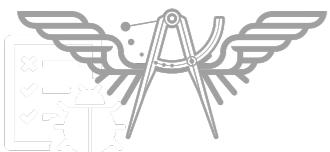




Development

**.NET C#
Test-Driven-Development
AI coding assistant**



Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Intro

Software Craftsmanship

Software Practitioner at สยามชัมนาณกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Somkiat Puisungnoen 15 mins · Bangkok · ⚙️

Java and Bigdata



Page

Messages

Notifications 3

Insights

Publishing Tools

Settings

Help ▾



somkiat.cc

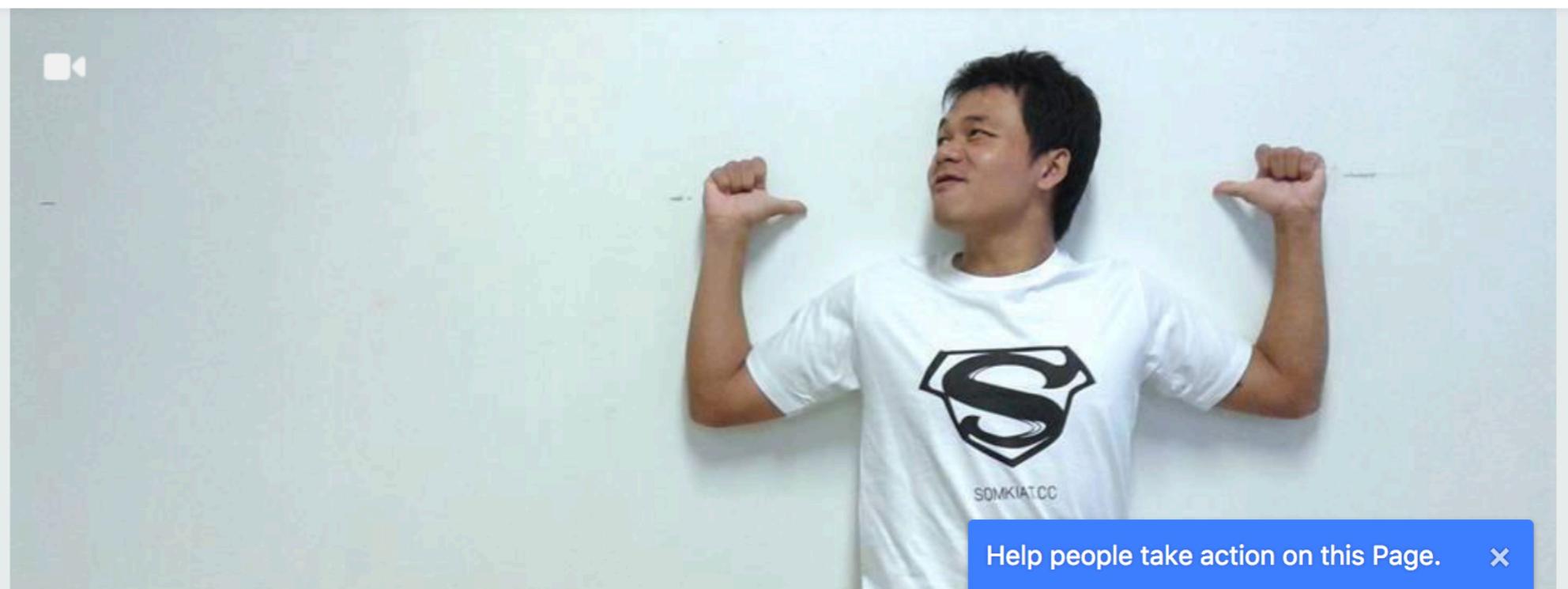
@somkiat.cc

Home

Posts

Videos

Photos



**[https://github.com/up1/
course-dotnet-tdd](https://github.com/up1/course-dotnet-tdd)**



Section 1

1. Test-Driven Development (TDD)
2. Working with .NET 8 C#
3. Write test case with xUnit
4. Dependency management
5. Test doubles
6. Workshop



Section 2

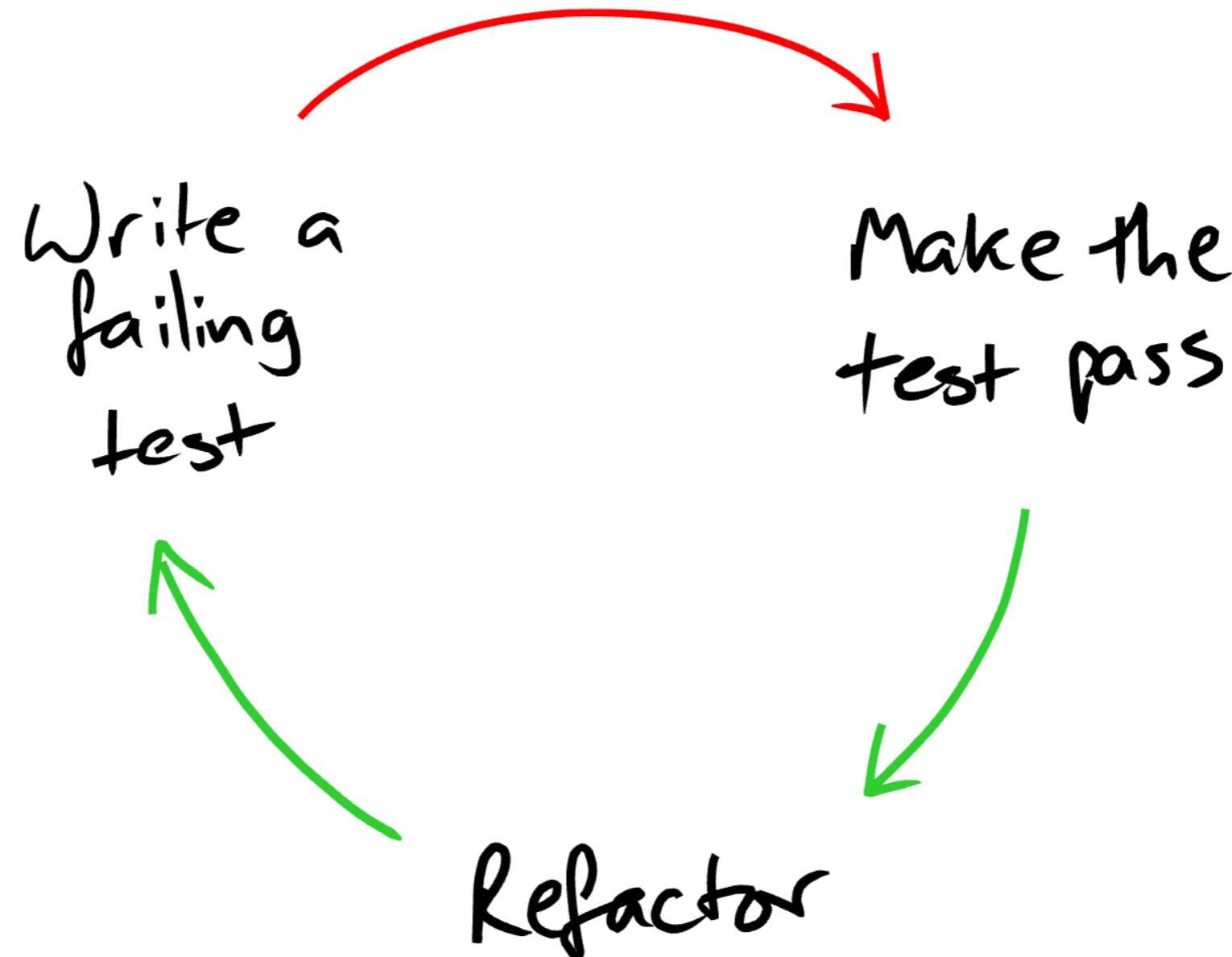
1. Integrate AI with coding and TDD
2. Development workflow with Generative AI
3. Working with Github Copilot Chat in VSCode
4. Workshop



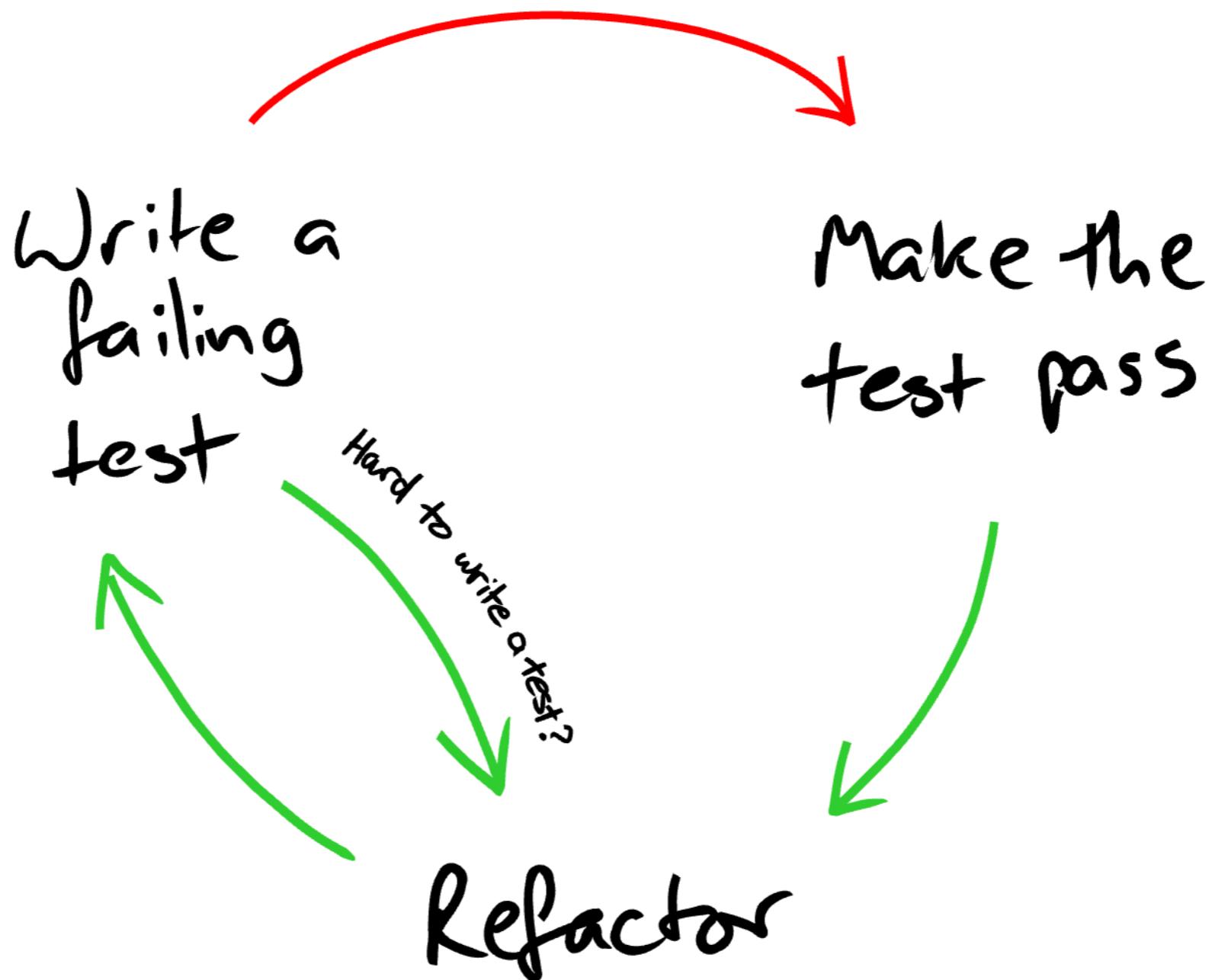
Test-Driven Development



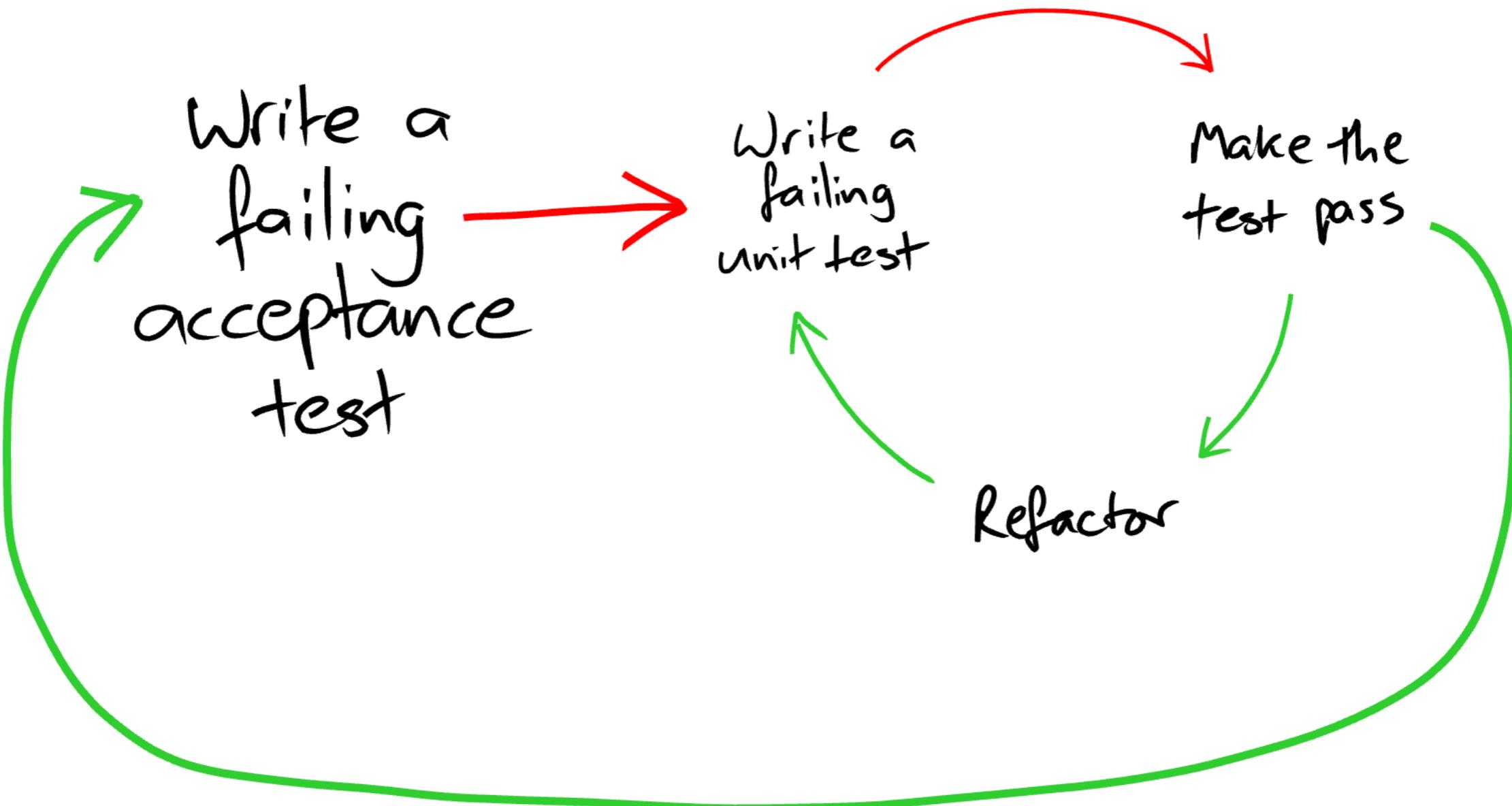
Simple TDD Cycle



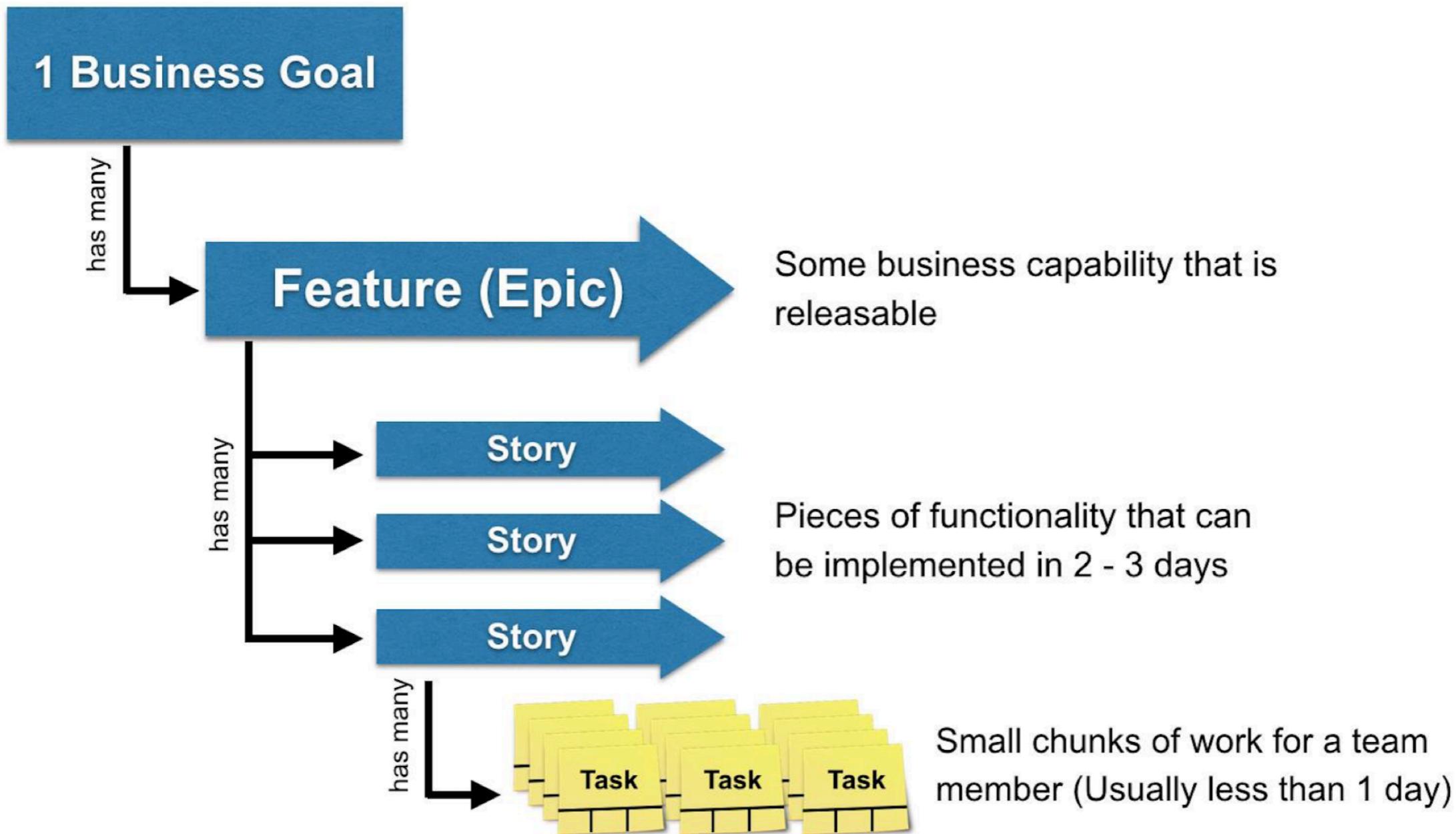
Listen to the test



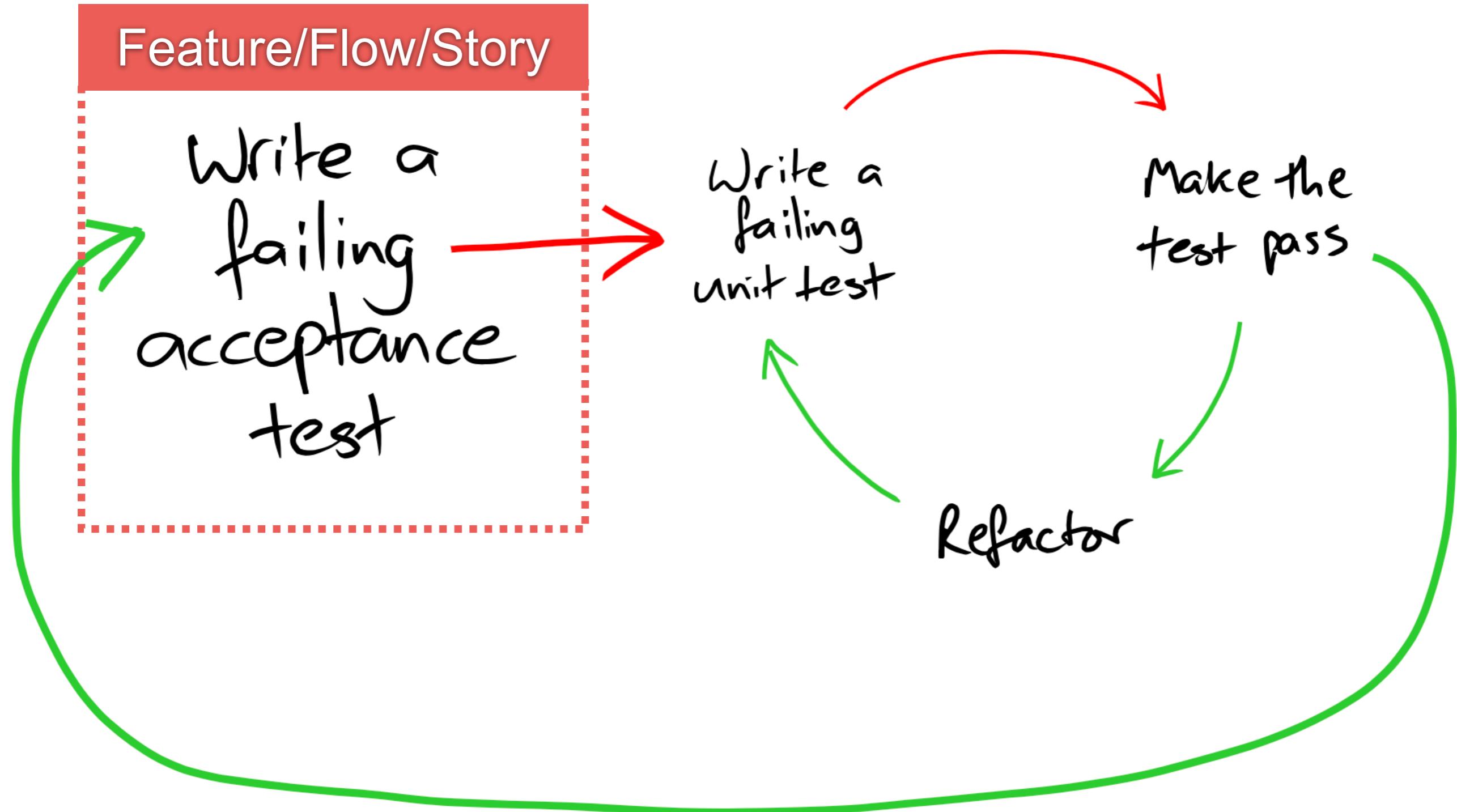
TDD with Acceptance test



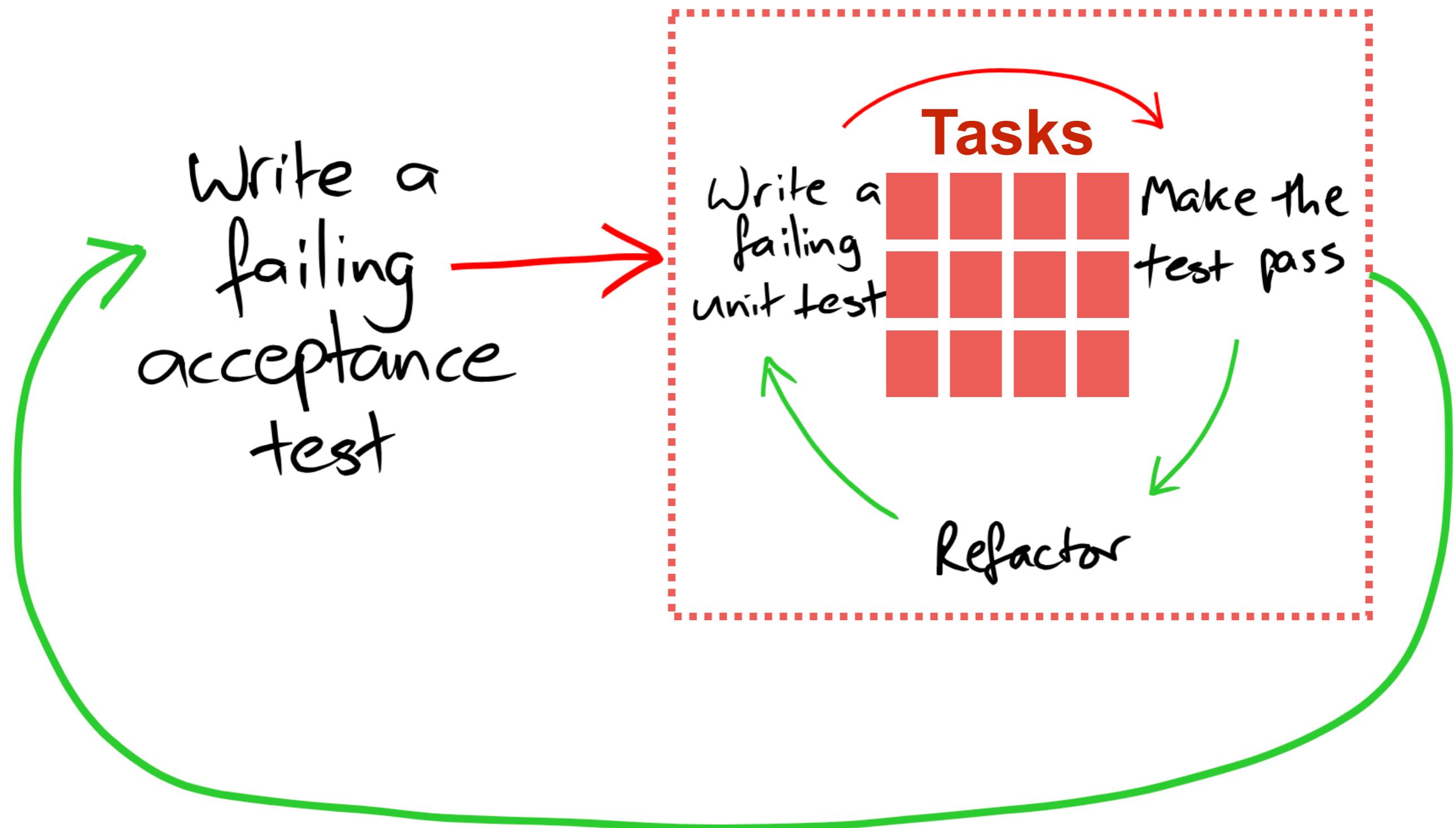
Work Break Down



Outside-in develop/test



Outside-in develop/test



Good Test ?



Good Test

F.I.R.S.T + U

Fast

Isolate

Repeat

Self-verify

Timely

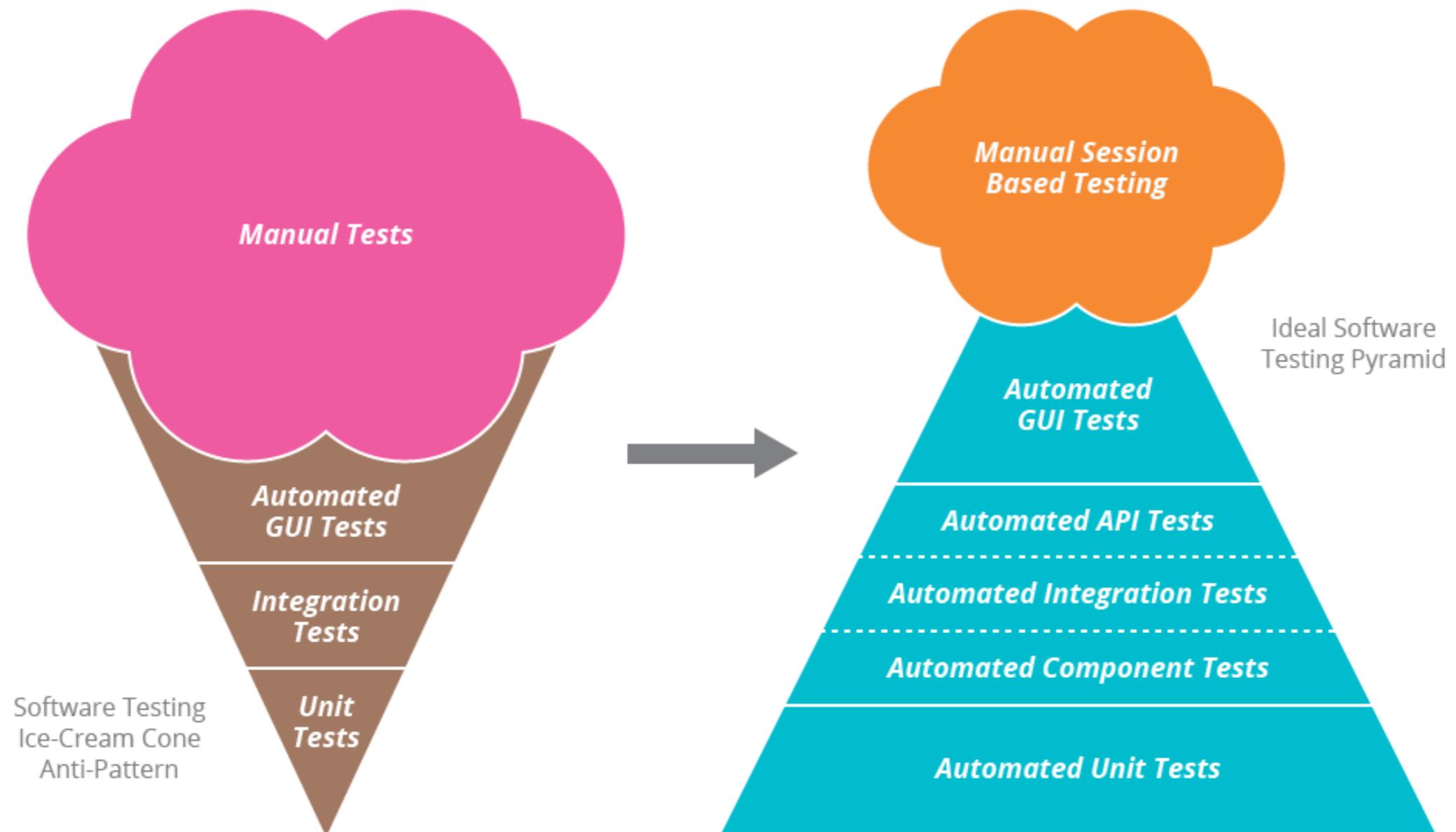
Understand



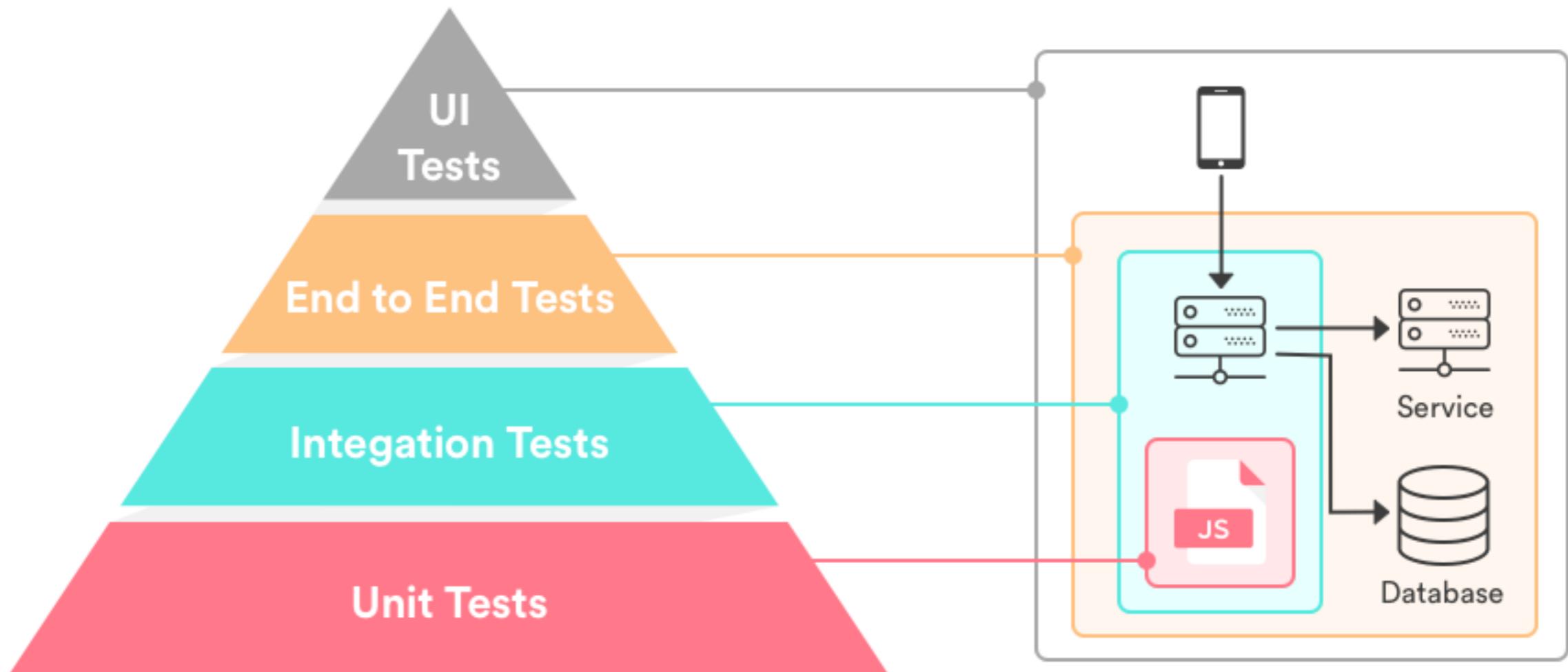
Test Strategies ?



Pyramid Testing ?



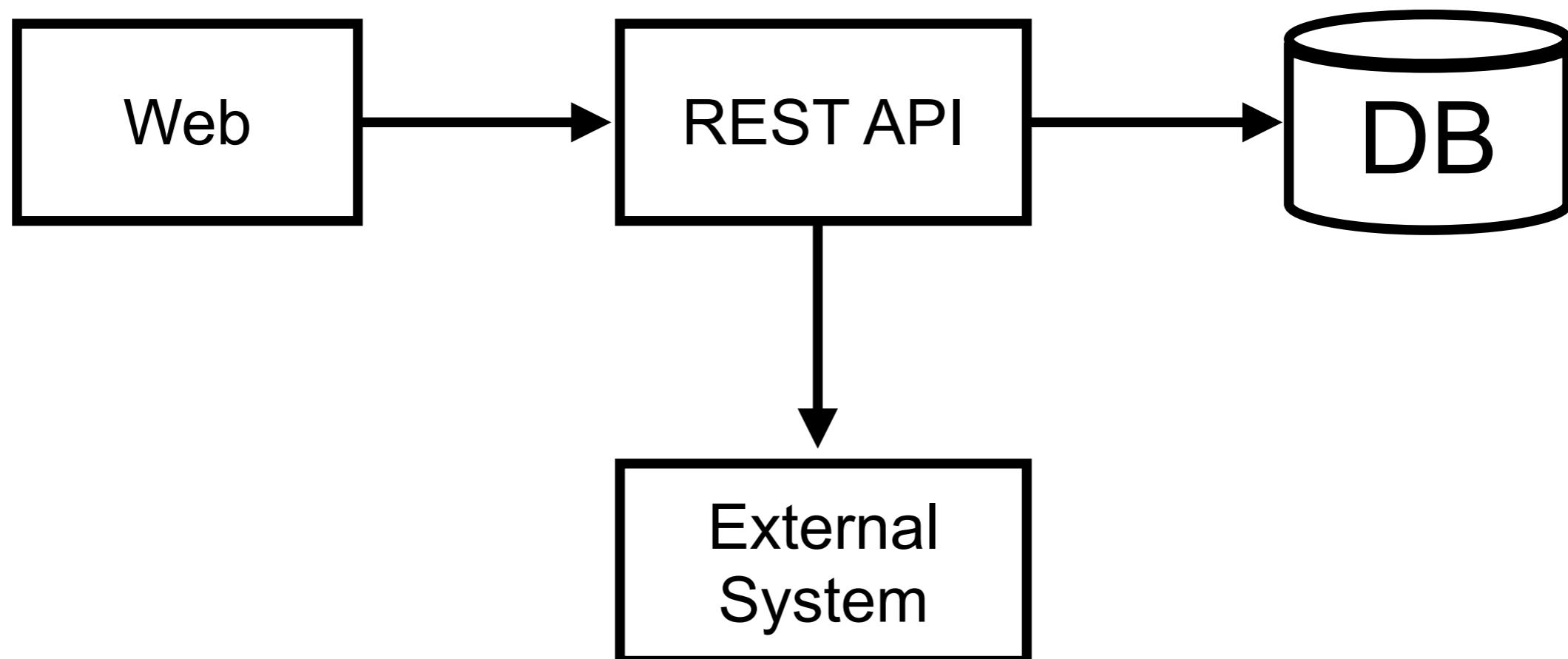
Pyramid Testing ?



Start with Architecture



Simple Architecture



How to test ?



Feature Register

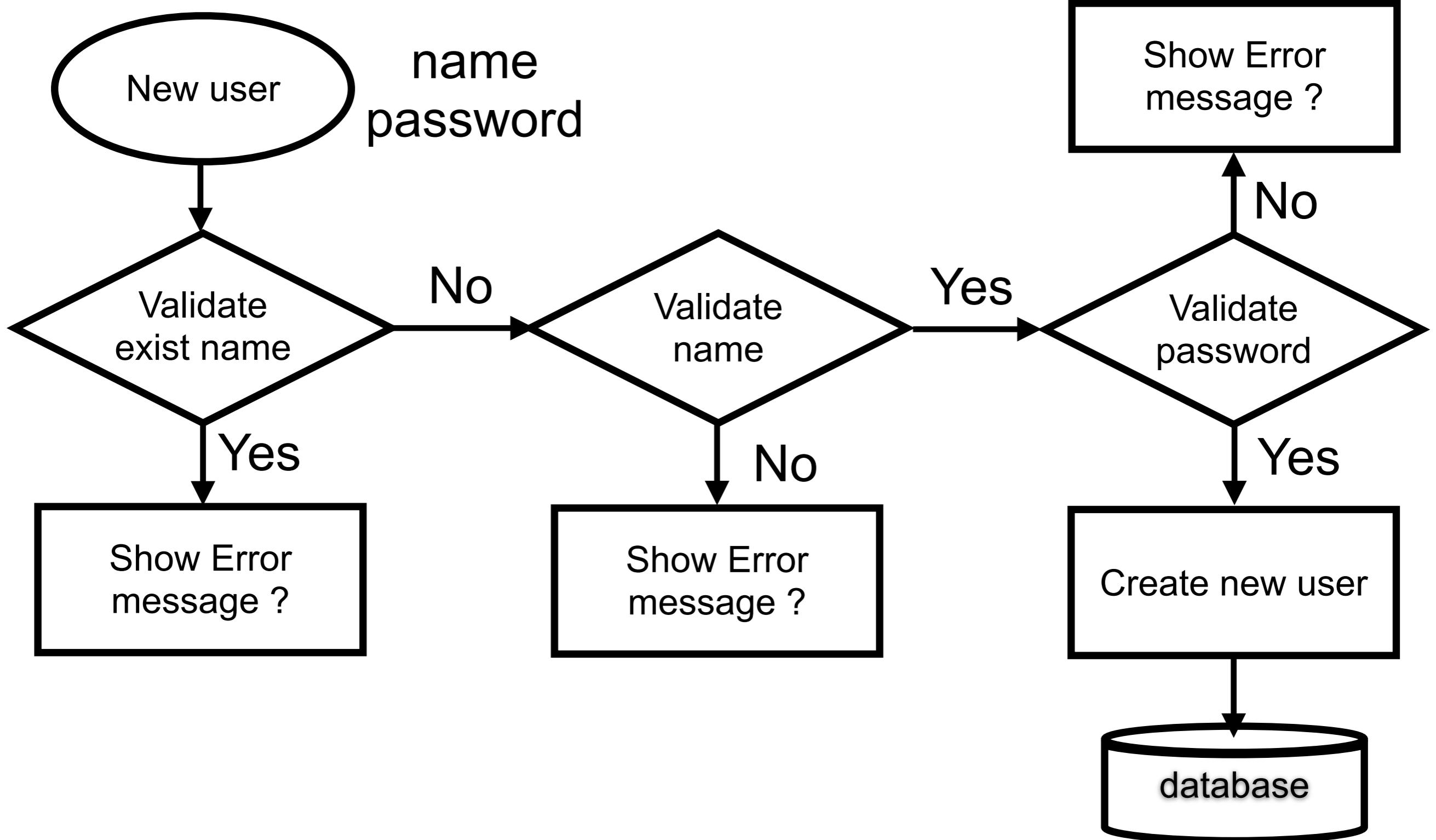
As a new user,

**I want to create an account with
User name and password**

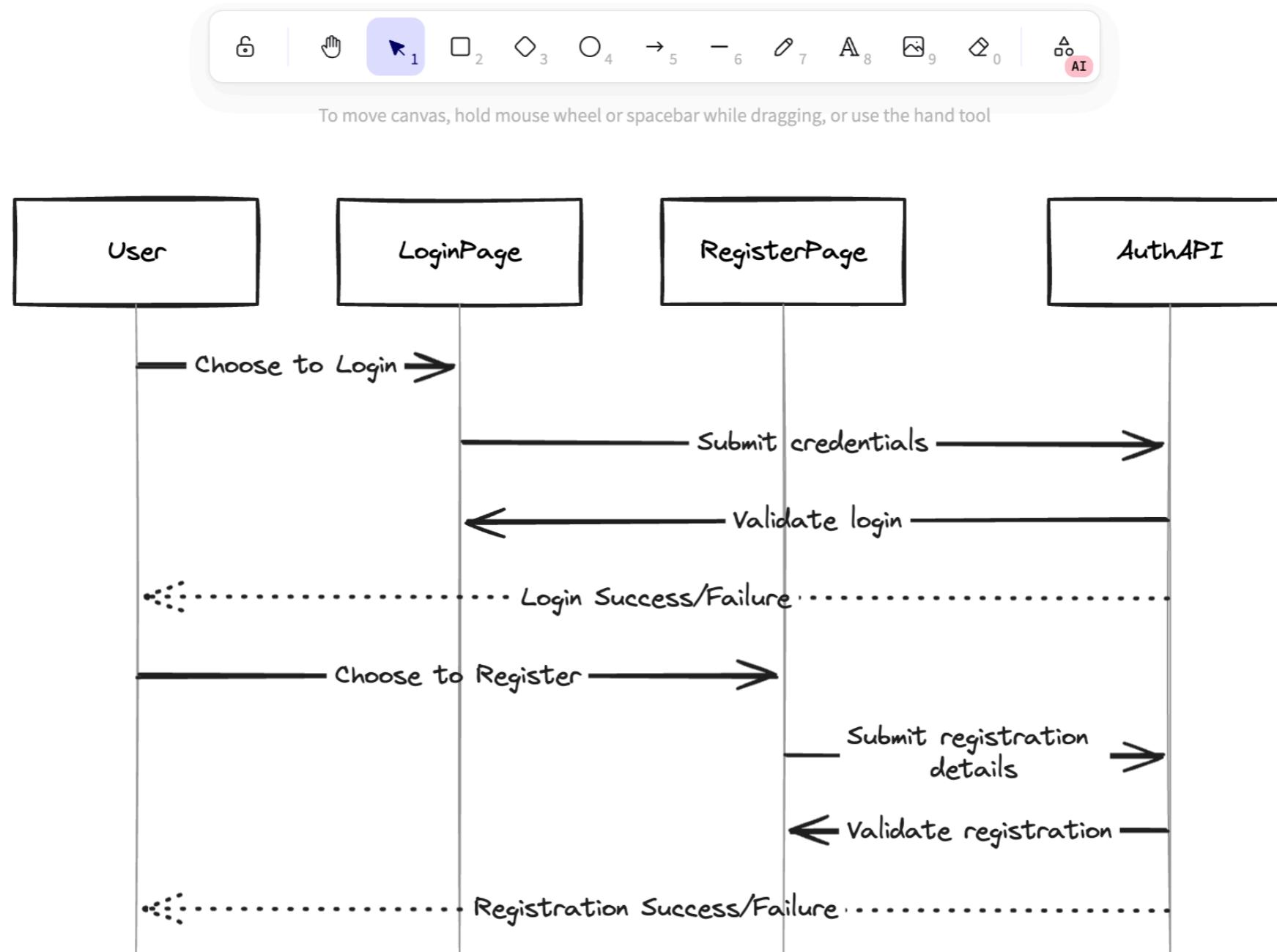
So that only I can access my information



Flow ?



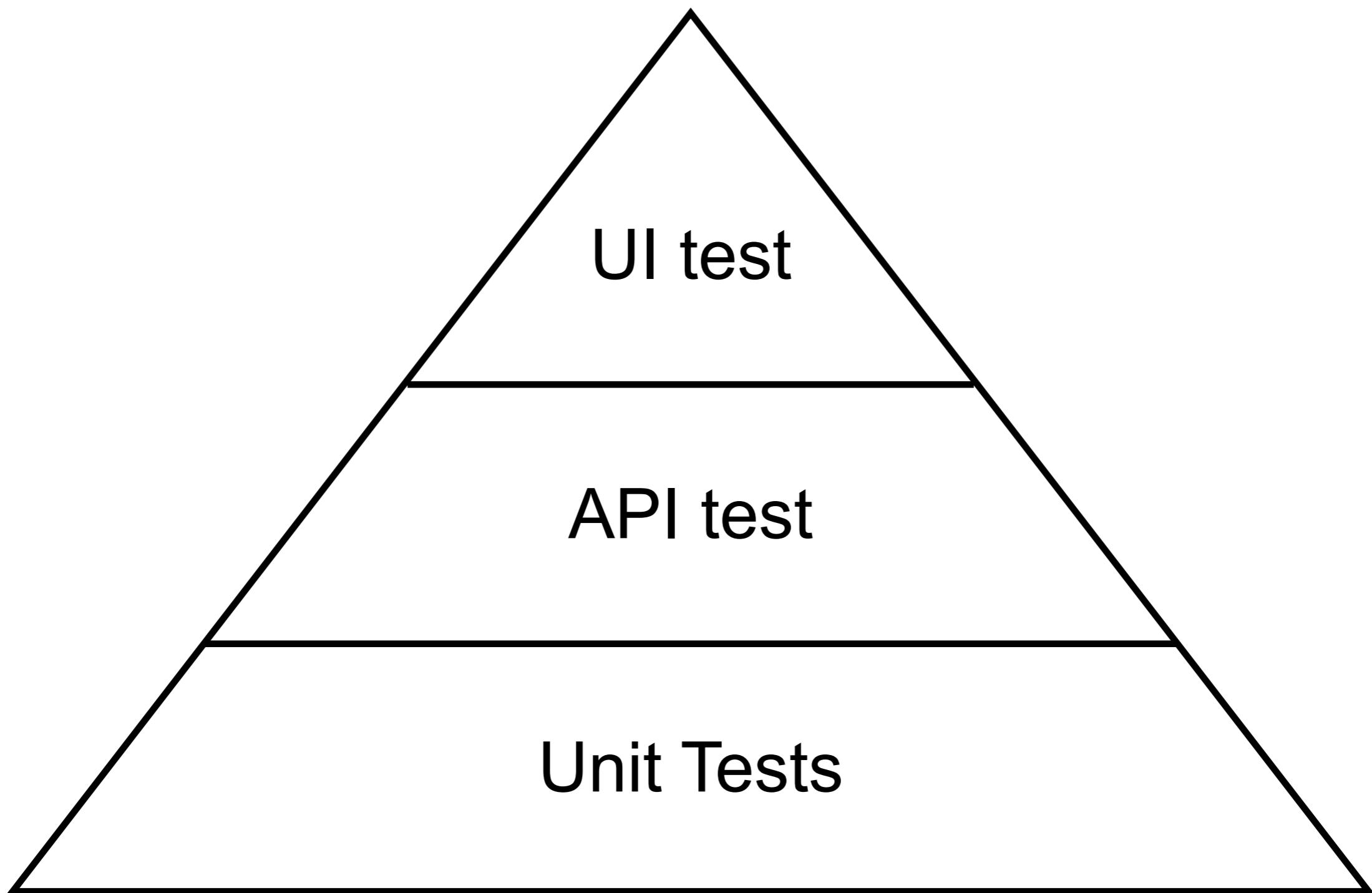
Generate Flow with AI



<https://excalidraw.com/>



Test cases with Test Level ?



Testing Tools

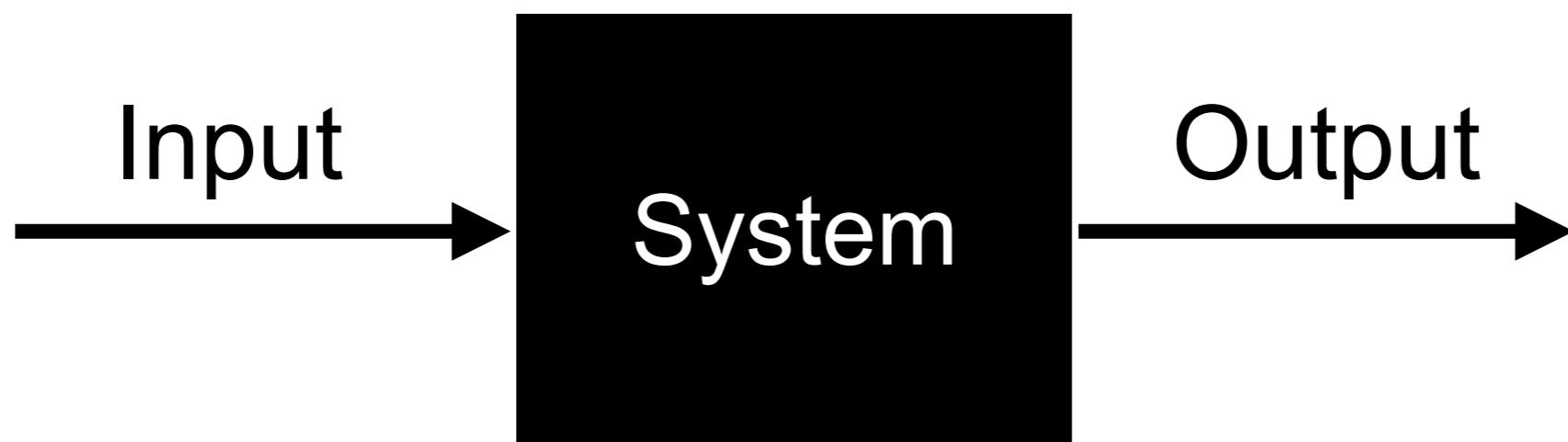
External
testing

Internal
testing



External Testing

Black box testing

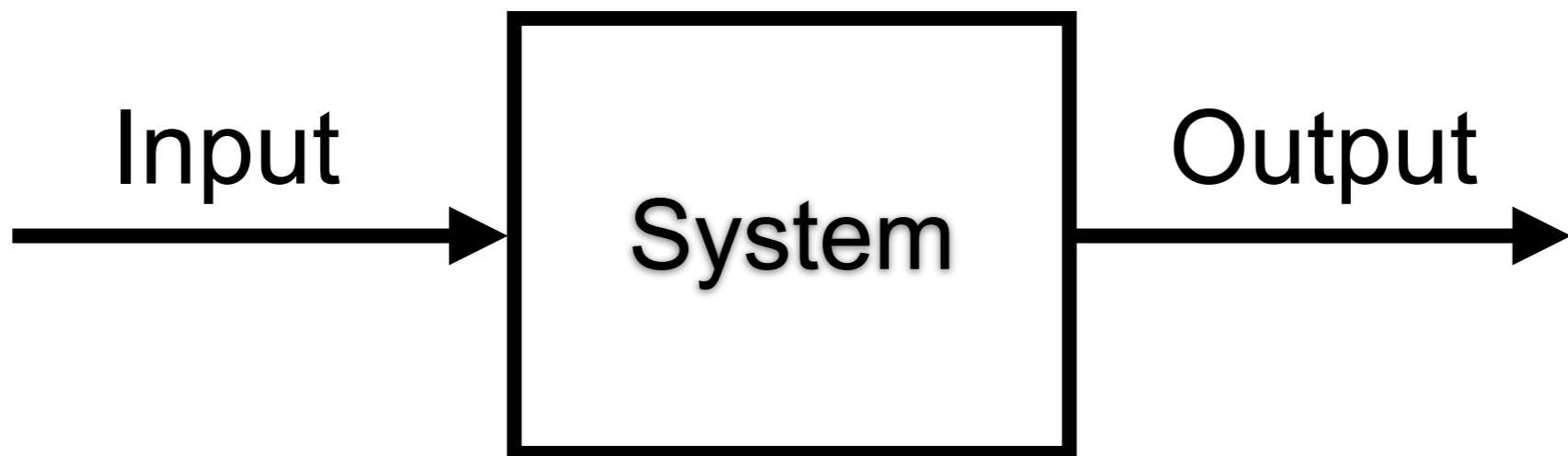


End-to-end testing



Internal Testing

White box testing



**Unit, Integration,
Contract, component testing**

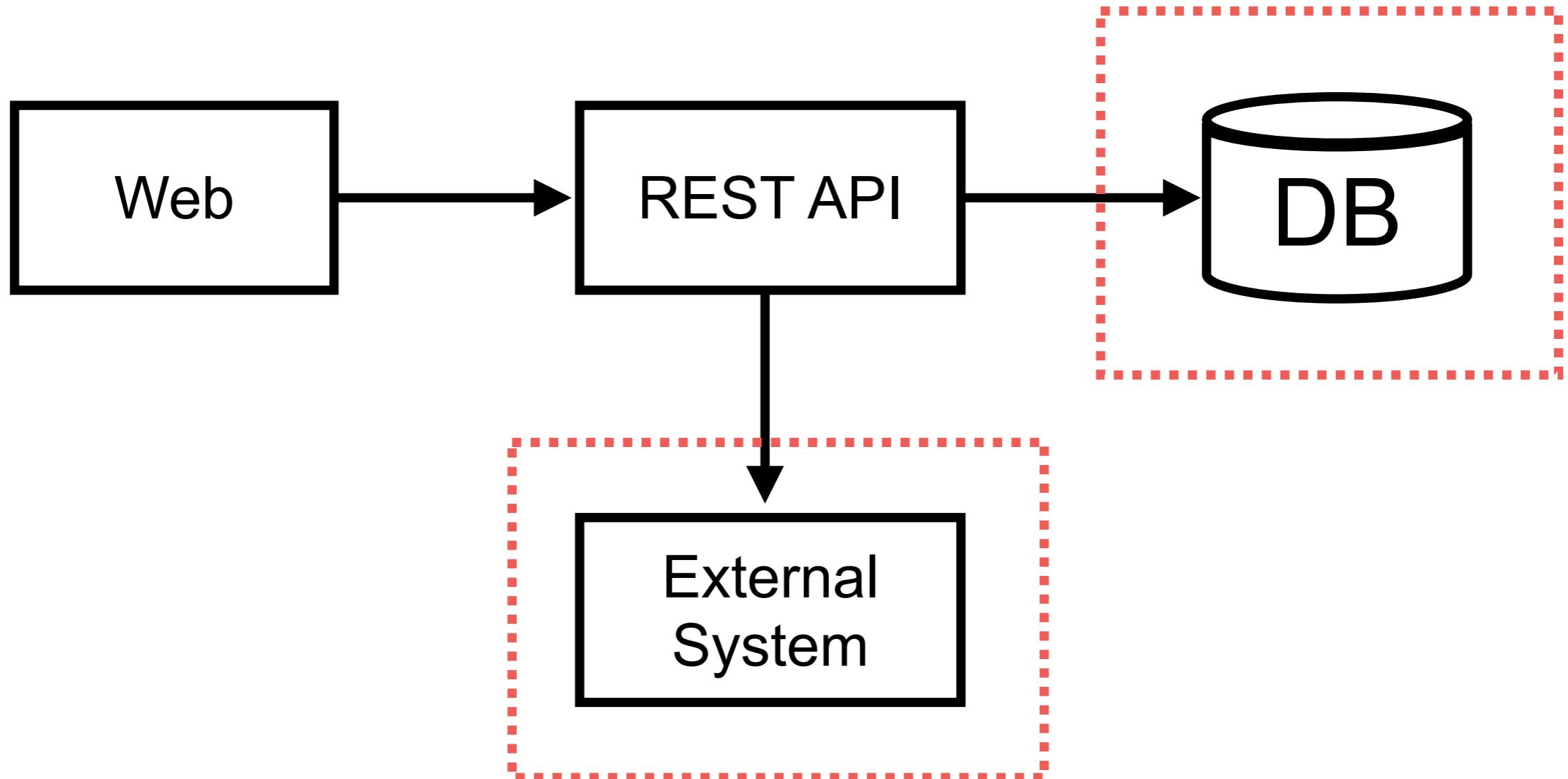




Manage Dependencies ?



Manage Dependencies ?



Test Double

Dummy

Stub

Spy

Mock

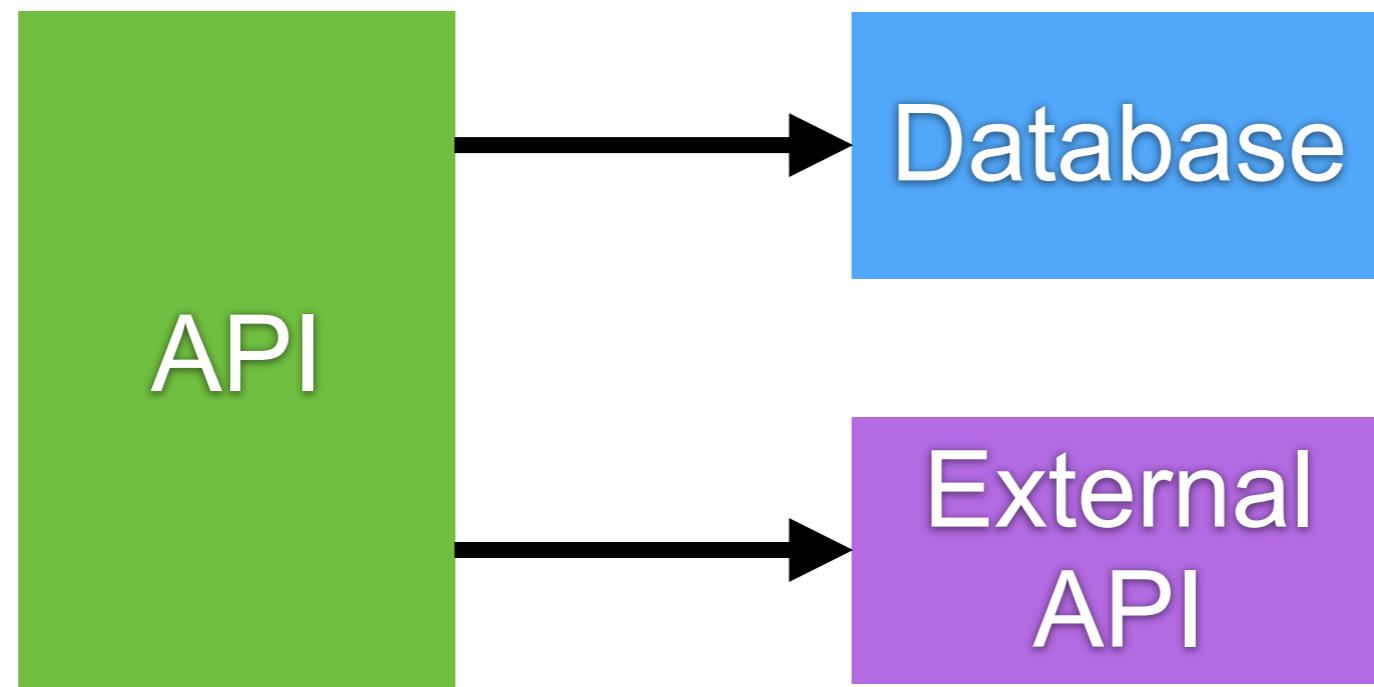
Fake

<http://xunitpatterns.com/Test%20Double.html>



Manage dependencies of API

Database
External API



Database

In-memory database
Container-based
Real database

Tables

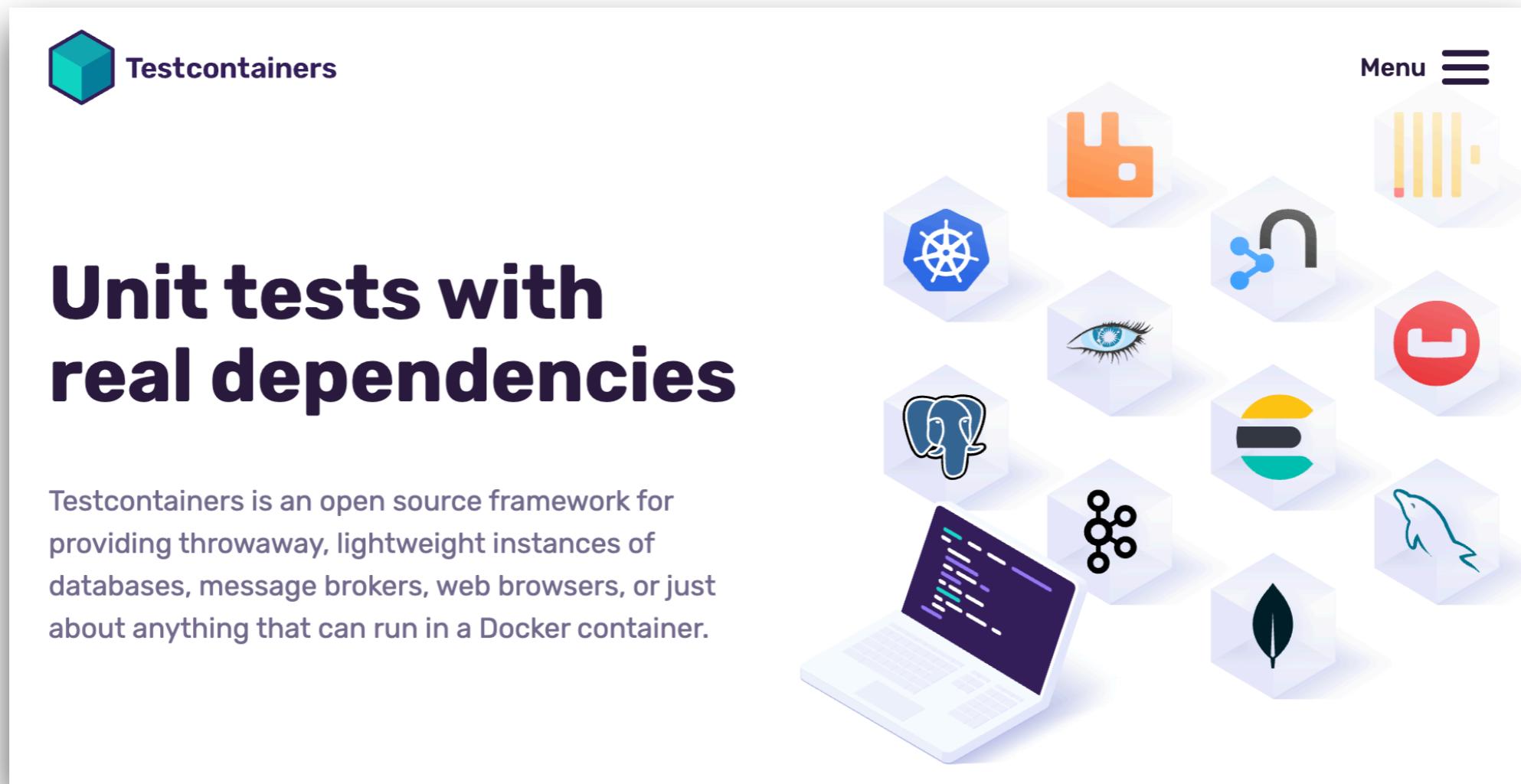
Data for test

Database migration tools



Test Container

Testing with real dependencies



The screenshot shows the official website for Testcontainers. At the top left is the Testcontainers logo (a teal cube icon) and the word "Testcontainers". At the top right is a "Menu" button with three horizontal lines. The main heading "Unit tests with real dependencies" is prominently displayed in large, bold, dark blue text. Below the heading is a paragraph of text: "Testcontainers is an open source framework for providing throwaway, lightweight instances of databases, message brokers, web browsers, or just about anything that can run in a Docker container." To the right of the text is a central laptop screen showing a grid of various dependency icons in hexagonal containers, including a blue eye, an orange padlock, a red terminal, a green elephant, a blue ship's wheel, a yellow bar chart, a black network graph, a blue cursor, and a green leaf.

<https://testcontainers.com/>



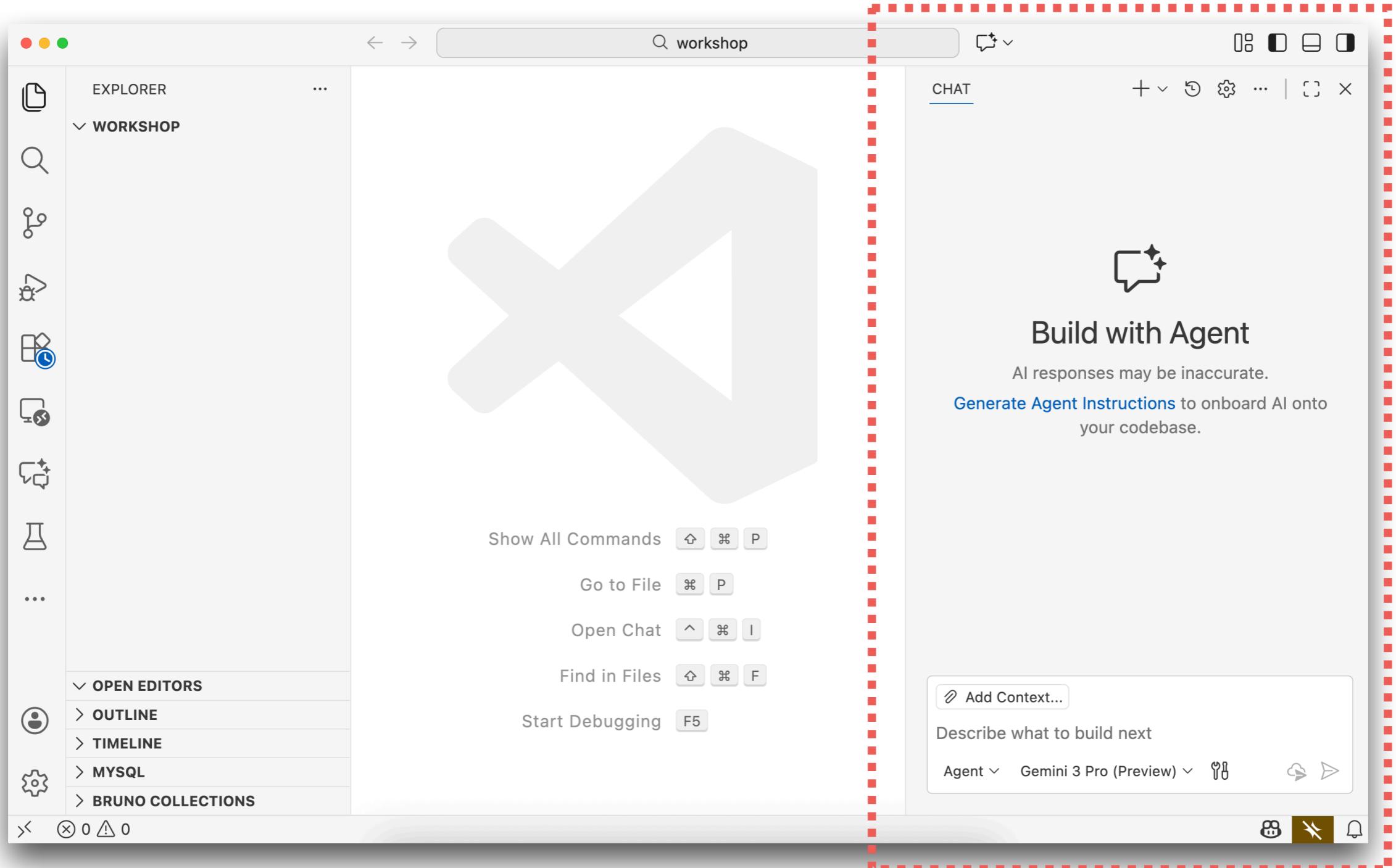
Ai-Coding Assistant



GitHub
Copilot



GitHub Copilot Chat in VSCode



<https://code.visualstudio.com/docs/copilot/chat/getting-started-chat>



Development

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Build-in Chat Modes

Types of interactions and tasks within the development workflow

Agent

Plan

Ask

Edit

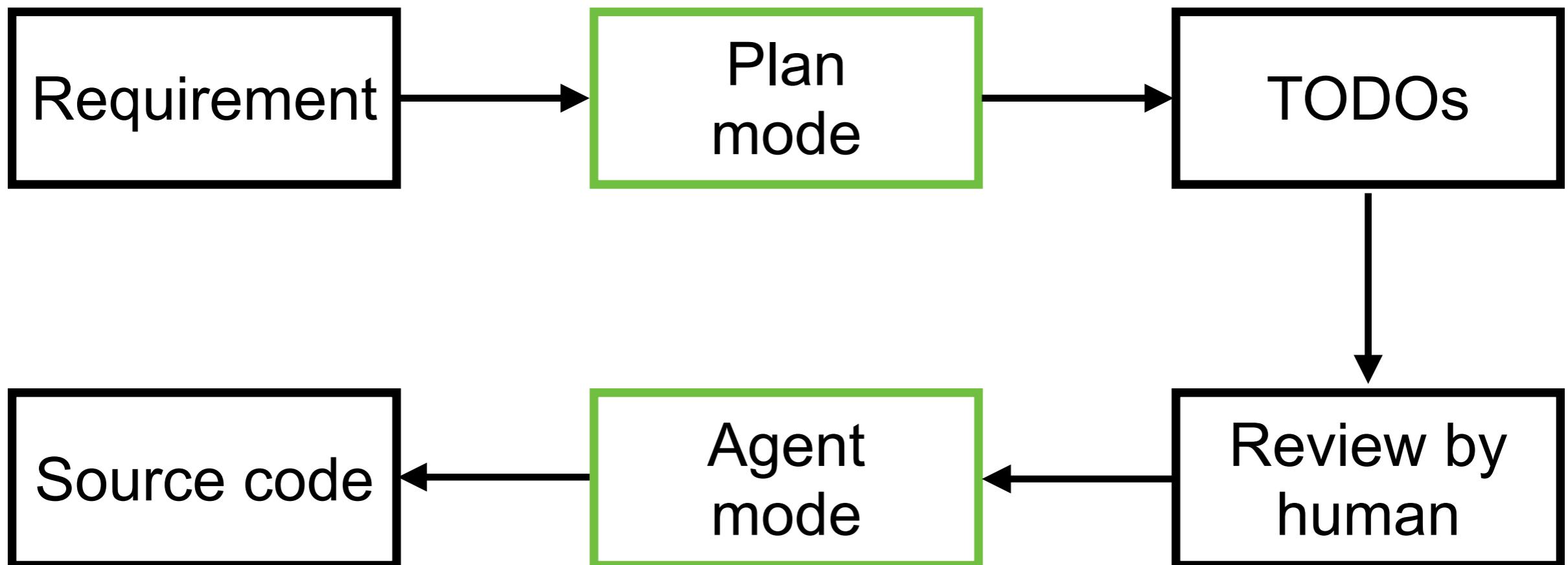
Custom ..

<https://code.visualstudio.com/docs/copilot/chat/chat-planning>

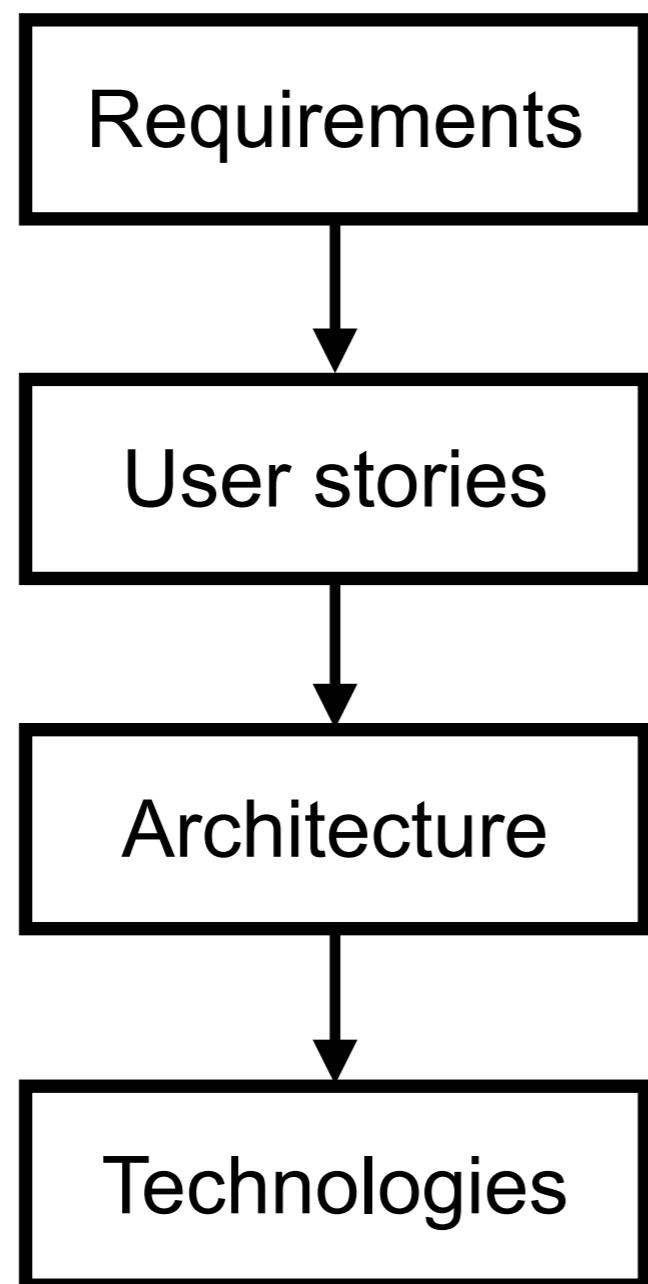


Better Process !!

Improve quality of source code from AI



Planning workflow



What you're build ?

What use need ?

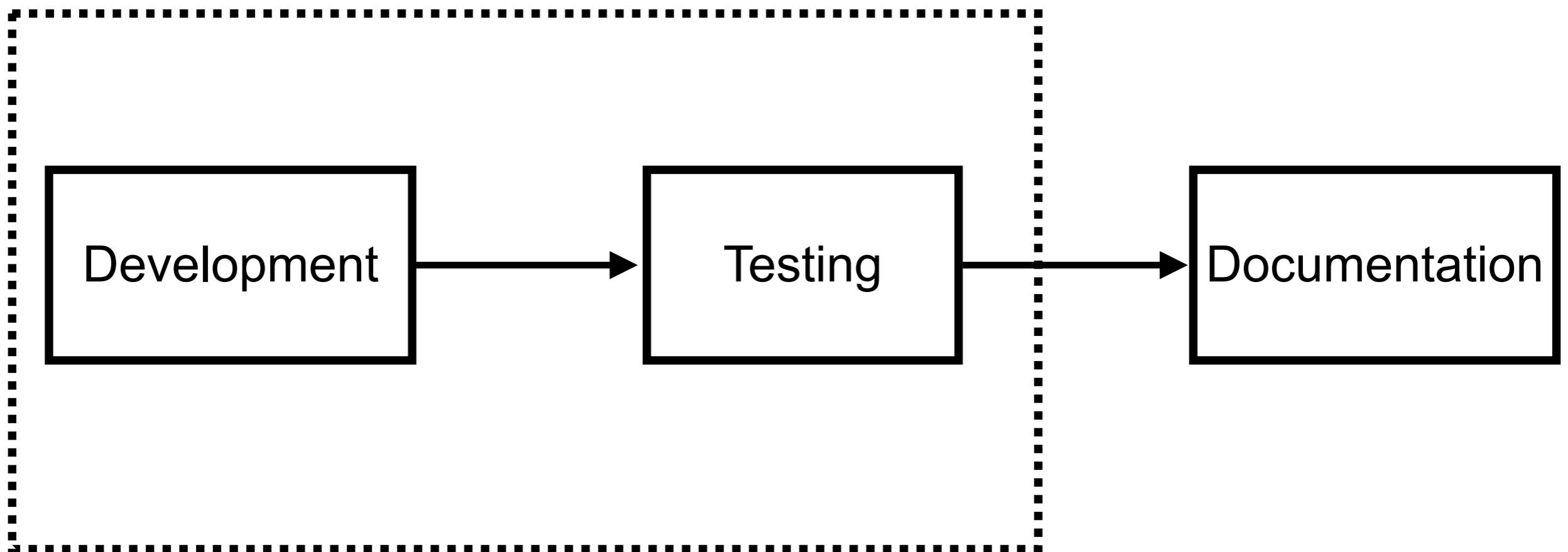
Selection ?

Selection ?

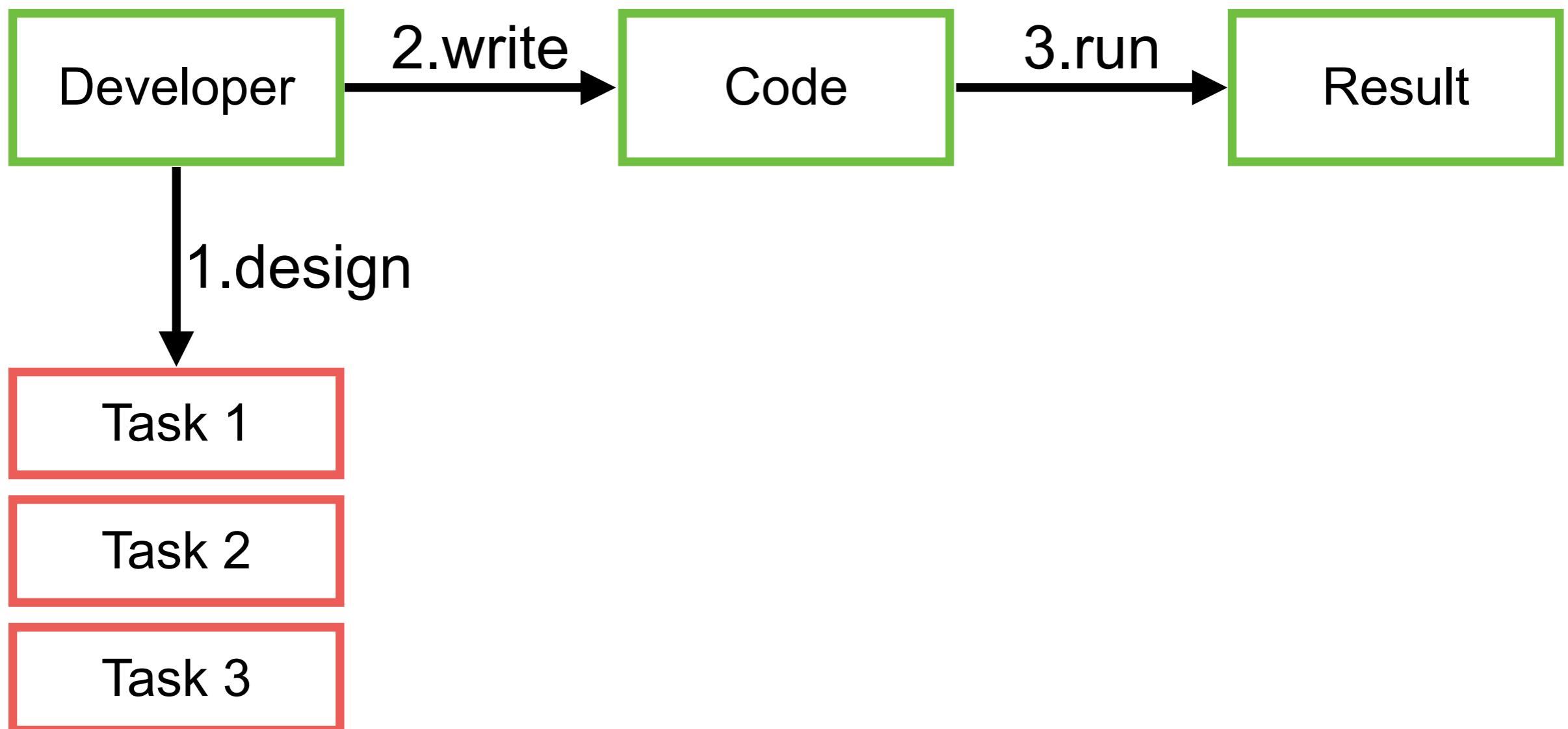


Processes

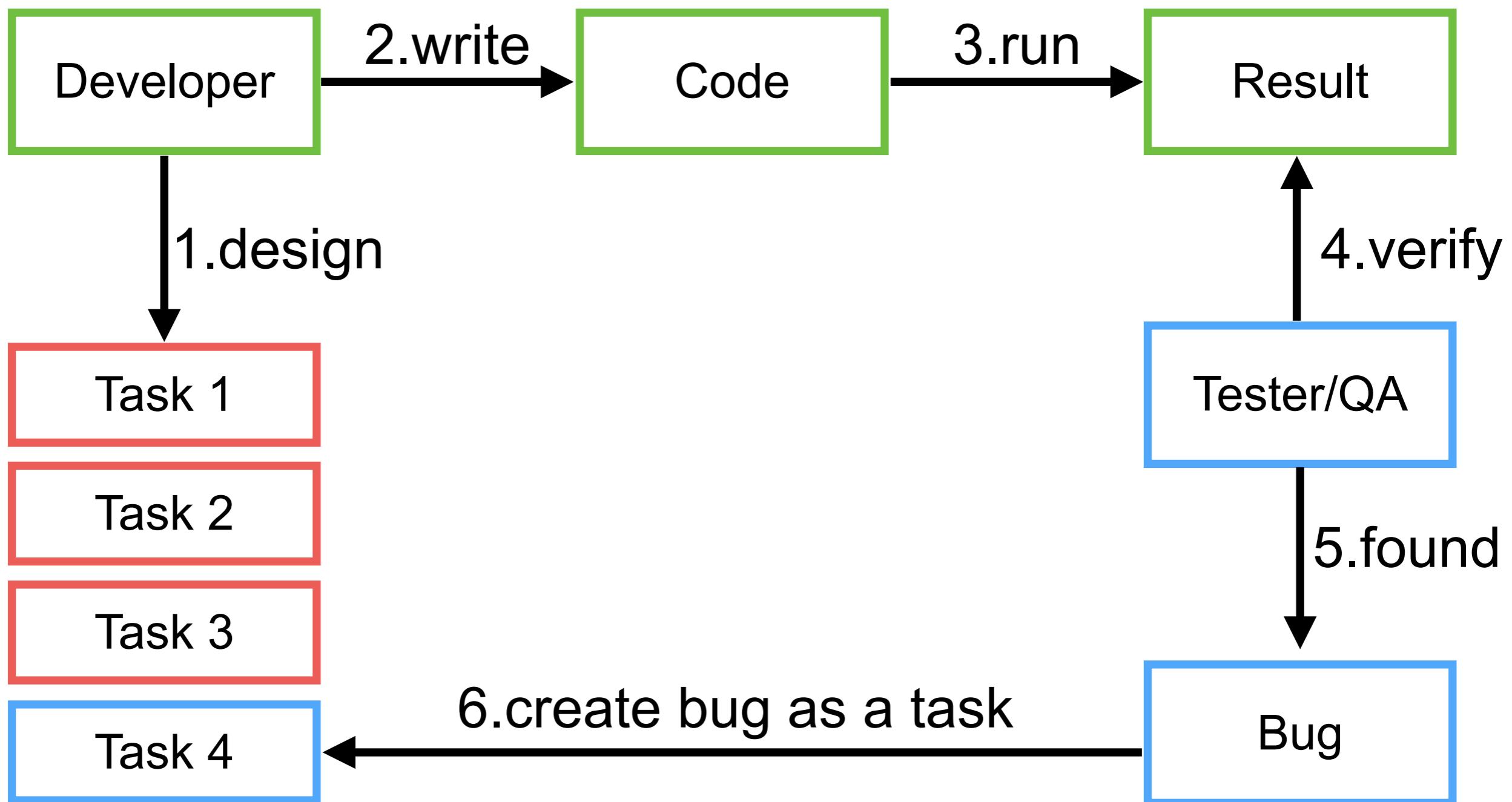
Iterative and incremental process



Development



Development + Testing



Test-Driven-Generation (TDG)

Development practice that integrate Generative AI into the development life-cycle

TDD

+

Pair
programming

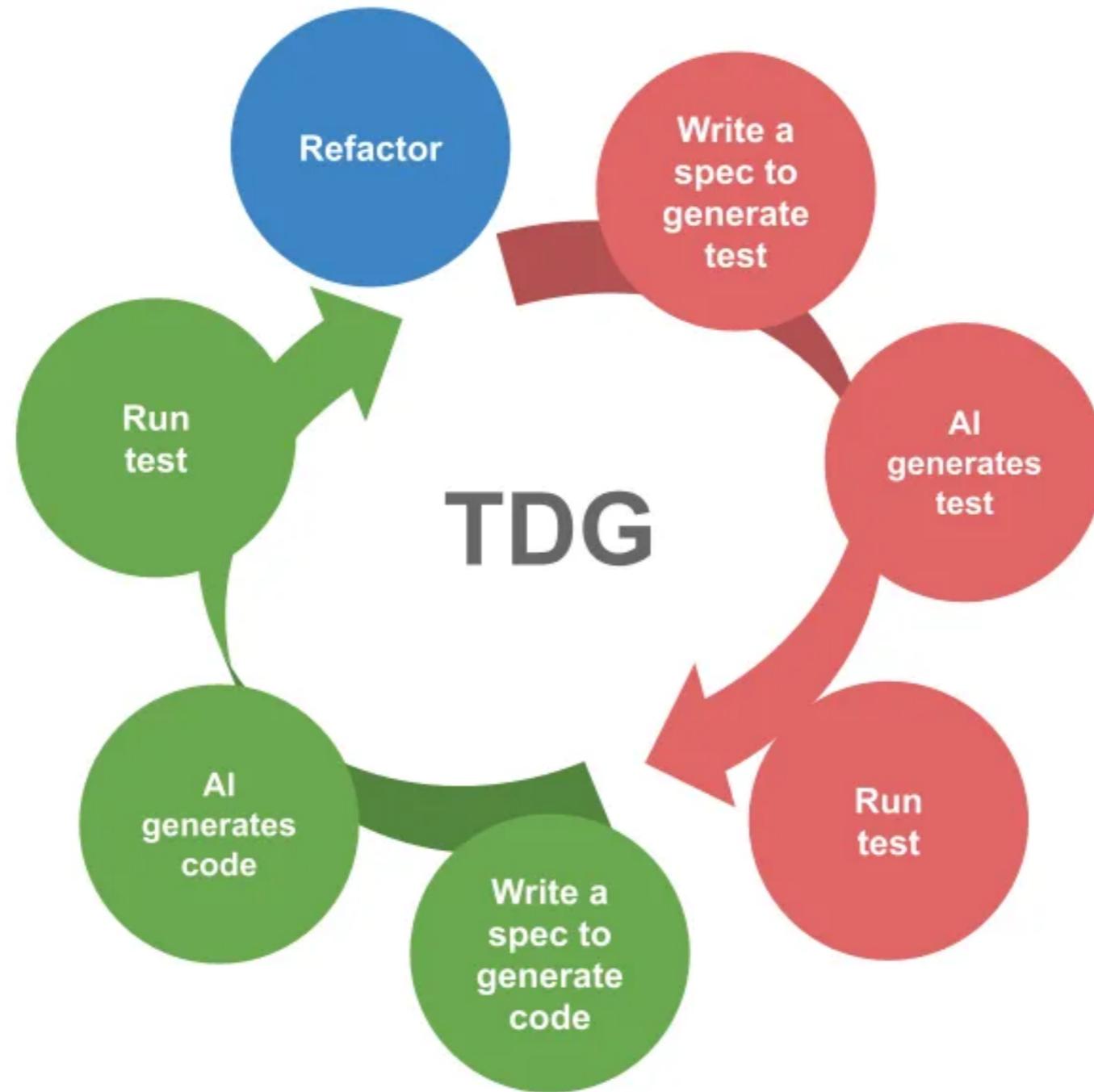
+

Generative AI

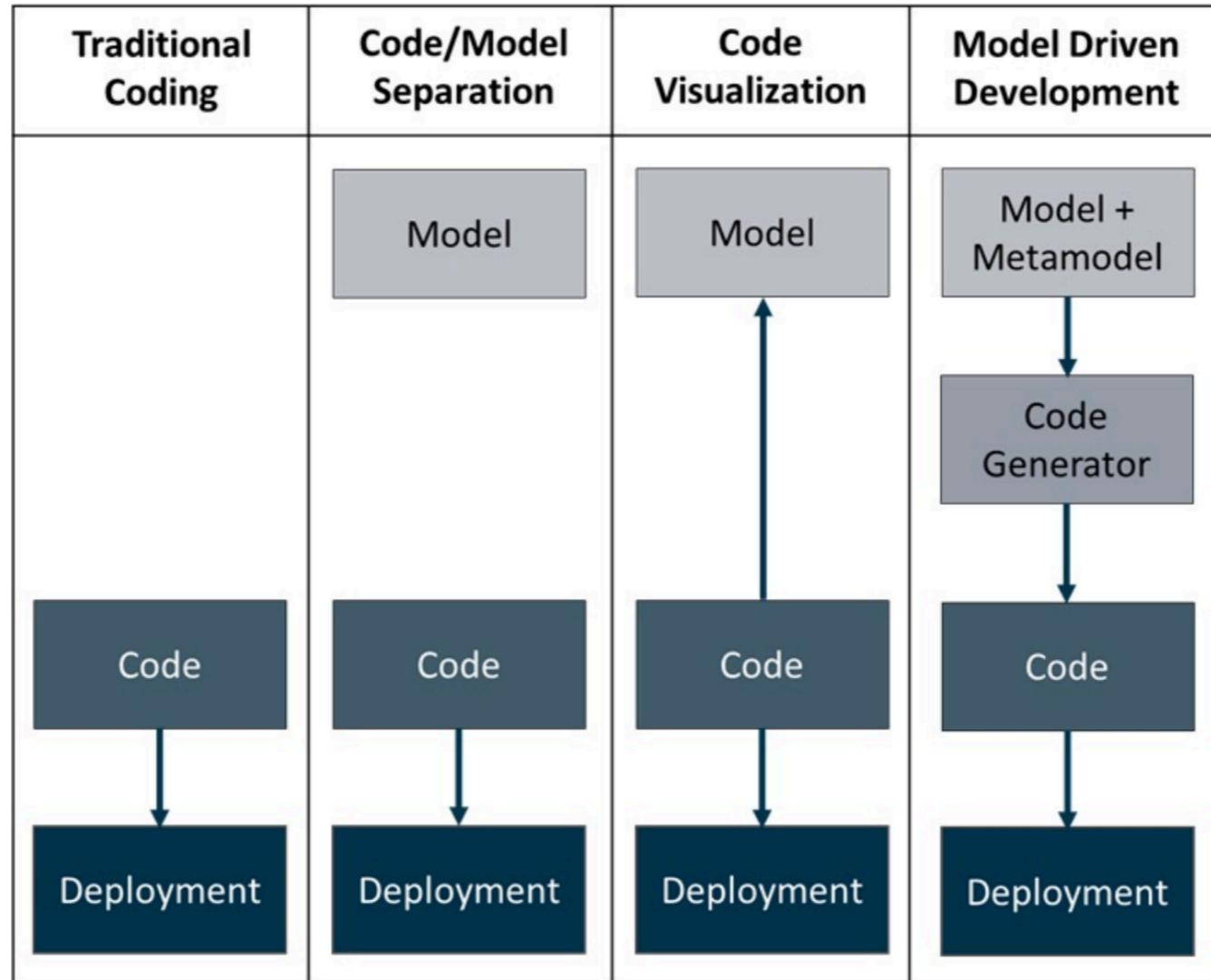
<https://chanwit.medium.com/test-driven-generation-tdg-adopting-tdd-again-this-time-with-gen-ai-27f986bed6f8>



Test-Driven-Generation (TDG)

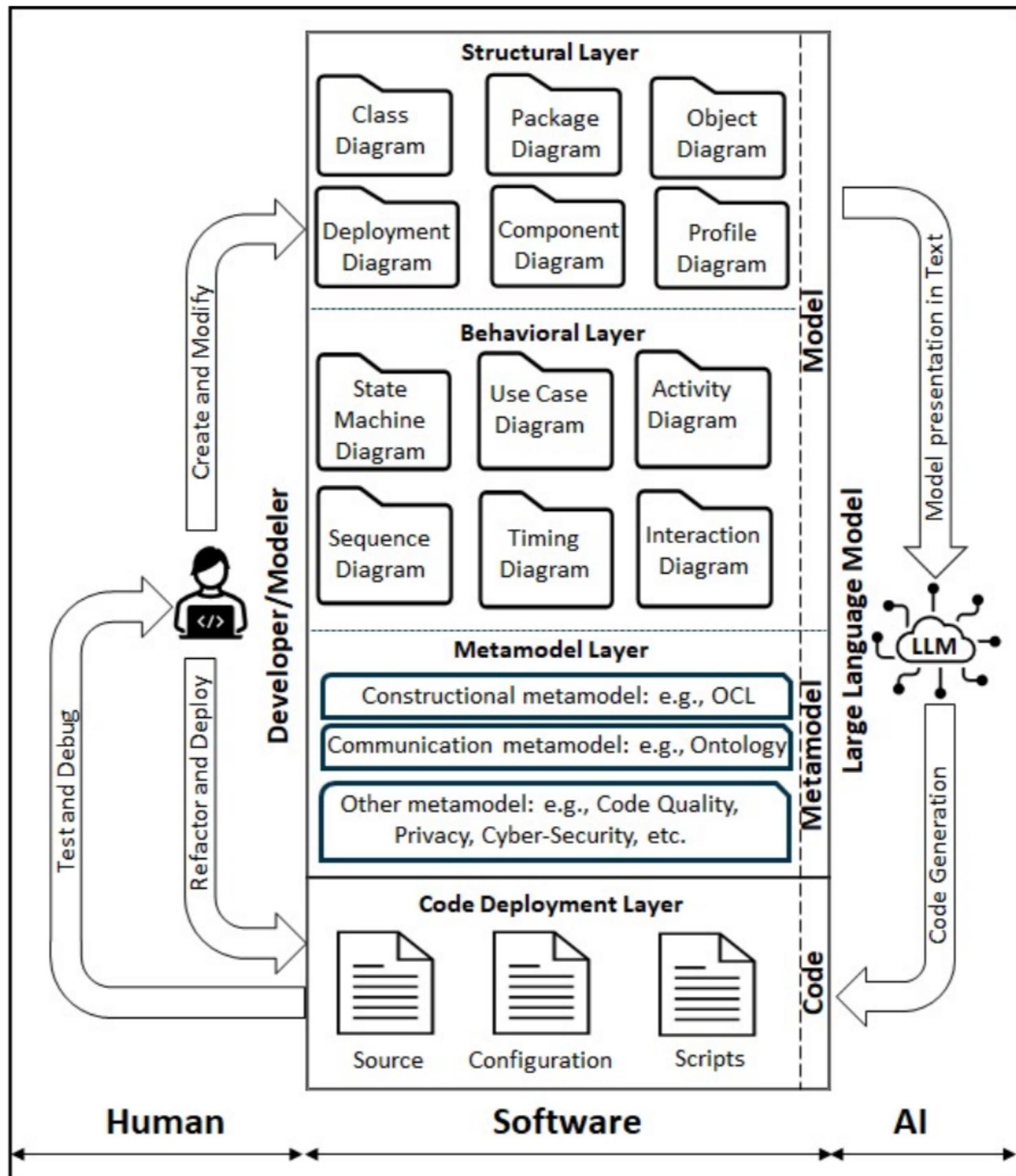


Model-Driven-Development (MDD)



<https://arxiv.org/pdf/2410.18489>

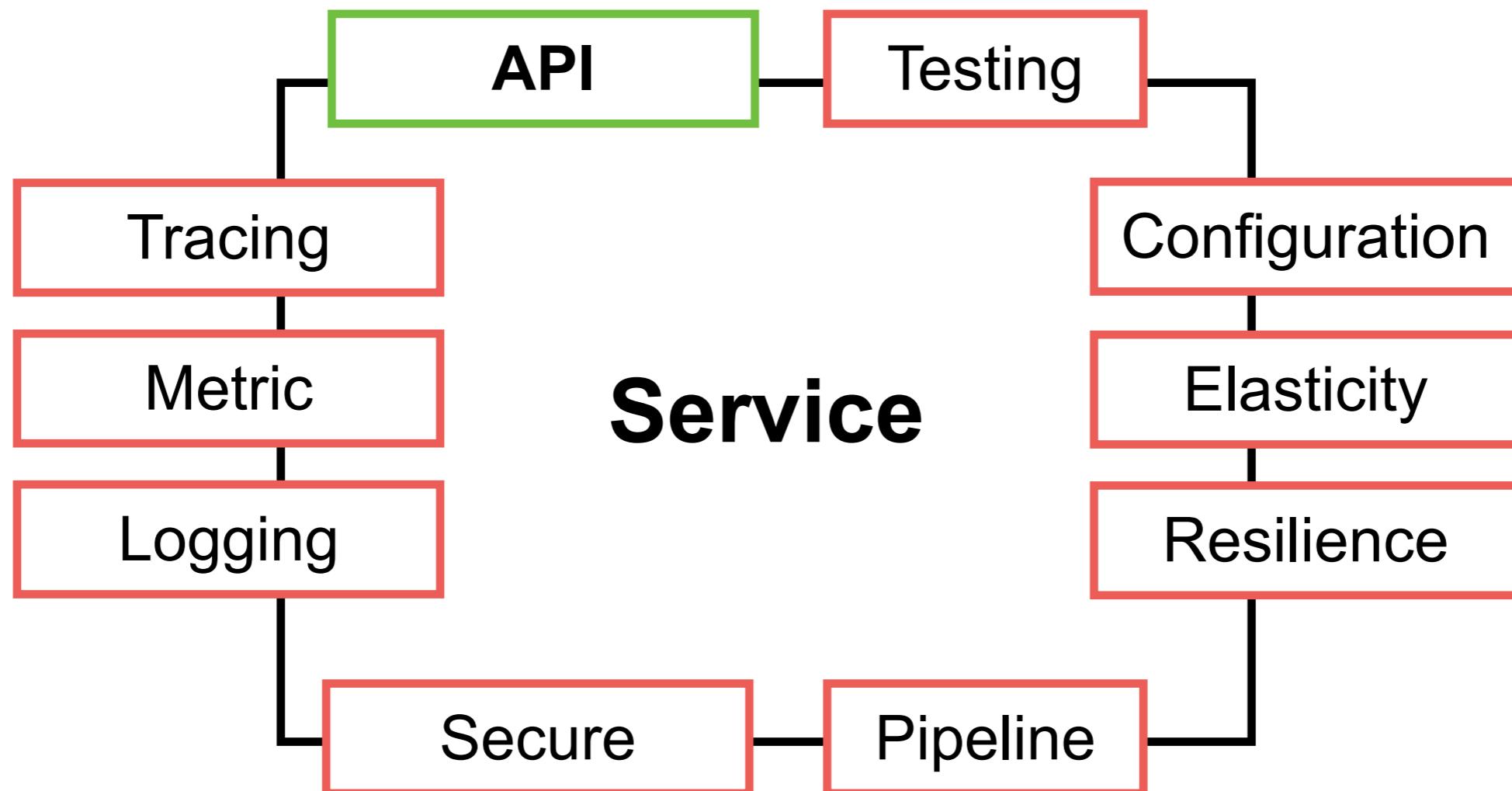




Develop REST API with .NET C#



Properties of REST APIs



APIs

Application Programming Interface

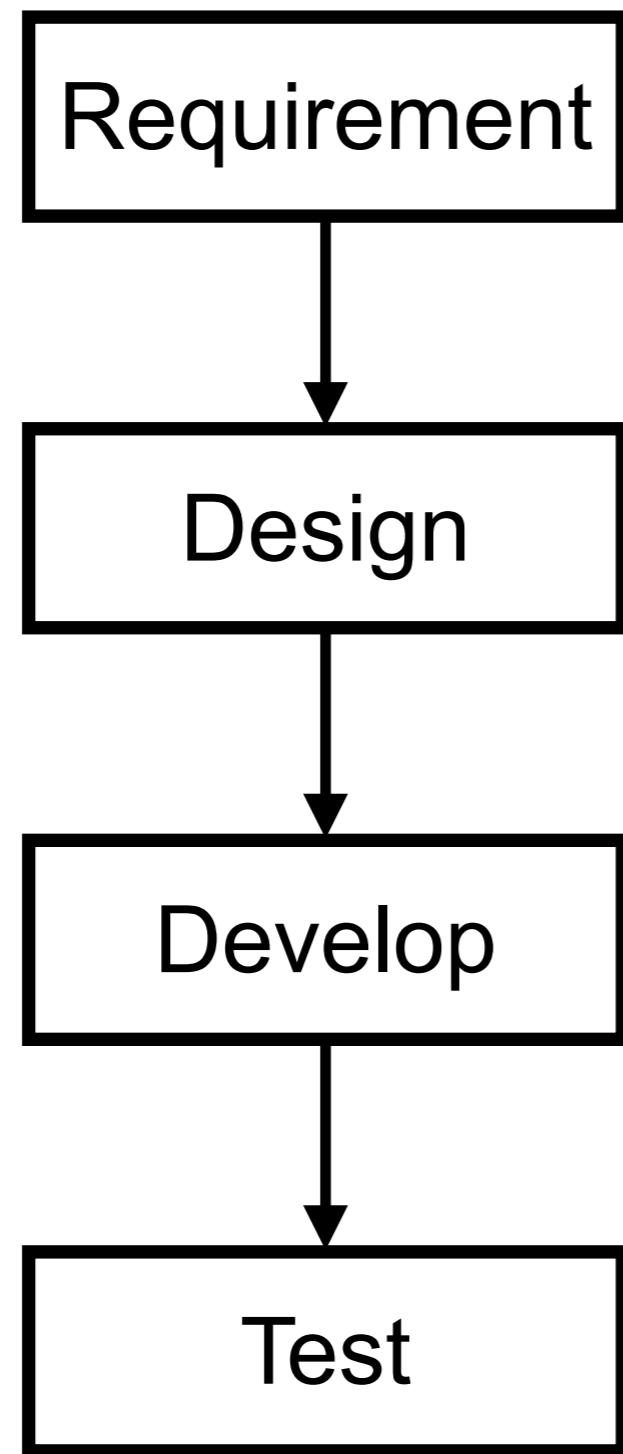
REST API

gRPC

Messaging



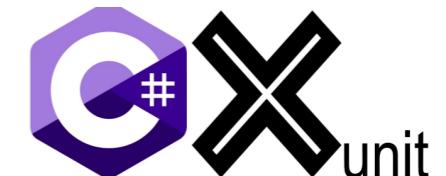
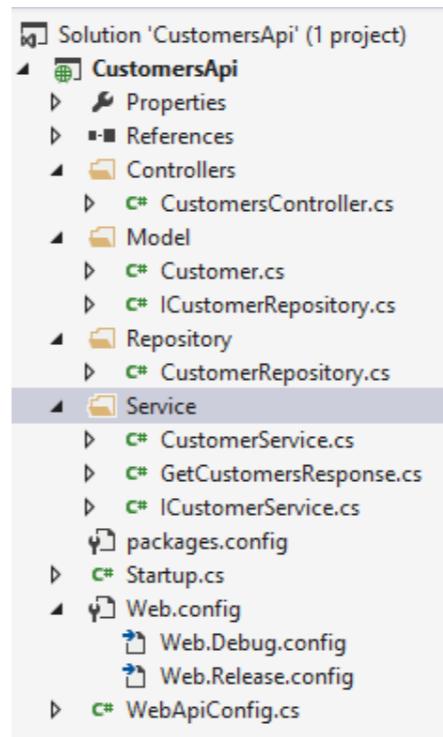
Steps of workshop



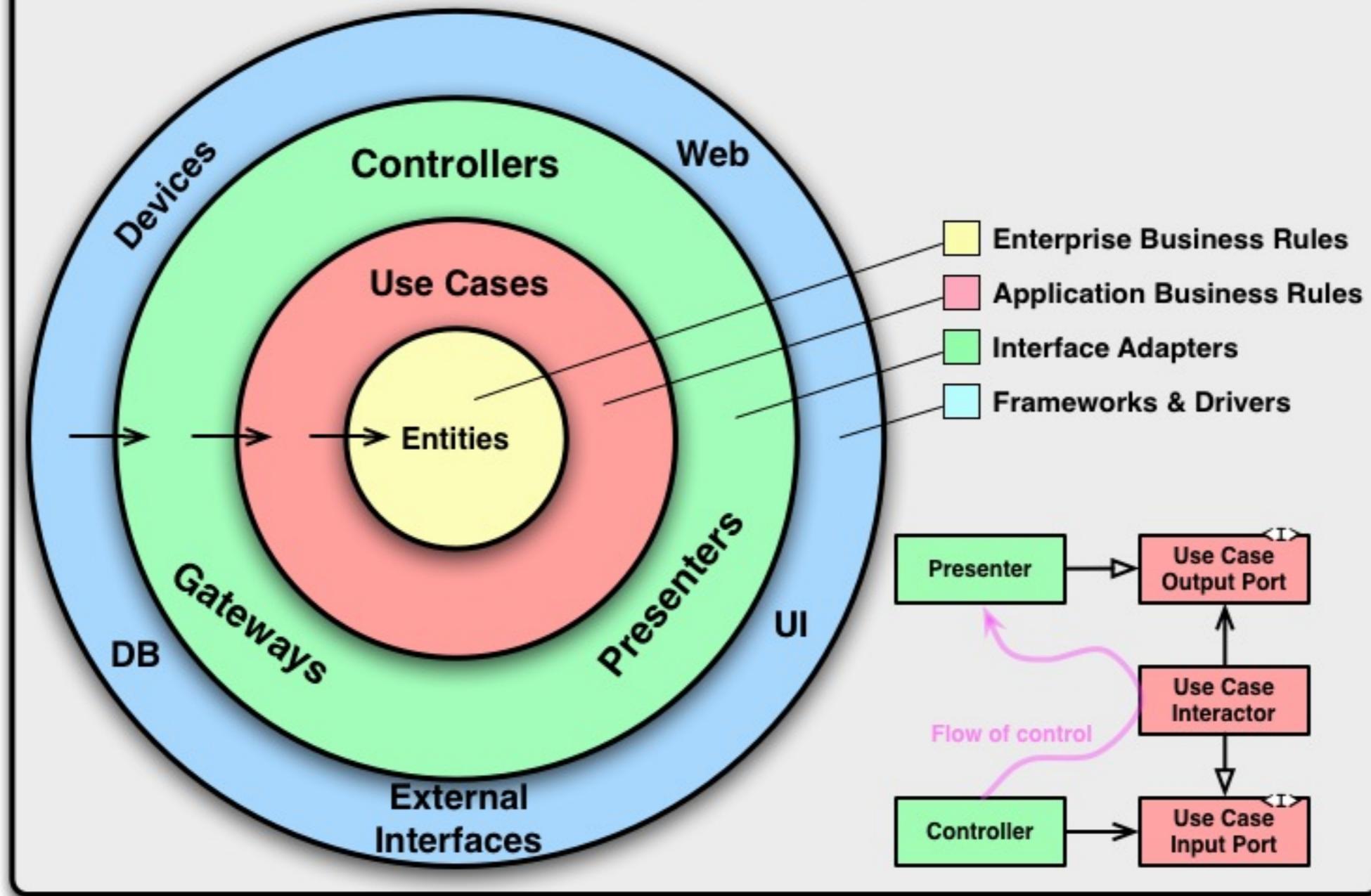
Coding standard



Swagger™



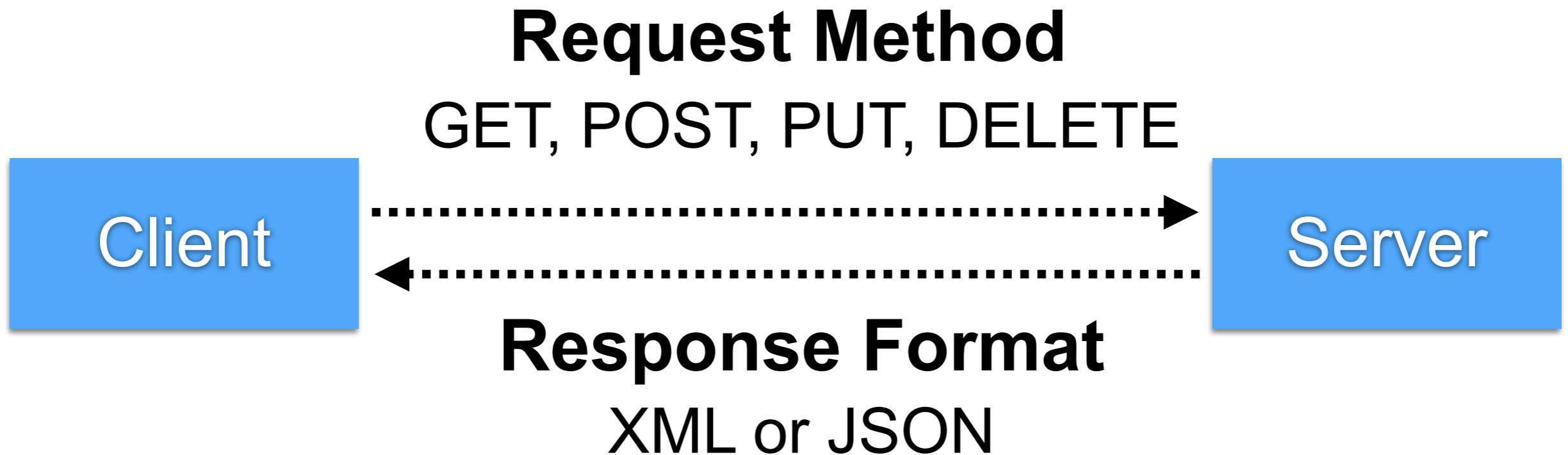
The Clean Architecture



<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>



REST Request & Response



HTTP Methods meaning

Method	Meaning
GET	Read data
POST	Create/Insert new data
PUT/PATCH	Update data or insert if a new id
DELETE	Delete data



Response format ?



Status Code

Code	Description
1xx	Information
2xx	Successful
3xx	Redirection
4xx	Client error
5xx	Server error

<https://developer.mozilla.org/en-US/docs/Web/HTTP>Status>



Good APIs ?



Principles of API Design

Consistency

Statelessness

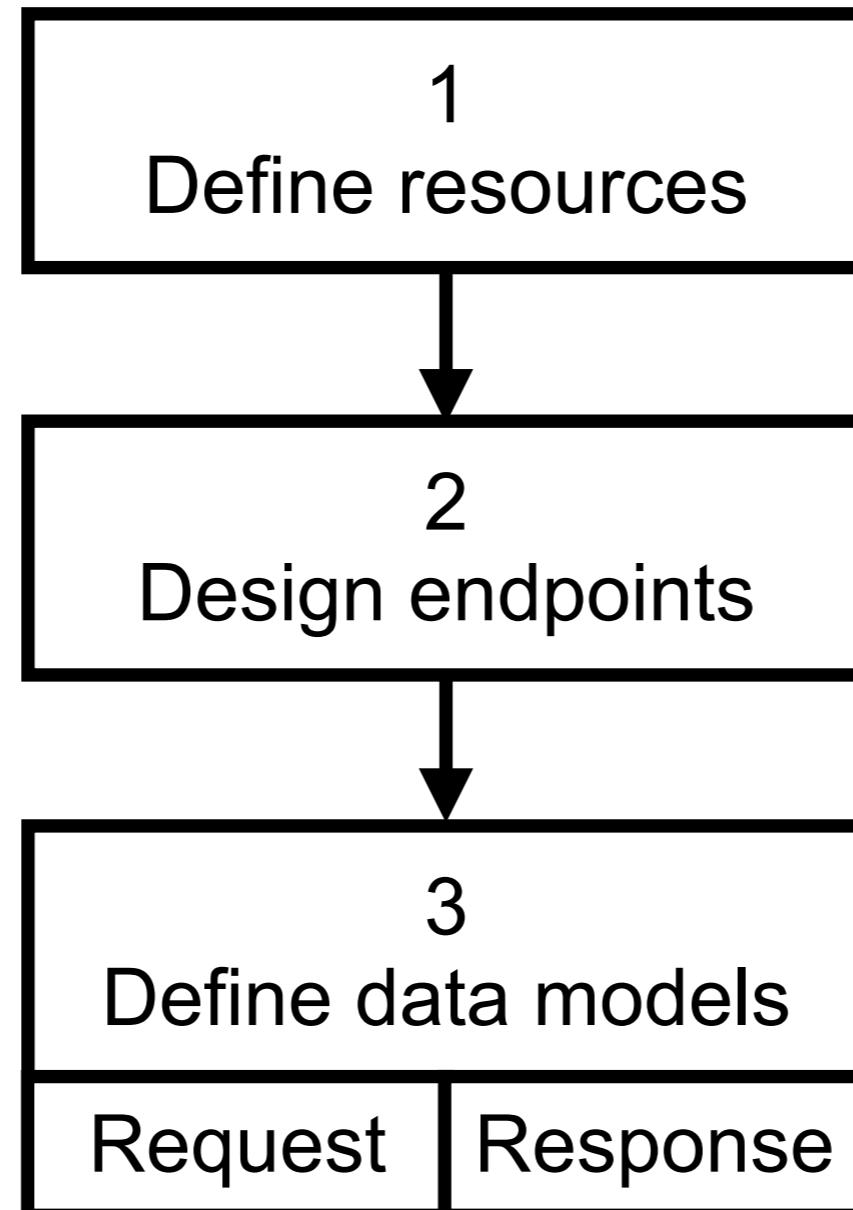
Resource-oriented design

Use standard
HTTP methods

Versioning



Steps to Design APIs



More !!

Authentication
Authorization

Pagination and
filtering

Rate Limiting

Pagination and
filtering

Error handling

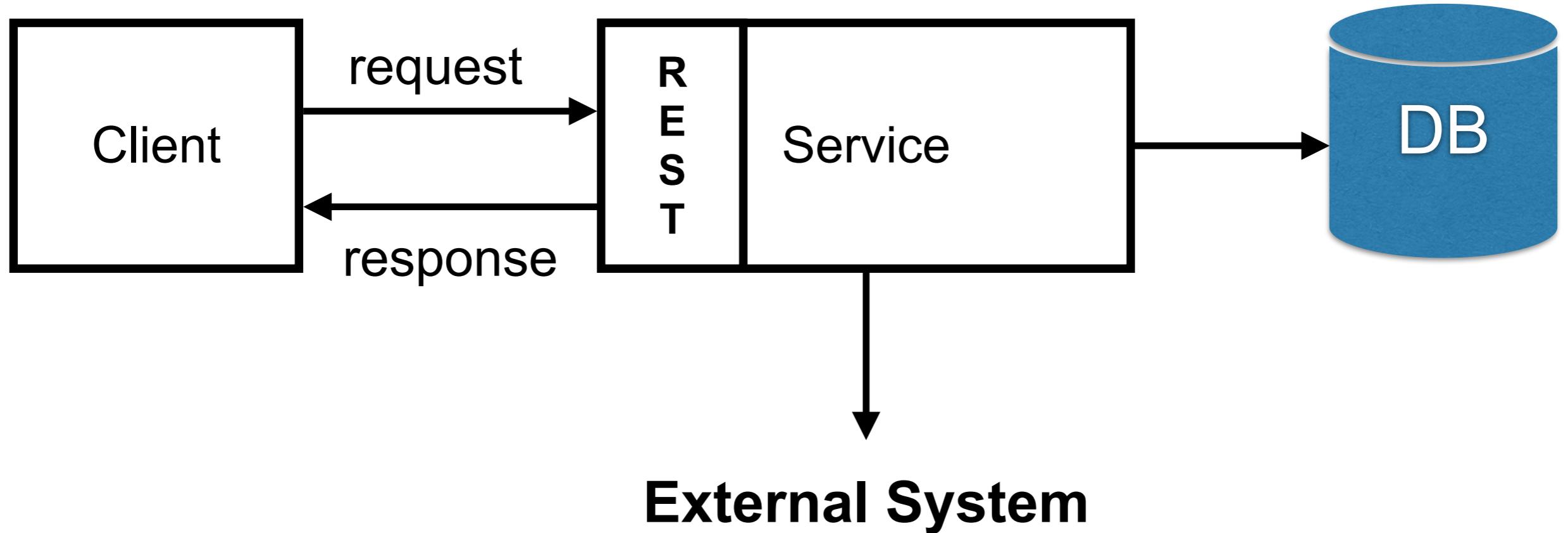
Documentation

Testing

Monitoring and
Observability



Structure of Service



API Documentation

pet Everything about your Pets

POST /pet/{petId}/uploadImage uploads an image 

POST /pet Add a new pet to the store 

PUT /pet Update an existing pet 

GET /pet/findByStatus Finds Pets by status 

GET /pet/{petId} Find pet by ID 

POST /pet/{petId} Updates a pet in the store with form data 

DELETE /pet/{petId} Deletes a pet 

<https://swagger.io/>



SOLID principle



SOLID principle



SOLID

Software development is not a Jenga game.



1. Single Responsibility Principle



Single Responsibility Principle

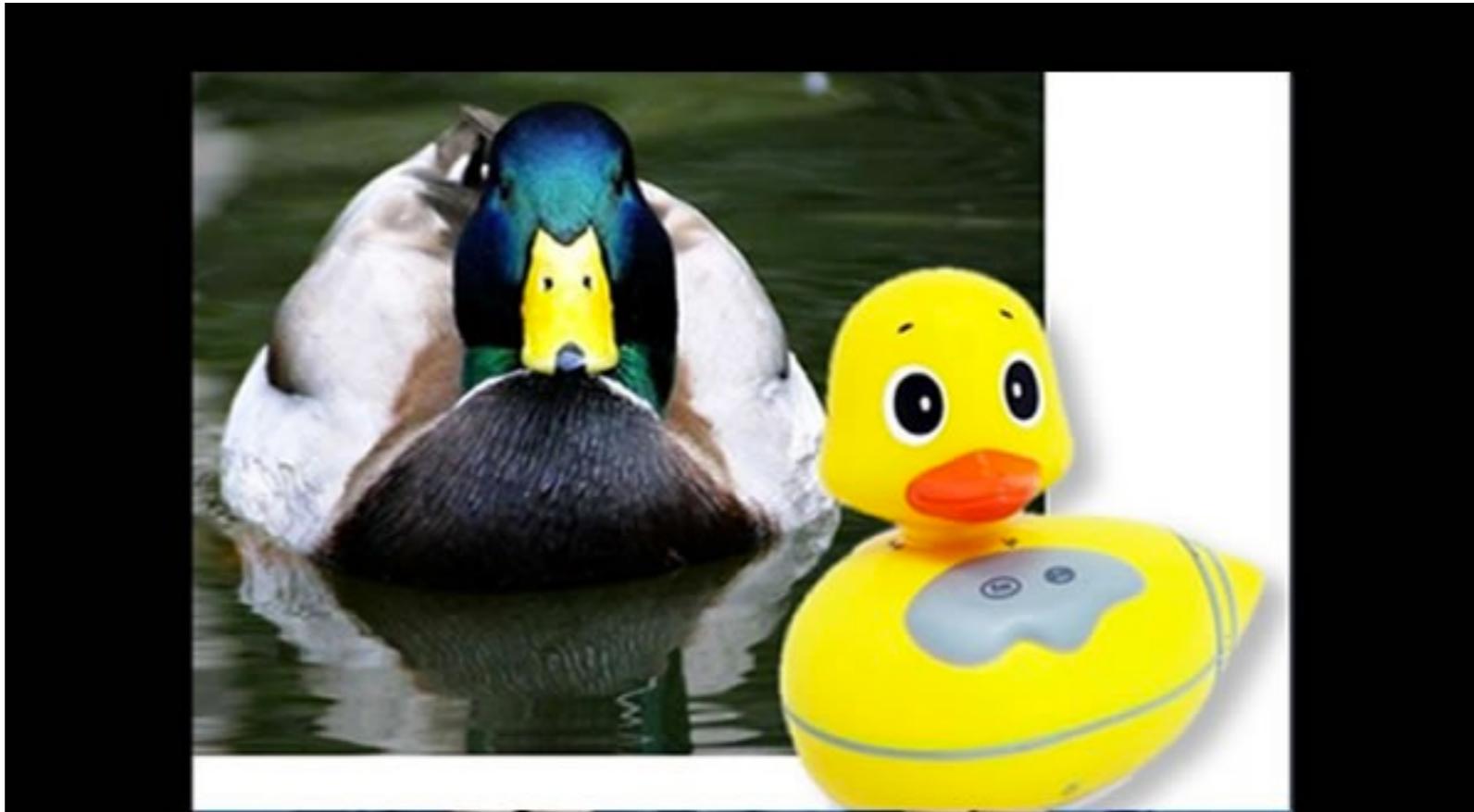
Just because you *can* doesn't mean you *should*.



2. Open-Closed Principle



3. Liskov Substitution Principle

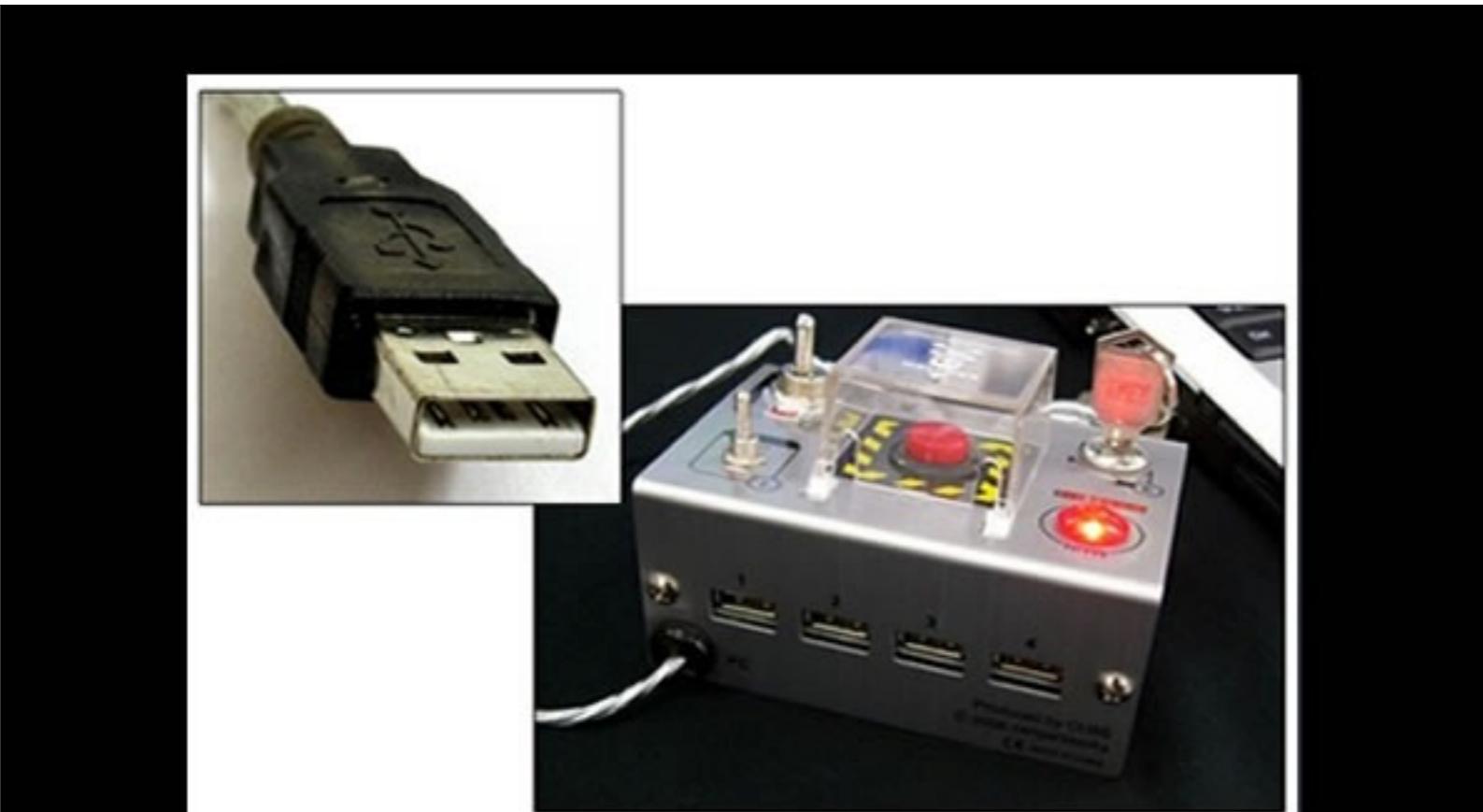


Liskov Substitution Principle

If it looks like a duck and quacks like a duck but needs batteries,
you probably have the wrong abstraction.



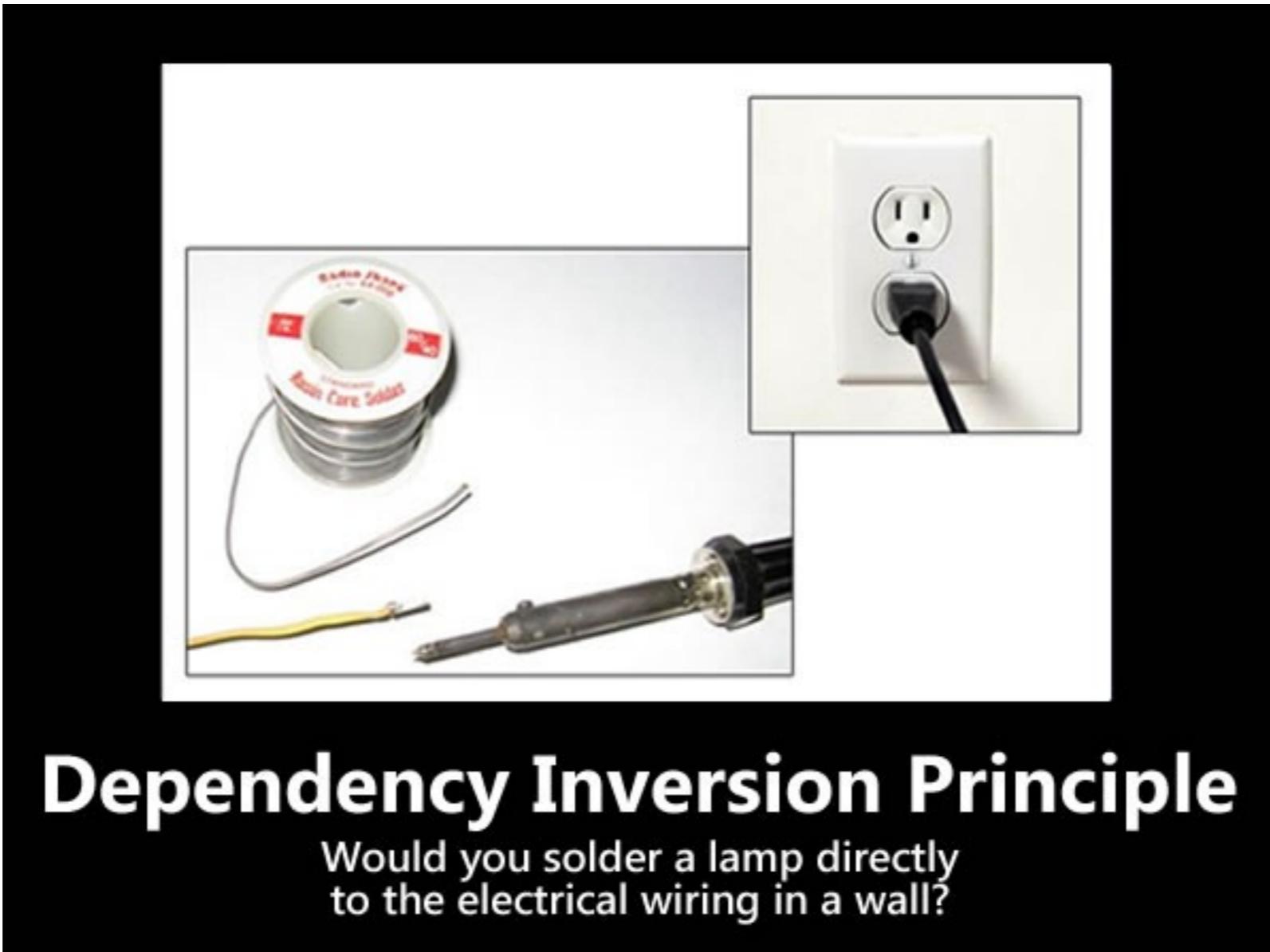
4. Interface Segregation Principle



Interface Segregation Principle
You want me to plug this in *where?*



5. Dependency Inversion Principle



Dependency Inversion Principle

Would you solder a lamp directly
to the electrical wiring in a wall?



Code Smell & Refactoring

https://en.wikipedia.org/wiki/Code_smell



Code Smell



Code Smell



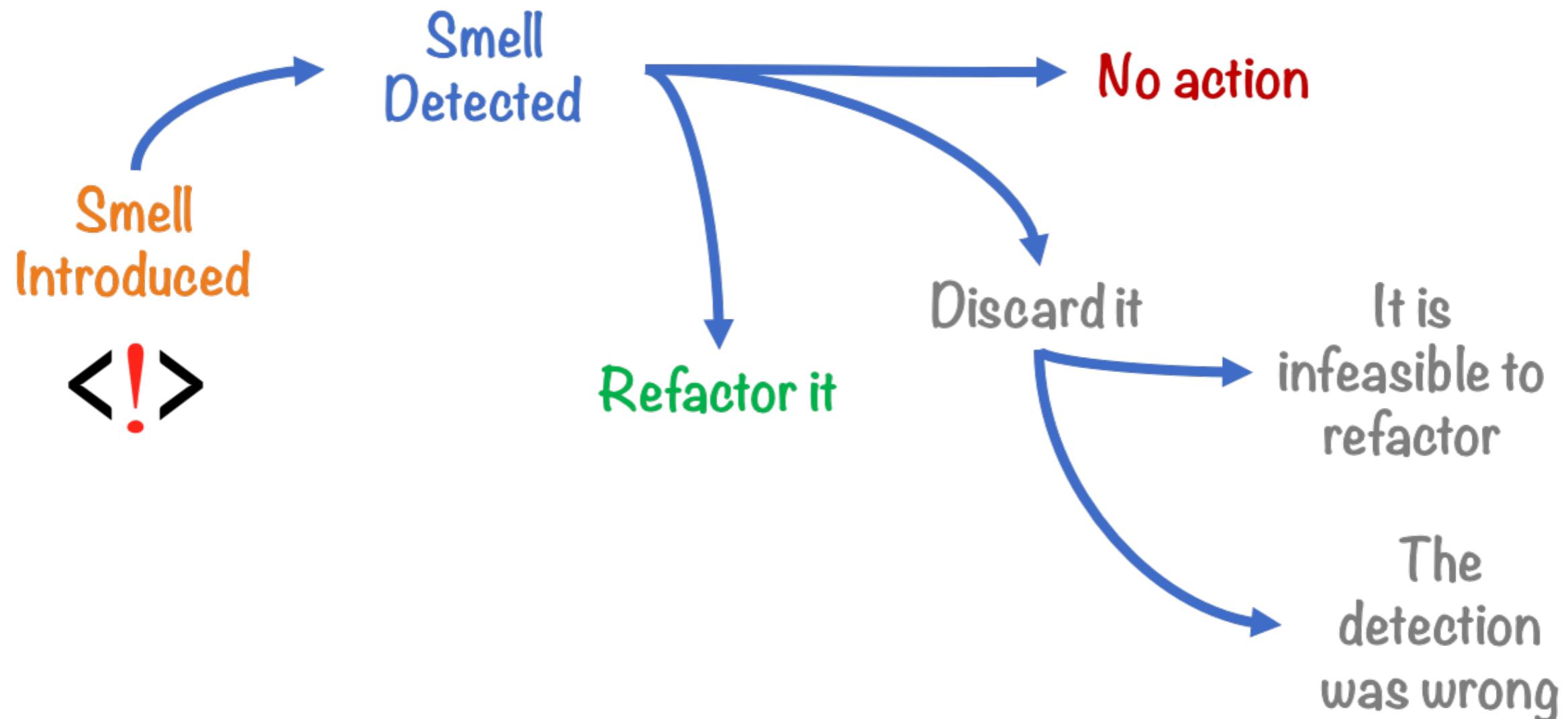
Code Smell make problems

```
41 @ ...
42     static MappedField validateQuery(final Class clazz, final Mapper mapper, final StringBuilder origProp, final FilterOperator op, final
43         MappedField mf = null;
44         final String prop = origProp.toString();
45         boolean hasTranslations = false;
46         if (!origProp.substring(0, 1).equals("$")) {
47             final String[] parts = prop.split(regex: "\\".__);
48             if (clazz == null) { return null; }
49             MappedClass mc = mapper.getMappedClass(clazz);
50             //CHECKSTYLE:OFF
51             for (int i = 0; ; ) { --Eek!
52                 //CHECKSTYLE:ON
53                 final String part = parts[i];
54                 boolean fieldIsArrayOperator = part.equals("$");
55                 mf = mc.getMappedField(part);
56                 //translate from java field name to stored field name
57                 if (mf == null && !fieldIsArrayOperator) {
58                     mf = mc.getMappedFieldByJavaField(part);
59                     if (validateNames && mf == null) {
60                         throw new ValidationException(format("The field '%s' could not be found in '%s' while validating - %s; if you wis
61                     }
62                     hasTranslations = true;
63                     if (mf != null) {
64                         parts[i] = mf.getNameToStore();
65                     }
66                 }
67                 i++;
68                 if (mf != null && mf.isMap()) {
69                     //skip the map key validation, and move to the next part
70                     i++;
71                 }
72                 if (i >= parts.length) {
73                     break;
74                 }
75                 if (!fieldIsArrayOperator) {
76                     //catch people trying to search/update into @Reference/@Serialized fields
77                     if (validateNames && !canQueryPast(mf)) {
78                         throw new ValidationException(format("Cannot use dot-notation past '%s' in '%s'; found while validating - %s", pa
79                     }
80                     if (mf == null && mc.isInterface()) {
81                         break;
82                     } else if (mf == null) {
83                         throw new ValidationException(format("The field '%s' could not be found in '%s'", prop, mc.getClazz().getName()));
84                     }
85                     //get the next MappedClass for the next field validation
86                     mc = mapper.getMappedClass((mf.isSingleValue()) ? mf.getType() : mf.getSubClass());
87                 }
88             }
89             //record new property string if there has been a translation to any part
90             if (hasTranslations) {
91                 origProp.setLength(0); // clear existing content
92                 origProp.append(parts[0]);
93                 for (int i = 1; i < parts.length; i++) {
```

What's a prop?
What's a part?
Eek!
Why all the null checks?
Control the loop
Comments, because code is unclear
Parameter mutation!



Do you know Code smell ?



Code Smell (example)

- Duplication code
- Long method
- Large class (God class)
- Comments
- Long parameter list
- Switch/If statements
- Divergent change (one class)
- Shotgun surgery (one class)

<https://sourcemaking.com/refactoring/smells>



Code Smell Workshop

<https://github.com/emilybache/Tennis-Refactoring-Kata>



Q/A

