Refactoring Tips and Principles

Andrejs Doronins
TEST AUTOMATION ENGINEER

Refactoring

What?

Why?

When?

When not?

How to be efficient at it?

Tips & Principles

Tools

Further Study

Tips and Principles

Understand your code well before refactoring

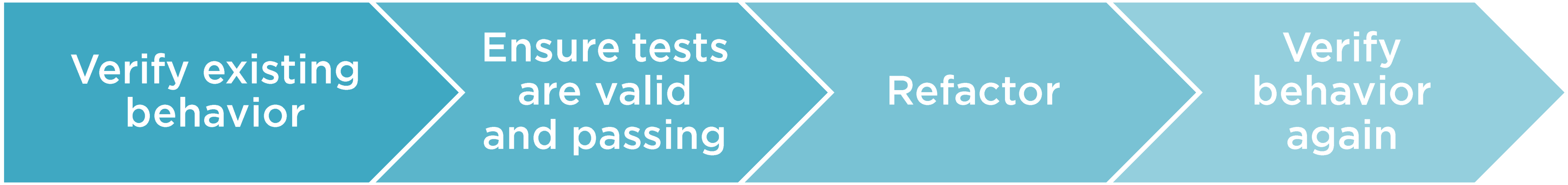
Create or run existing tests

Keep each refactoring small

Manage the scope of refactoring

Seek help and advice

Standard Refactoring Process



Verify existing behavior

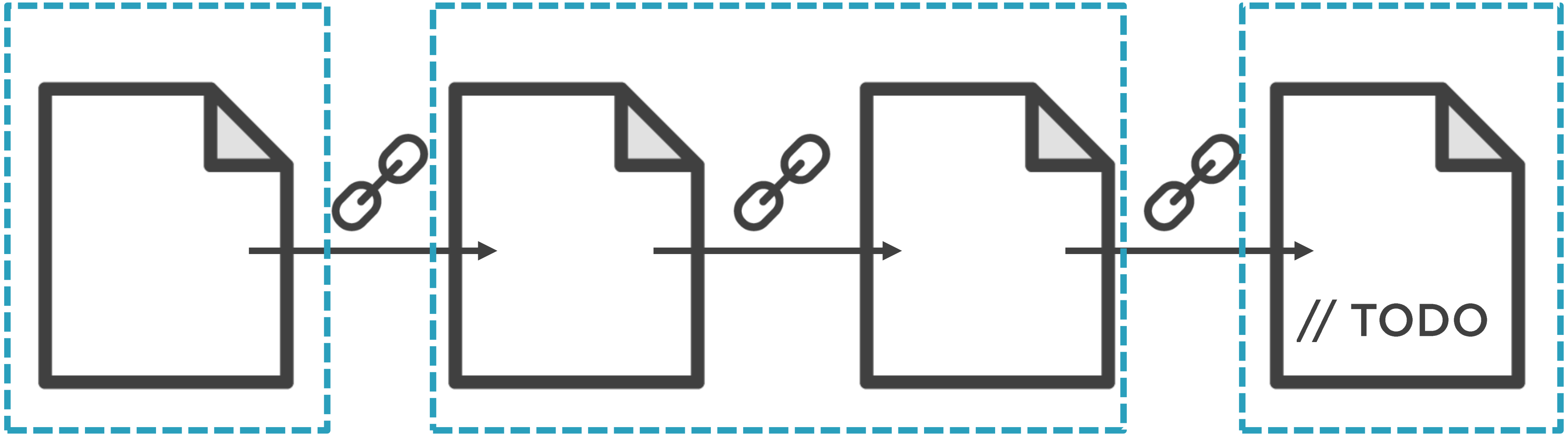
Ensure tests are valid and passing

Refactor

Verify behavior again

> “Refactoring in small steps helps prevent the introduction of defects.”

Joshua Kerievsky


// TODO

> “To get the best refactoring results, you’ll want the help of many eyes.”

Joshua Kerievsky

Use Code Review

Static Analysis Tools

IDE

SonarLint

A tool is not a replacement for skills

Further Material

Refactoring (Book) by Martin F.

SOLID Principles of Object Oriented Design

Java: Writing Readable and Maintainable Code

Summary

- ✓ Refactor one step at a time
- ✓ IDE and static analysis tools are your friends
- ✓ Continue learning and practicing