

FLOWING[®]

—

Strategic Test-driven Development



Ego Slide

Francesco Strazzullo
(AKA **Strazz**)

[@TheStrazz86](https://twitter.com/TheStrazz86)

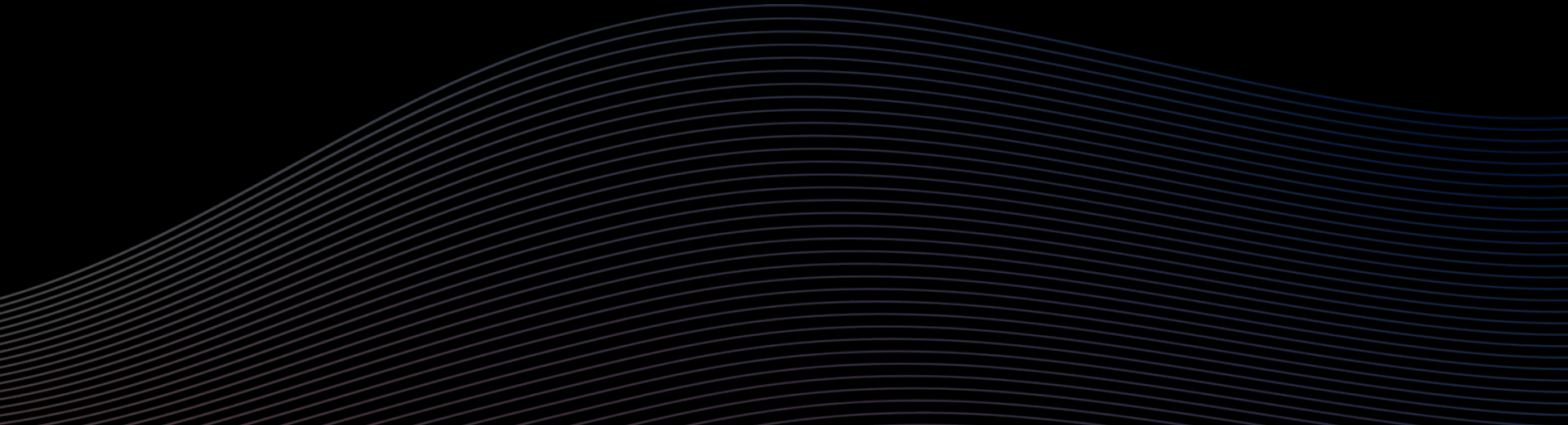
(Follow me on Twitter!)

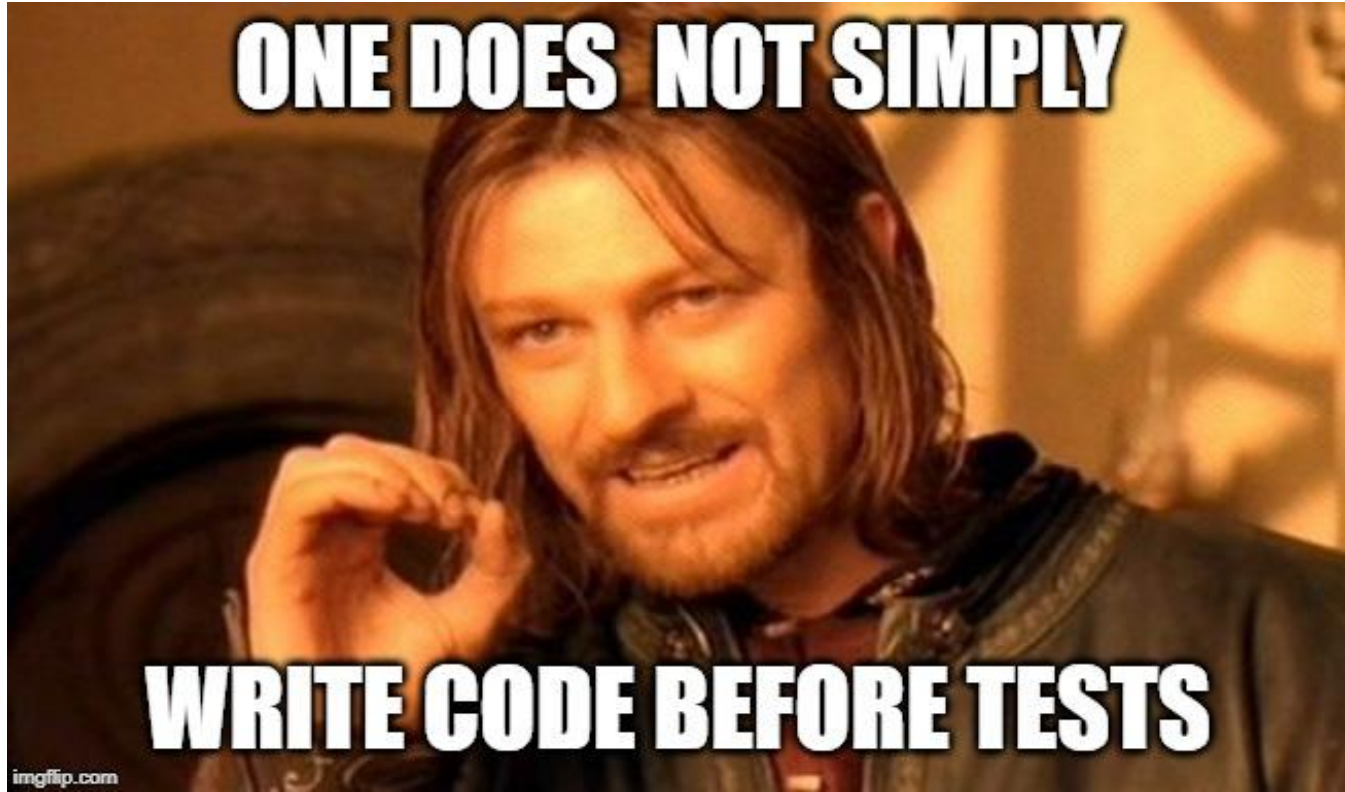


FLOWING®

—

What is Test-driven Development?

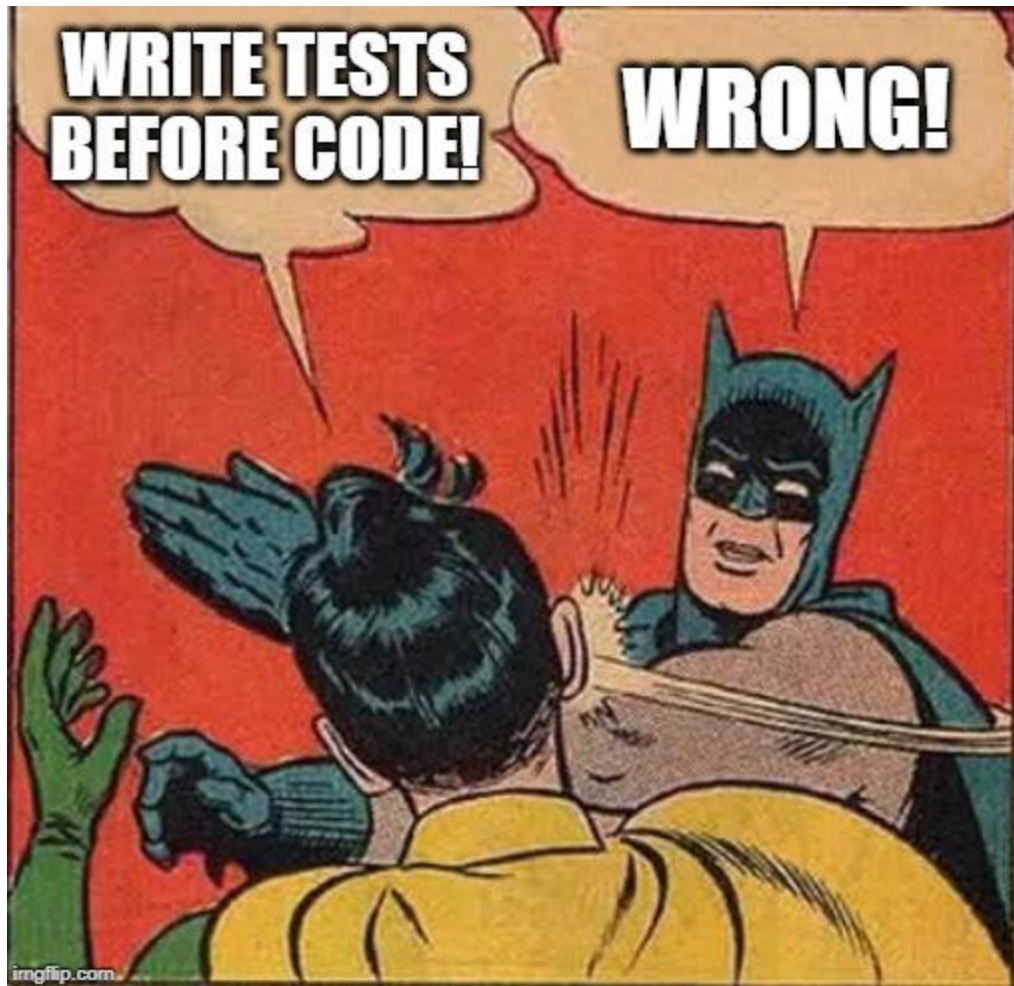




FLOWING®

**WRITE TESTS
BEFORE CODE!**

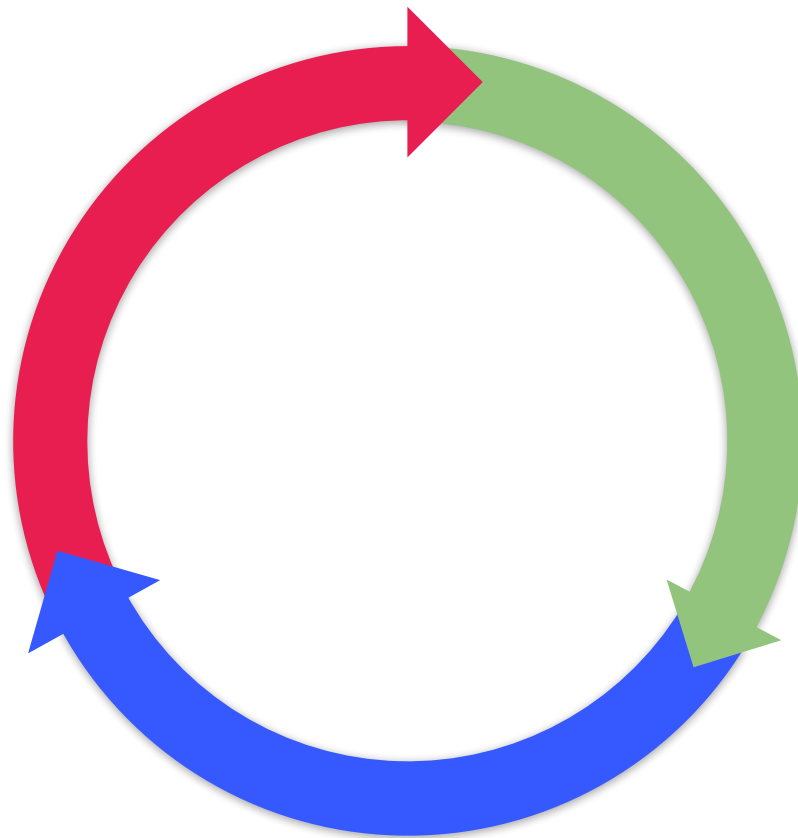
WRONG!



FLOWING®

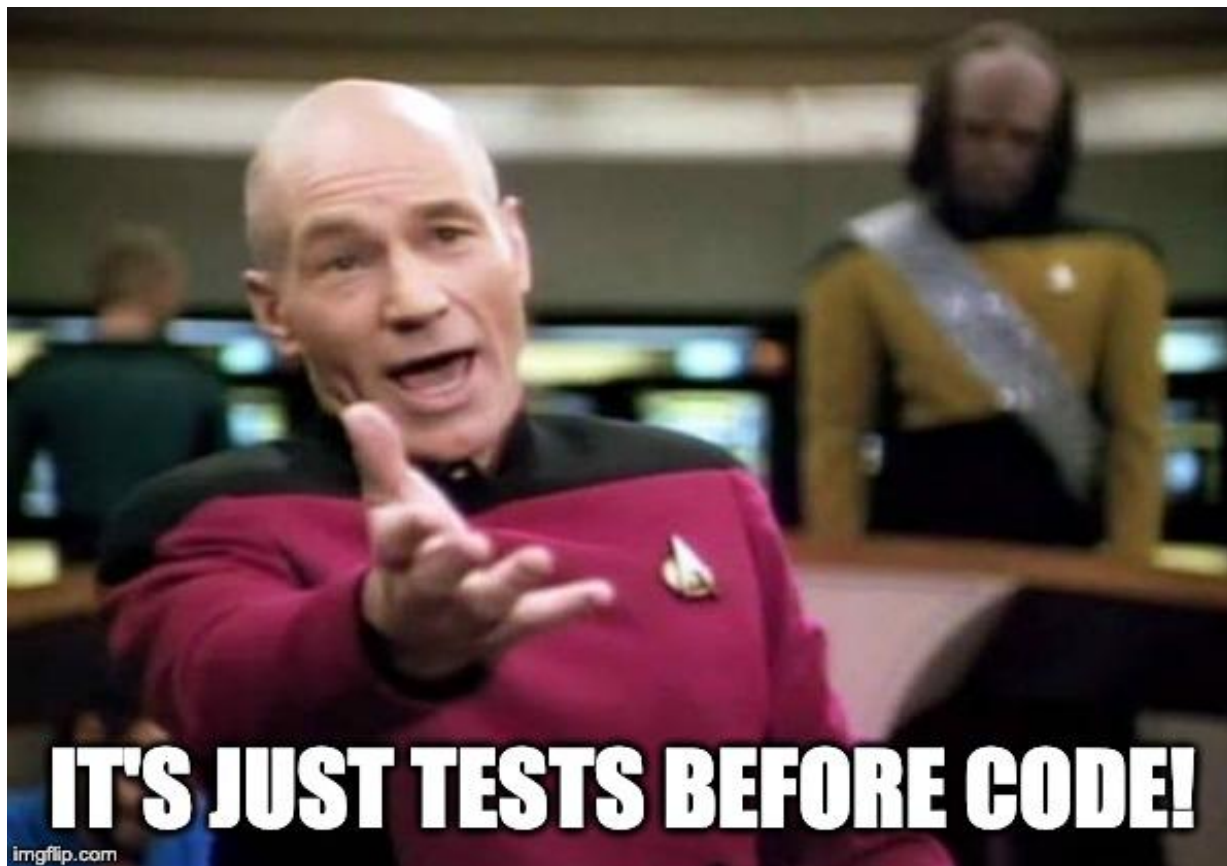
Write a failing
test

Write code to
pass that test



FLOWING®

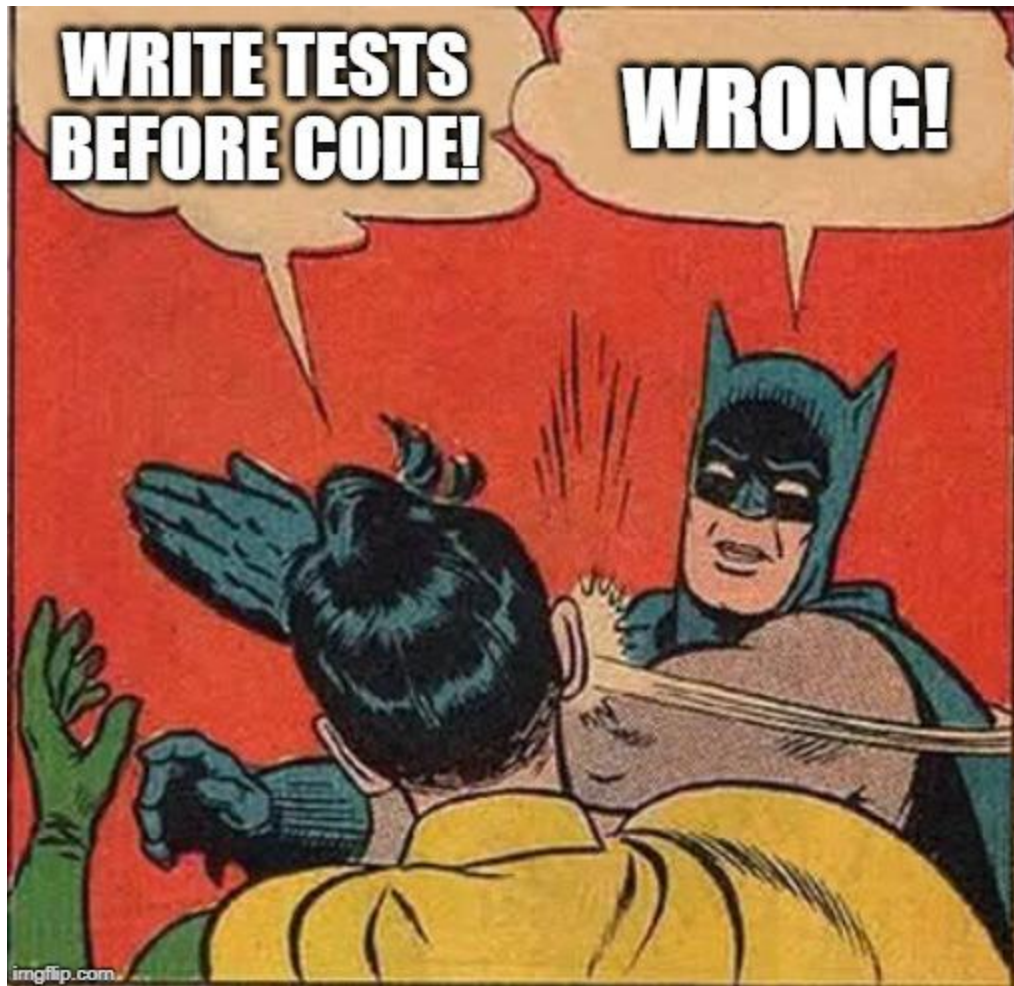
Refactor keeping the test green



FLOWING®

**WRITE TESTS
BEFORE CODE!**

WRONG!



FLOWING®

imgflip.com

FLOWING®



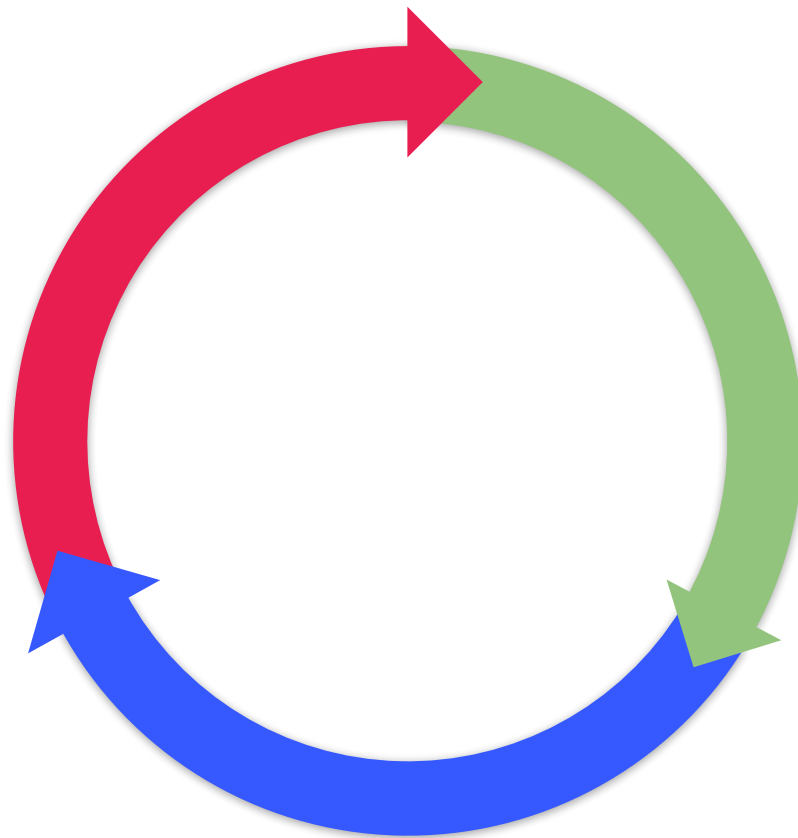
Test **Driven** Development

FLOWING®

How can Development be driven by Tests?

Write a failing
test

Write code to
pass that test



FLOWING®

Refactor keeping the test green

**A Big Decision is hidden in
plain sight**

**What is the next test to
write?**

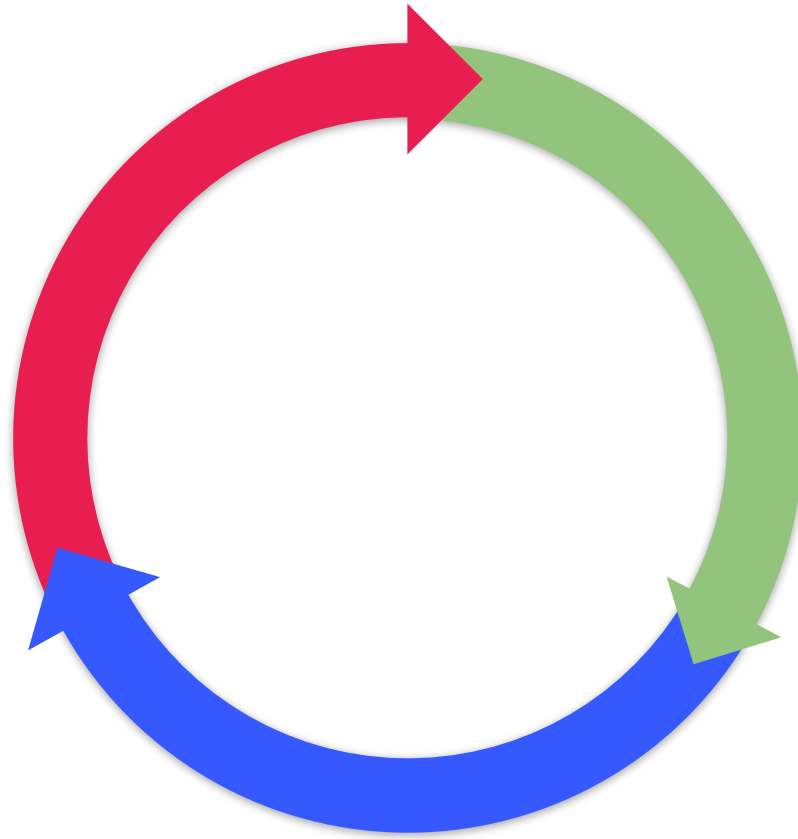
*“Software development is a learning process,
working code is a side effect. If that’s true...
how can we maximize learning?”*

Alberto Brandolini

**Write the test that lets you
learn the most about the
problem**

**Write the test that lets you
reach your goal as fast as
possible**

Write a test that
lets you
approach the
goal



Write code to
pass that test

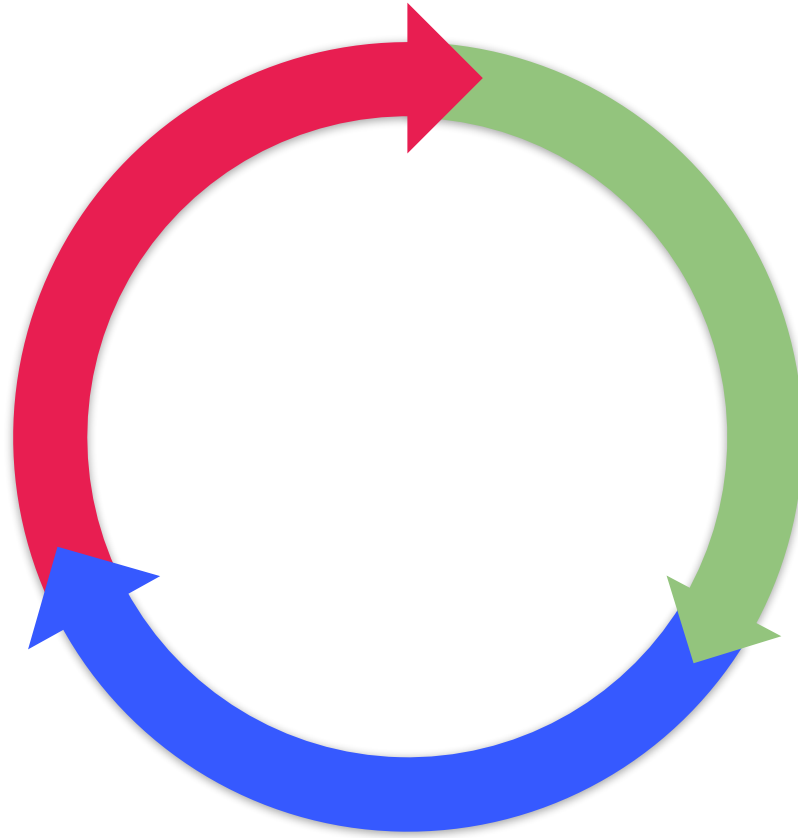
FLOWING®

Refactor keeping the test green

**Another Big Decision is
hidden in plain sight**

**Write just the code that you
need to pass the new test**

Write a test that
lets you
approach the
goal



Write just the
code needed to
pass that test

FLOWING®

Refactor keeping the test green

**Your Design emerges from
tests**

—

**TDD is a goal oriented
decision-making tool
based on tests**

The bottom half of the slide features a decorative graphic consisting of numerous thin, light blue lines that curve and ripple across the width of the page, creating a sense of motion and depth.

Test Driven Development

FLOWING®

What is a Test?

*“Tests are just one form of
feedback”*

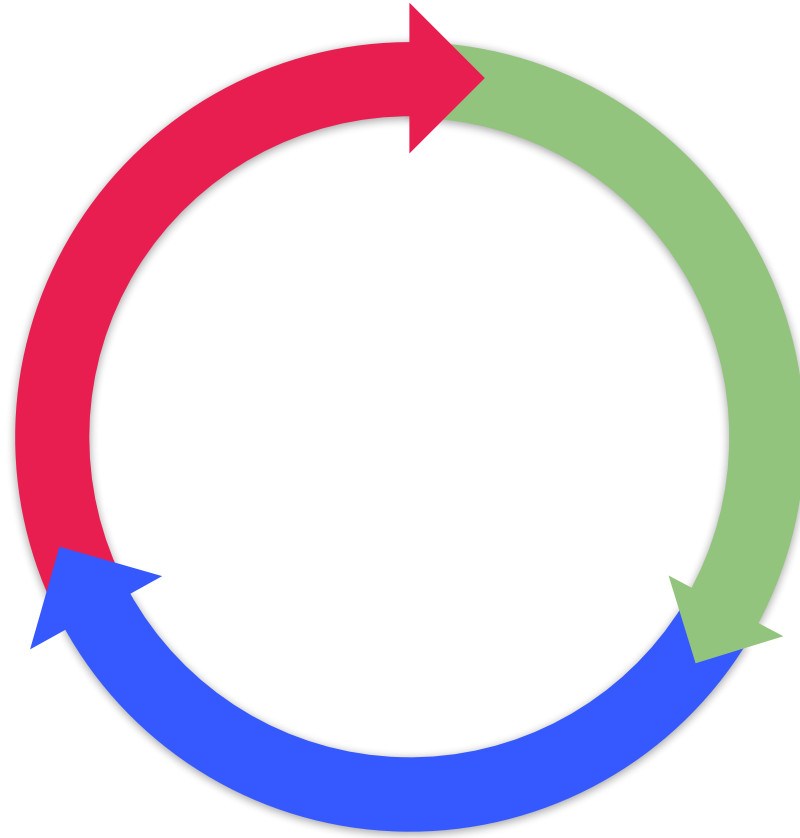
Kent Beck

—

Feedback Driven Development

FLOWING®

Define a feedback
that lets you
approach the goal



Build something
“small”

FLOWING®

Validate your code with “fast” feedbacks

—

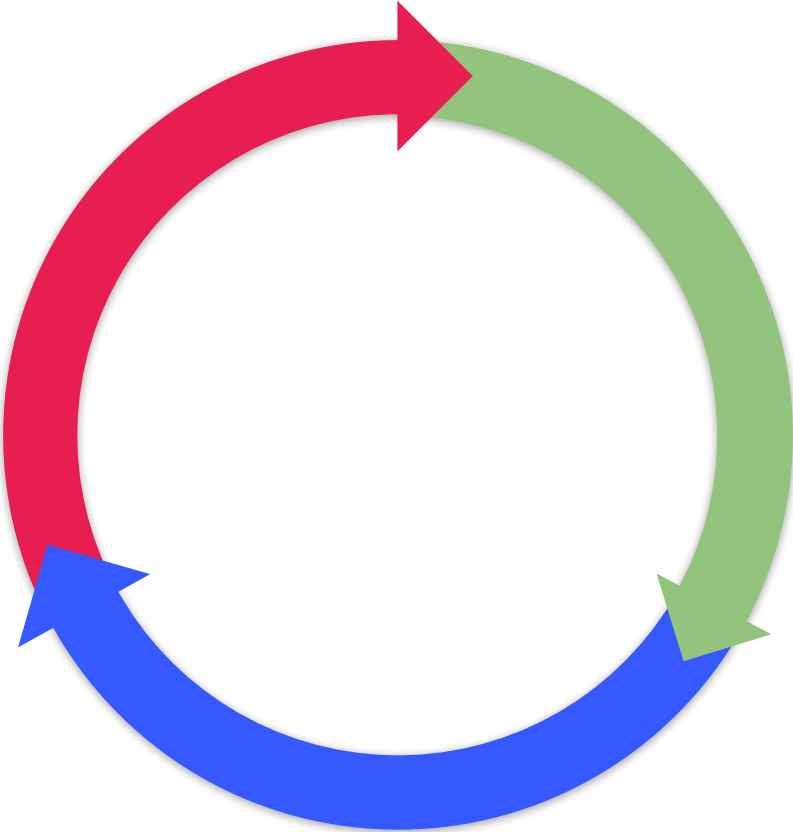
**TDD is a goal oriented
decision-making tool
based on feedbacks**

The bottom portion of the slide features a decorative graphic consisting of numerous thin, dark blue lines that curve and ripple across the width of the page, creating a sense of motion and depth.



Goal

Small Outcome



FLOWING®

Feedback

—

What kind of decisions can you make with TDD?

FLOWING®

Strategic long term decisions

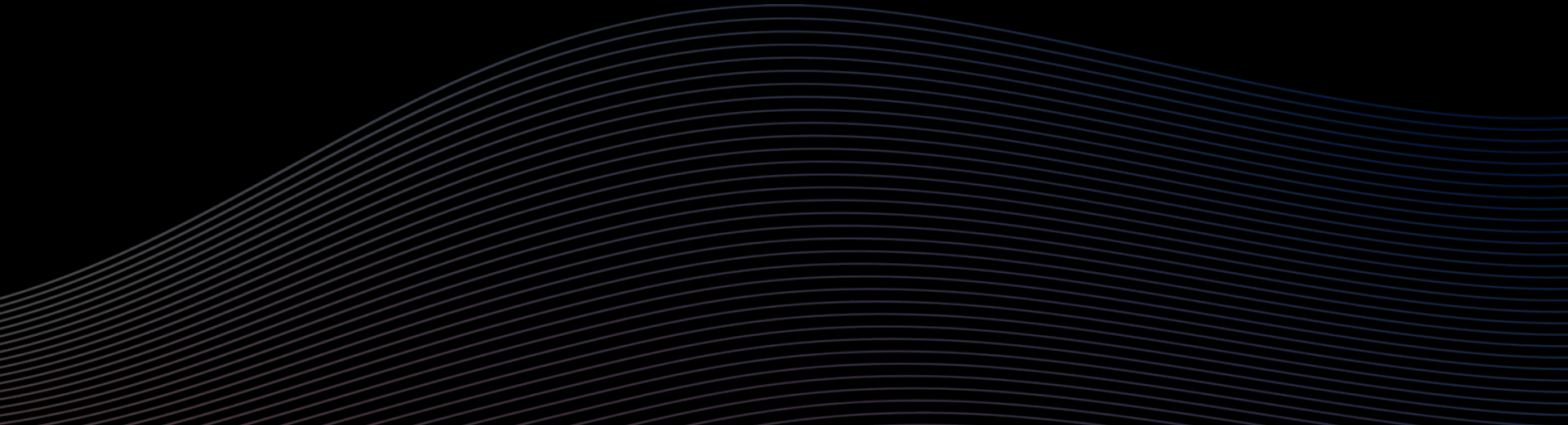
Tactical put strategy into action

Operational made daily

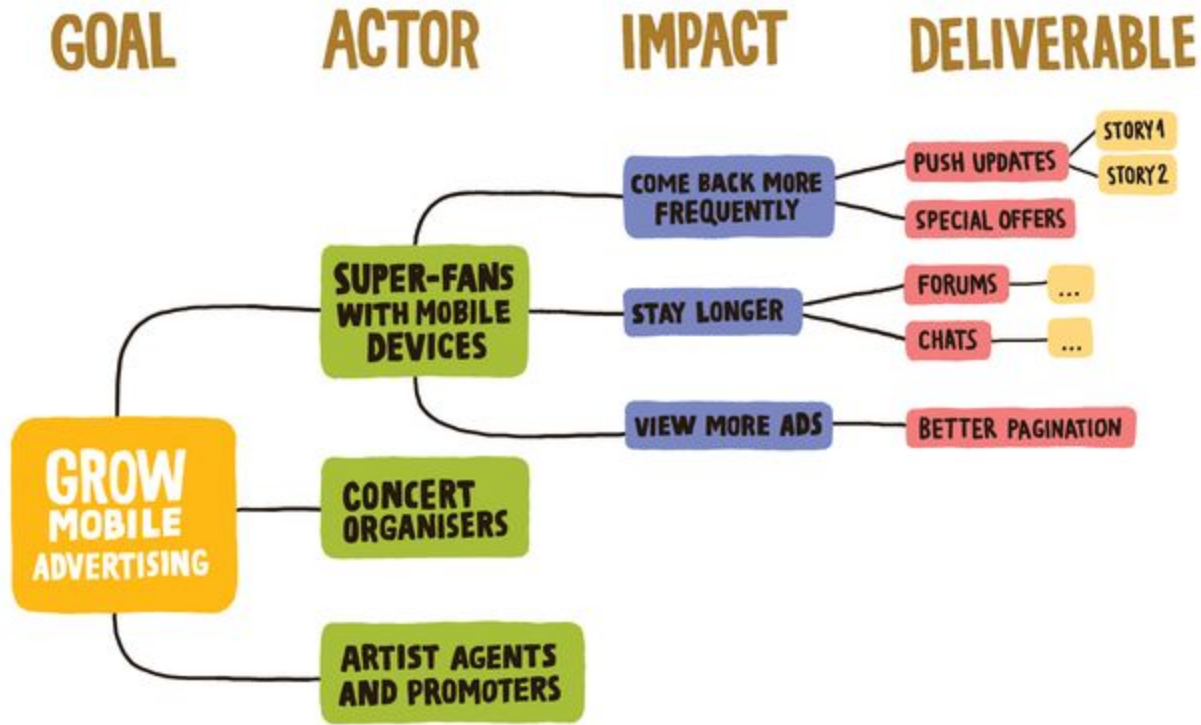
How to use TDD “mindset” for Tactical or Strategic Decisions?

—

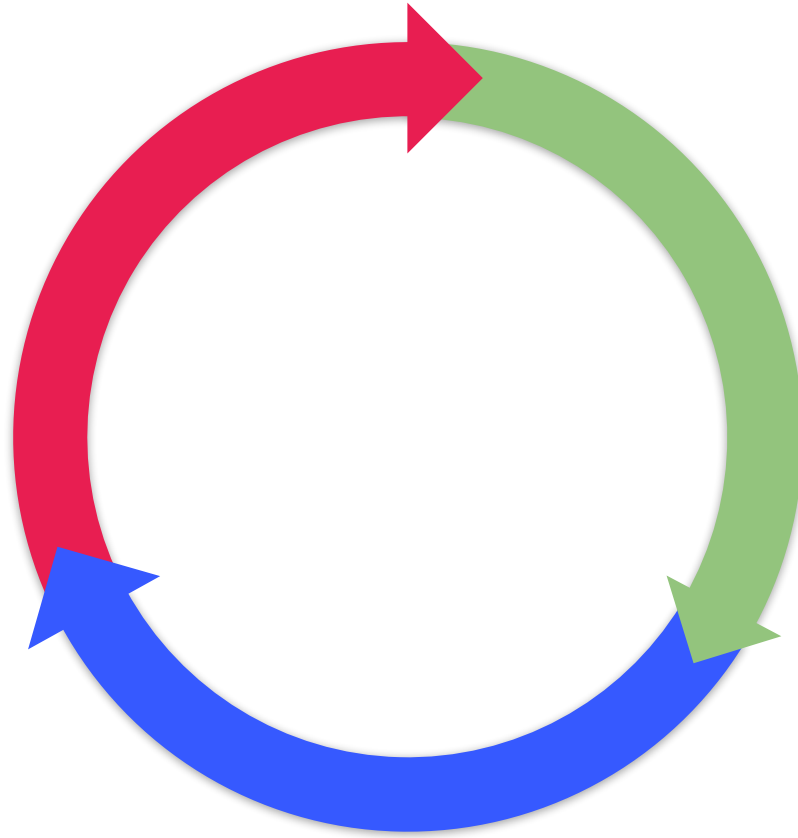
Choosing the next feature



Impact Mapping



Choose the impact
that you believe
will help the most
to reach the goal



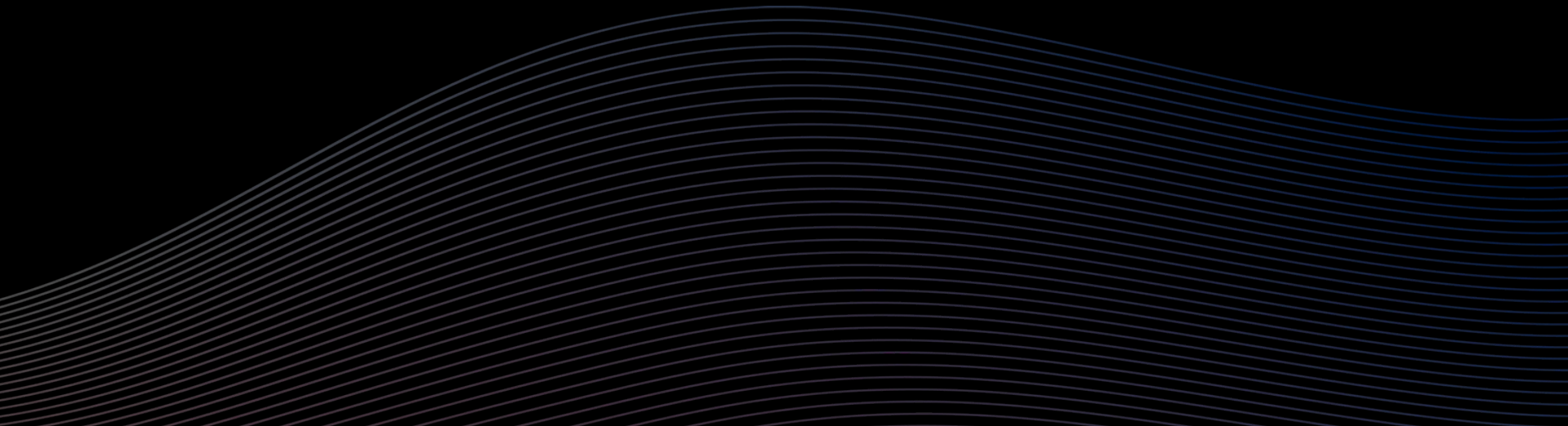
Build a small
deliverable

FLOWING®

Use Metrics to validate your assumptions

—

Software Architecture



What is the goal of Software Architecture?

—

Non-functional Requirements

FLOWING®

*“A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to **judge the operation of a system, rather than specific behaviors.**”*

Wikipedia

accessibility	accountability	accuracy	adaptability	reliability
administrability	affordability	agility	auditability	responsiveness
autonomy	availability	compatibility	composability	seamlessness
configurability	correctness	credibility	customizability	simplicity
debugability	self-sustainability	robustness	dependability	sustainability
deployability	determinability	discoverability	distributability	usability
durability	effectiveness	efficiency	extensibility	repeatability
fault tolerance	fidelity	flexibility	inspectability	reusability
reproducibility	integrity	interoperability	learnability	securability
maintainability	manageability	mobility	modifiability	stability
modularity	operability	evolvability	performances	tailorability
portability	precision	predictability	resilience	transparency
producibility	safety	recoverability	relevance	testability

What are the feedback for Software Architecture?

—

Fitness Functions

FLOWING[®]

What is a “small outcome” for Software Architecture?

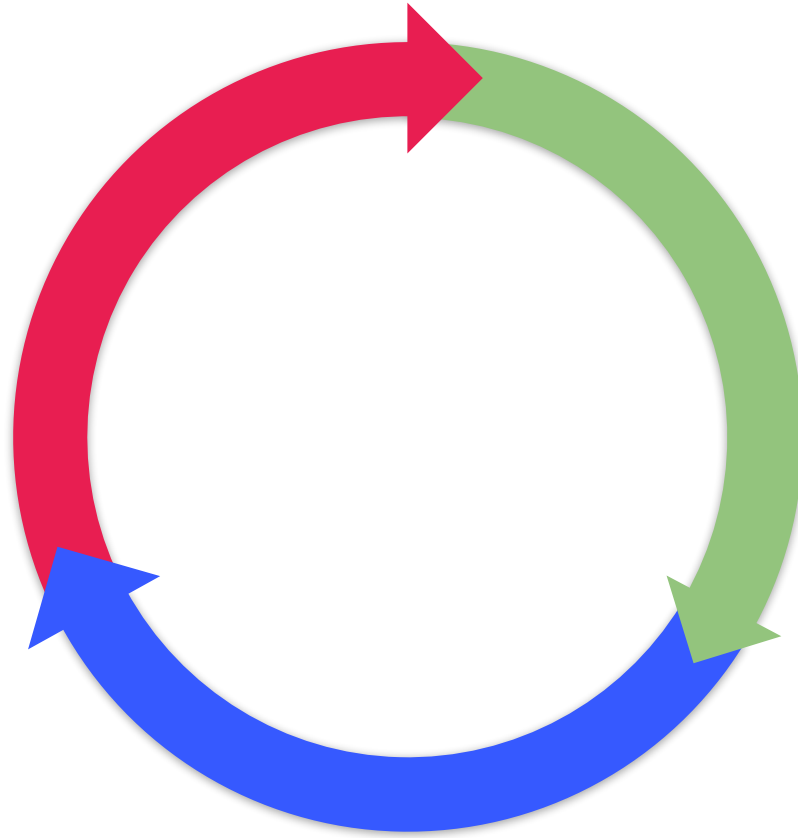
Walking Skeleton

FLOWING®

“A Walking Skeleton is a tiny implementation of the system that performs a small end-to-end function. It need not use the final architecture, but it should link together the main architectural components. The architecture and the functionality can then evolve in parallel.”

Alistair Cockburn

Define a set of
NFRs to check
with fitness
functions

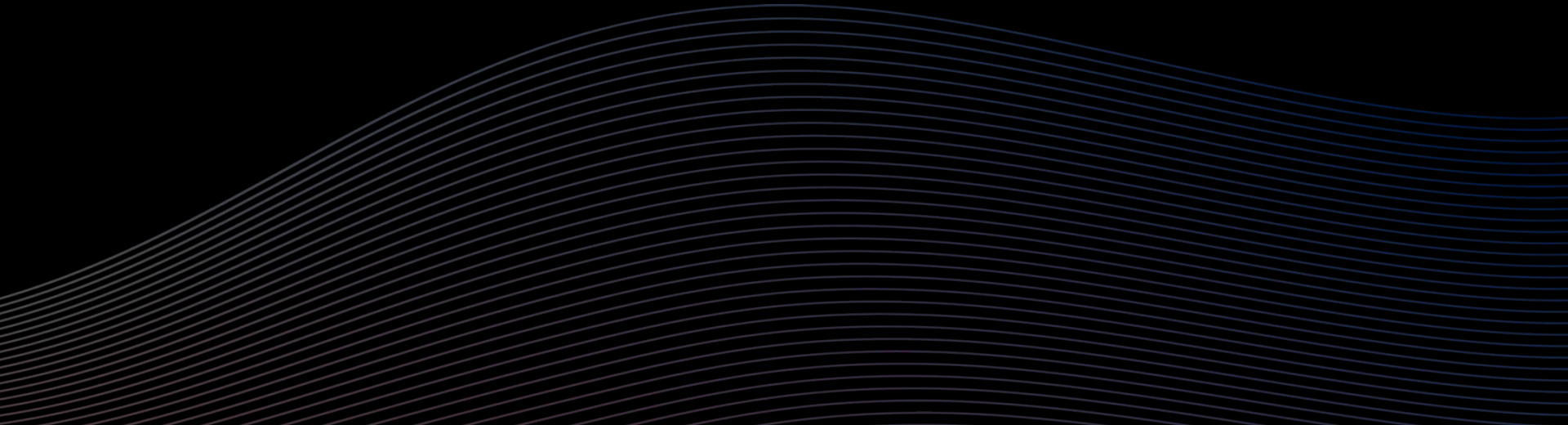


Build a “walking
skeleton”

FLOWING® Evolve architecture with fitness functions

—

**We should build this
software?**



**What kind of feedback do we
need?**

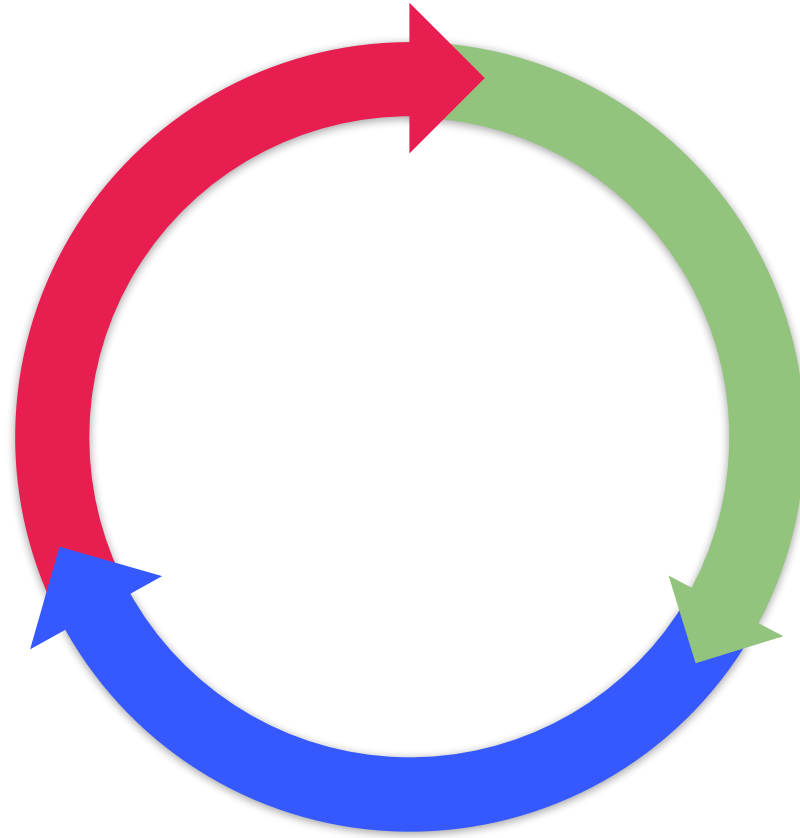
—

User Research

FLOWING®

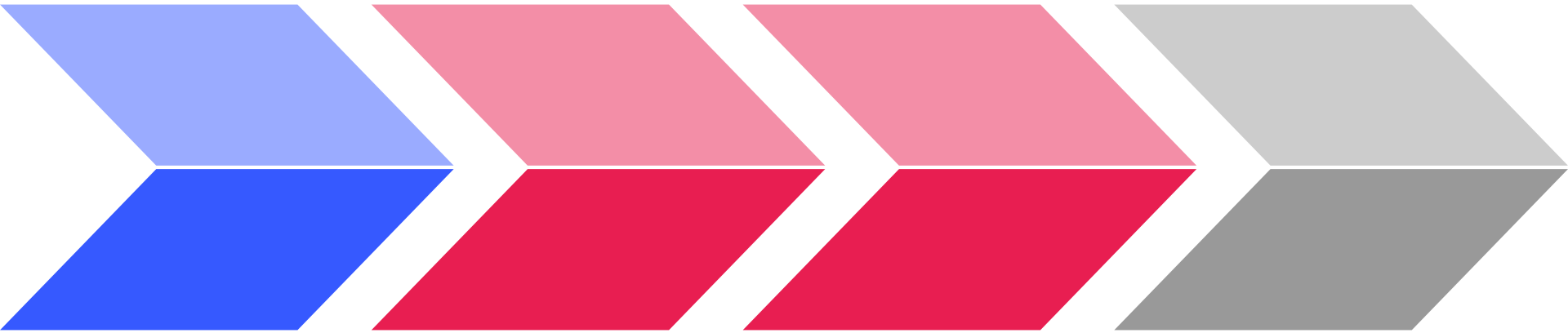
Define your idea

Small Outcome



FLOWING®

Use User Feedbacks to validate your idea



Strategic

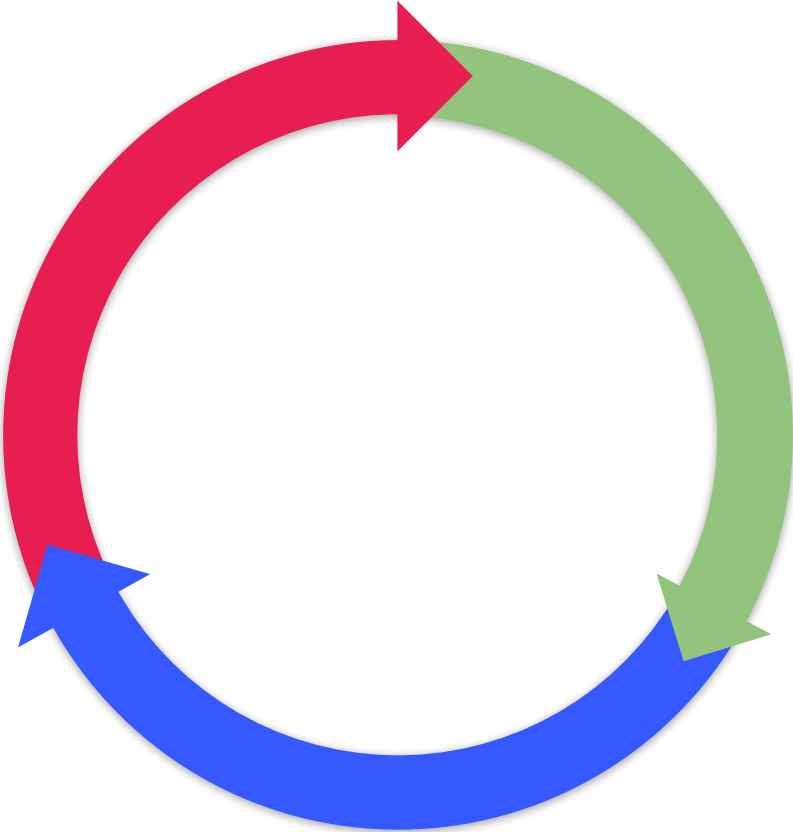
Tactical

Operational

FLOWING®

Goal

Small Outcome

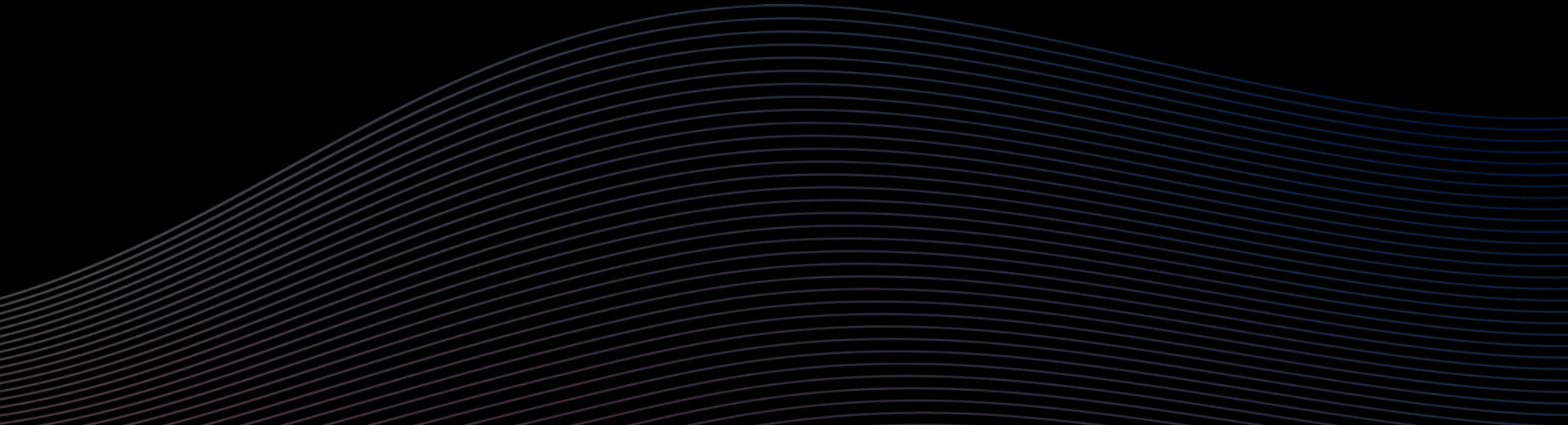


FLOWING®

Feedback

—

In a nutshell...



Learn the rules of the tool

Find the principles that are hidden in the tool

Explore interrelationships between tools

Thanks!

FLOWING[®]

Francesco Strazzullo

strazz@flowing.it