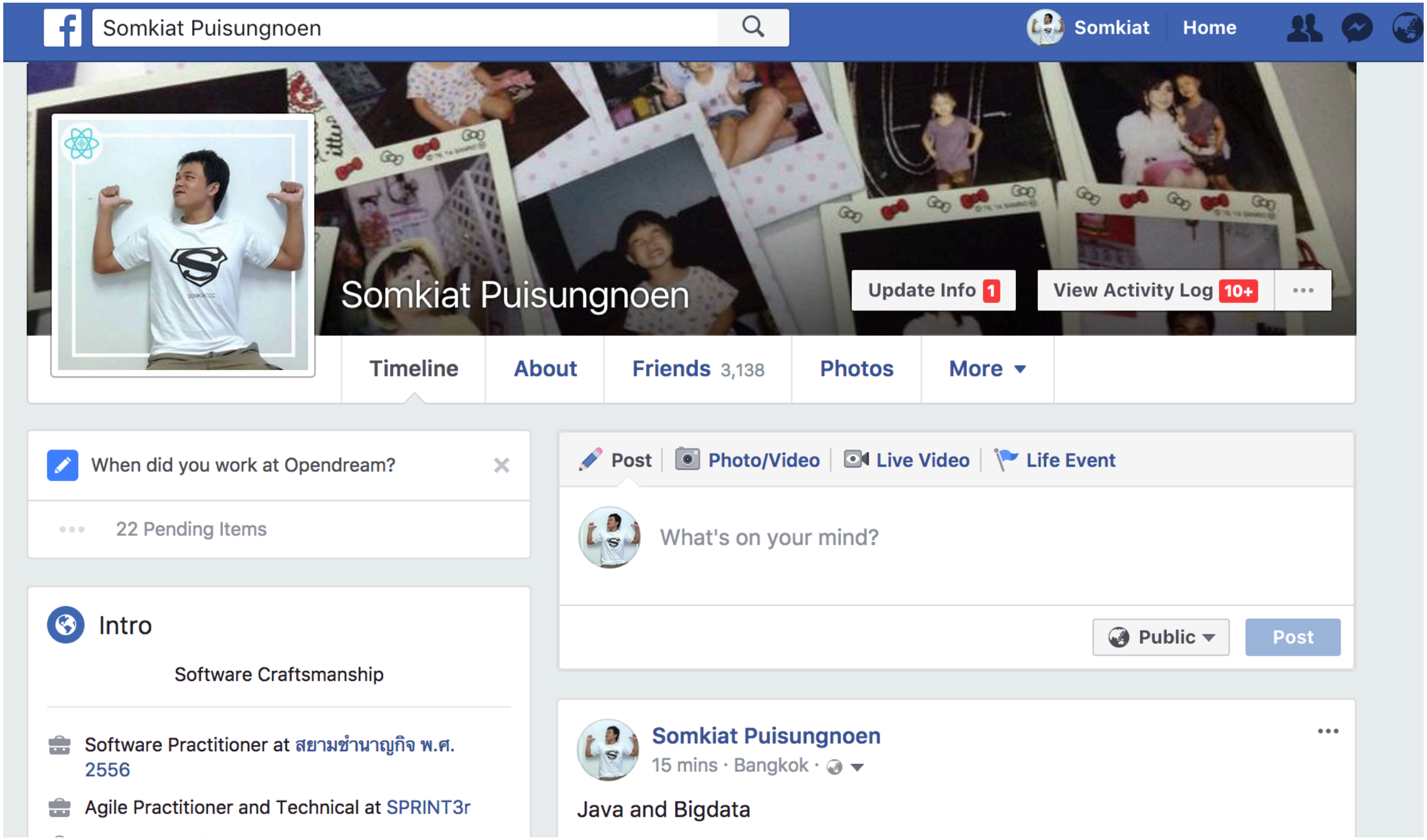
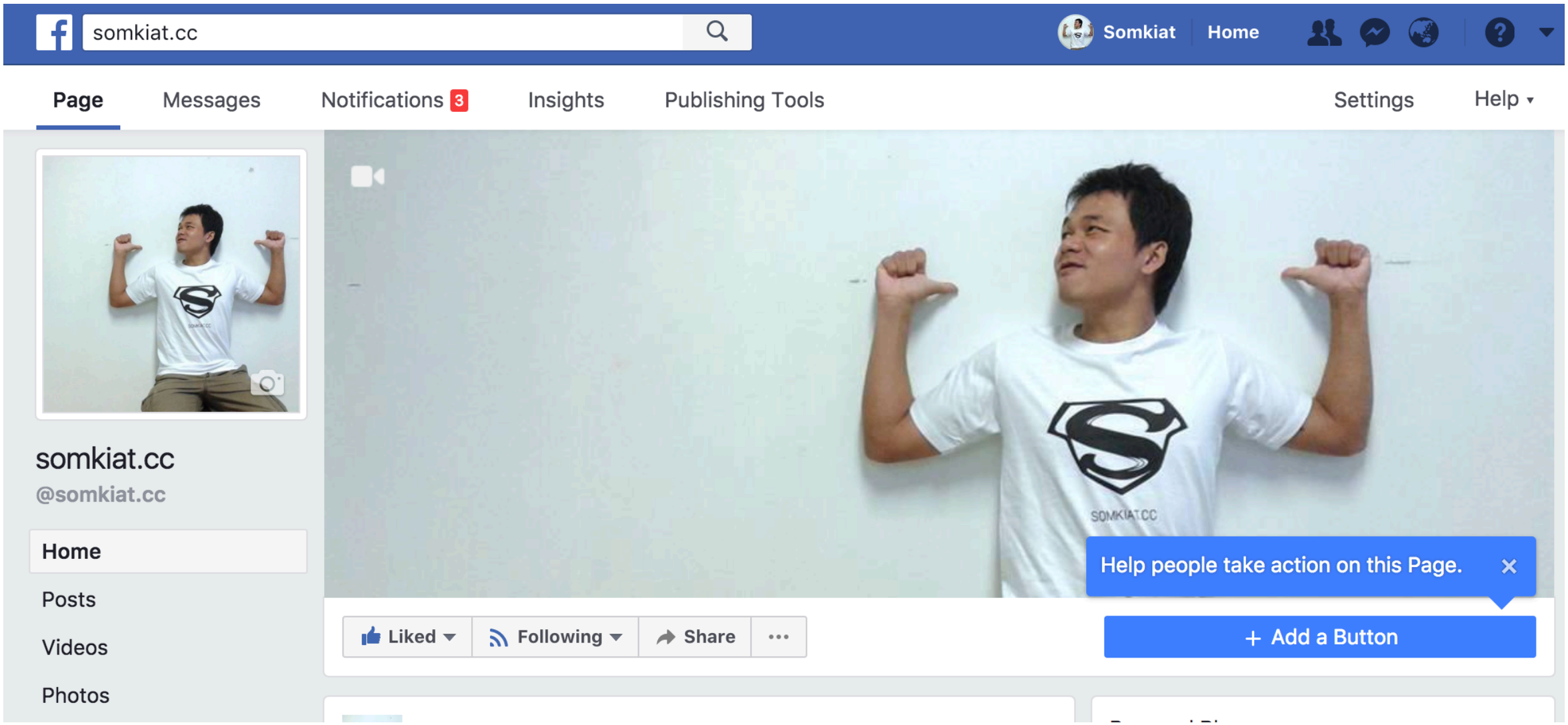
สยามชำนาญกิจ
SIAM CHAMNAN KIT

Advance JAVA



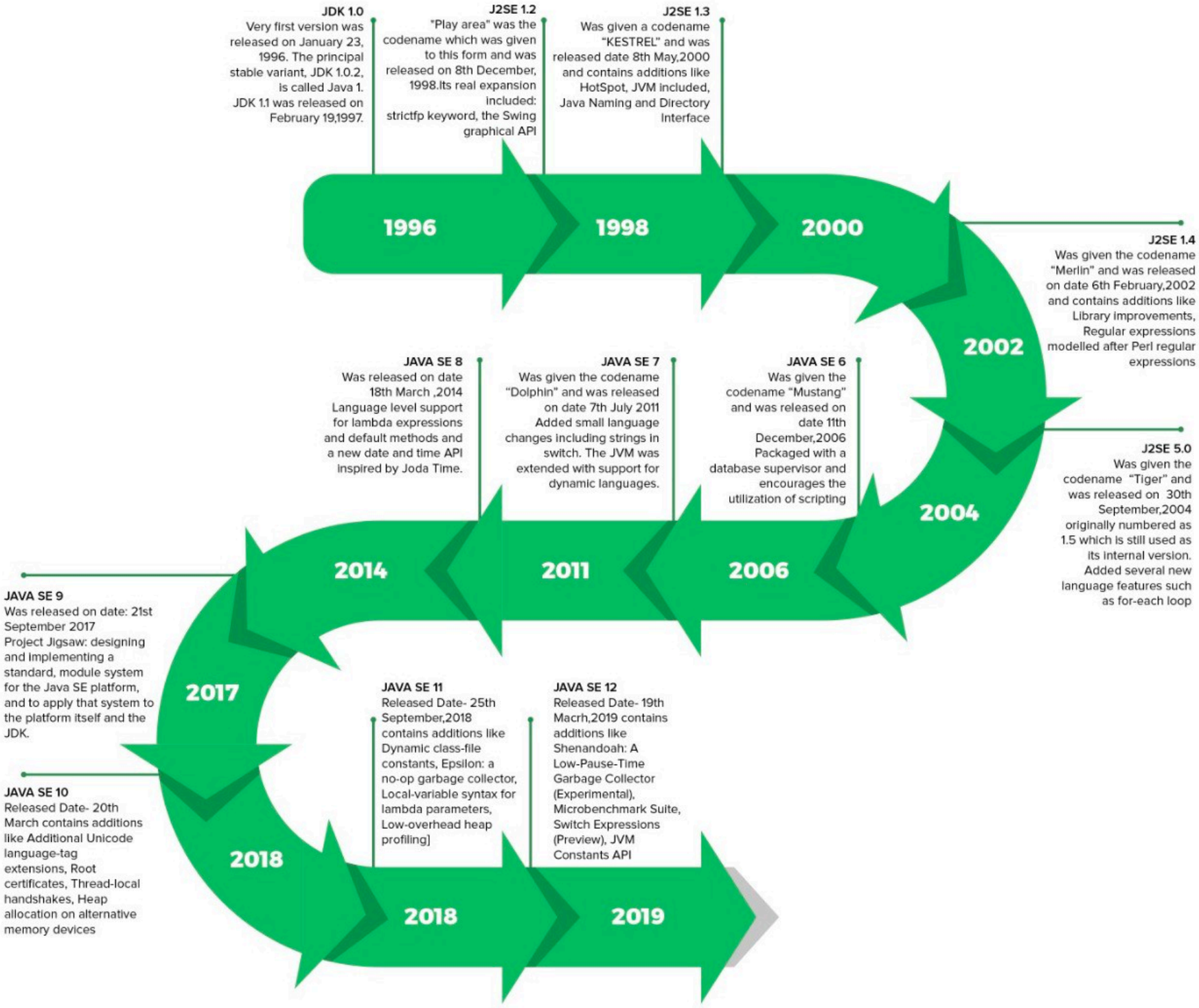


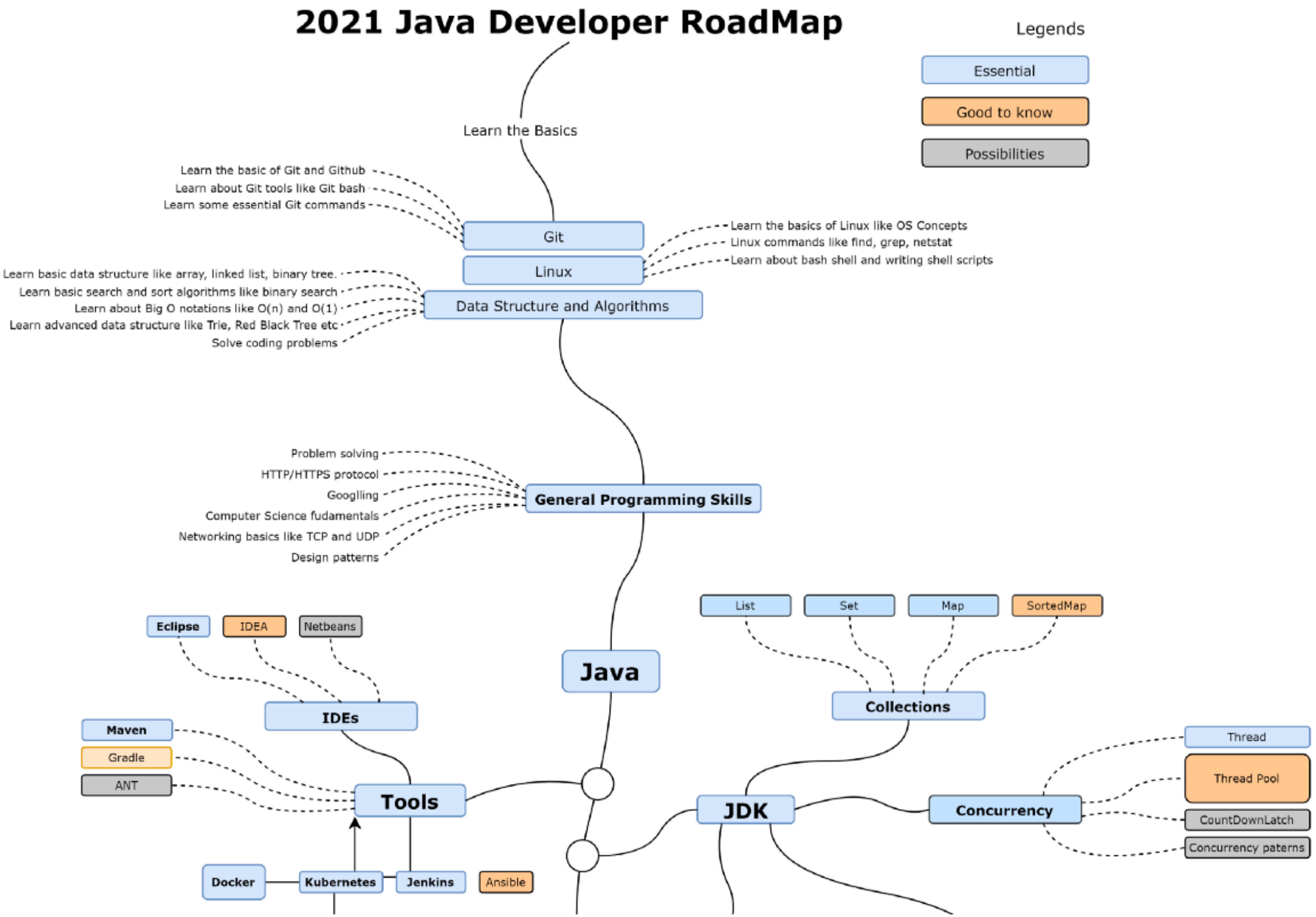
https://github.com/up1/course-advance-java

Agenda

- Test-Driven Develop (TDD)
- Clean Code
- Refactoring
- Object-Oriented Design
- S.O.L.I.D vs S.T.U.P.I.D
- Design patterns
- Working with Spring Boot

History of Java

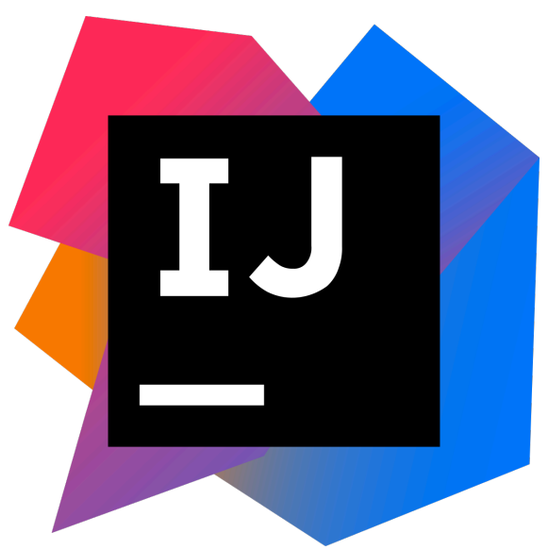




https://medium.com/javarevisited/the-java-programmer-roadmap-f9db163ef2c2

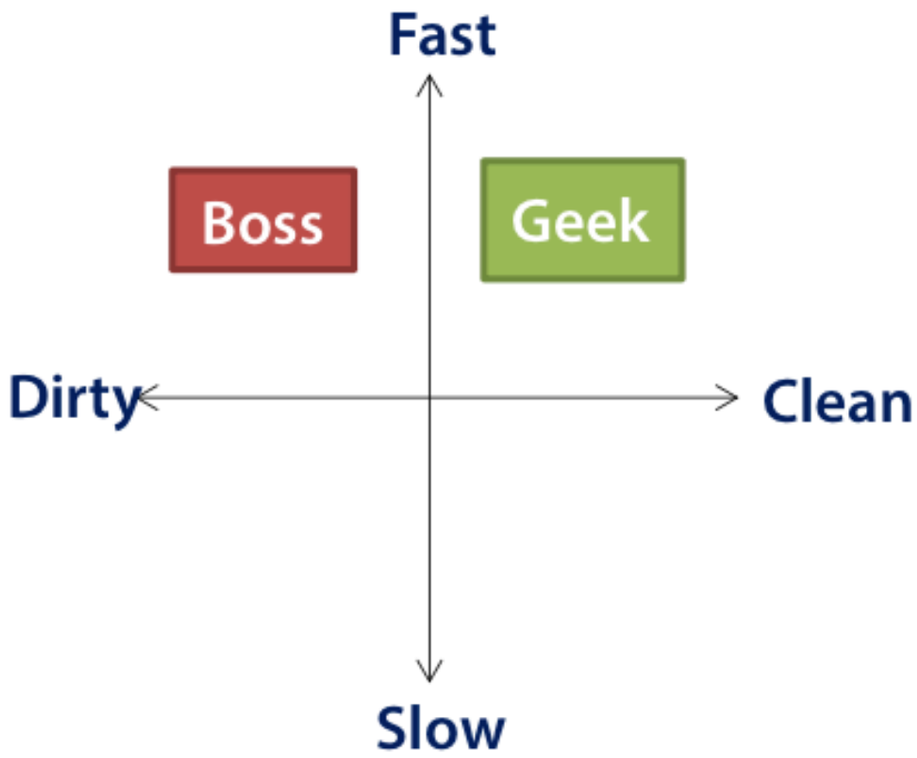
Build tools

IDE WARs

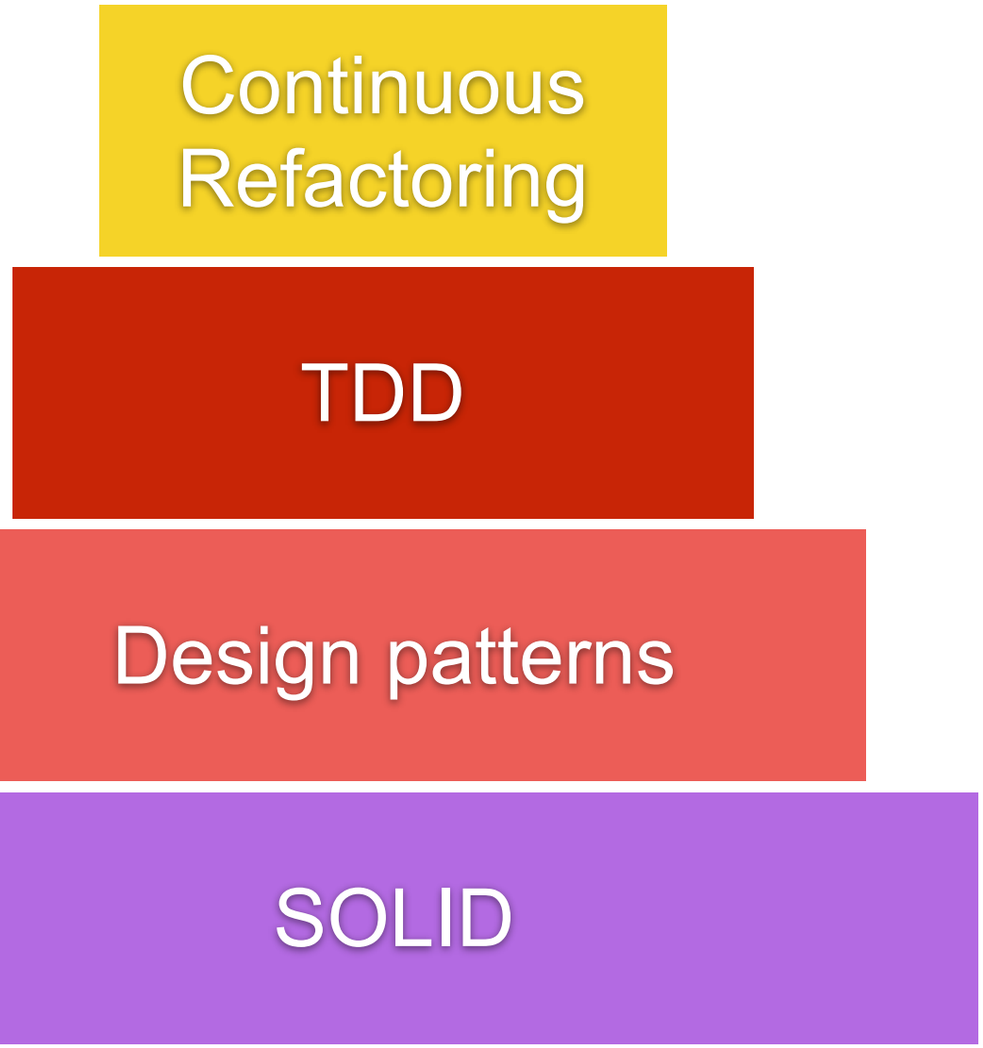


Coding is for Humans

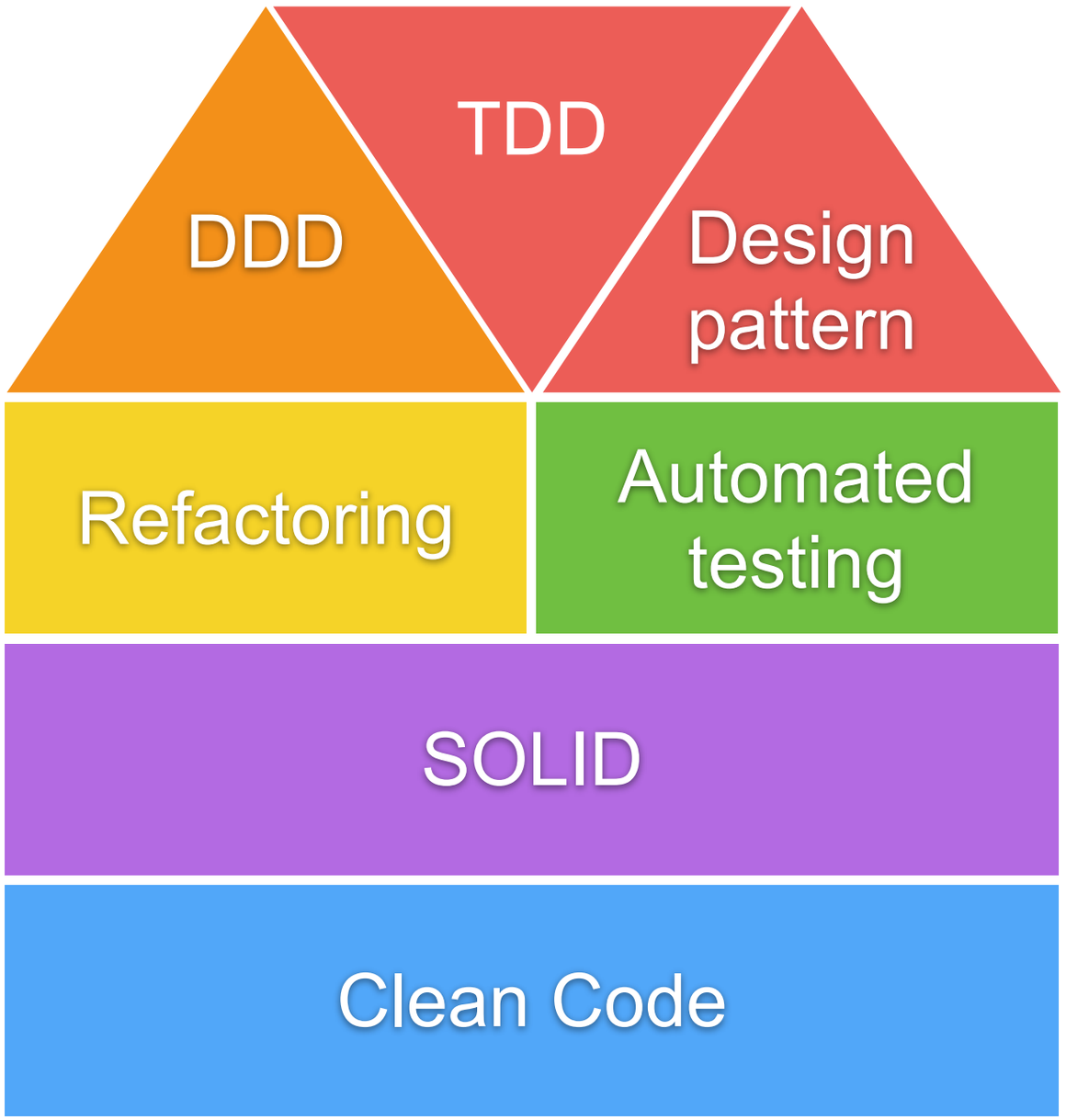
Why should care ?



Pyramid of Clean Code

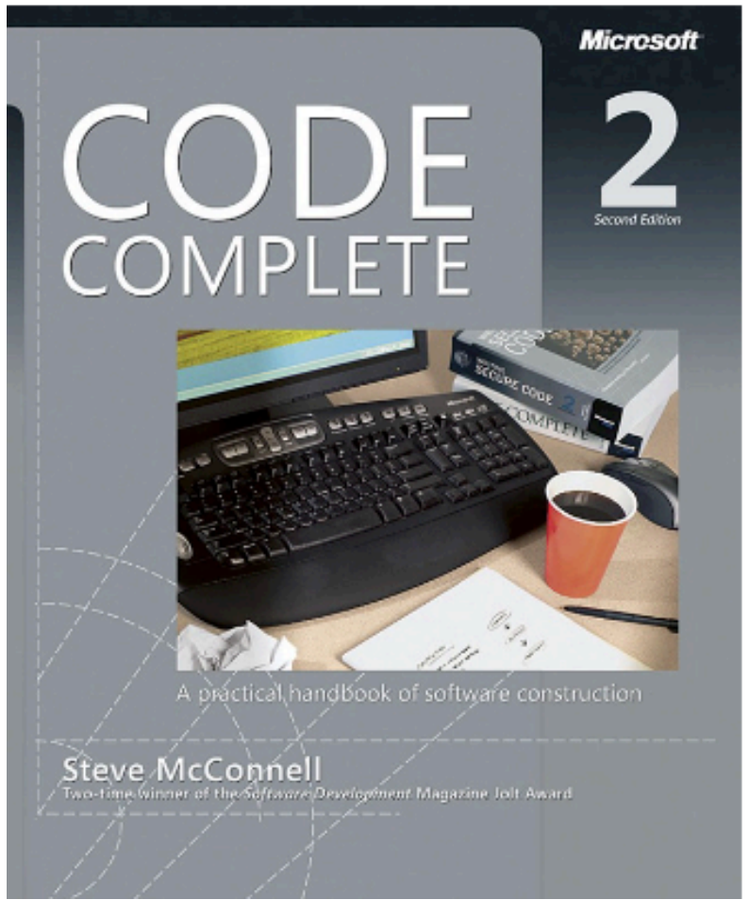


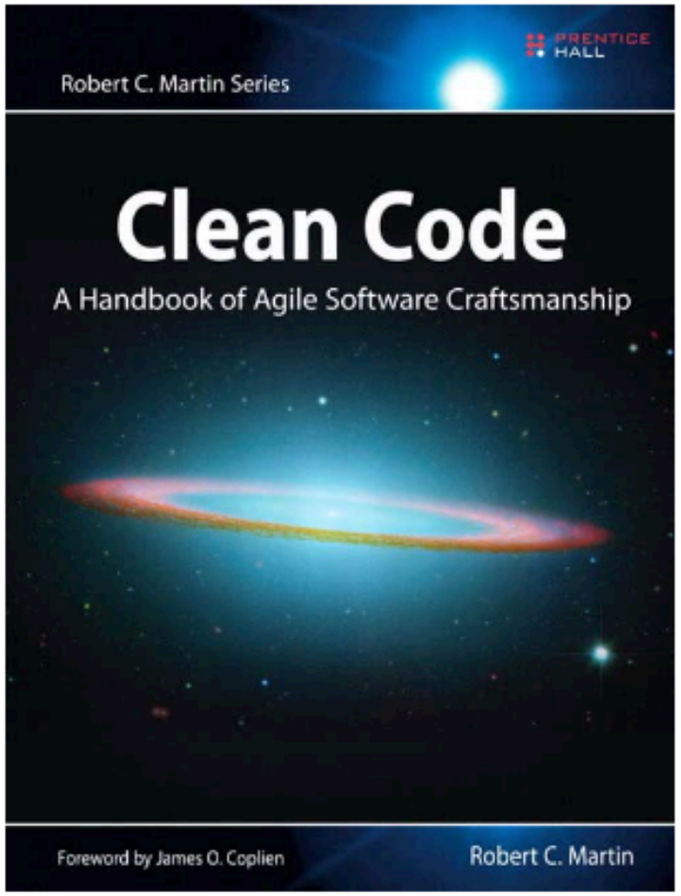
Foundation

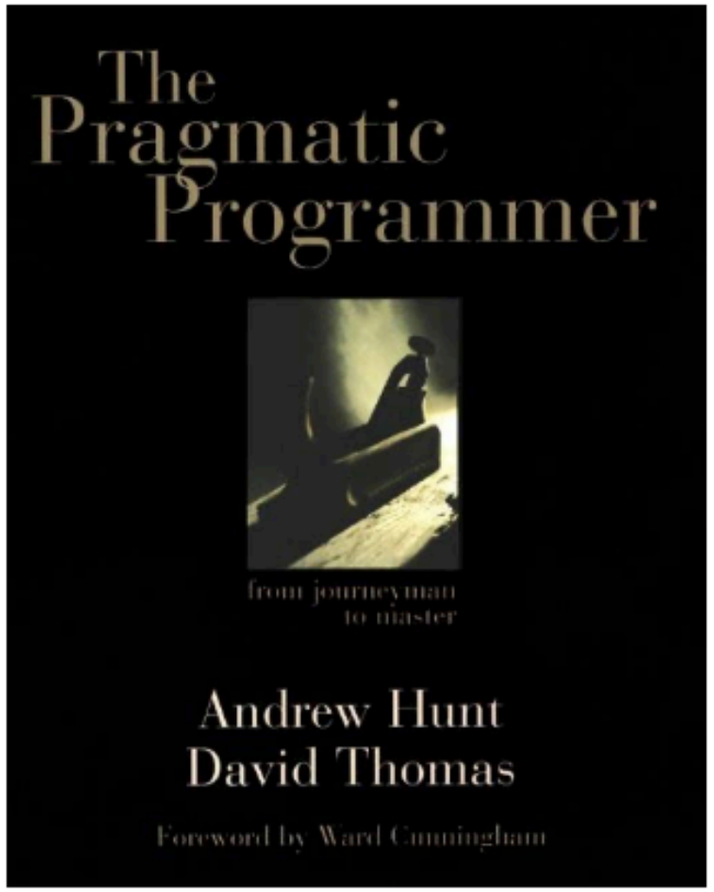


Start with Clean Code

Books





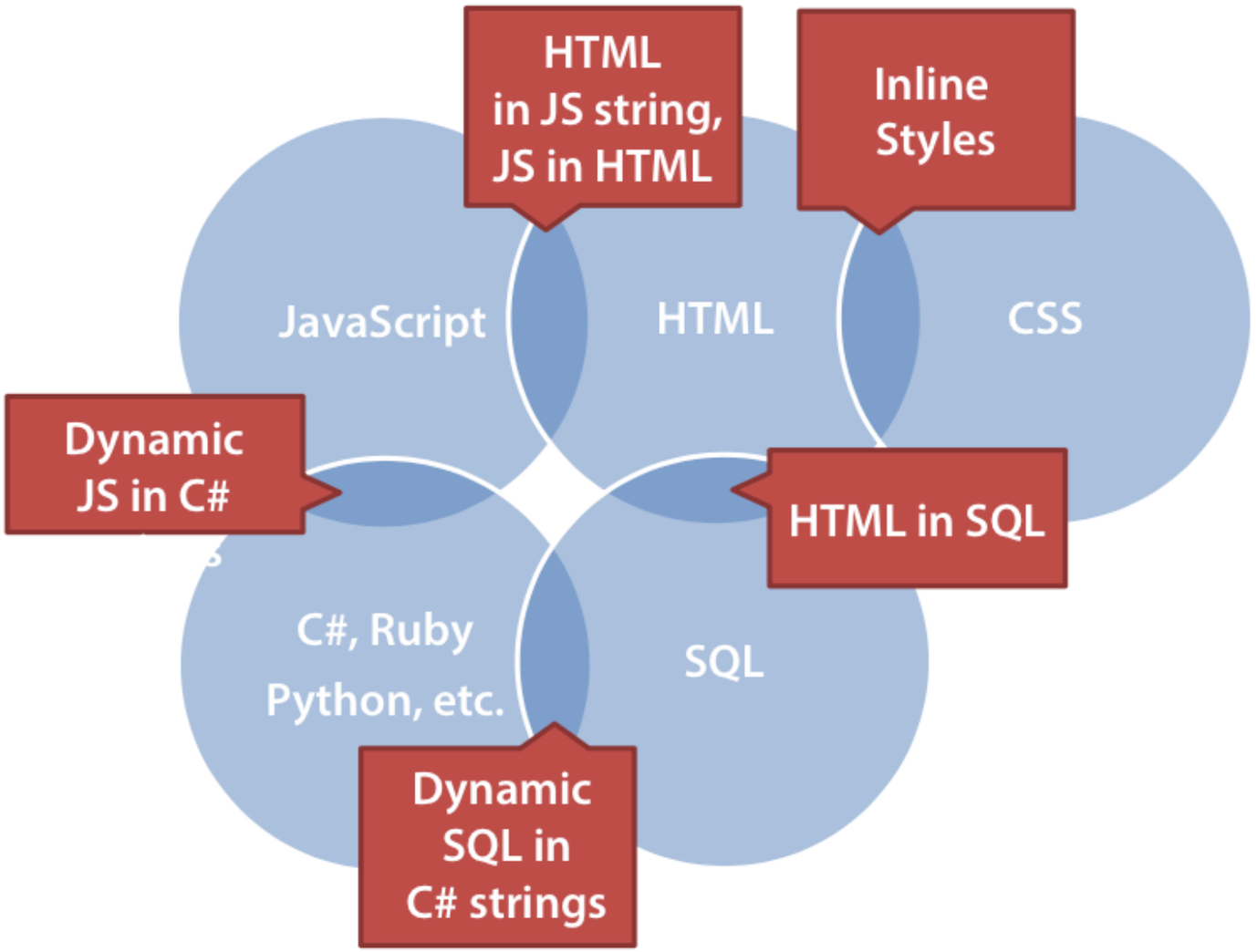


Clean Code

Right tool for the job
Maximize signal to noise ratio
Self-documenting code

1. Right tool for the job

Boundary matter



Boundary / Stay native

Dirty

```
string script = @"<script type=""text/javascript"" defer=""defer"">
                    //<![CDATA[
                        var _gaq = _gaq || [];
                        _gaq.push(['_setAccount', '" + ws.GoogleAnalyticsID + @"']);
                        _gaq.push(['_trackPageview']);

                        (function() {
                        var ga = document.createElement('script');
                        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
                        '.google-analytics.com/ga.js';
                        ga.setAttribute('async', 'true');
                        document.documentElement.firstChild.appendChild(ga);
                        })();
                    //]]>
                    </script>";
this.Header.Controls.Add(new LiteralControl("\r\n" + script));
```

Clean

```
<!--In document head-->
<script type="text/javascript">
    var WebSiteSetup = { "GoogleAnalyticsKey": "JDSGI832JDUG9831" };
</script>
```

```
//In GoogleAnalytics.js
var _gaq = _gaq || [];
_gaq.push(['_setAccount', WebSiteSetup.GoogleAnalyticsKey]);
_gaq.push(['_trackPageview']);

(function () {
   var ga = document.createElement('script');
   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
   '.google-analytics.com/ga.js';
   ga.setAttribute('async', 'true');
   document.documentElement.firstChild.appendChild(ga);
})();
```

Boundary / Stay native

Dirty

```
string script = @"<script type=""text/javascript"" defer=""defer"">
                    //<![CDATA[
                        var _gaq = _gaq || [];
                        _gaq.push(['_setAccount', '" + ws.GoogleAnalyticsID + @"']);
                        _gaq.push(['_trackPageview']);

                        (function() {
                        var ga = document.createElement('script');
                        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
                        '.google-analytics.com/ga.js';
                        ga.setAttribute('async', 'true');
                        document.documentElement.firstChild.appendChild(ga);
                        })();
                    //]]>
                    </script>";
this.Header.Controls.Add(new LiteralControl("\r\n" + script));
```

Clean

```
<!--In document head-->
<script type="text/javascript">
    var WebSiteSetup = { "GoogleAnalyticsKey": "JDSGI832JDUG9831" };
</script>
```

```
//In GoogleAnalytics.js
var _gaq = _gaq || [];
_gaq.push(['_setAccount', WebSiteSetup.GoogleAnalyticsKey]);
_gaq.push(['_trackPageview']);

(function () {
   var ga = document.createElement('script');
   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
   '.google-analytics.com/ga.js';
   ga.setAttribute('async', 'true');
   document.documentElement.firstChild.appendChild(ga);
})();
```

Boundary / Stay native

Dirty

```
string script = @"<script type=""text/javascript"" defer=""defer"">
                    //<![CDATA[
                        var _gaq = _gaq || [];
                        _gaq.push(['_setAccount', '" + ws.GoogleAnalyticsID + @"']);
                        _gaq.push(['_trackPageview']);

                        (function() {
                        var ga = document.createElement('script');
                        ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
                        '.google-analytics.com/ga.js';
                        ga.setAttribute('async', 'true');
                        document.documentElement.firstChild.appendChild(ga);
                        })();
                    //]]>
                    </script>";
this.Header.Controls.Add(new LiteralControl("\r\n" + script));
```

Clean

```
<!--In document head-->
<script type="text/javascript">
    var WebSiteSetup = { "GoogleAnalyticsKey": "JDSGI832JDUG9831" };
</script>
```

```
//In GoogleAnalytics.js
var _gaq = _gaq || [];
_gaq.push(['_setAccount', WebSiteSetup.GoogleAnalyticsKey]);
_gaq.push(['_trackPageview']);

(function () {
   var ga = document.createElement('script');
   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
   '.google-analytics.com/ga.js';
   ga.setAttribute('async', 'true');
   document.documentElement.firstChild.appendChild(ga);
})();
```

Boundary / Stay native

Separation of Concern

Reusable

Cached

2. Maximize signal to noise ratio

Signal
Noise

Maximize Signal

Logic that follows the **TED** rule

Terse

Expressive

Do one thing

Noise

High cyclomatic complexity
Large classes
Long methods
Duplication code
Dead code/ Zombie code
Poorly named/structures
Unnecessary comments

https://sourcemaking.com/refactoring/smells

3. Self-documenting code

High cyclomatic complexity
Large classes

3. Self-documenting code

Understanding the original programmer’s intent is the most difficult problem

Writing the better code

Clear intent
Layers of abstractions
Format for readability
Favour code over comments

Naming

Dirty

```
List<decimal> p = new List<decimal>() { 5.50m, 10.48m, 12.69m };
decimal t = 0;
foreach (var i in p)
{
    t += i;
}

return t;
```

Could you read this book?

P was very angry with G for insulting her M. G kicked P in the A. He slept on the C.

Clean

```
List<decimal> prices = new List<decimal>() { 5.50m, 10.48m, 12.69m };
decimal total = 0;
foreach (var price in prices)
{
    total += price;
}

return total;
```

Naming Classes

Dirty

- WebsiteBO
- Utility
- Common
- MyFunctions
- JimmysObjects
- *Manager / *Processor/*Info

Clean

- User
- Account
- QueryBuilder
- ProductRepository

Guidelines:

1. Noun
2. Be specific
3. Single Responsibility
4. Avoid generic suffixes

Specific names lead to smaller more cohesive classes

Naming Classes

Dirty

- WebsiteBO
- Utility
- Common
- MyFunctions
- JimmysObjects
- *Manager / *Processor/*Info

Guidelines:

1. Noun
2. Be specific
3. Single Responsibility
4. Avoid generic suffixes

Clean

- User
- Account
- QueryBuilder
- ProductRepository

Specific names lead to smaller more cohesive classes

Method name should say it all

Say what?

- Get
- Process
- Pending
- Dolt
- Start
- On_Init, Page_Load, etc.

Right on.

- GetRegisteredUsers
- IsValidSubmission
- ImportDocument
- SendEmail

Refactor until the method name completely describes what it does

Naming Variables

Avoid abbreviate name

Boolean names should sound like true/false questions

Dirty

- open
- start
- status
- login

```
if (login)
{

}
```

Clean

- isOpen
- done
- isActive
- loggedIn

```
if (loggedIn)
{

}
```

Naming Variables

Be symmetrical

Dirty

- on/disable
- quick/slow
- lock/open
- slow/max

Clean

- on/off
- fast/slow
- lock/unlock
- min/max

Conditional

Use positive conditionals

Dirty

```
if (!isNotLoggedIn)
```

Clean

```
if (loggedIn)
```

Magic Number !!

Dirty

```
if (age > 21)
{
    //body here
}
```

Clean

```
const int legalDrinkingAge = 21;
if (age > legalDrinkingAge)
{
    //body here
}
```

Dirty

```
if (status == 2)
{
    //body here
}
```

Clean

```
if (status == Status.Active)
{
    //body here
}
```

Complex conditionals

```
if (car.Year > 1980
    && (car.Make == "Ford" || car.Make == "Chevrolet")
    && car.Odometer < 100000
    && car.Vin.StartsWith("V2") || car.Vin.StartsWith("IA3"))
{
    //do lots of things here.
}
```

Problems and Solutions ?

Intermediate variables

Dirty

```
if (employee.Age > 55
    && employee.YearsEmployed > 10
    && employee.IsRetired == true)
{
    //logic here
}
```

What question is this trying to answer?

Clean

```
bool eligibleForPension = employee.Age > MinRetirementAge
    && employee.YearsEmployed > MinPensionEmploymentYears
    && employee.IsRetired;
```

Encapsulate complex condition

Dirty

```
//Check for valid file extensions. Confirm admin or active
if (fileExtension == "mp4" ||
    fileExtension == "mpg" ||
    fileExtension == "avi")
    && (isAdmin || isActiveFile);
```

Principle: Favor expressive code over comments

Clean

```
if (ValidFileRequest(fileExtension, isActiveFile, isAdmin))
```

```
private bool ValidFileRequest(string fileExtension, bool isActiveFile, bool isAdmin)
{
    return (fileExtension == "mp4" ||
        fileExtension == "mpg" ||
        fileExtension == "avi")
        && (isAdmin || isActiveFile);
}
    return validFileType && userIsAllowedToViewFile;
}
```

Polymorphism

Dirty

```
public void LoginUser(User user)
{
    switch (user.Status)
    {
        case Status.Active:
            //logic for active users
            break;
        case Status.Inactive:
            //logic for inactive users
            break;
        case Status.Locked:
            //logic for locked users
            break;
    }
}
```

Clean

```
public void LoginUser(User user)
{
    user.Login();
}
```

```
public abstract class User
{
    public string FirstName;
    public string LastName;
    public Status Status;
    public int AccountBalance;

    public abstract void Login();
}
```

Methods

When to create a method ?

Duplication
Unclear intent
More tasks

Avoid nested if and loop !!

```
function iIds(startAt, showSessionRoot, iNewNmVal, endActionsVal, iStringVal, seqProp, htmlEncodeRegEx) {
    if (SbUtil.dateDisplayType === 'relative') {
        iRange();
    } else {
        iSelActionType();
    }
    iStringVal = notifyWindowTab;
    startAt = addSessionConfigs.sbRange();
    showSessionRoot = addSessionConfigs.elHiddenVal();
    var headerDataPrevious = function(tabArray, iNm) {
        iPredicateVal.SBDB.deferCurrentSessionNotifyVal(function(evalOutMatchedTabUrlsVal) {
            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                iPredicateVal.SBDB.normalizeTabList(function(appMsg) {
                    if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                        iPredicateVal.SBDB.detailTxt(function(evalOrientationVal) {
                            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                iPredicateVal.SBDB.neutralizeWindowFocus(function(iTokenAddedCallback) {
                                    if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                        iPredicateVal.SBDB.evalSessionConfig2(function(sessionNm) {
                                            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                iPredicateVal.SBDB.iWindow2TabIdx(function(iURLsStringVal) {
                                                    if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                        iPredicateVal.SBDB.idx7Val(undefined, iStringVal, function(getWindowIndex) {
                                                            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                                addTabList(getWindowIndex.rows, iStringVal, showSessionRoot && showSessionRoot.length > 0 ? show
                                                                    if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                                        evalSAllowLogging(tabArray, iStringVal, showSessionRoot && showSessionRoot.length > 0 ?
                                                                            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                                                BrowserAPI.getAllWindowsAndTabs(function(iSession1Val) {
                                                                                    if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                                                        SbUtil.currentSessionSrc(iSession1Val, undefined, function(initCurrentSe
                                                                                            if (!htmlEncodeRegEx || htmlEncodeRegEx == iContextTo) {
                                                                                                addSessionConfigs.render(matchText(iSession1Val, iStringVal, eva
                                                                                                    id: -13,
                                                                                                    unfilteredWindowCount: initCurrentSessionCache,
                                                                                                    filteredWindowCount: iCtrl,
                                                                                                    unfilteredTabCount: parseTabConfig,
                                                                                                    filteredTabCount: evalRegisterValue5Val
                                                                                                }] : [], cacheSessionWindow, evalRateActionQualifier, undefined,
                                                                                                    if (seqProp) {
                                                                                                        seqProp();
                                                                                                    }
                                                                                                });
                                                                                            }
                                                                                        });
                                                                                    }
                                                                                });
                                                                            }
                                                                        });
                                                                    }
                                                                }, showSessionRoot && showSessionRoot.length > 0 ? showSessionRoot : startAt ? [startAt] : []);
                                                            }
                                                        });
                                                    }
                                                });
                                            }
                                        });
                                    }
                                });
                            }
                        });
                    }
                });
            }
        });
    }
});
```

Avoid nested if and loop !!

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

icompile.eladkarako.com

Method …

Extract method
Fail fast
Return early

Extract to new method

Before

```
if
    if
        while
            do
            some
            complicated
            thing
        end while
    end if
end if
```

After

```
if
    if
        doComplicatedThing()
    end if
end if
```

```
doComplicatedThing()
{
    while
        do some complicated thing
    end while
}
```

Fail fast

Dirty

```
public void RegisterUser(string username, string password)
{
    if (!string.IsNullOrWhiteSpace(username))
    {
        if (!string.IsNullOrWhiteSpace(password))
        {
            //register user here.
        }
        else
        {
            throw new ArgumentException("Username is required.");
        }
    }
    else
    {
        throw new ArgumentException("Password is required");
    }
}
```

Clean

```
public void RegisterUser(string username, string password)
{
    if (string.IsNullOrWhiteSpace(username)) throw new ArgumentException("Username is required.");
    if (string.IsNullOrWhiteSpace(password)) throw new ArgumentException("Password is required");

    //register user here.
}
```

Return early

```
private bool ValidUsername(string username)
{
    bool isValid = false;

    const int MinUsernameLength = 6;
    if (username.Length >= MinUsernameLength)
    {
        const int MaxUsernameLength = 25;
        if (username.Length <= MaxUsernameLength)
        {
            bool isAlphaNumeric = username.All(Char.IsLetterOrDigit);
            if (isAlphaNumeric)
            {
                if (!ContainsCurseWords(username))
                {
                    isValid = IsUniqueUsername(username);
                }
            }
        }
    }
    return isValid;
}
```

Return early

```
private bool ValidUsername(string username)
{
    const int MinUsernameLength = 6;
    if (username.Length < MinUsernameLength) return false;

    const int MaxUsernameLength = 25;
    if (username.Length > MaxUsernameLength) return false;

    bool isAlphaNumeric = username.All(Char.IsLetterOrDigit);
    if (!isAlphaNumeric) return false;

    if (ContainsCurseWords(username)) return false;

    return IsUniqueUsername(username);
}
```

Try/Catch/Log = fail slow

Dirty

```
try
{
    RegisterSpeaker();
}
catch(Exception e)
{
    LogError(e);
}

EmailSpeaker();
```

Clean

```
RegisterSpeaker();
EmailSpeaker();
```

Try/Catch body

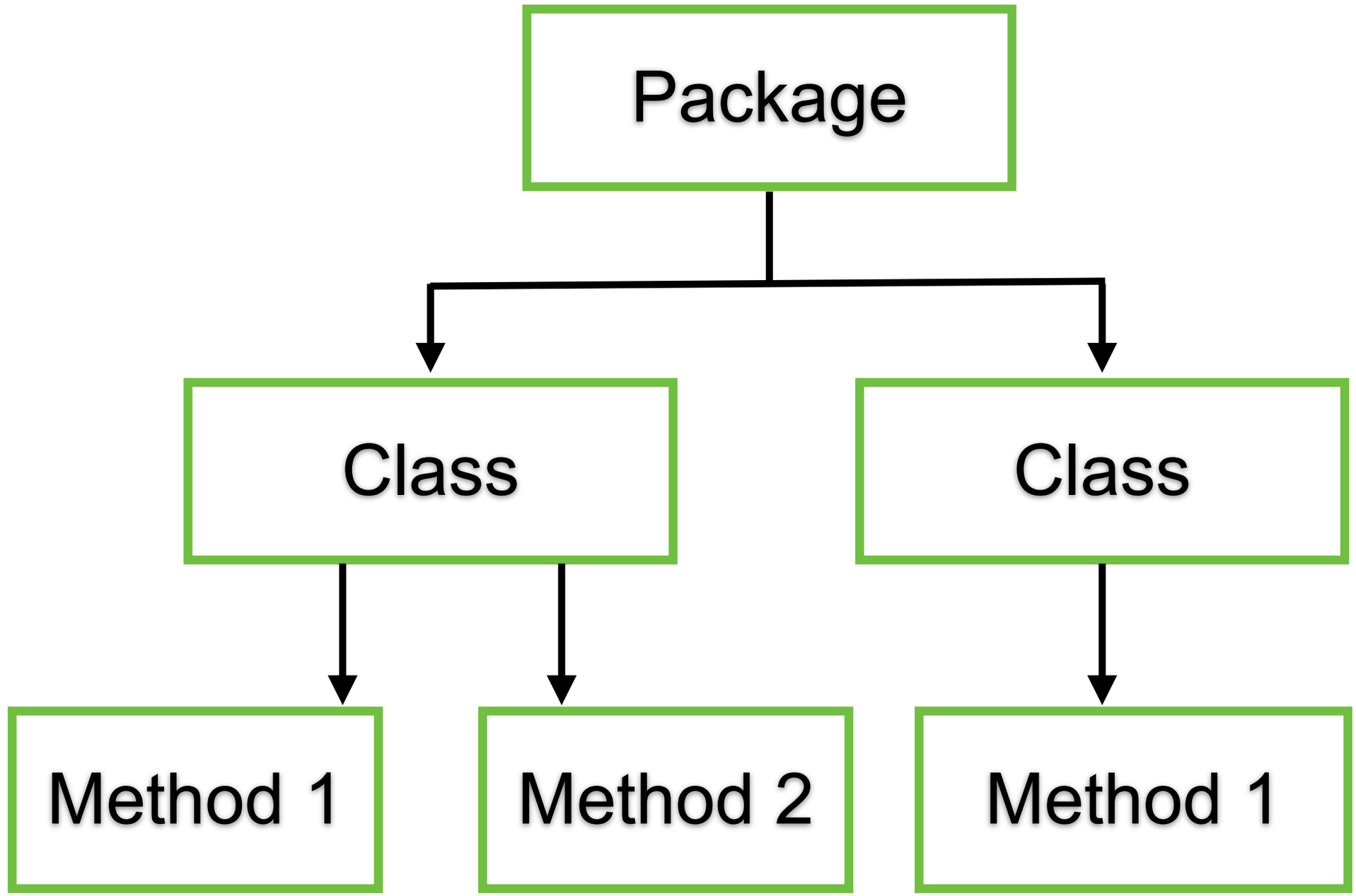
Dirty

```
try
{
    //many
    //lines
    //of
    //complicated
    //and
    //verbose
    //logic
    //here
}
catch (ArgumentOutOfRangeException)
{
    //do something here
}
```

Clean

```
try
{
    SaveThePlanet();
}
catch (ArgumentOutOfRangeException)
{
    //do something here
}

private void SaveThePlanet()
{
    //many
    //lines
    //of
    //complicated
    //and
    //verbose
    //logic
    //here
}
```

Classes



When to create a class ?

When to create a class ?

Abstract or real world
Low cohesion
Promote reuse
Reduce complexity
Clarify parameters

Stay clean

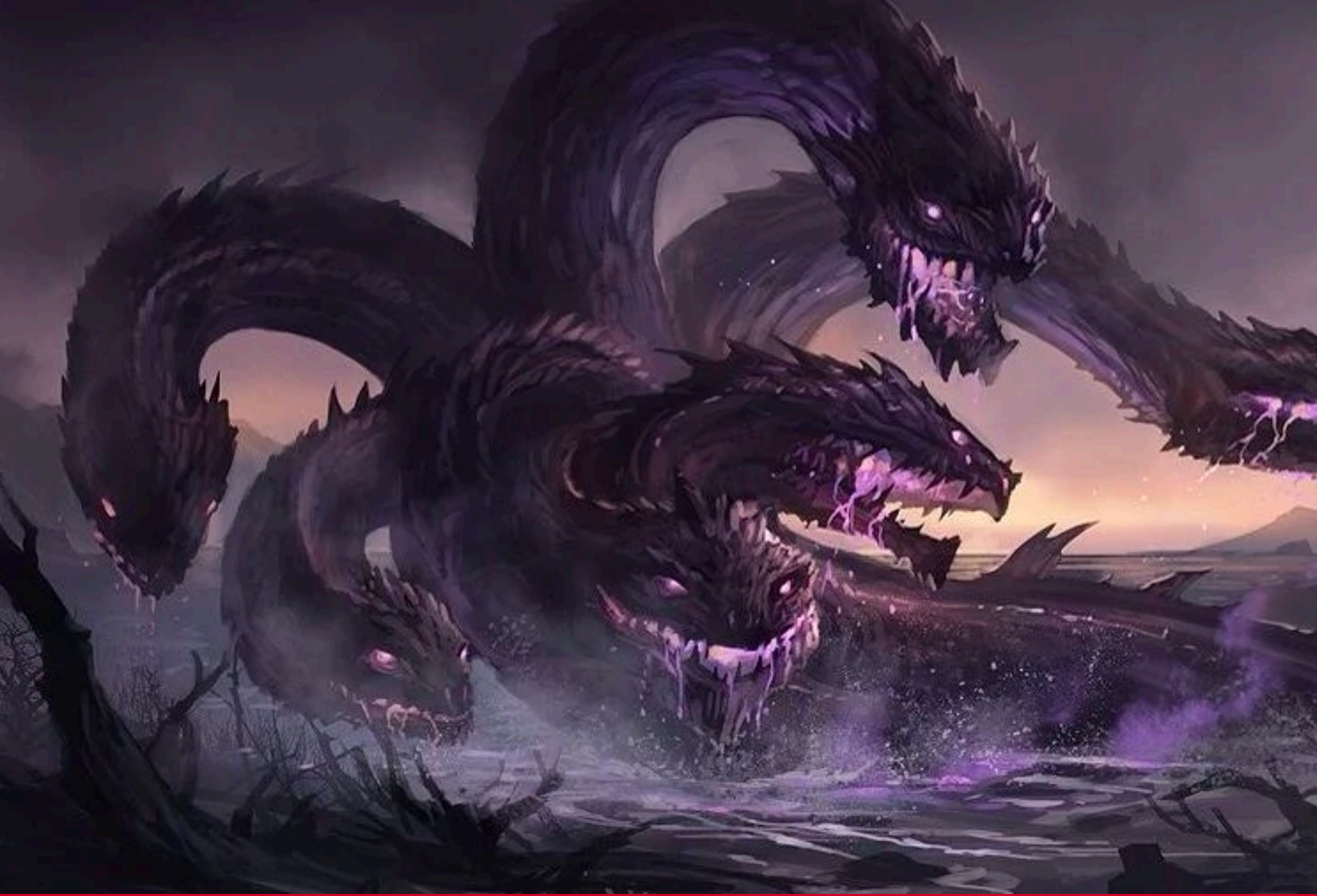
Trust in your code ?

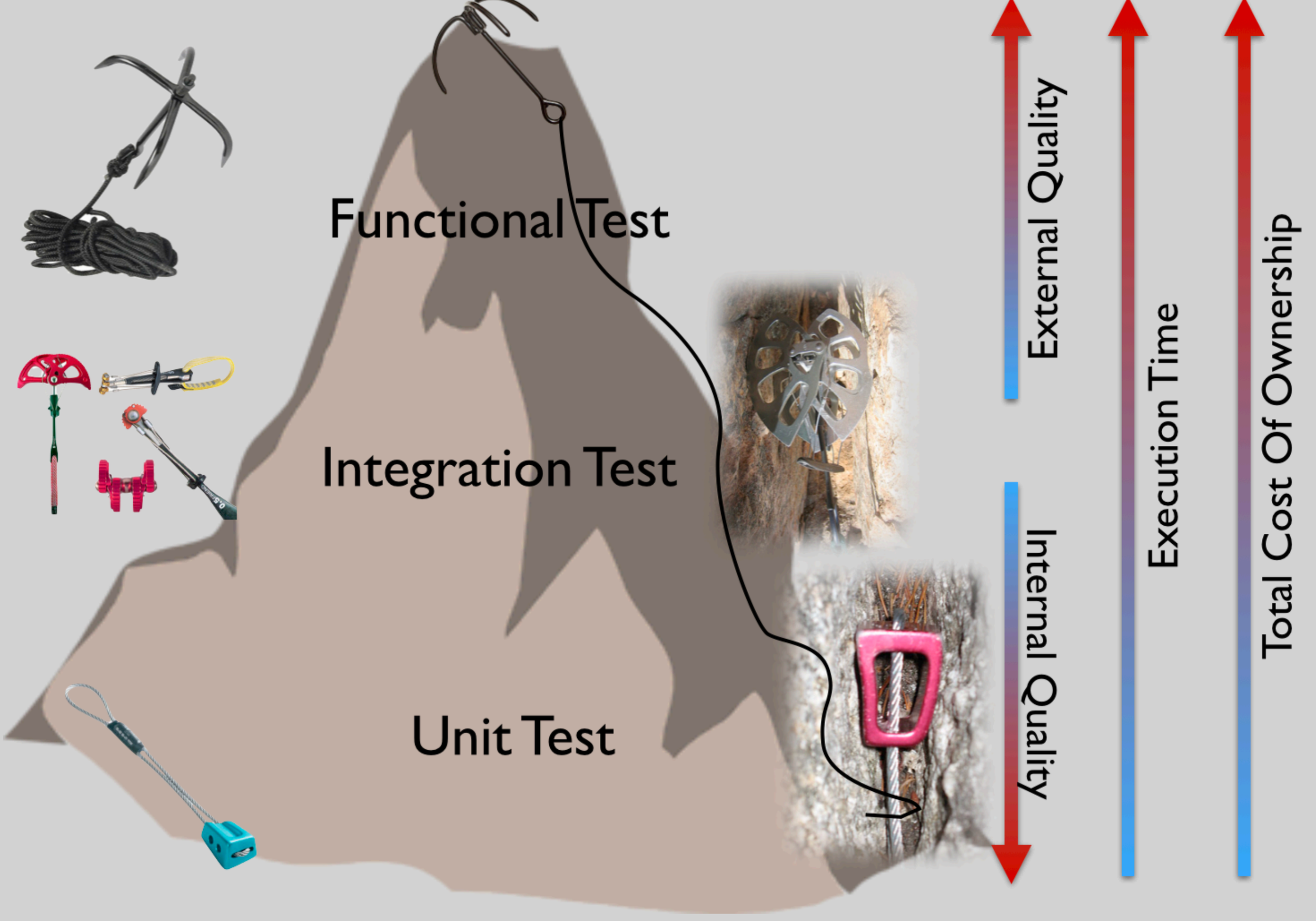
Write Unit tests

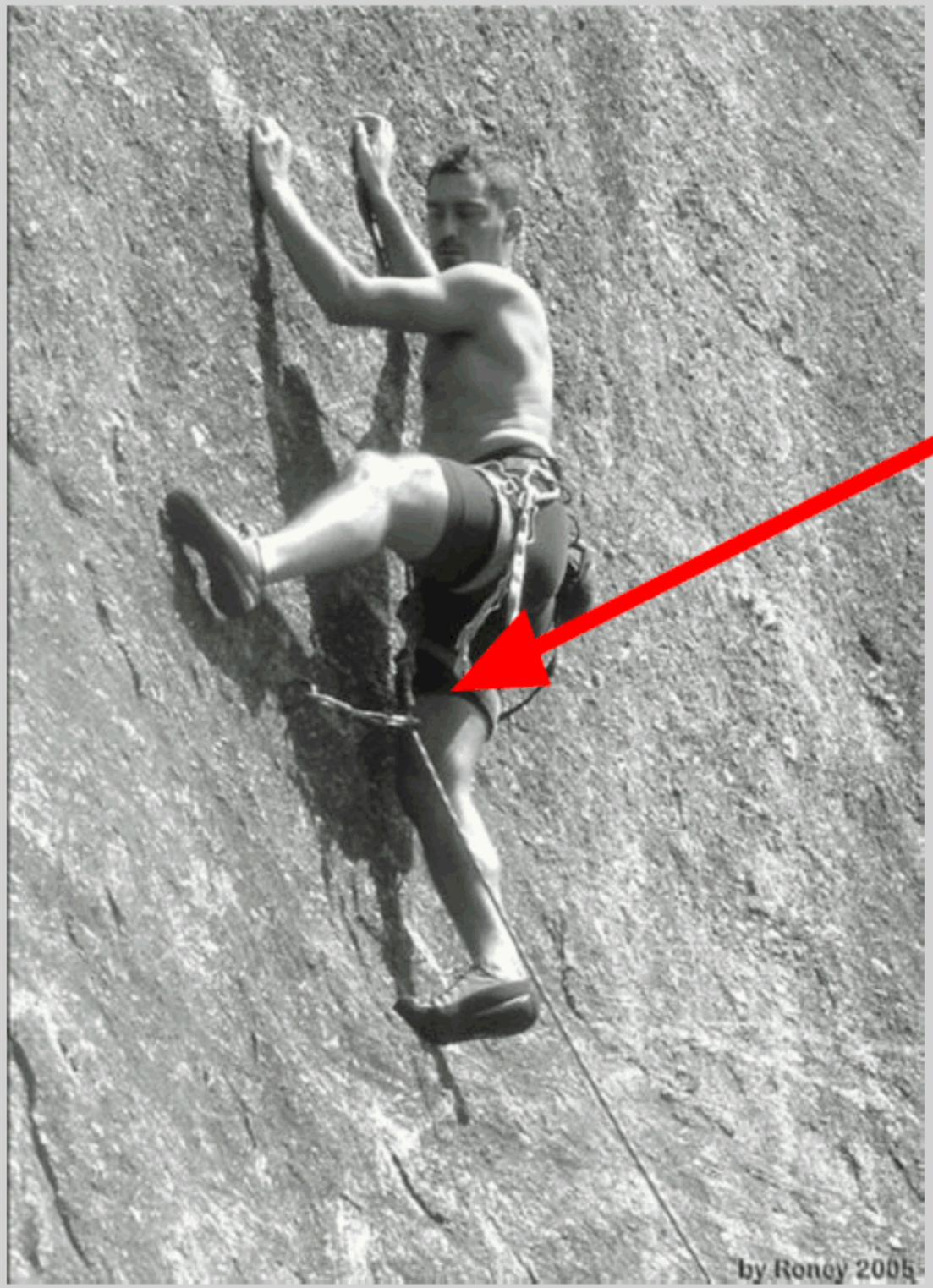
https://junit.org/

Hello with Unit testing

Your code not break anything ?



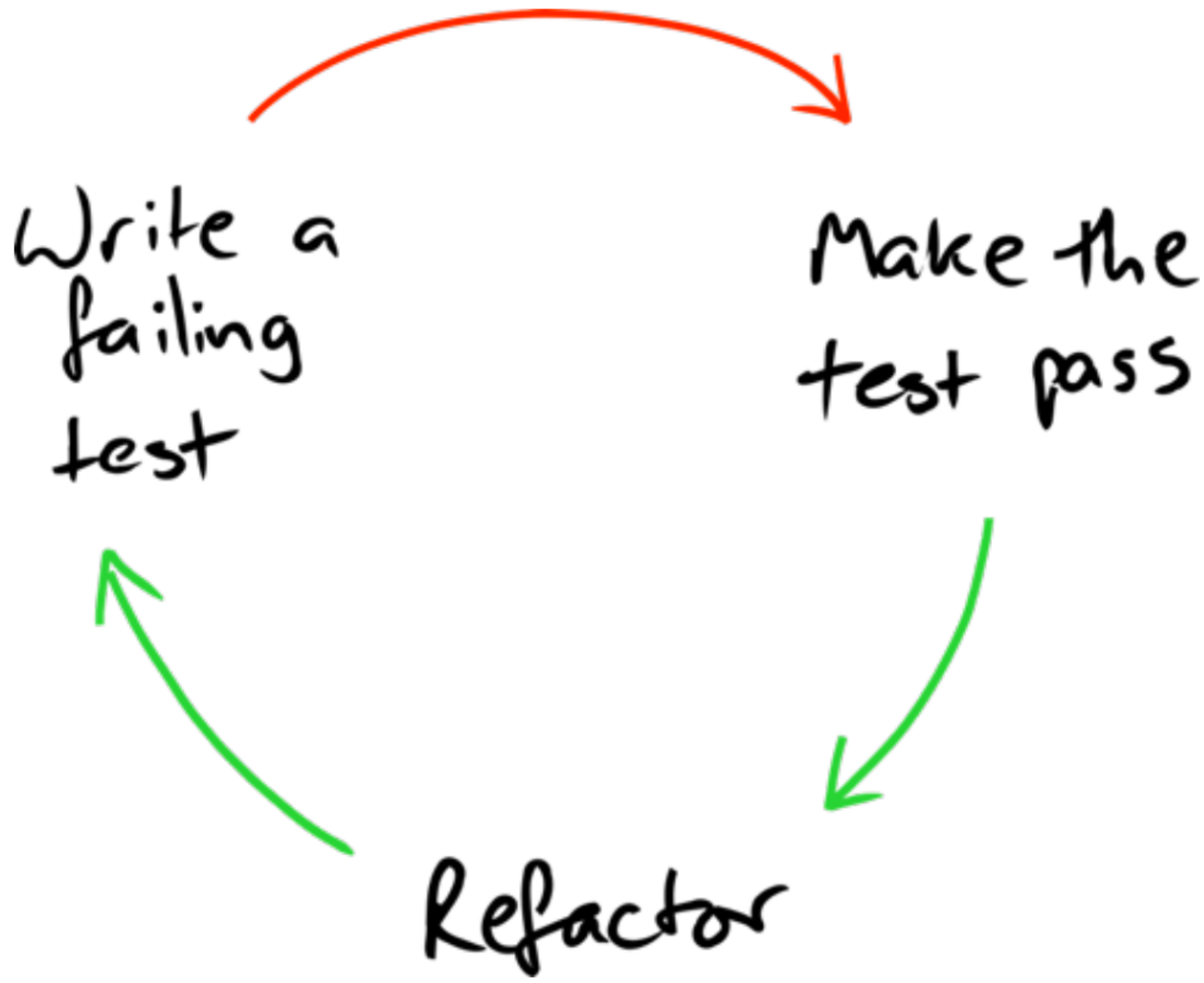




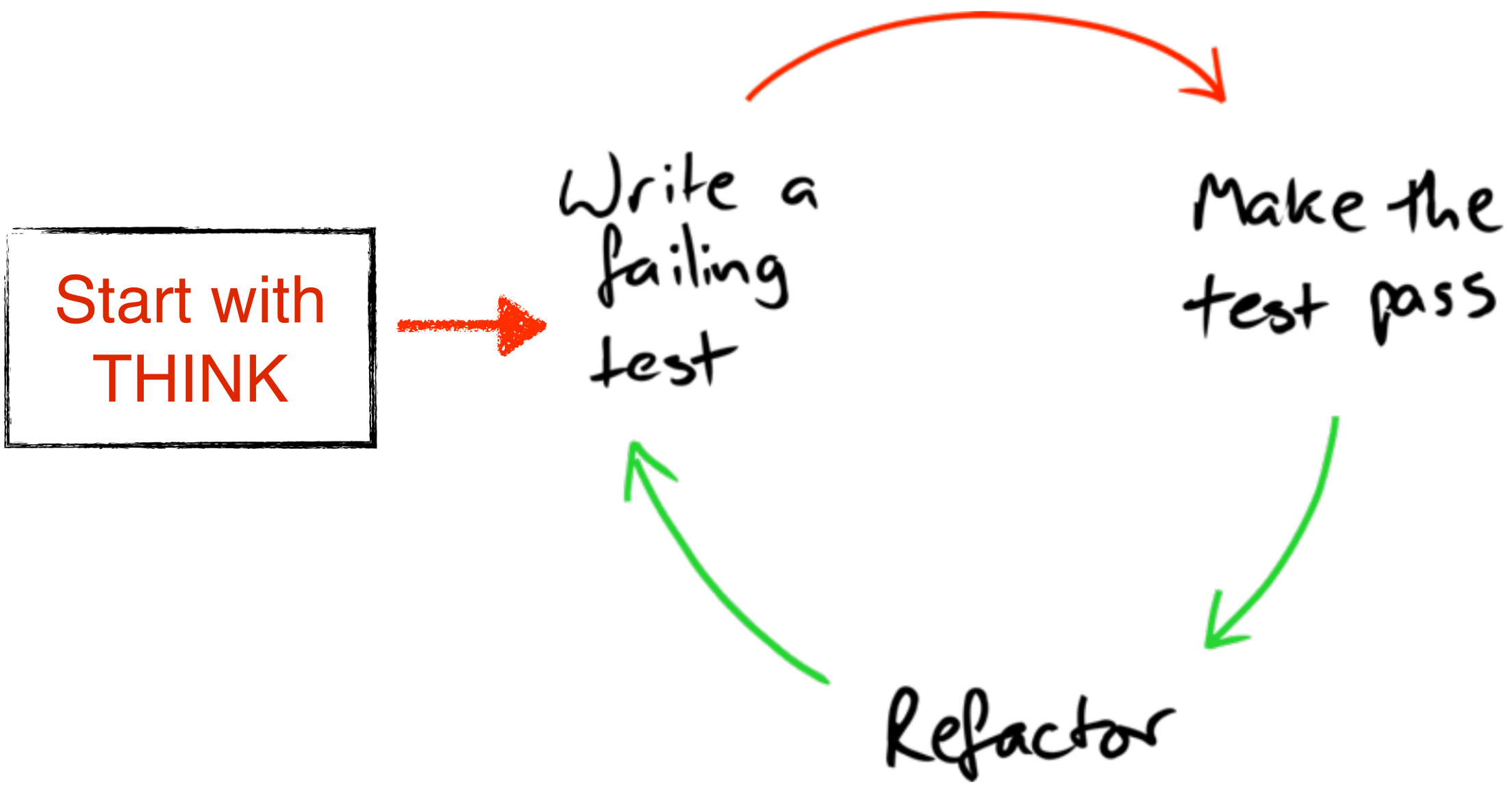
Good Unit Test

Fast
Isolated / **I**ndependent
Repeatable
Self-validating
Timely

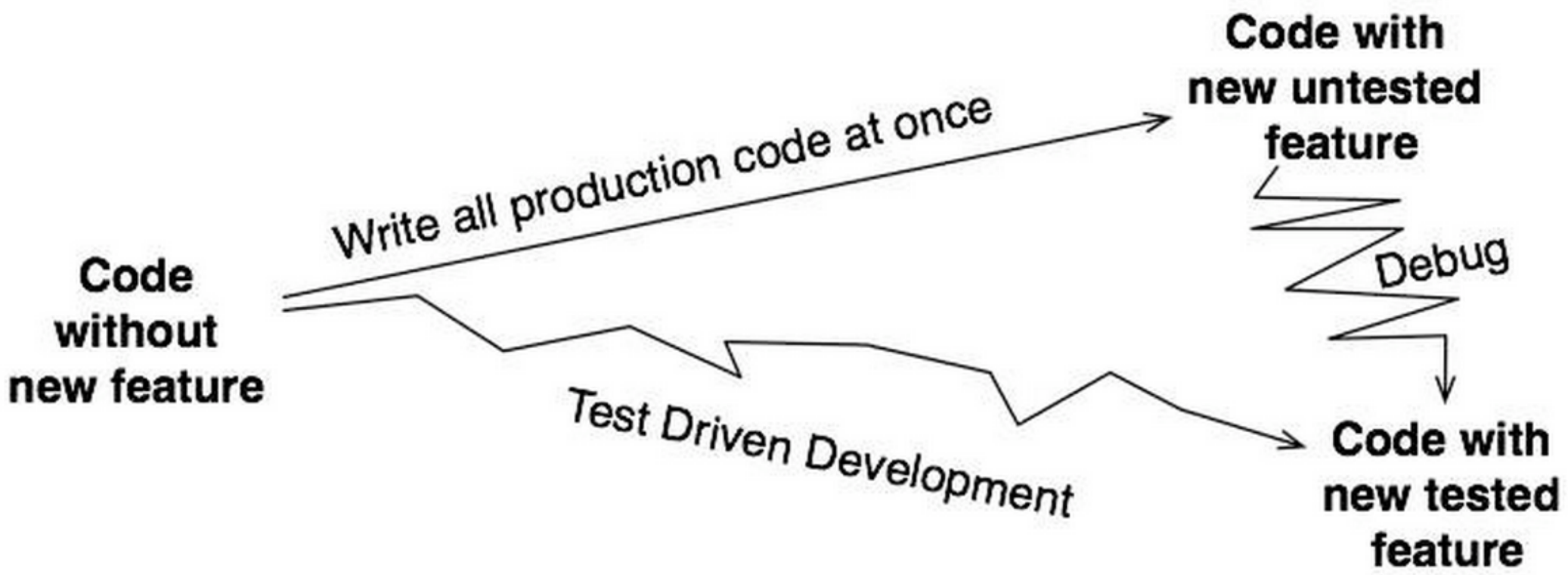
TDD



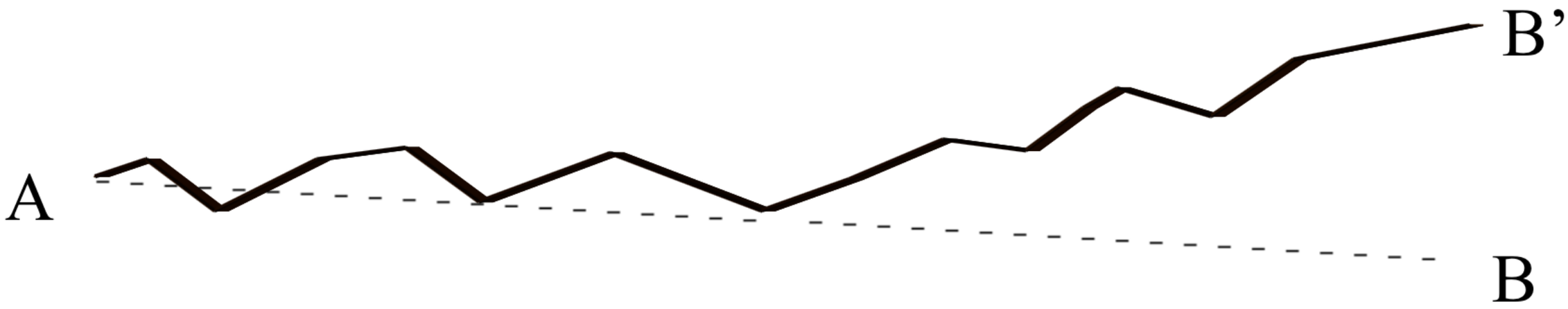
TDD



TDD vs DLP



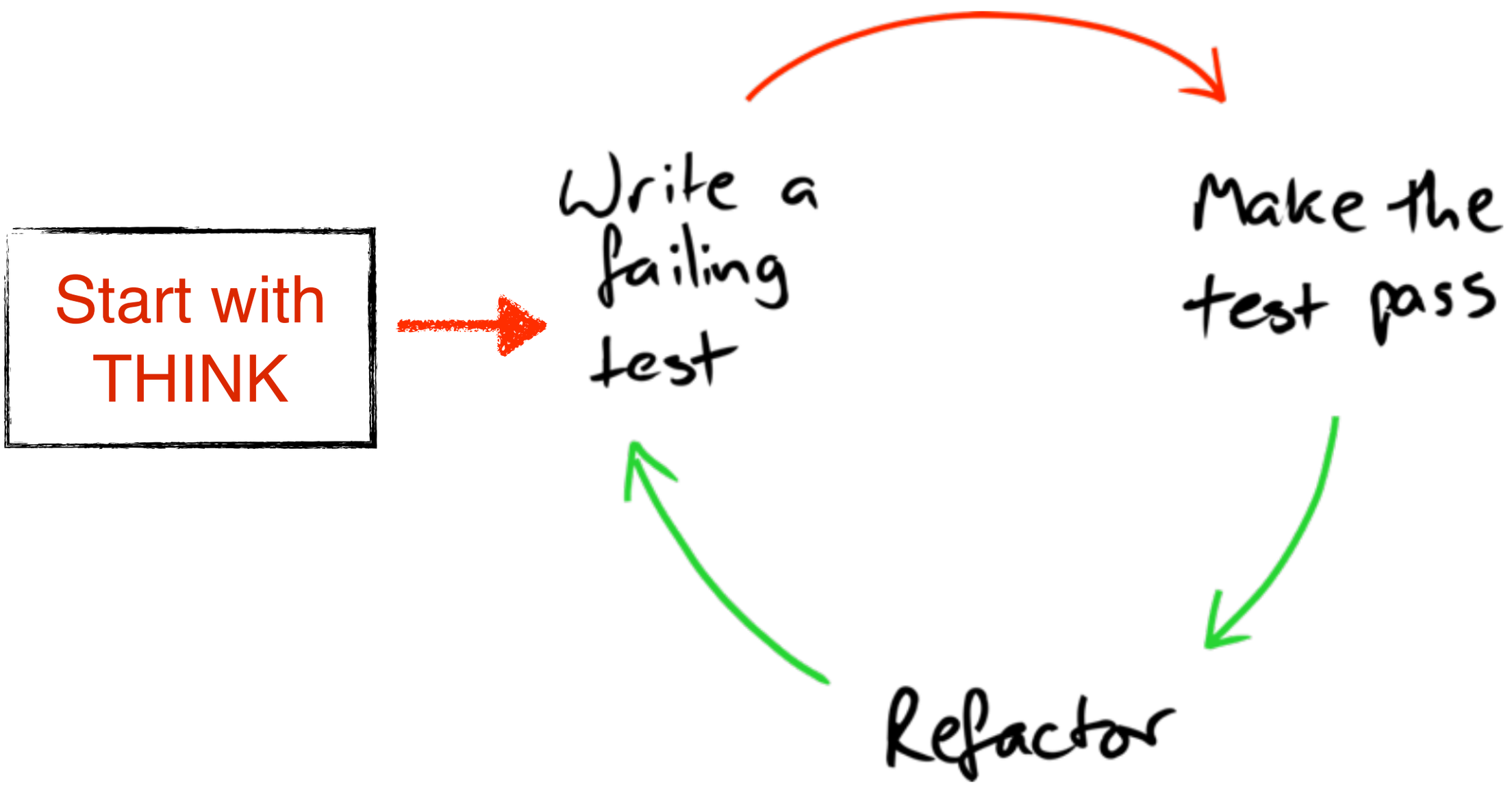
Start with small step



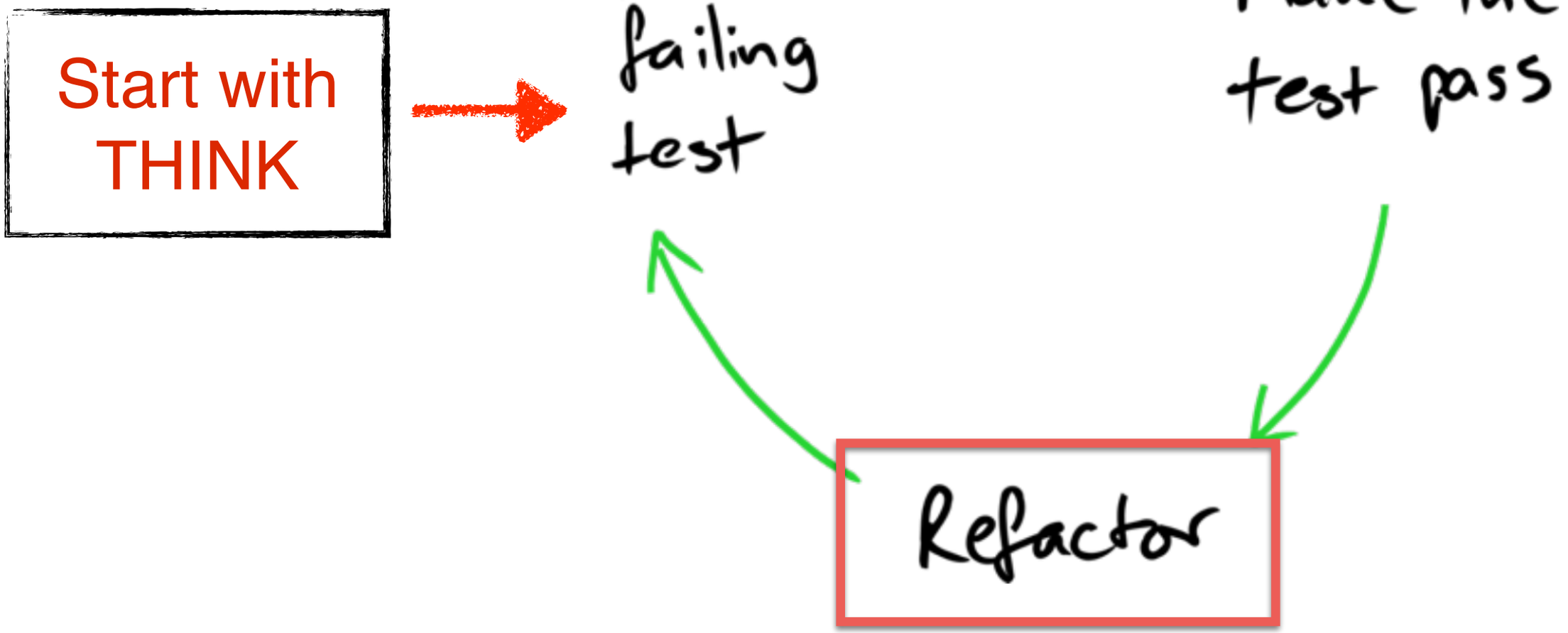
Workshop
Circular Buffer

Workshop Range

TDD



Refactor



Code Smell & Refactoring

https://sourcemaking.com/refactoring/smells

When to Refactor ?

When to Refactor ?

You need to work with this code
Difficult to change
Test coverage to protect from regression

No Broken Windows

Code Smell

https://sourcemaking.com/refactoring/smells

Code Smell

Code Smell

Don’t write S.T.U.P.I.D code

Singleton
Tight coupling
Untestability
Premature optimization
Indescriptive naming
Duplication

Code Smell Workshop

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

Working with Bad Code

https://github.com/up1/workshop-java-badcode

Object-Oriented Design

S.O.L.I.D principle

S.O.L.I.D principle

The foundation on which we can build
Clean and **maintainable** architecture

Problems

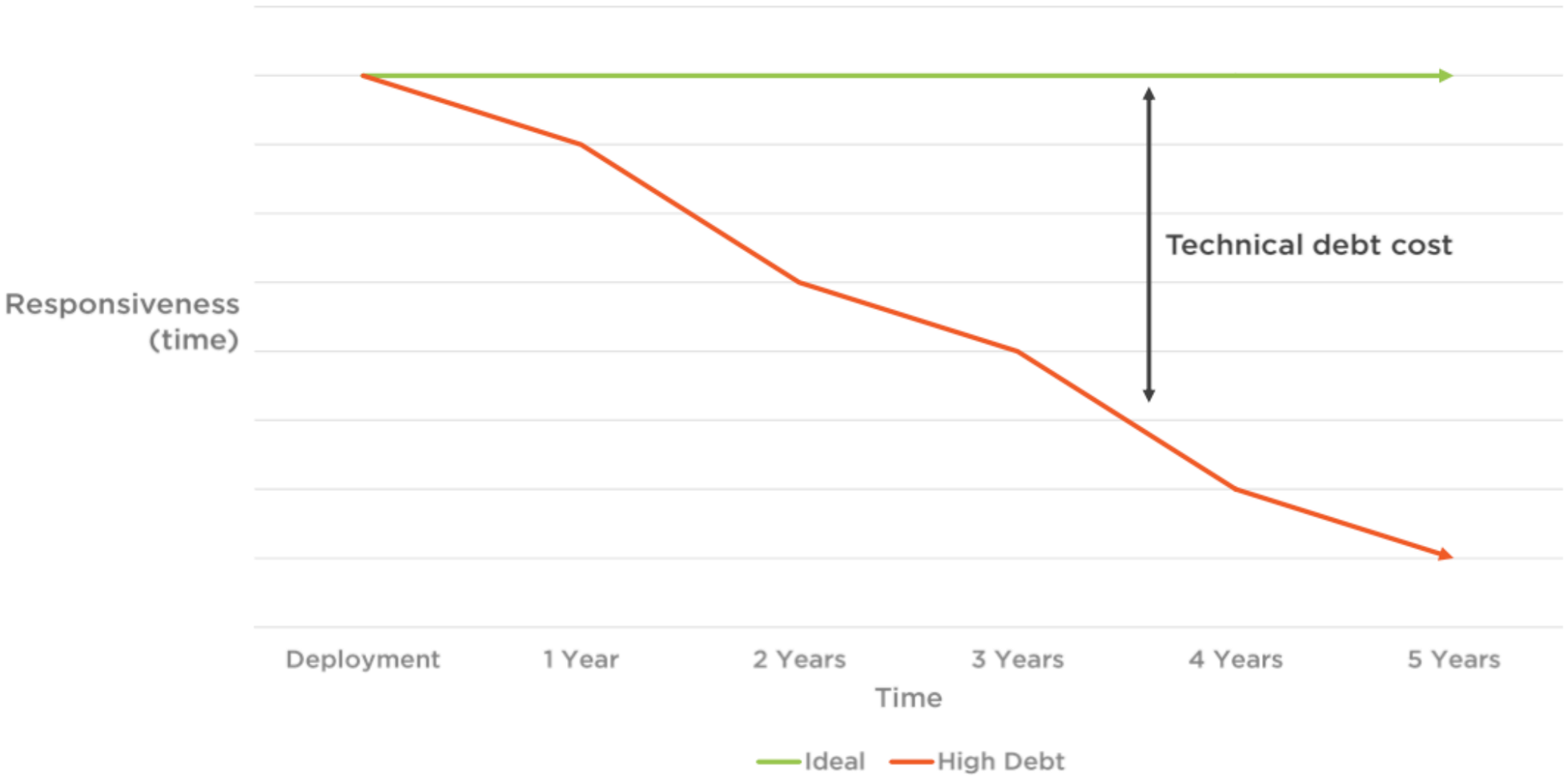
Change request
Implement change
Deploy application
Bugs in other system or sub-system

Technical Debt

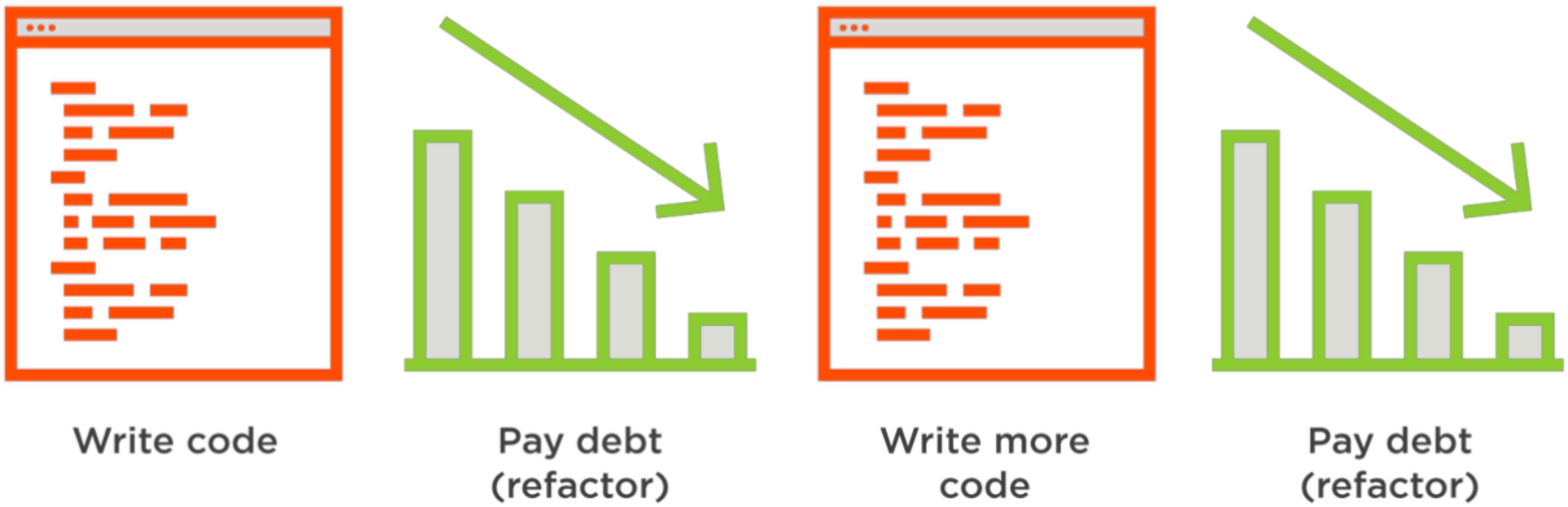
The cost of prioritizing
fast delivery over **code quality**
for long periods of time

Cost of Change

Customer Responsiveness



Control Technical Debt



SOLID principle

Help to keep technical debt under control

Goals of SOLID code

Easy to understand
Changes are faster and minimal risk level
Highly maintainable
Cost effective

Other ways

Continuous refactoring
Design patterns
Unit testing / TDD

Start with S.O.L.I.D

1. Single Responsibility Principle

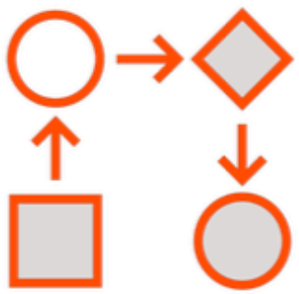
Responsibility

Business logic

User interface

Persistence

Logging

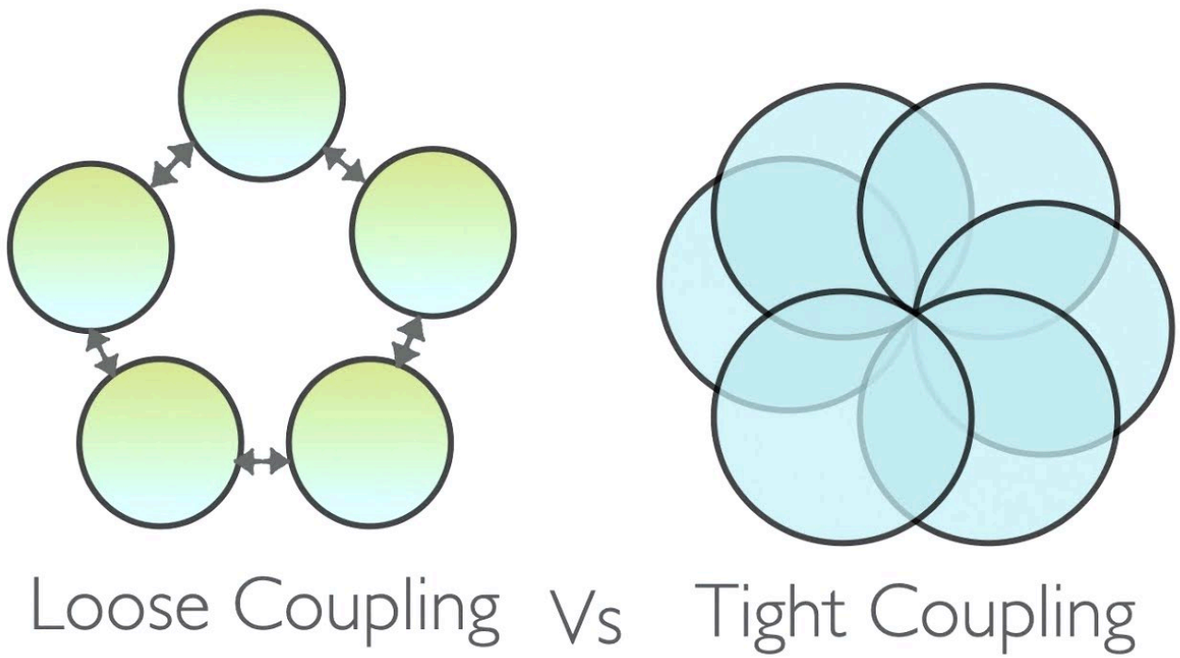


Orchestration

Users

Why SRP ?

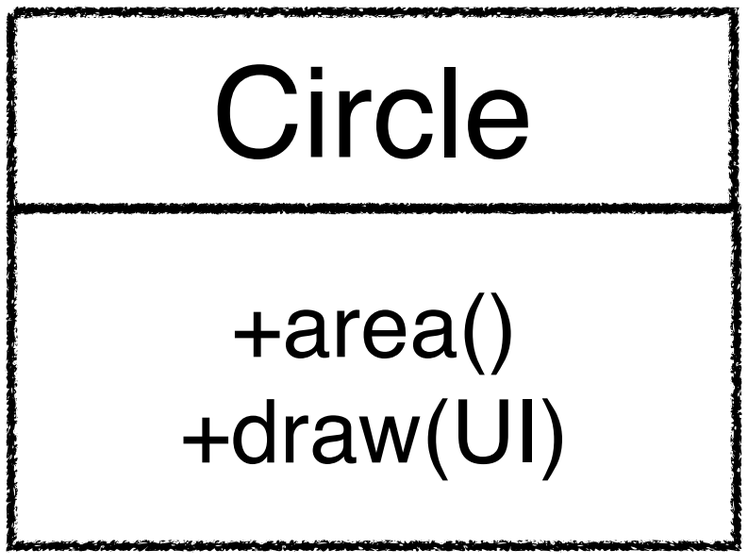
Easy to understand, change, fix and maintain
Classes are **less coupled**
More **testable** design



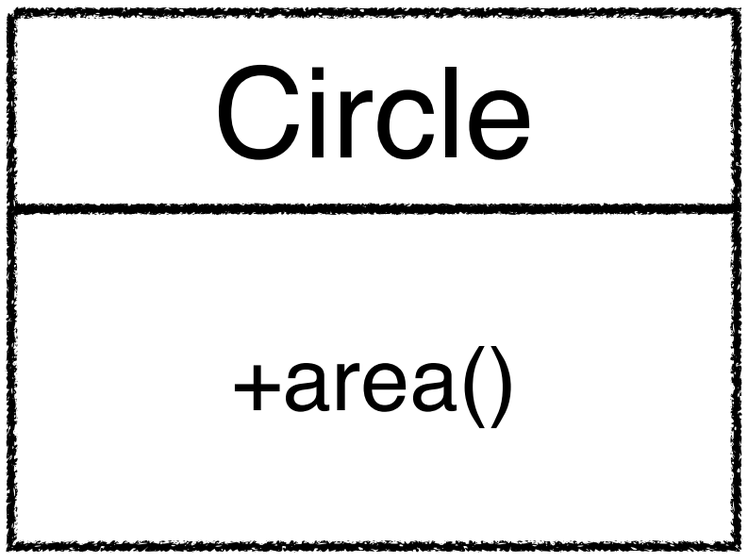
Identify multiple reason to change !!

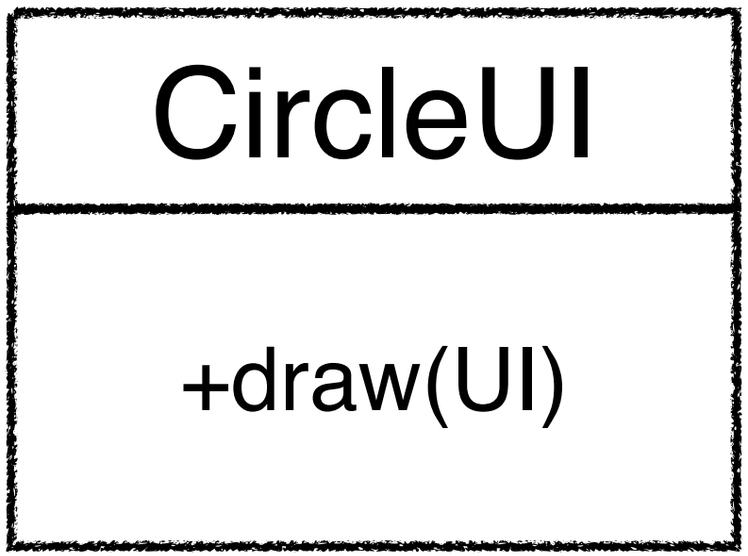
If/Switch
Monster/large method
God class
User !!

Example



Example



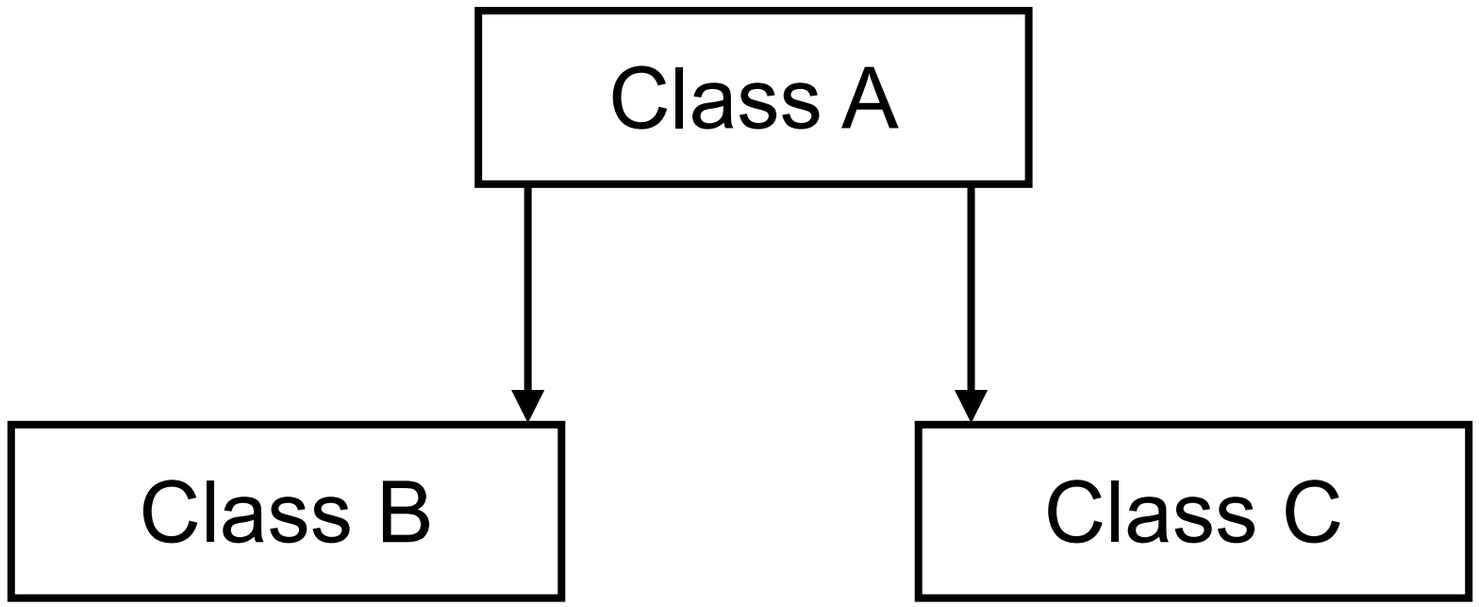


Demo and workshop

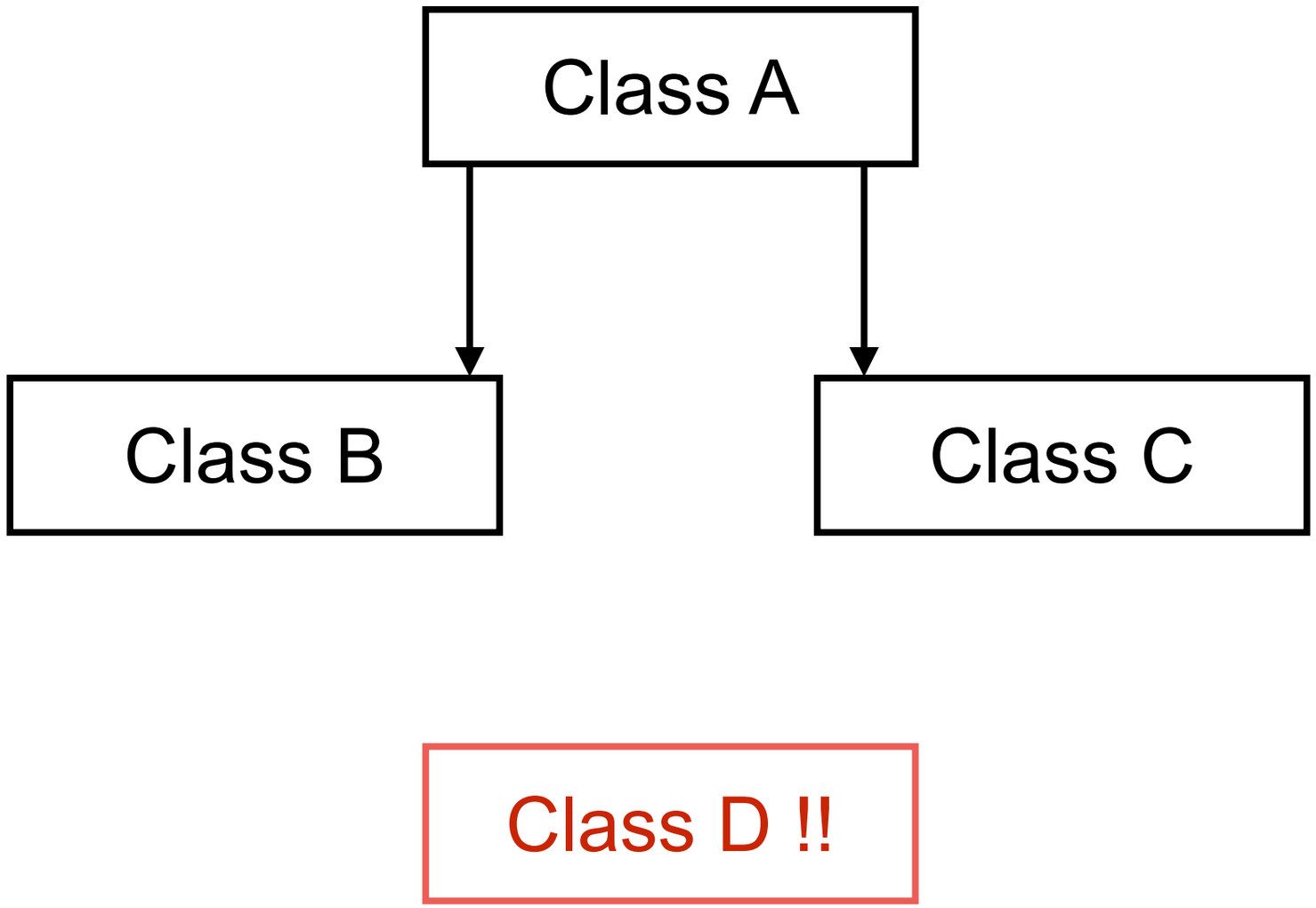
2. Open-Closed Principle

OCP is all about changes

Change == Risk !!



Change == Risk !!



Why OCP ?

Easy to add new feature with minimal cost

Minimize the risk

Enforce decouple by isolate changes in specific components

How to apply the OCP ?

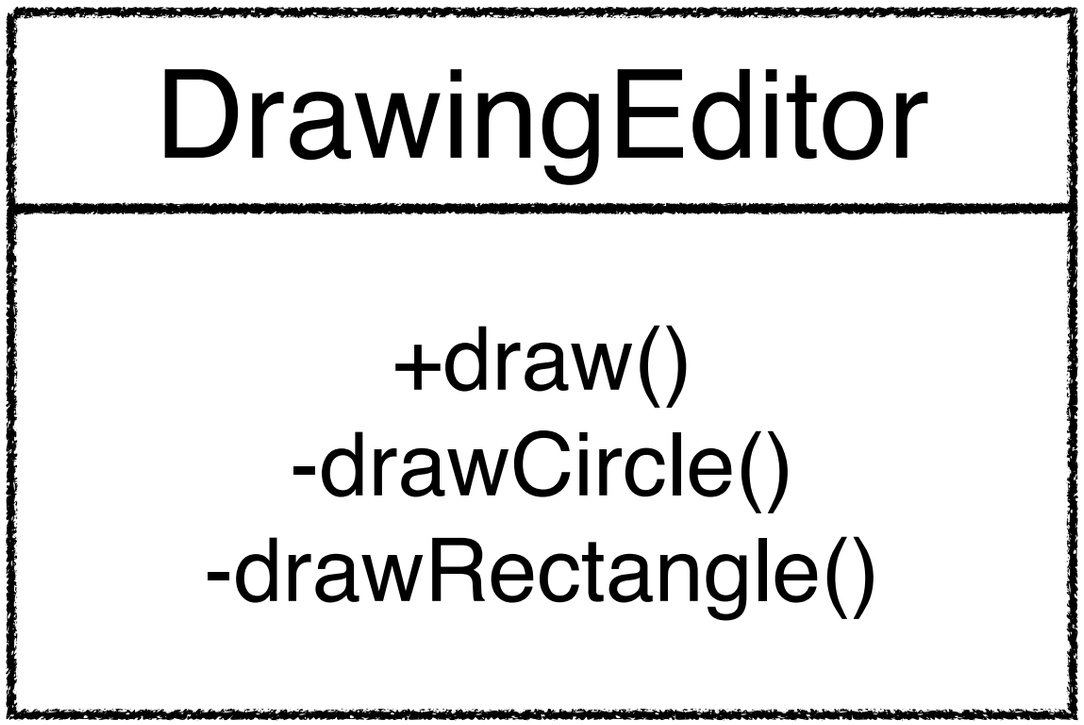
Start small

Change/Bug-driven development

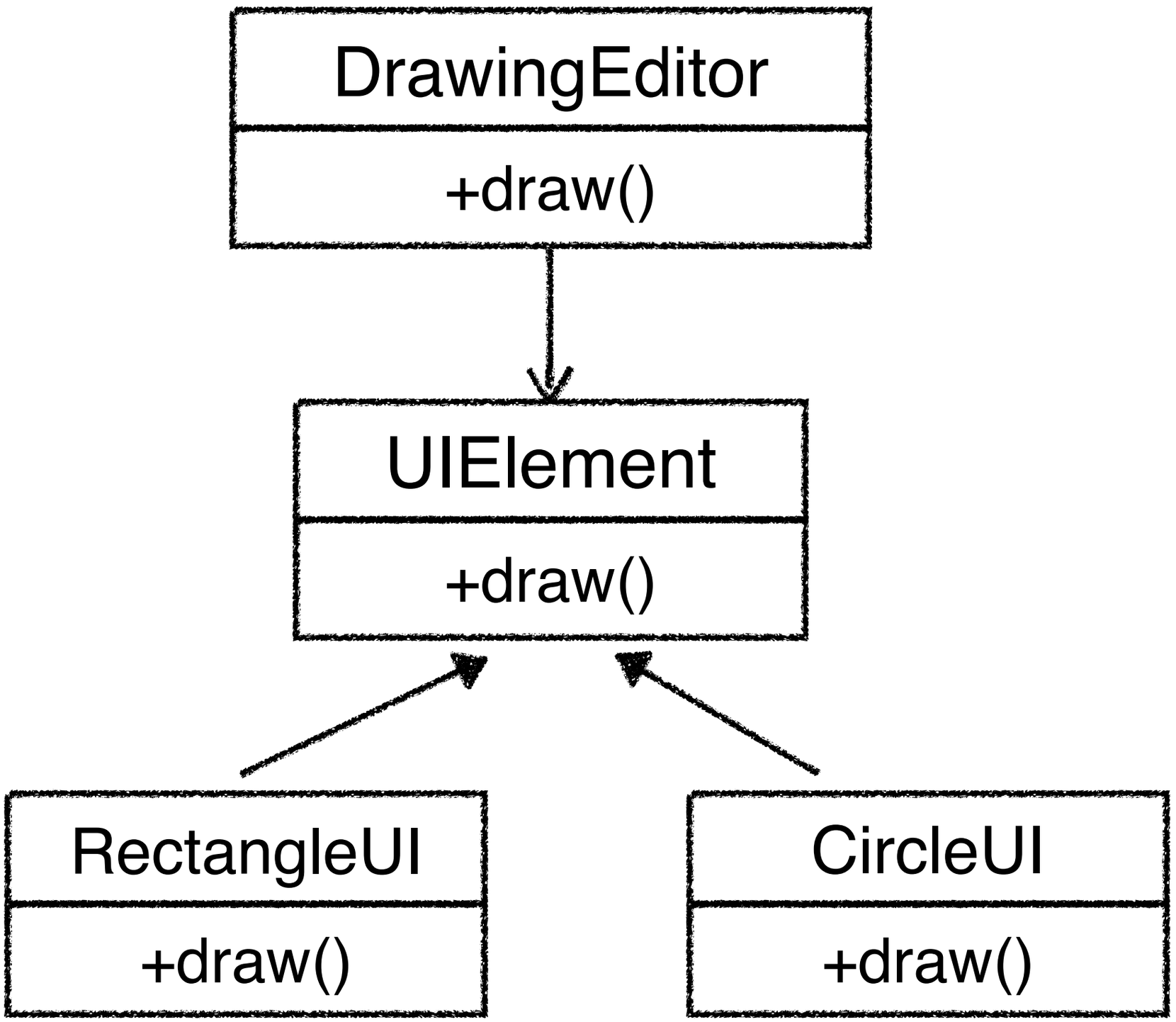
More changes !!

Many changes and Dynamic decision

Example



Example



Workshop

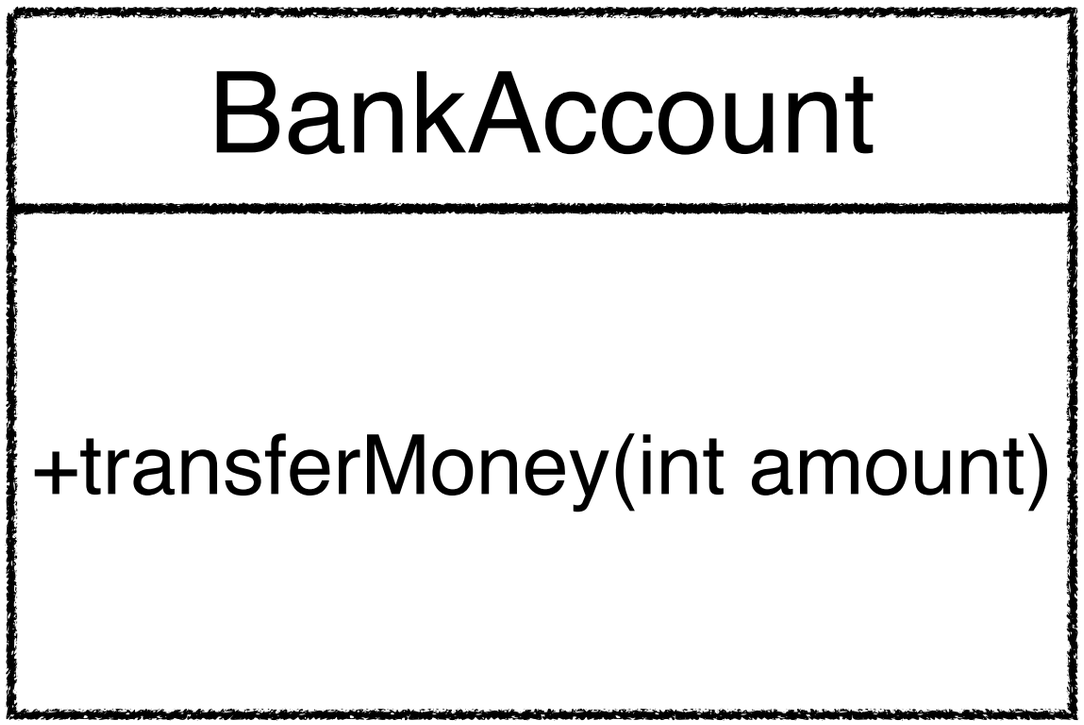
FizzBuzz

Input	Expected Result
1	1
2	2
3	Fizz
4	4
5	Buzz
6	Fizz
7	7
8	8
9	Fizz
10	Buzz
15	FizzBuzz

7 => HELLO

More workshop

Transfer money



Transfer money !!

Local

International

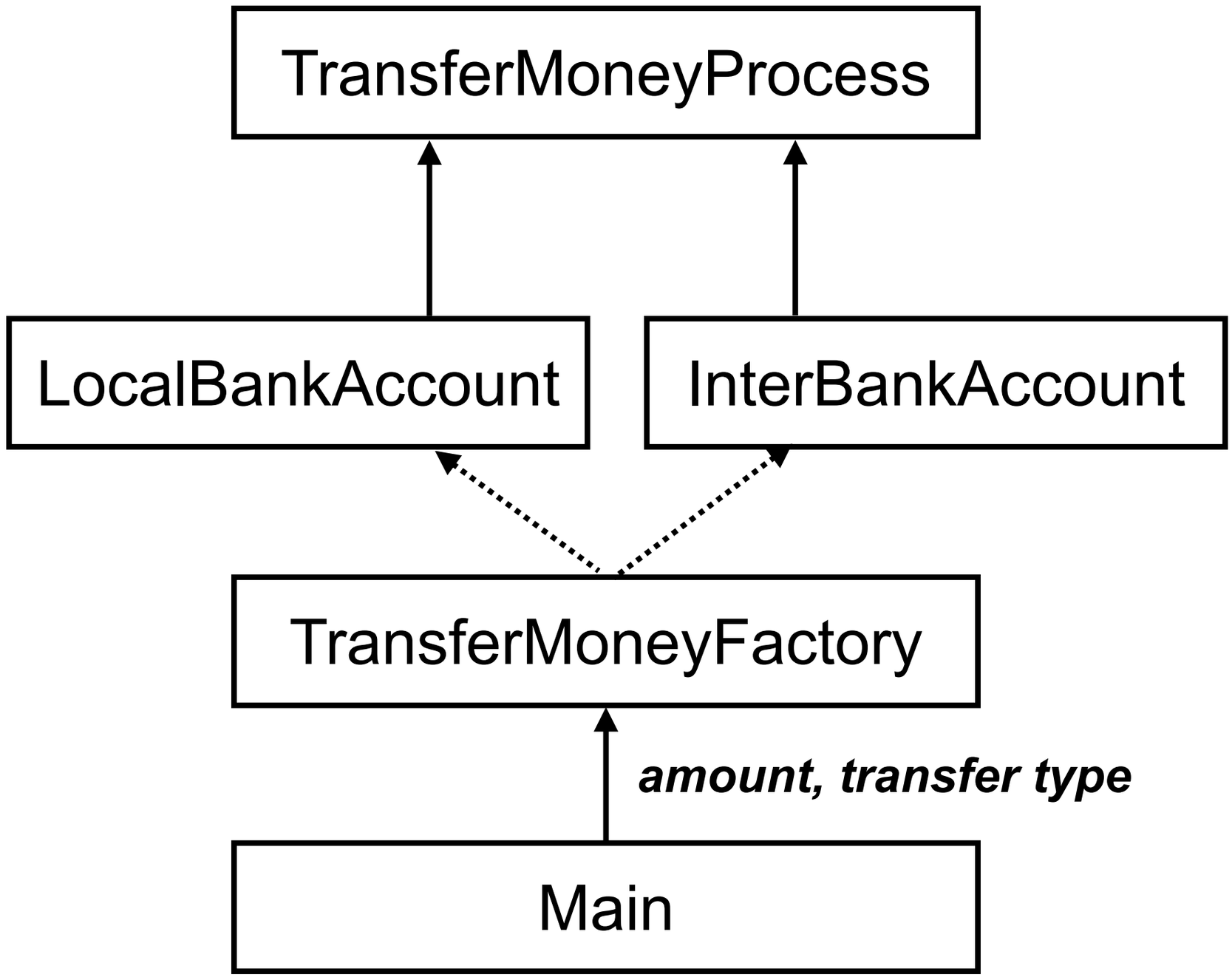
Transfer money !!

If-else

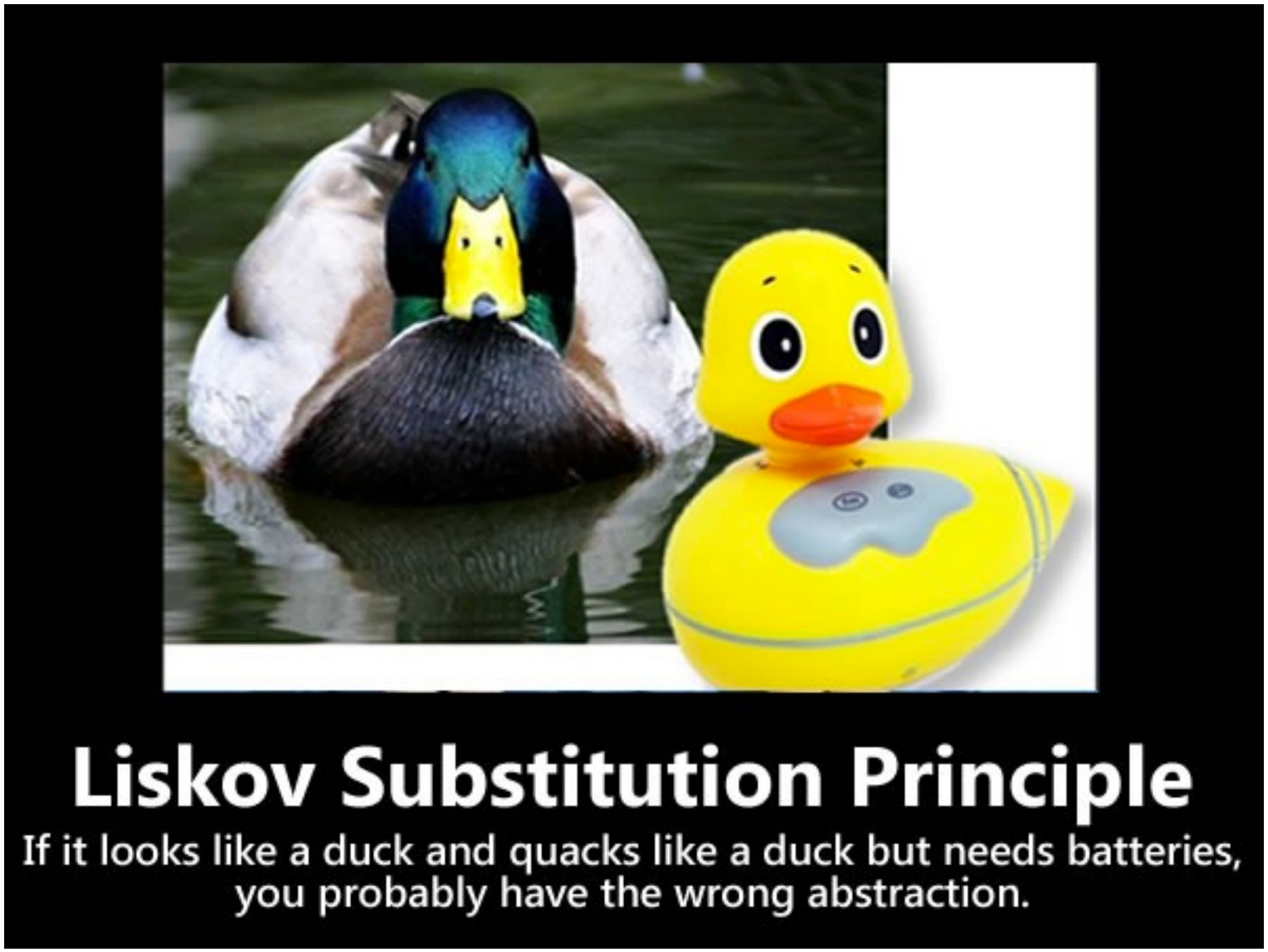
Inheritance

Interface and Strategy pattern

Strategy pattern



3. Liskov Substitution Principle



Wrong relationships between types cause unexpected bugs or side effects

Violations of LSP

Empty methods

Harden preconditions

Partial implemented interface

Type checking

4. Interface Segregation Principle

ISP !!

Identify Fat interface
Refactor code that depend on large interface

Client should not be forces to depend on methods that not use !!

Interface pollution

Interface with lots of methods
Interface with low cohesion
Client throws exception instead of implement
Client provide empty implementation

Fat interface

```
interface LoginService{
    void signIn();
    void signOut();
    void updateRememberMeCookie();
    User getUserDetails();
    void setSessionExpiration(int seconds);
    void validateToken(Jwt token);
    …
}
```

Fat interface

```
interface Account{
    double getBalance();
    void processLocalPayment(double amount);
    void processInternationalPayment(double amount);
}
```

Split to small interface

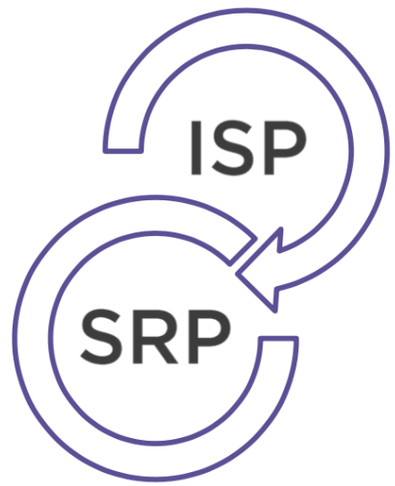
```
interface BaseAccount{
    double getBalance();
}
interface LocalMoneyTransferCapability{
    void processLocalPayment(double amount);
}
interface IntlMoneyTransferCapability{
    void processInternationalPayment(double amount);
}
```

Throw exception !!

```
class GoogleLoginService implements LoginService{
    …
    public void updateRememberMeCookie(){
        throw new UnsupportedOperationException();
    }
    public void setSessionExpiration(int seconds){
        throw new UnsupportedOperationException();
    }
```

ISP

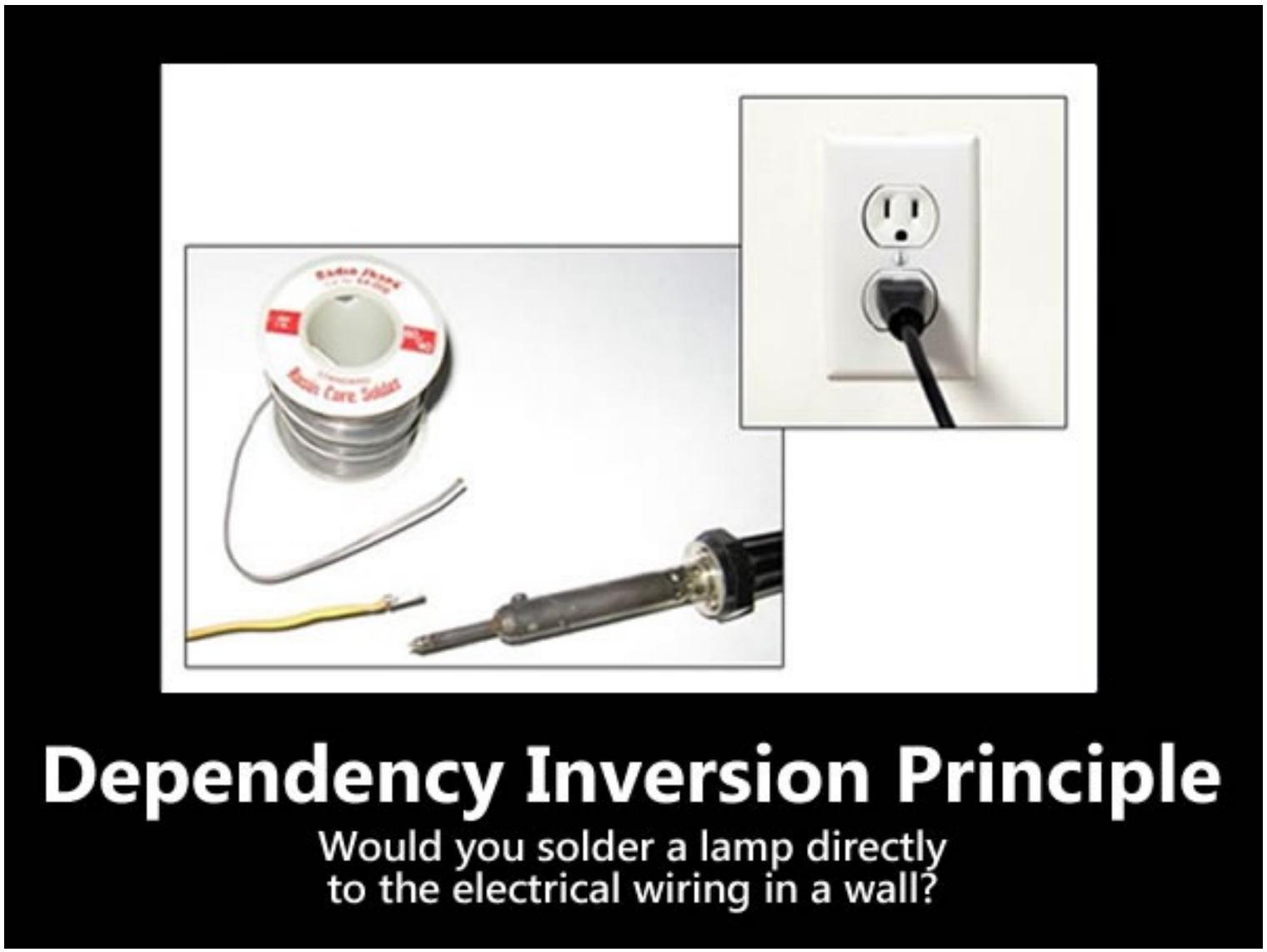
By keeping interfaces small, the classes that implement them have a higher chance to fully substitute the interface



Classes that implement small interfaces are more focused and tend to have a single purpose

Composition over Inheritance

5. Dependency Inversion Principle



DIP ?

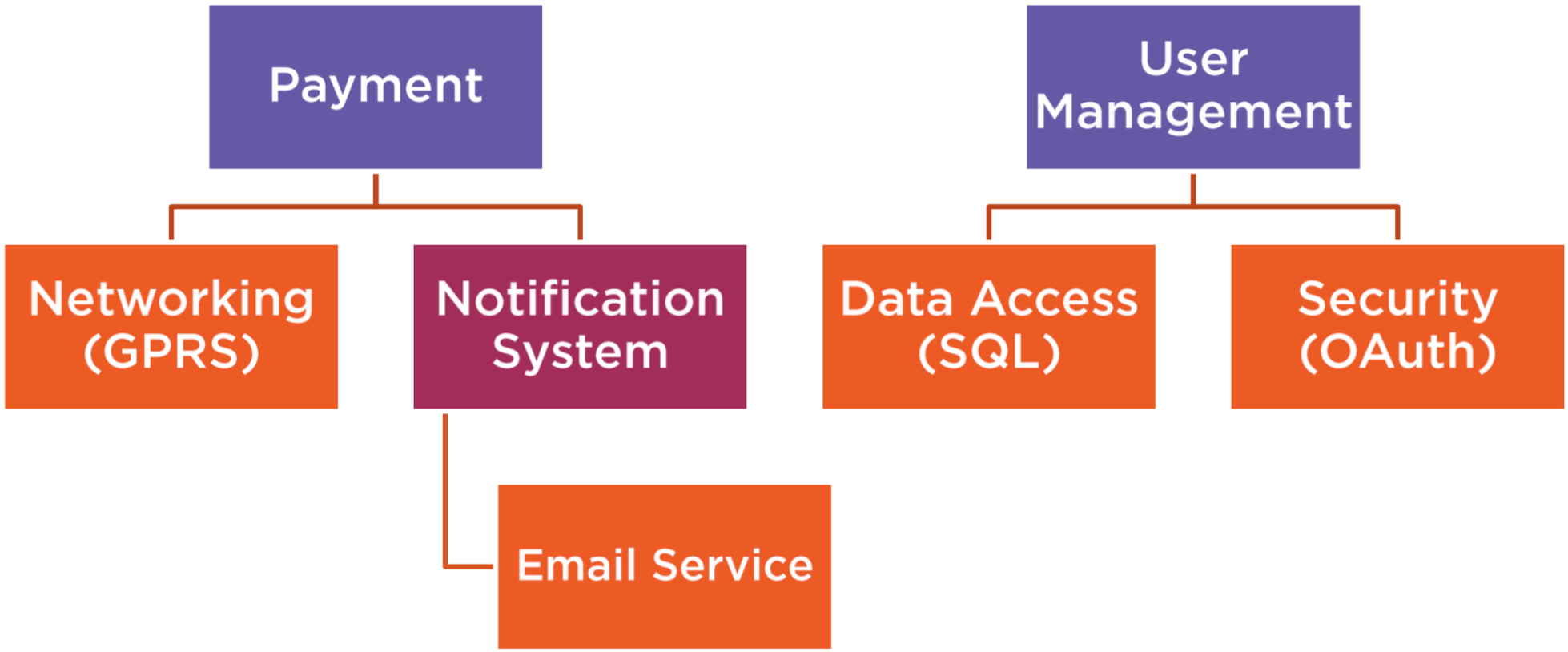
Modules should **not depend** on other modules
Abstraction should not depend on details

DIP ?

Modules should **not depend** on other modules
Abstraction should not depend on details

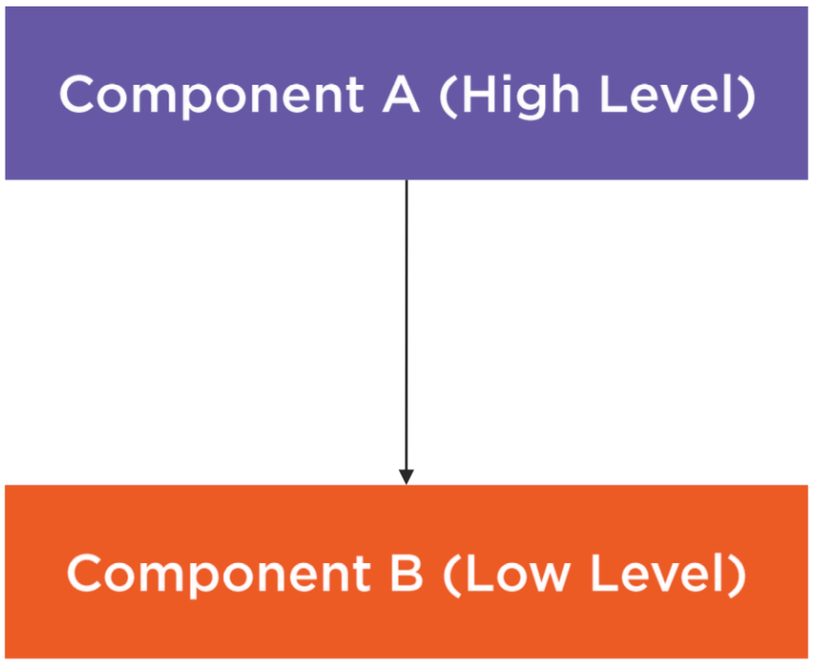
Logging
Data access/ Repository
Network
I/O

Example

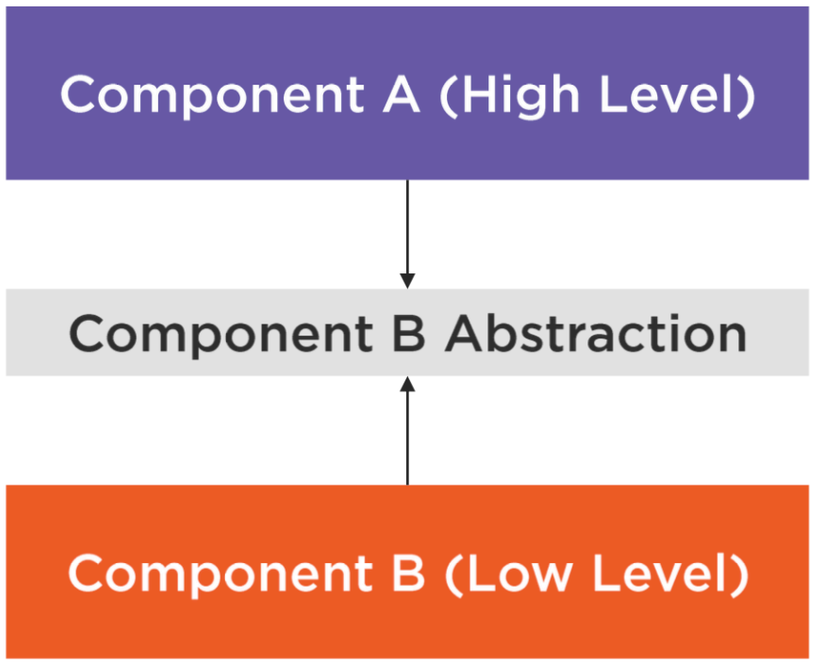


Solutions

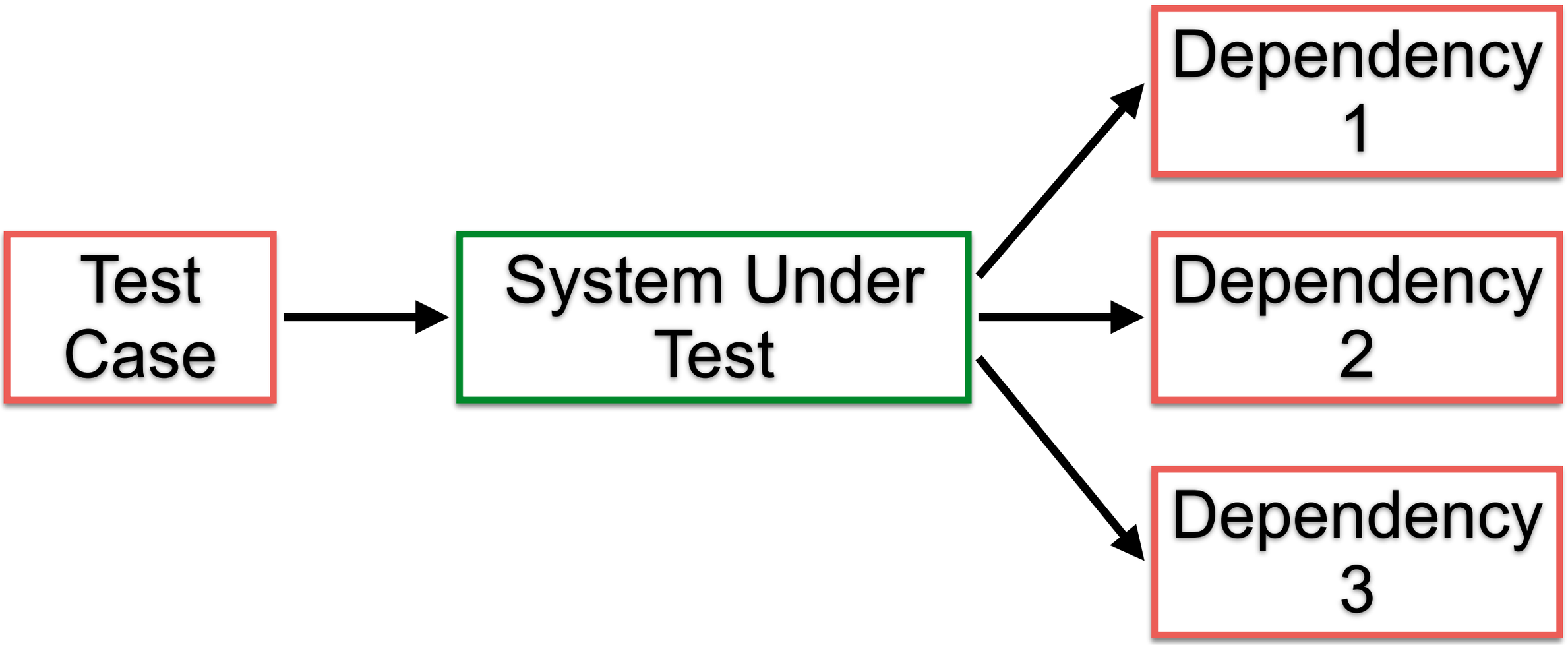
Depend on details



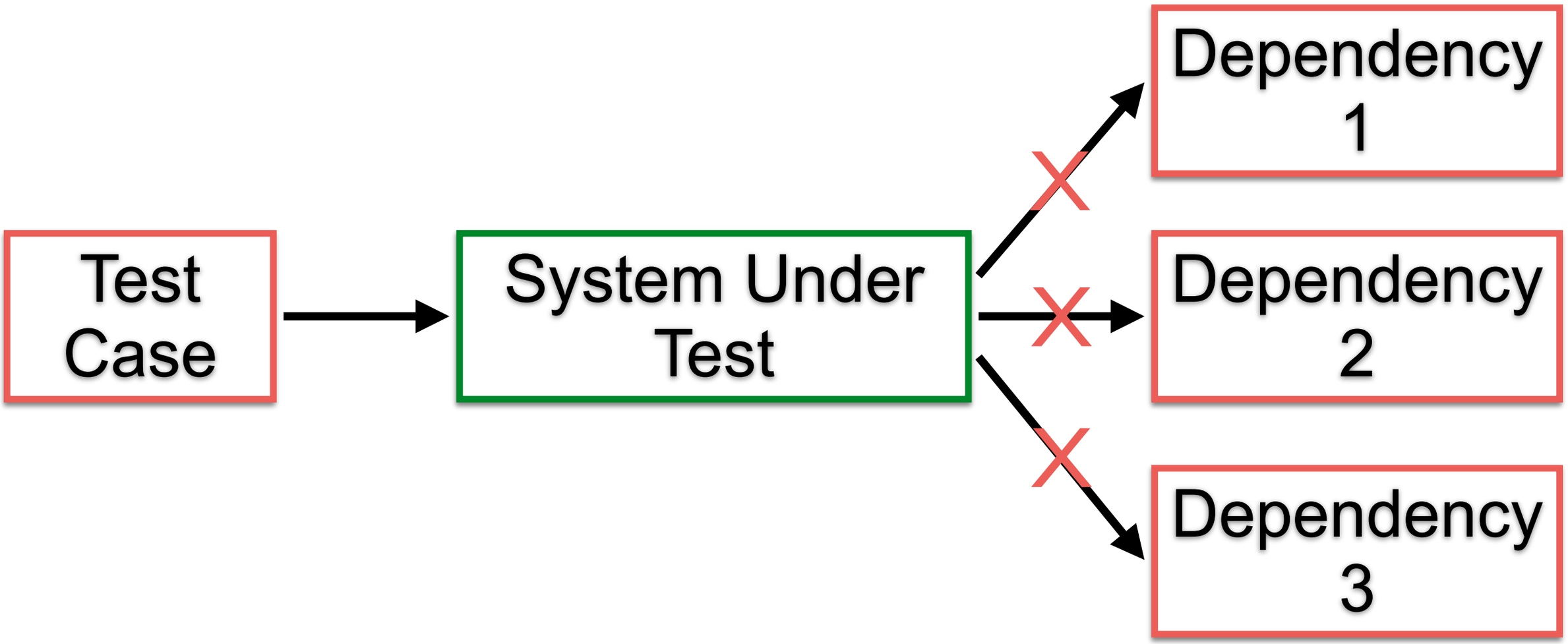
Depend on abstractions



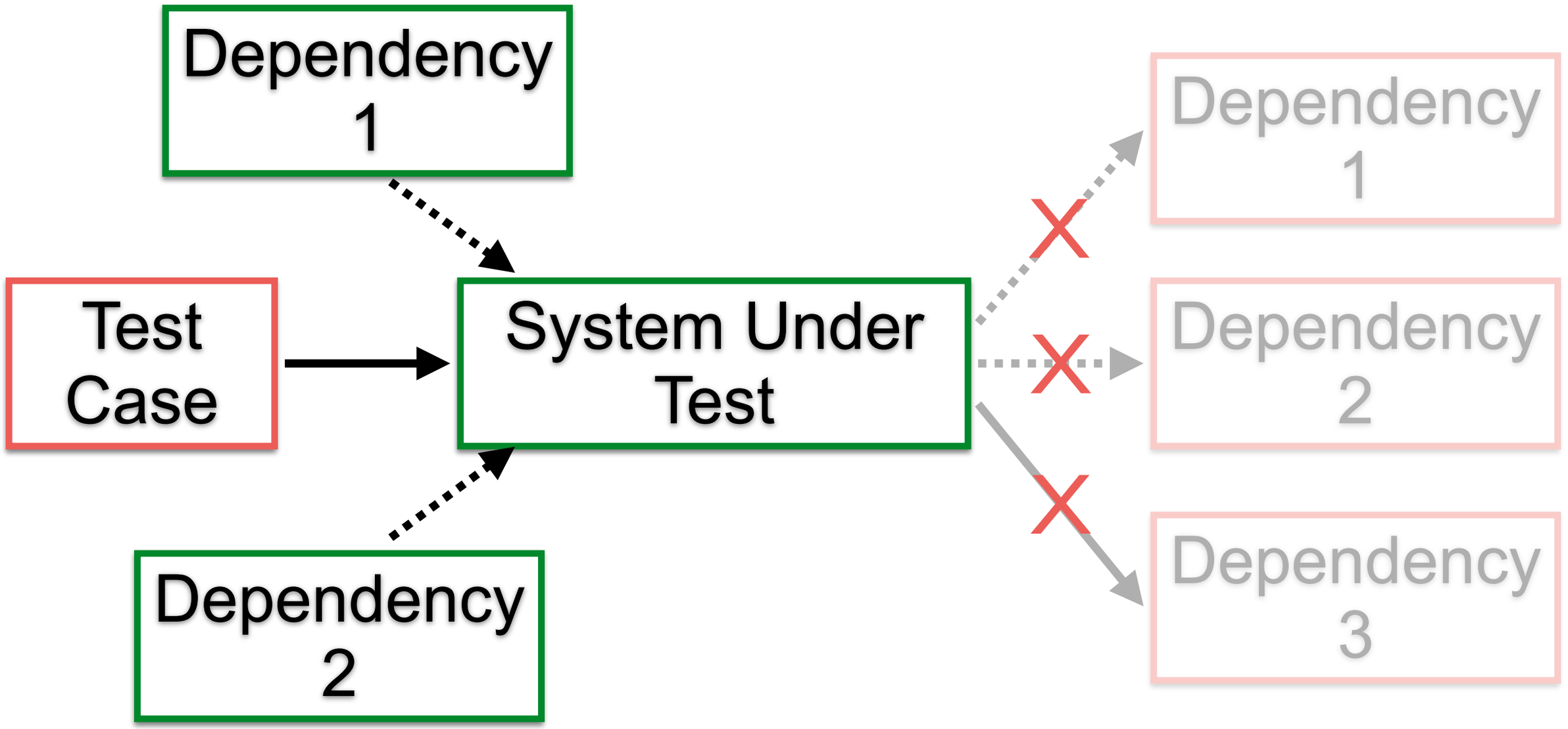
How to test ?



Remove dependencies !!



Inject dependencies !!



Dependency Injection (DI)

Constructor
Property/Setter
Method

Test Double

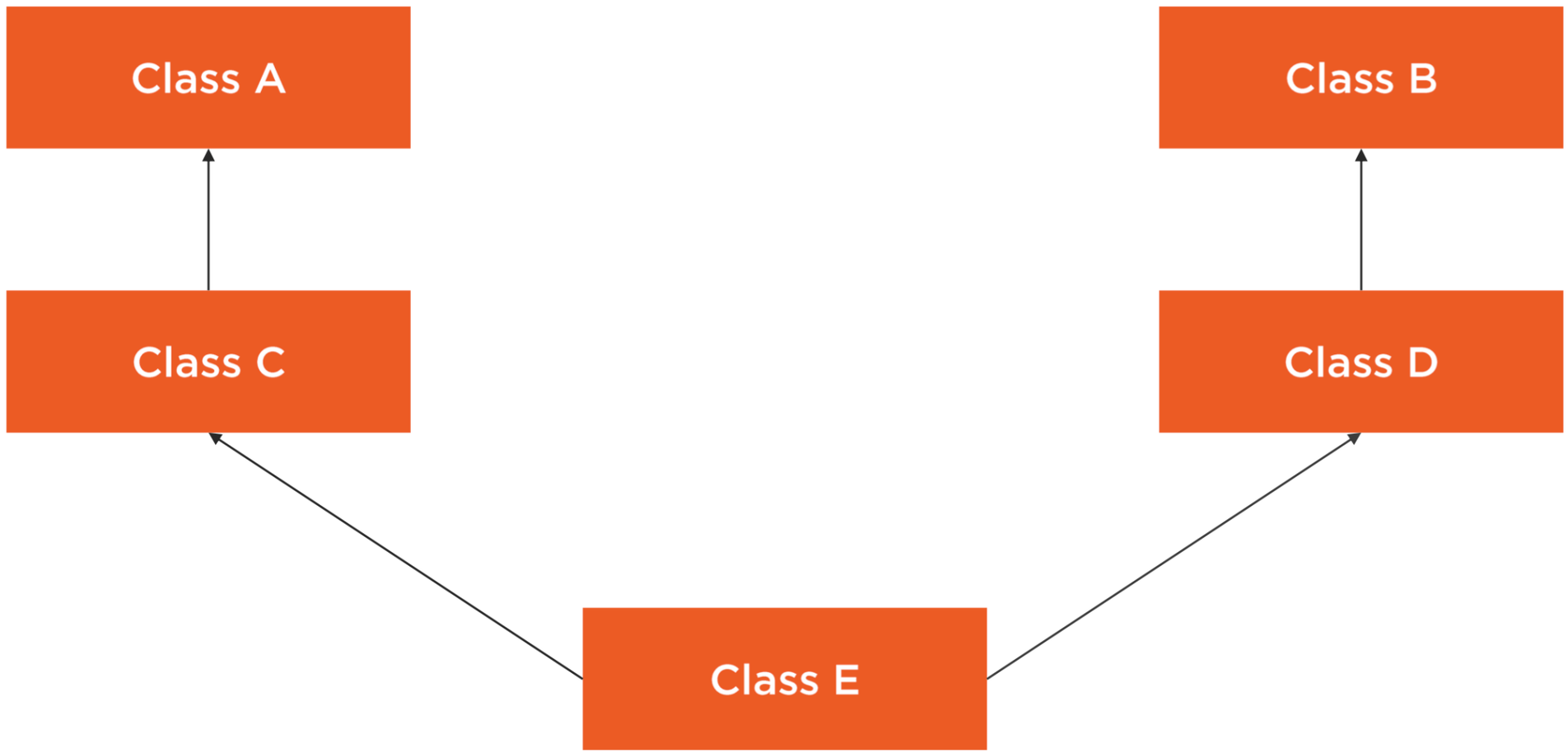
Dummy | Stub | Spy | Mock | Fake

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

Workshop with Test Double

More complex dependencies

Example

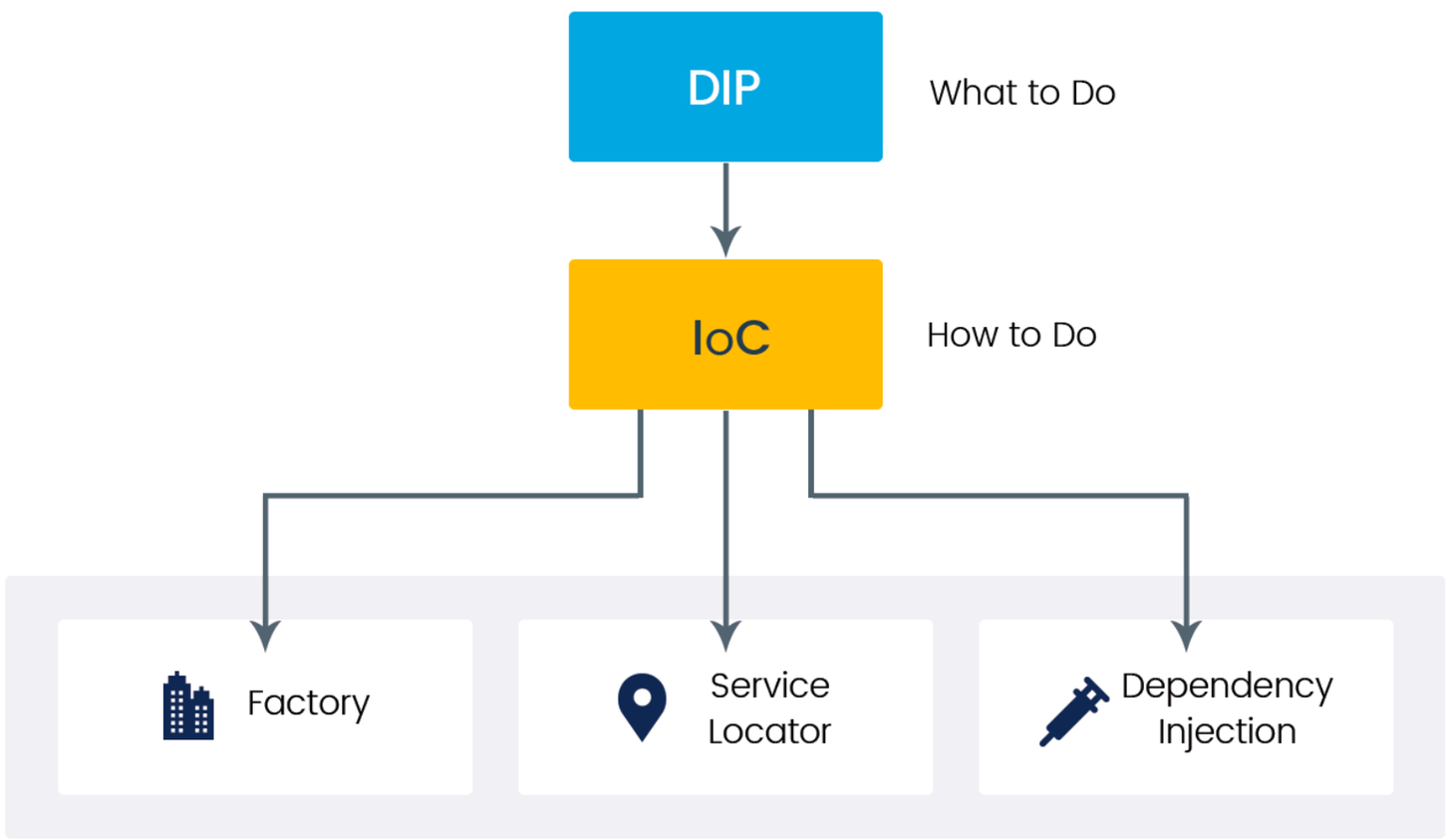


???

```
A a = new A();

B b = new B();

C c = new C(a);

D d = new D(b);

E e = new E(c,d);


e.doSomething();
```

Inversion of Control (IoC)

Control of object creation, configuration and lifecycle is passed to a container



DIP, DI and IoC

Effective ways to eliminate code coupling and keep system easy to maintain and evolve

OOP with Harry Potter

Design

Properties/Behaviors

Book

BookItem

Order

PriceCalculator

DiscountCalculator

Relationships ?

Develop

Develop

1. Create book class
2. Create Order class
3. Create BookItem class
4. Create Price Calculator
5. Create Discount Calculator

1. Create Book

How to create instance ?

1. With constructor
2. With Creation Method
3. With Builder pattern

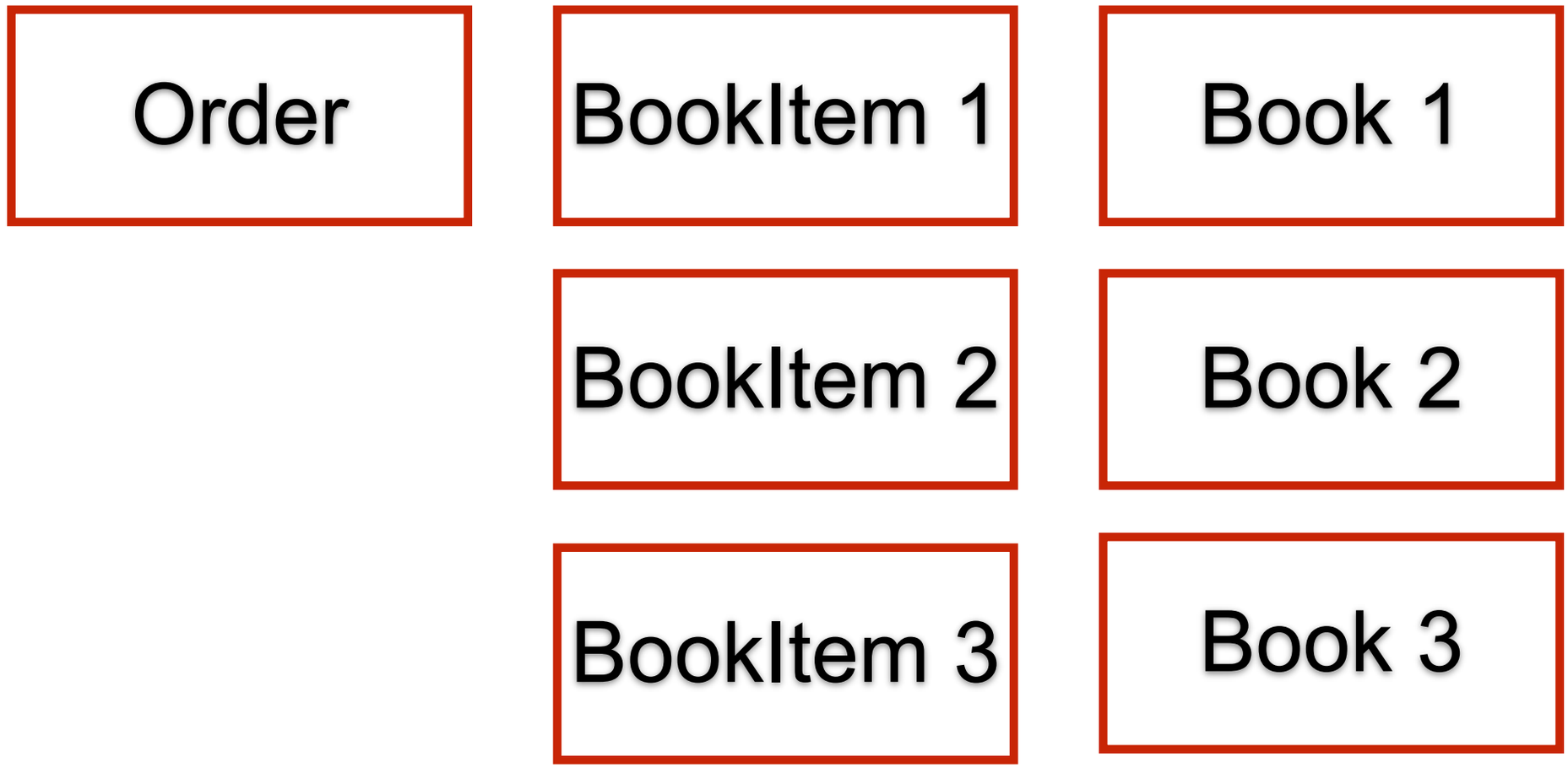
Create Book

```java
@Test
public void createBookWithConstructor() {
    Book book1 = new Book("H1", 8, 2);
}

@Test
public void createBookWithCreationMethod() {
    Book book1 = Book.createHarryPotterOne();
    Book book2 = Book.createHarryPotterTwo();
}

@Test
public void createBookWithBuilderPattern() {
    Book book1 = new BookBuilder()
                        .setPrice(8)
                        .setName("H1")
                        .build();
```

2. Create Order

Create Order

One order can have 0-N item(s)



Start with Empty Order

```
@Test
public void createEmptyOrder() {
    Order order = new Order();
    order.process();
    assertEquals(0, order.getBookAmount());
    assertEquals(0, order.getTotalPrice(), 0.00);
    assertEquals(0, order.getDiscount(), 0.00);
    assertEquals(0, order.getNetPrice(), 0.00);
}
```

Order with 1 item

```
@Test
public void createOrderWithOneBook() {
    Book book1 = new Book("H1", 8, 2);
    BookItem bookItem = new BookItem(book1, 1);
    Order order = new Order();
    order.addItem(bookItem);
    order.process();

    assertEquals(1, order.getBookAmount());
    assertEquals(8, order.getTotalPrice(), 0.00);
    assertEquals(0, order.getDiscount(), 0.00);
    assertEquals(8, order.getNetPrice(), 0.00);
}
```

Order with 2 item = 5%

```
@Test
public void createOrderWithTwoBook() {
    Book book1 = new Book("H1", 8, 2);
    Book book2 = new Book("H2", 8, 2);
    BookItem bookItem1 = new BookItem(book1, 1);
    BookItem bookItem2 = new BookItem(book2, 1);
    Order order = new Order();
    order.addItem(bookItem1);
    order.addItem(bookItem2);
    order.process();

    assertEquals(2, order.getBookAmount());
    assertEquals(16, order.getTotalPrice(), 0.00);
    assertEquals("Discount not correct", 16 * 0.05, order.getDiscount(), 0.00);
    assertEquals(16 - (16 * 0.05), order.getNetPrice(), 0.00);
}
```

3. Create BookItem

4. Price Calculation

5. Discount Calculation

Working with Spring Boot

Spring Initializr

spring initializr

Project
- Maven Project
- Gradle Project

Language
- Java
- Kotlin
- Groovy

Spring Boot
- 2.5.0 (SNAPSHOT)
- 2.5.0 (M1)
- 2.4.3 (SNAPSHOT)
- 2.4.2
- 2.3.9 (SNAPSHOT)
- 2.3.8

Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot
Package name	com.example.demo
Packaging	Jar / War
Java	15 / 11 / 8

Dependencies — ADD DEPENDENCIES... ⌘ + B

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

H2 Database SQL
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

https://start.spring.io/

Workshop