Testing
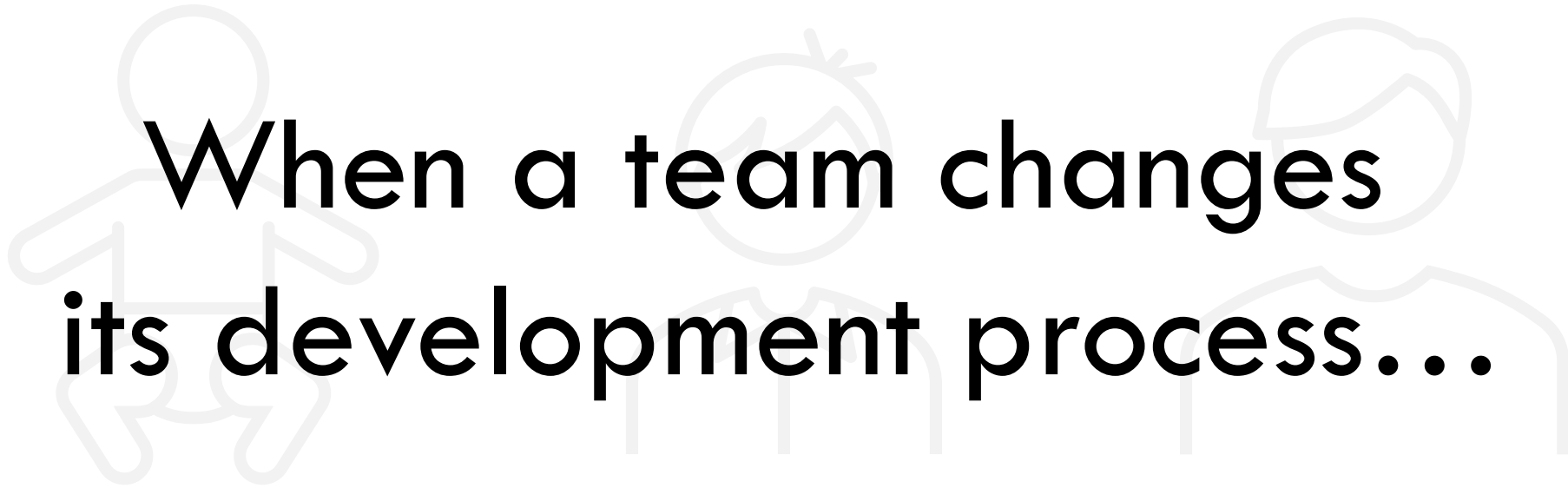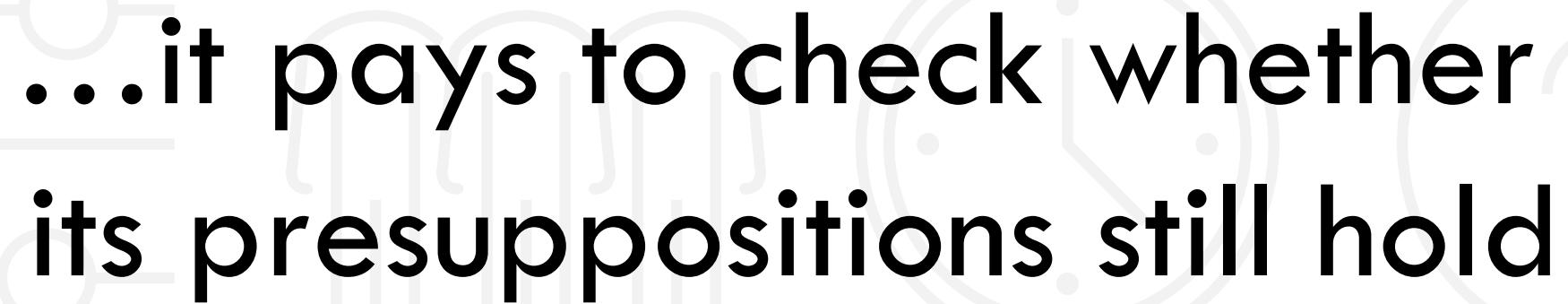# A Philosophical Retrospective

Karl van Heijster

www.karlvanheijster.com

When a team changes
its development process…

…it pays to check whether its presuppositions still hold

What did our changing testing strategy teach us about testing?

Our original development process was deceptively simple:

The programmers program a feature, the testers test it

This presupposed a lot about

1) how
2) who
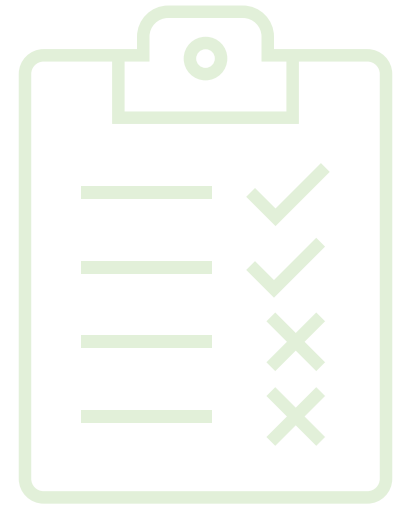3) when, and
4) why we tested software

We saw testing as
1) a manual task
2) done by testers
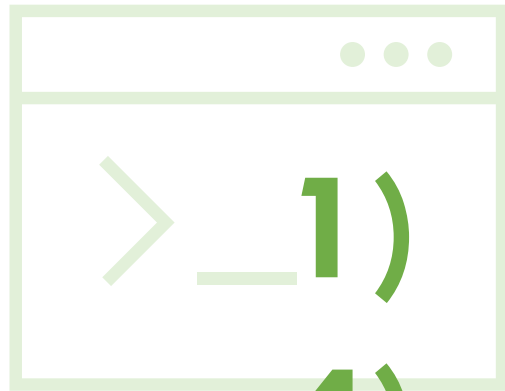3) after coding
4) to check the dev's work

Problem

Our way of working lead to week long regression test sprints

Solution

Automate the test scripts

We became aware of some (now) incorrect presuppositions

Good tests are
1) automated
4) to prevent regressions

Problem

The tester could not keep up with the work

Solution

Make the programmers (partly) responsible for automating tests

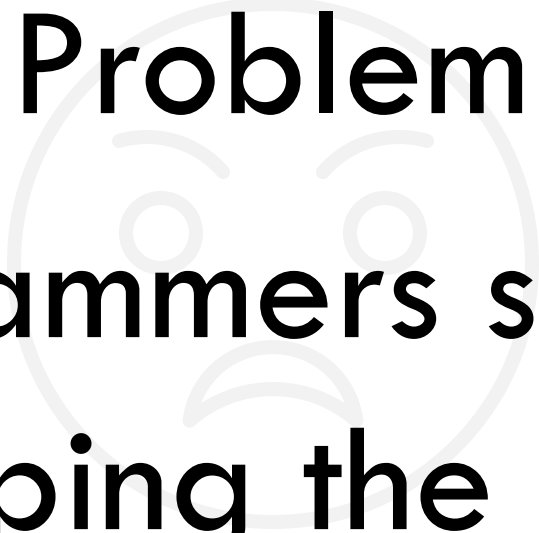This made us notice some other incorrect presuppositions

Tests are written

2) by testers *and* programmers
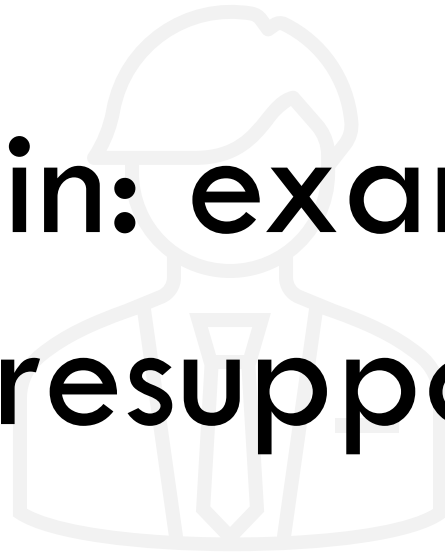
4) to document the code

Problem

Programmers started
skipping the tests

Solution

Teach them the ways of Test-Driven Development

Again: examine
your presuppositions

Testing is something you do

3) during development
4) to help you design the code

The development of our testing strategy shows the following trends:

o Automate ever more

o Test ever sooner

o Shift responsibility

Programming and testing
are two sides of the same coin
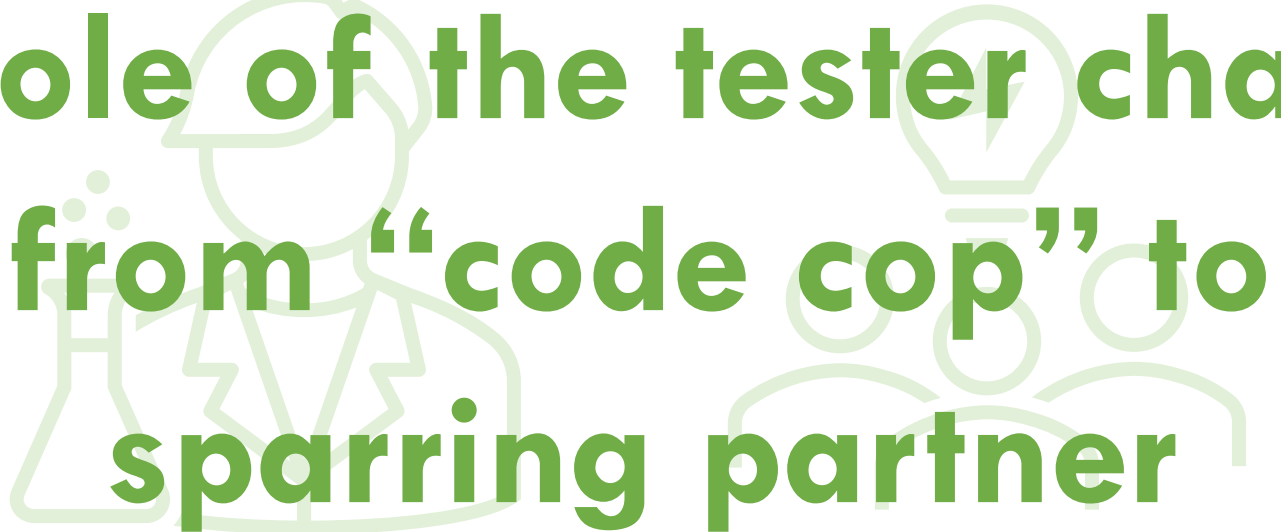
Where to go from here?
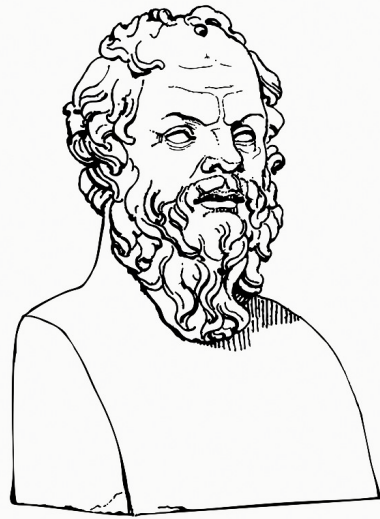
Problem

Programmers and testers are doing redundant work

Solution

Collaborate!

The role of the tester changes from "code cop" to sparring partner

But that won't be
the last presupposition
that will have to go…

The unexamined life is not worth living

- Socrates

Testing
A Philosophical Retrospective

Karl van Heijster

www.karlvanheijster.com