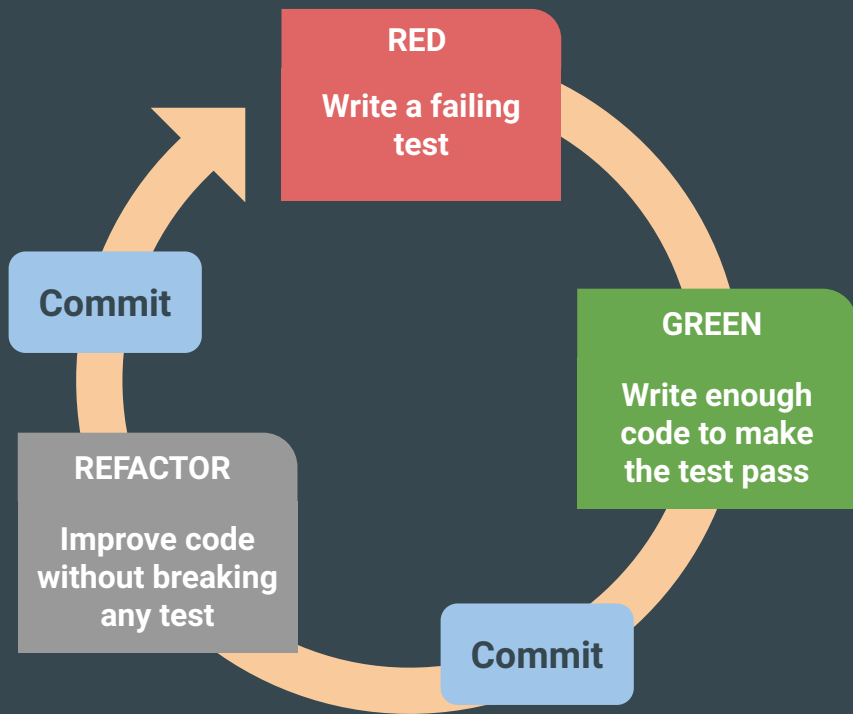# The power of the mob

André Silva, 01/07/2021

Introduction

- The three laws of TDD

- My experience with the three laws of TDD

- Mob programming

- Mob programming sessions: Retrospect

- Conclusion

The three laws of TDD

- You are not allowed to write any production code unless it is to make a failing unit test pass

- You are not allowed to write any more of a unit test than is sufficient to fail - compilation failures are also failures

- You are not allowed to write any more production code than is sufficient to pass the one failing unit test

My experience with the three laws of TDD

RED

Write a failing test

GREEN

Write enough code to make the test pass

REFACTOR

Improve code without breaking any test

Commit

Commit

Testing behavior and not the actual implementation!

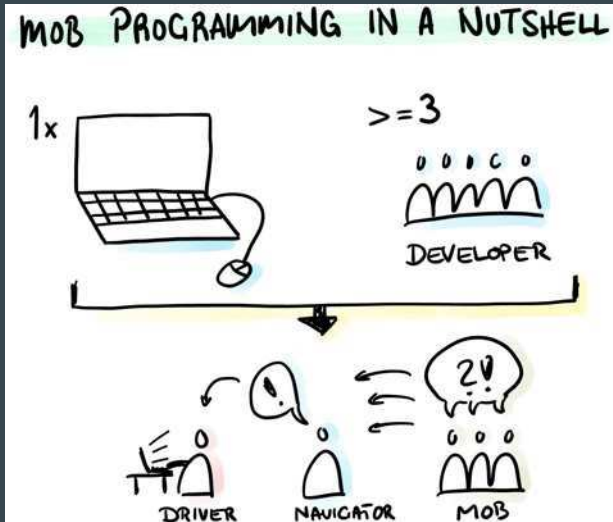As we write tests, we are adding constraints to the problem

How do we handle these constraints?

Core steps to move forward:

- Fake implementation
- Obvious implementation
- Triangulation with the next test

Mob programming

Is a software development approach where the entire team sits together and works on a given task at the same time.

This extends the collaboration to the whole team, using one computer to write code.

Mob programming

MOB PROGRAMMING IN A NUTSHELL

One keyboard/mouse

Driver: types the code. Listens to the navigator and may ask questions whenever an instruction was not clear

Navigator: gives instructions to the driver (e.g.: expressing a test scenario). May take insights/help from the mob and conveys it to the driver
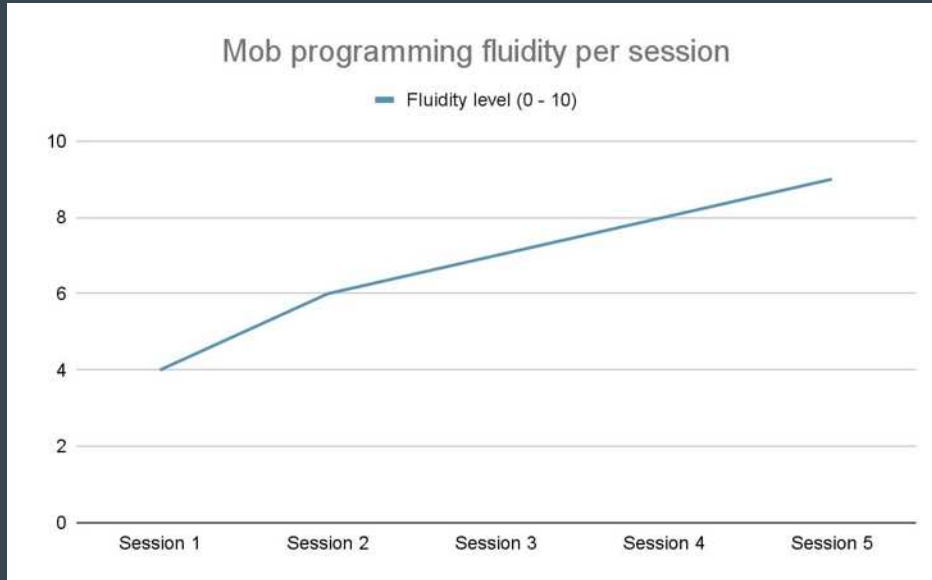
Mob: actively checks the driver-navigator dynamic and contributes when it's appropriate

Timed driver/navigator rotation (around 10 min)

Do a few breaks during the session

Mob programming sessions: Retrospect
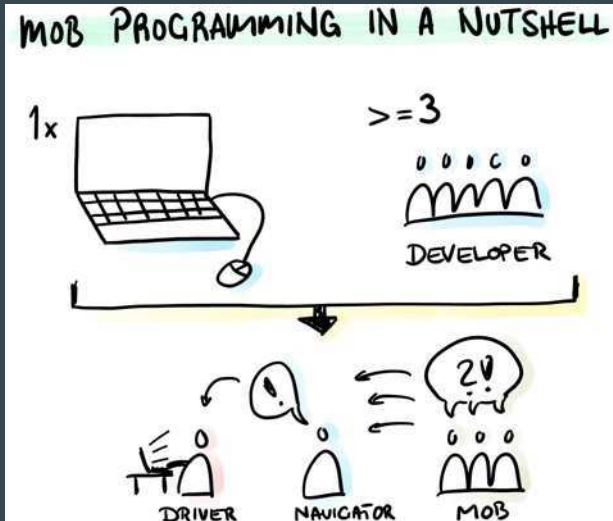
Mob programming fluidity per session
— Fluidity level (0 - 10)

1st session: slower pace, mainly due to the unpacking of the concepts

2nd session: our interactions started to feel more fluid. Less chaotic

3rd session onwards: gradual increment per session. New concepts were being put into practice and we were learning from previous sessions

Conclusion

At first, I was somewhat skeptical about the benefits (if any) of a mob programming session. However, based on the experience and knowledge I have gathered throughout these sessions, I can now see their value and how positive they can be towards solving a particular problem.

During these sessions I noticed that the TDD concepts were being consolidated and applied on each iteration which, at the same time, were improving our way of solving the given problems.

On top of that, I look at them as an environment where every team member, which might have different skill sets, has the opportunity to share knowledge, improve the codebase and mitigate unaware edge cases or bugs.

It is an approach that I would like to consider for my current project and try it out, as an experience, on a monthly basis.

A regular old Plumbus

Thank you!

- Alessandro and Marco, for providing and explaining these concepts and guiding us throughout the entire course.

- My colleagues, for their contribution to this journey. It was fun!

Questions?