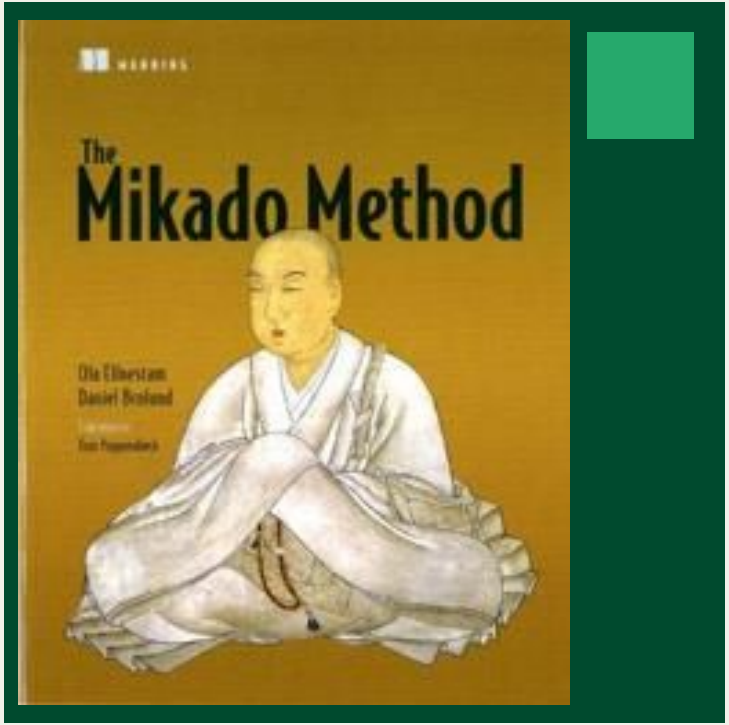
<3 Open Source e.g Ubuntu/Linux, Python, Inkscape...

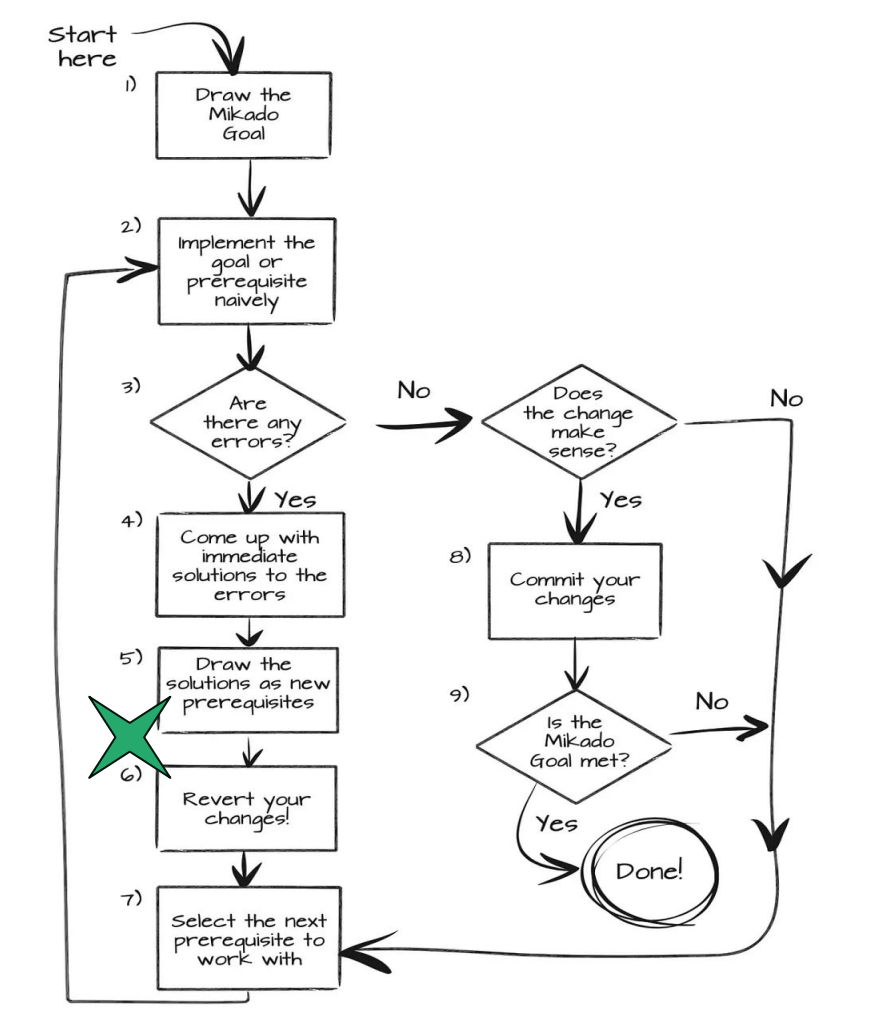
Olof Bjarnason

objarni @ github
olof.bjarnason@proton.me
olofbjarnason.se

The Limits of Mikado Method

POLL!



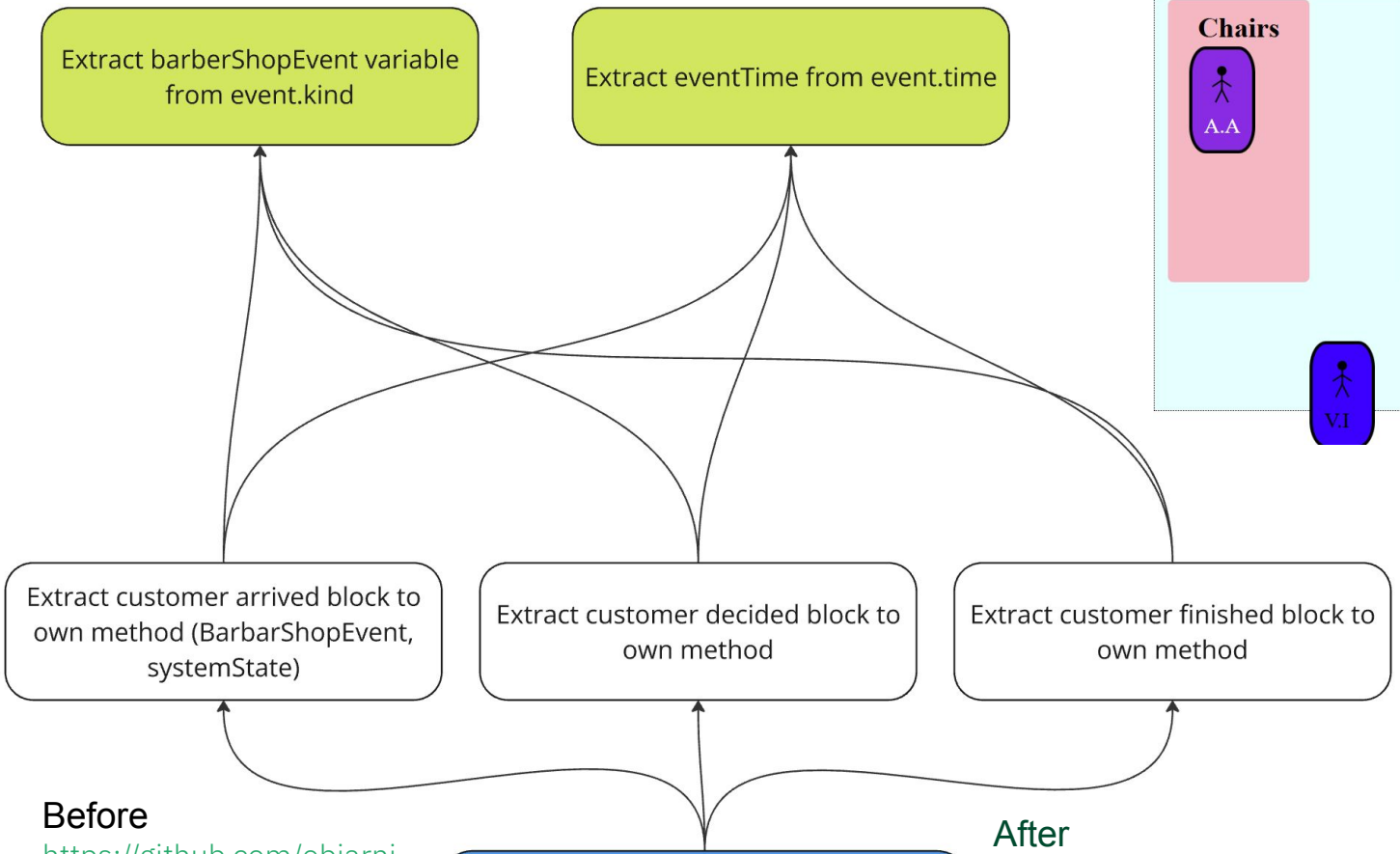


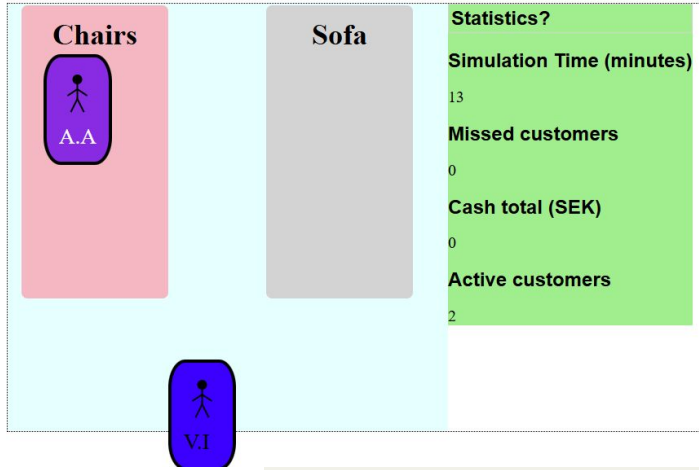
Mikado Method?

<- “recipe”

(read through 1 min)

[DEMO TIME]





Before

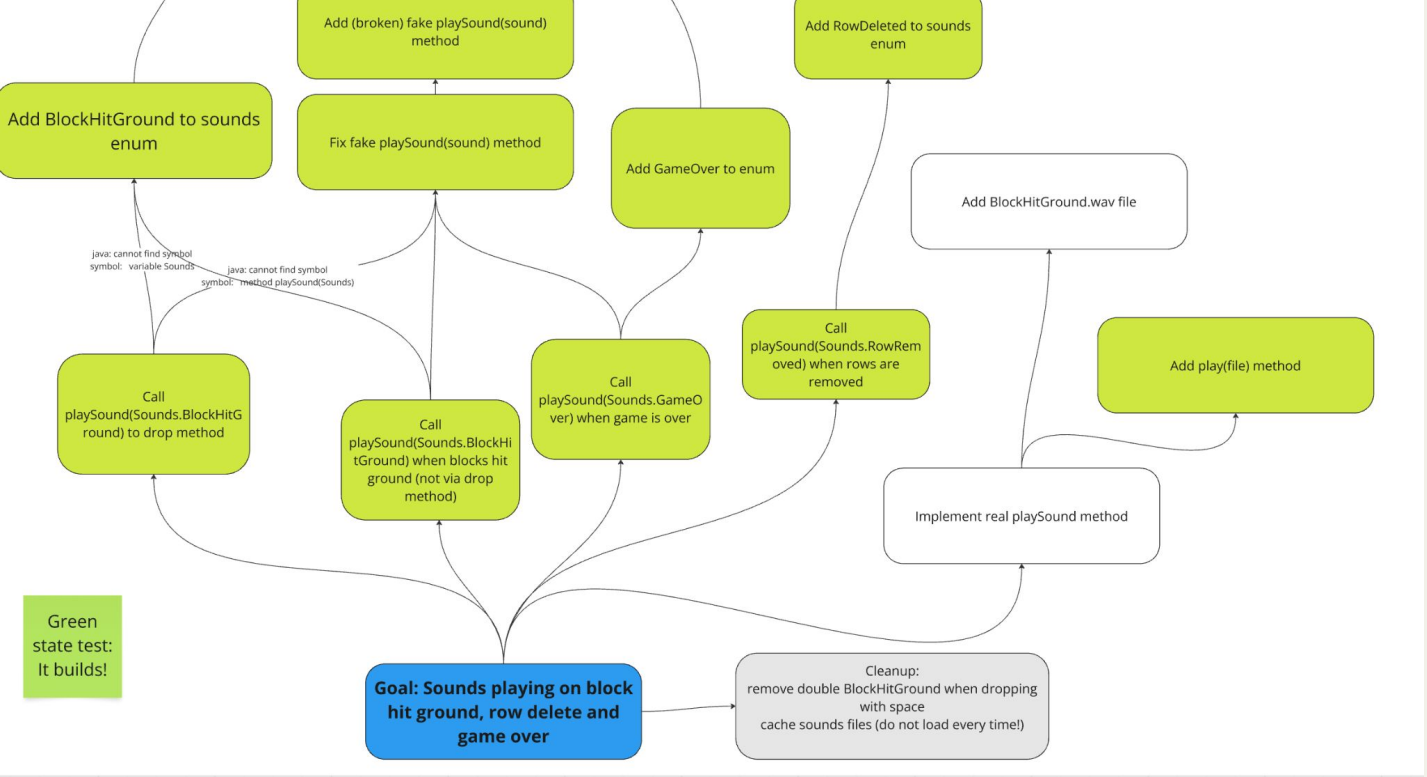
https://github.com/objarni/collaborating-birds/blob/44d3a4e30e9ca8f115d17be0a15f43c25f63c616/src/barbershop.ts#L58

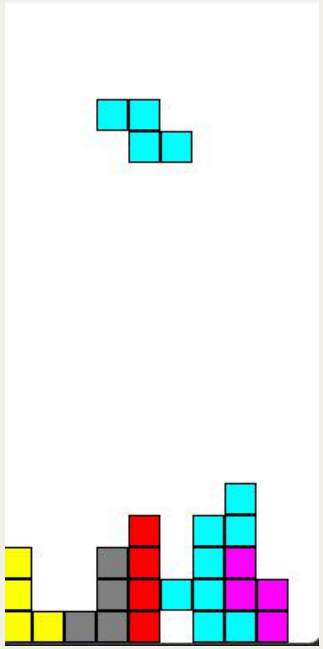
After

https://github.com/objarni/collaborating-birds/blob/0a8be75861d2428771a8e0f65ef2de8504ec2beda/src/barbershop.ts#L30

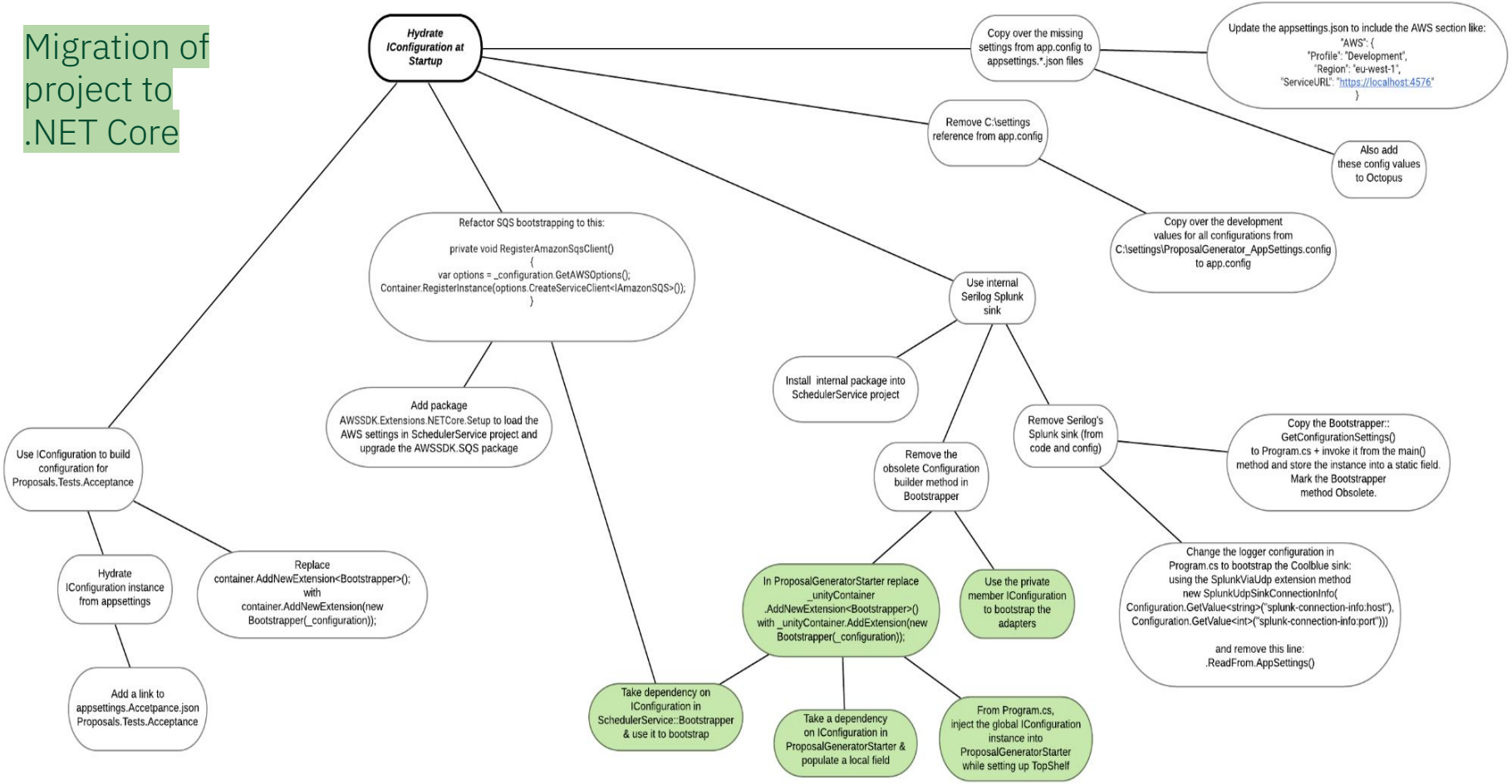
Discrete event simulation

"Barber shop"


Add (broken) fake playSound(sound) method
Add RowDeleted to sounds enum
Add BlockHitGround to sounds enum
Fix fake playSound(sound) method
Add GameOver to enum
Add BlockHitGround.wav file
java: cannot find symbol
symbol: variable Sounds
java: cannot find symbol
symbol: method playSound(Sounds)
Call playSound(Sounds.BlockHitGround) to drop method
Call playSound(Sounds.BlockHitGround) when blocks hit ground (not via drop method)
Call playSound(Sounds.GameOver) when game is over
Call playSound(Sounds.RowRemoved) when rows are removed
Add play(file) method
Implement real playSound method
Green state test: It builds!
Goal: Sounds playing on block hit ground, row delete and game over
Cleanup:
remove double BlockHitGround when dropping with space
cache sounds files (do not load every time!)



Migration of project to .NET Core



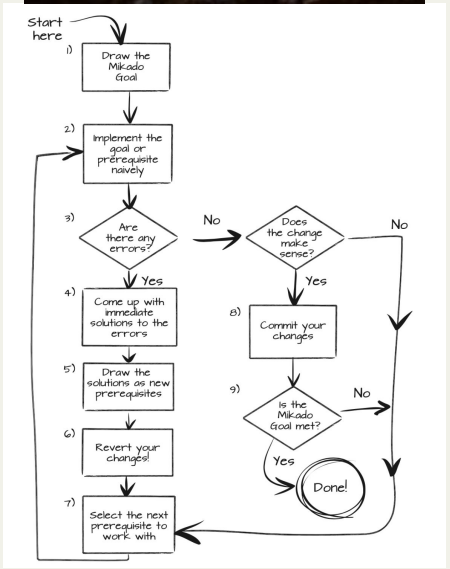
Benefits of Mikado Method

Get unstuck - avoid analysis paralysis

Small commits

Enable Trunk-Based Development

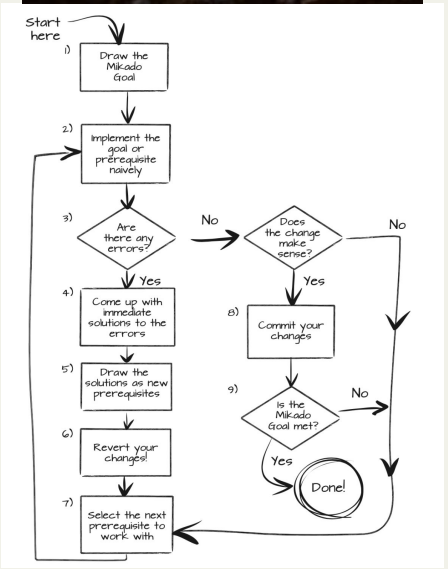
Transparency to team/org “What is going on?” during big change work



“No assumptions - relentlessly pragmatic”

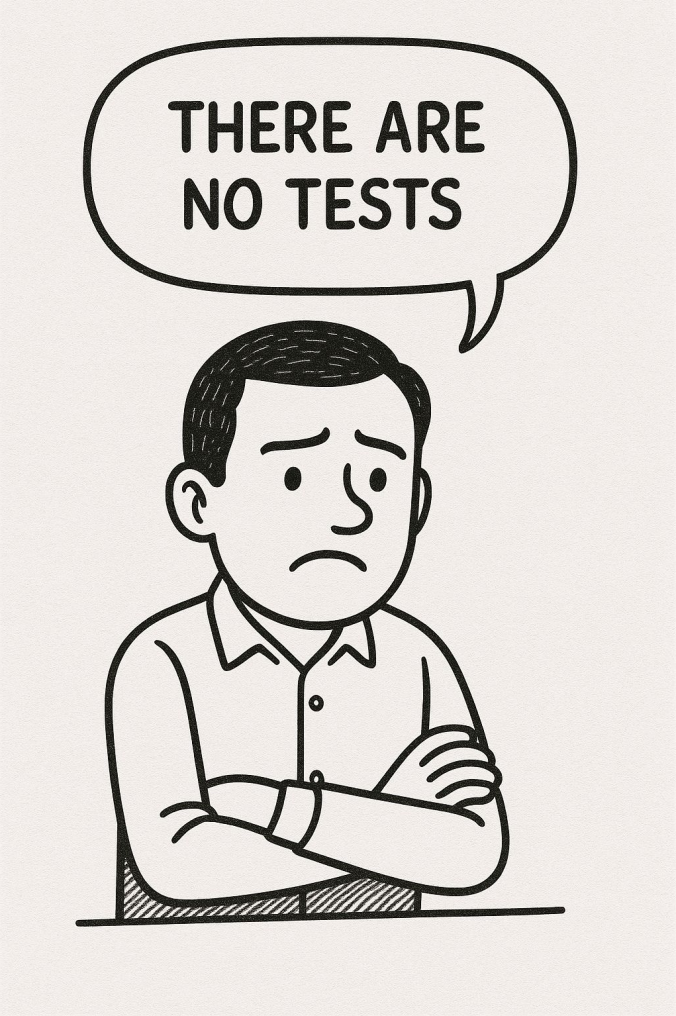
Reduce cognitive load - enable “get back to work with mental model intact - what was I doing again yesterday?”

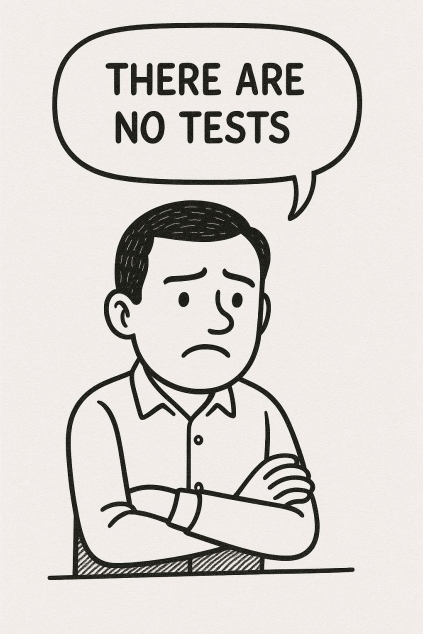
That sounds nice -
what's the catch...?



Limit

#01


THERE ARE
NO TESTS



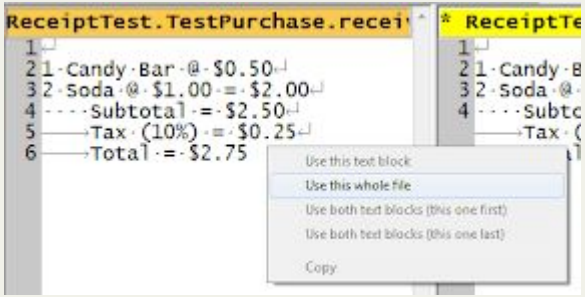
.. well, then add tests!

:)

SOLUTION - TRY THIS

Economic hint:

Try **Approval Tests*** to cover large swaths of code in one go



* also known as characterization tests, golden images tests or text tests

Limit

#02


THERE ARE RELIABLE TESTS – BUT THEY TAKE MORE THAN 1 MINUTE TO RUN!

May seem fine first couple of hours...

SOLUTION - TRY THIS

.. but with time, it becomes unwieldy. Cycle time is too long!

Find other ways to test the system automatically.

- parallelize test runs
- test subsets
- special tooling
- ... be creative!

Limit

#03

SOLUTION - TRY THIS

Audience turn! What todo?

?

(what is the pattern here?)

Yes - cycle time.

Questions?

FEEDBACK

call to action slide

Thanks for listening!