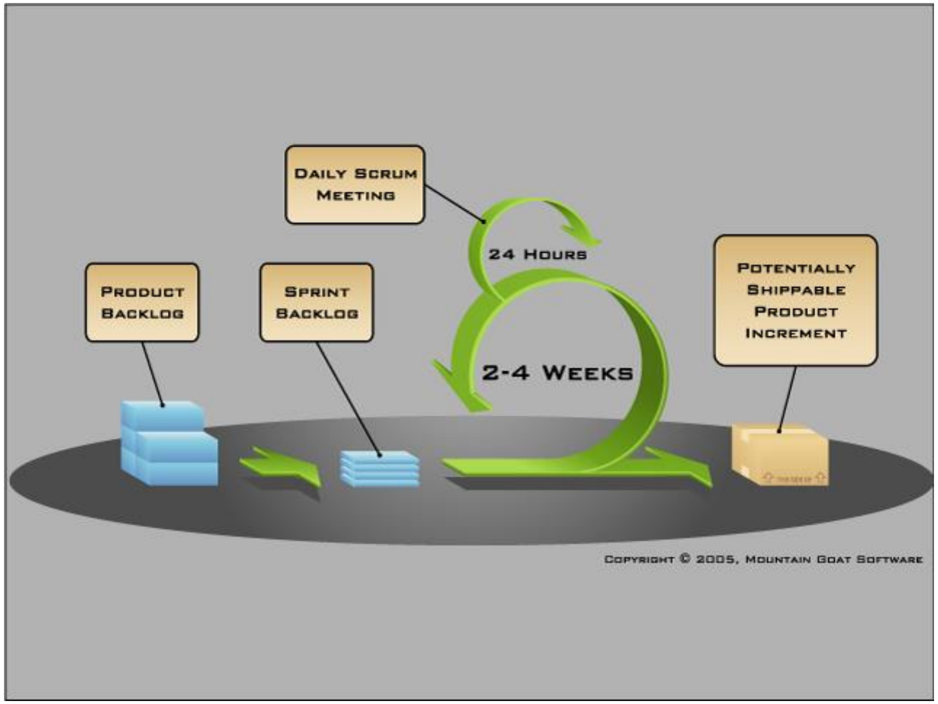
SCRUMCOP



Copyright © 2005, Mountain Goat Software

Agile Engineering Practices

By: Hiep Le

SCRUMCOP

About me

- Very normal developer
- 7 years in Scrum
- Agile Vietnam Board Member
- Founder of ScrumCOP
- International speaker
- CSM, CSP

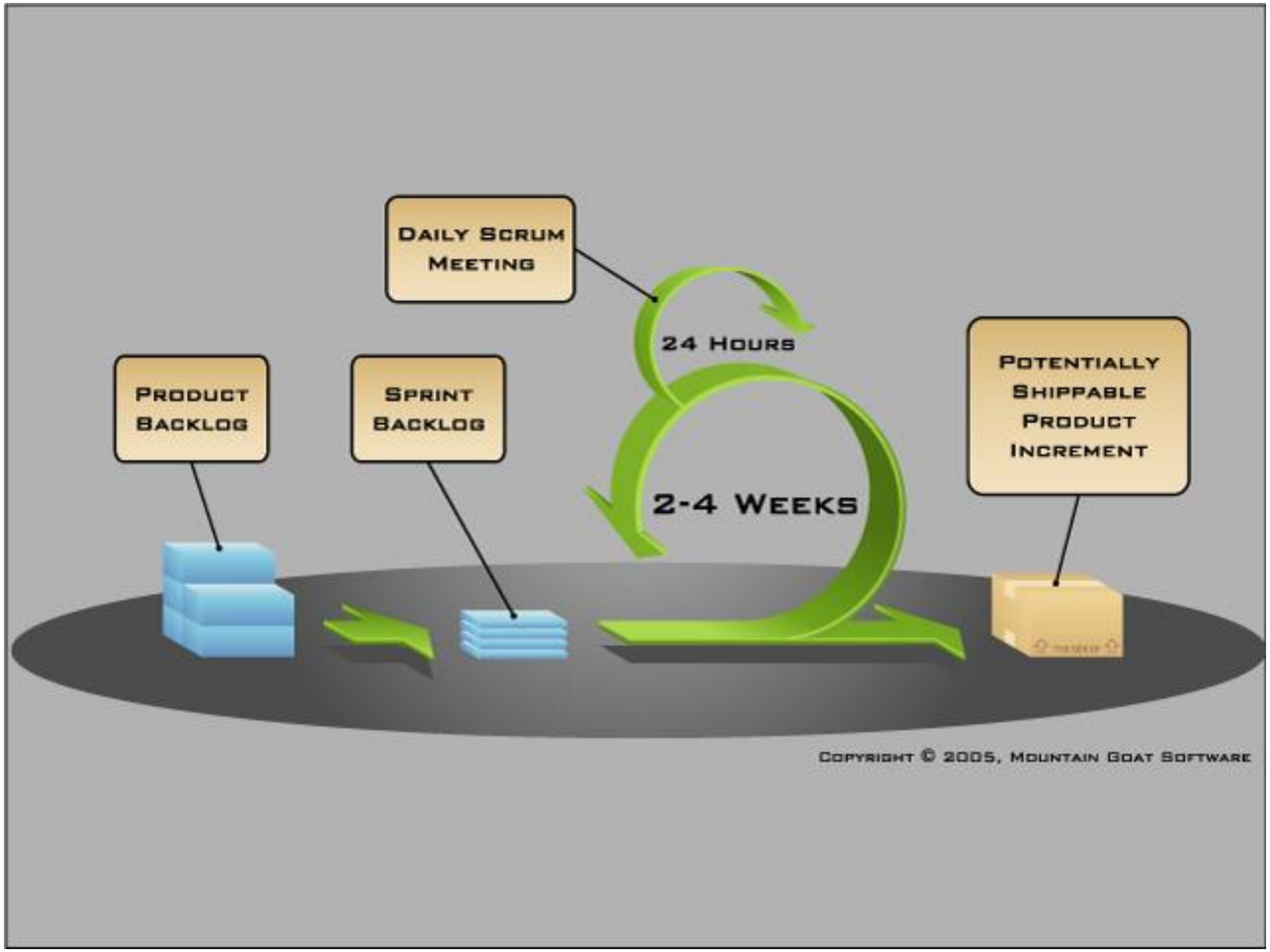
Agilevietnam
Building the community

Contents

- Sprint 1: Development Team in Scrum
- Sprint 2: Code for future aka TDD
- Sprint 3: Code for human aka DDD
- Sprint 4: Code for delivery aka ATDD

Sprint 1: Development Team

Scrum Framework


Daily Scrum Meeting
24 Hours
Product Backlog
Sprint Backlog
2-4 Weeks
Potentially Shippable Product Increment
Copyright © 2005, Mountain Goat Software

SCRUMCOP

Cross-functional feature team

Transparent and sustainable pace

Inspect and adapt

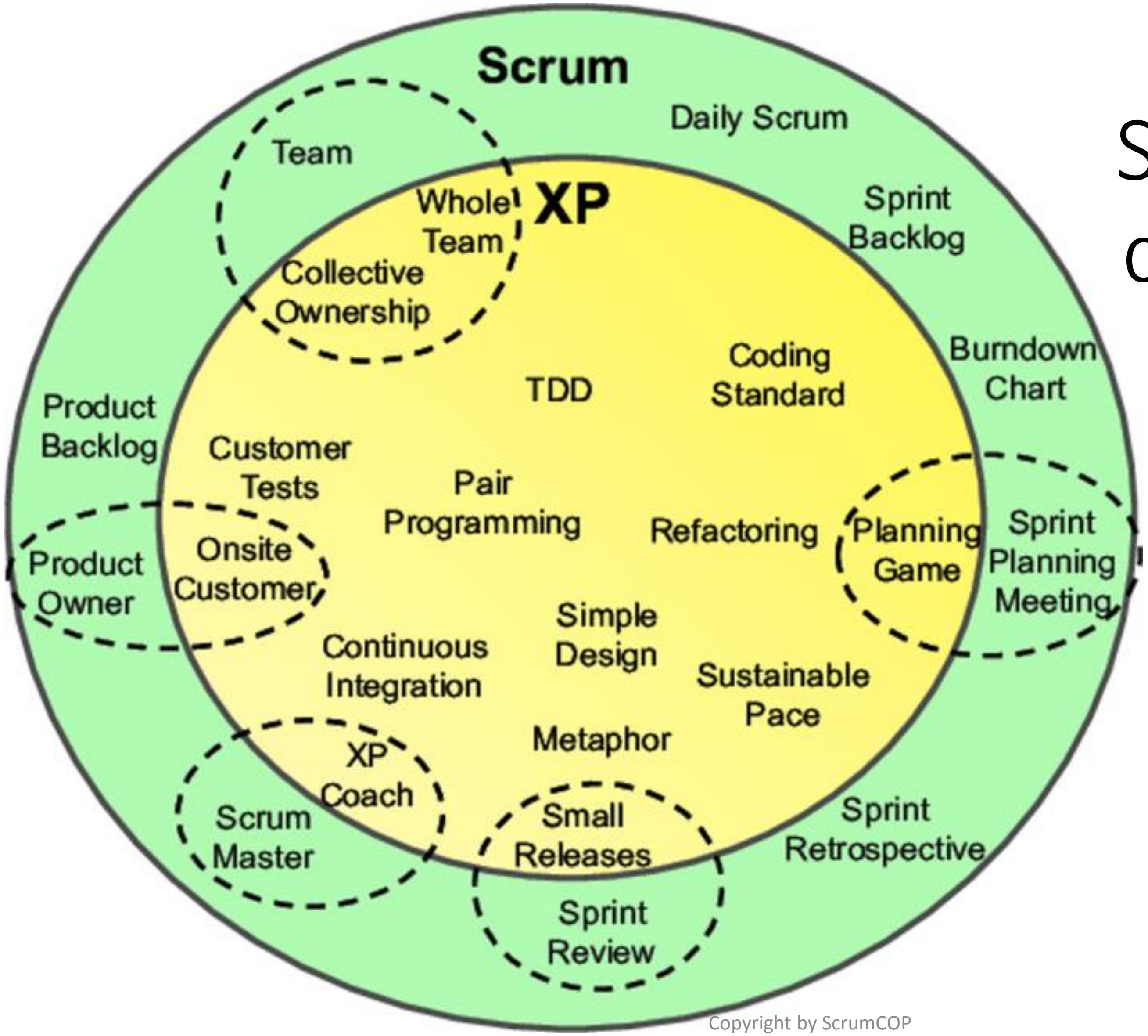
Shippable product

SCRUMCOOP

Scrum is deceptively simple!

Scrum is secretly difficult!

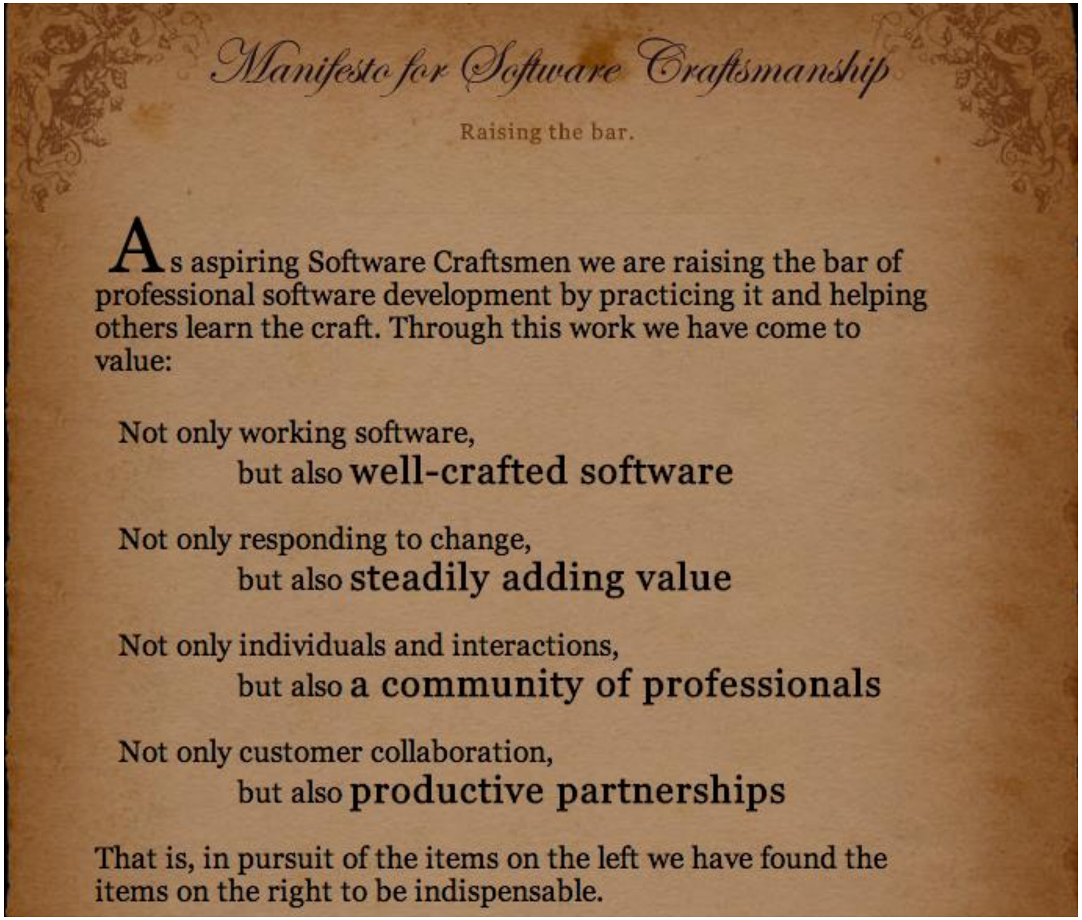
If we want Scrum to work



Scrum takes 1 day to set up!

XP takes years of practice!

Manifesto for Software Crafsmanship



Manifesto for Software Craftsmanship

Raising the bar.

As aspiring Software Craftsmen we are raising the bar of professional software development by practicing it and helping others learn the craft. Through this work we have come to value:

Not only working software,
but also **well-crafted software**

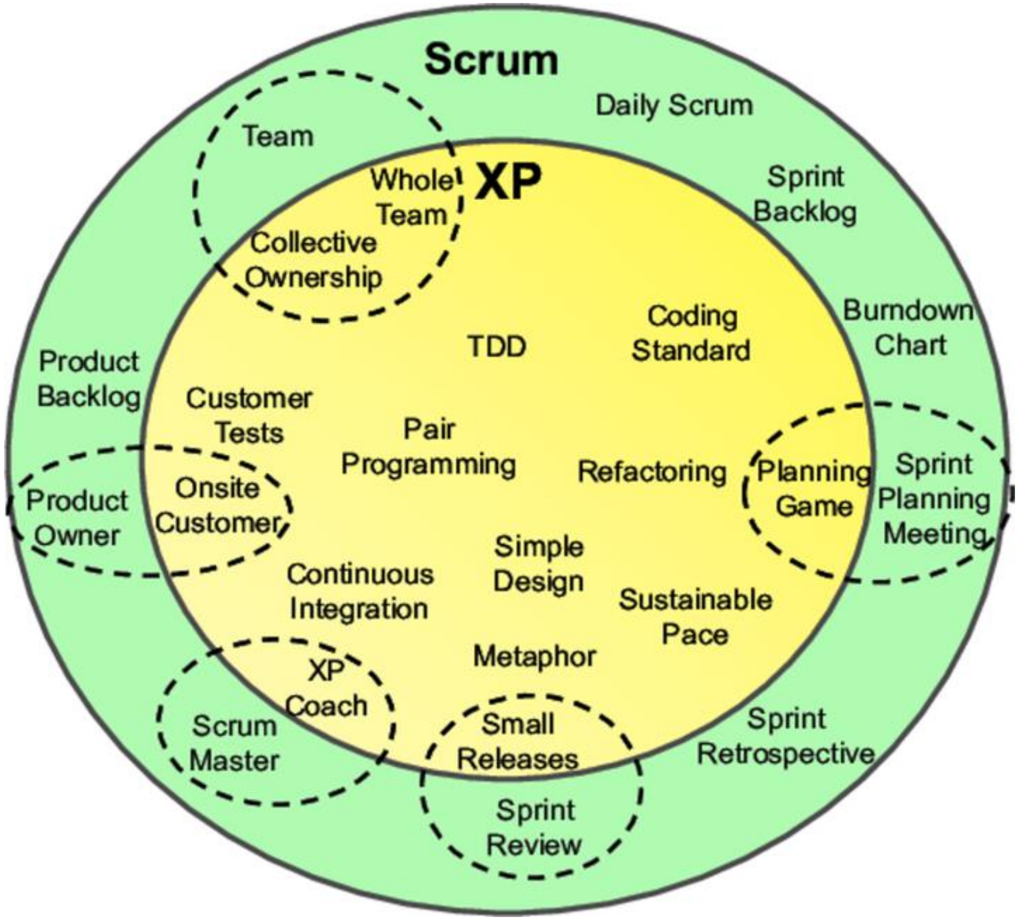
Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**

That is, in pursuit of the items on the left we have found the items on the right to be indispensable.

XP Practice – Day 1



Unit Test Ready?

```
assertEquals("Always fail", true, false );
```

Batman Application

- The application takes different strings of different lengths.
- If the number of vowels are equal or more than 30% of the string length, then replace ‘iambatman’ for each continuous set of vowels.
- Example:
  - “a”->”iambatman”
  - “baab”->”biambatmanb”
  - “jokkr”-> “jokkr”

Your first excercise

- Form a team of 2-3.
- Go to https://github.com/hieplenet/ScrumKata
- Download Excercise 1 – Batman of your favorite langage.
- Read the solution and test the code.

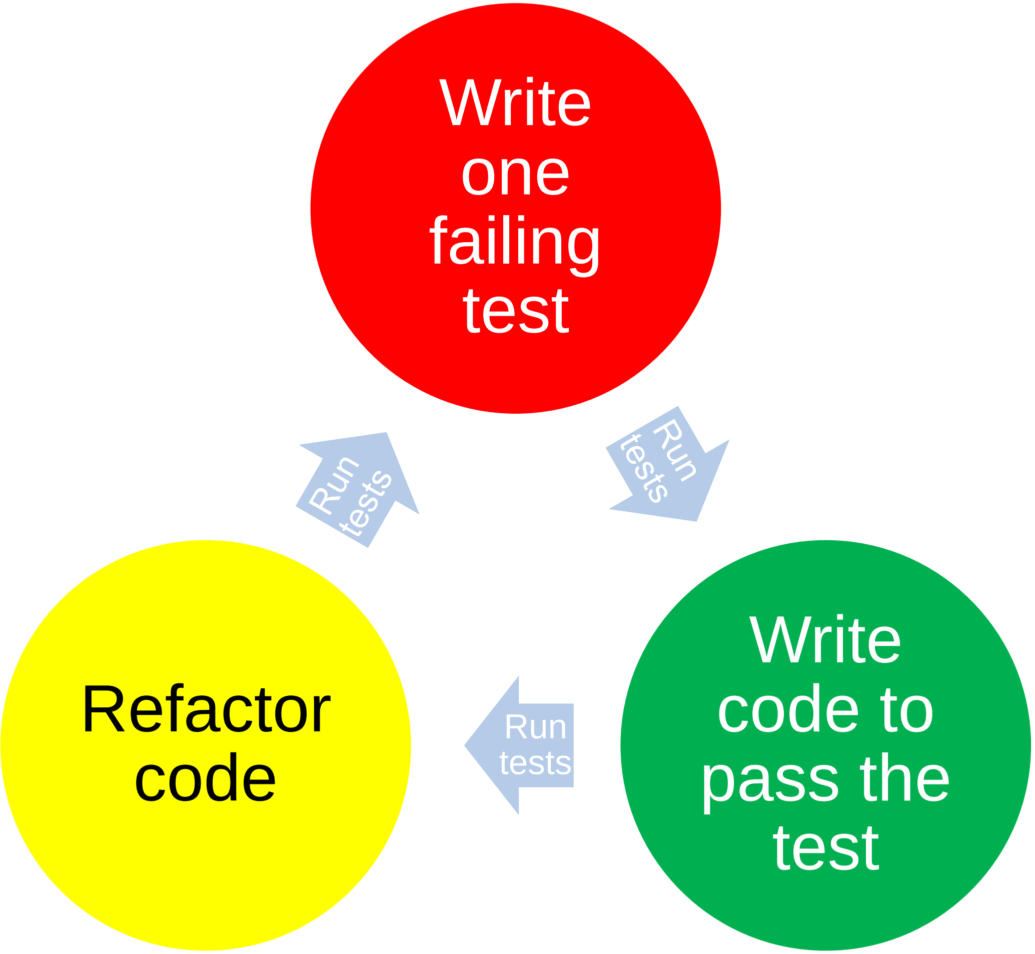
Copyright by ScrumCOP

What did you learn?

SCRUMCAMP

Sprint 2: Code for future aka TDD

Classic/Chicago/Detroit TDD



Practice TDD - Roman Numeral

- Write a method to convert an integer to a Roman Numeral without using your web browser.
- Example:
  - 1 -> I
  - 5 -> V
  - 10 -> X
- Hints:
  - Go from 1, 2, 3 to greater.
  - Temporary ignore exception case like 4, 9.

What did you learn?

Transformations Priority Premise

1. **({}–>nil)** no code at all->code that employs nil
2. **(nil->constant)**
3. **(constant->constant+)** a simple constant to a more complex constant
4. **(constant->scalar)** replacing a constant with a variable or an argument
5. **(statement->statements)** adding more unconditional statements.
6. **(unconditional->if)** splitting the execution path
7. **(scalar->array)**
8. **(array->container)**
9. **(statement->recursion)**
10. **(if->while)**
11. **(expression->function)** replacing an expression with a function or algorithm
12. **(variable->assignment)** replacing the value of a variable.

Credit : Robert Martin (Uncle Bob)

DDD Exercise - Leap Year

Client functional requirements:

- A year is a leap year if it is perfectly divisible by four. This part affects normal annual year. Example: 1996, 1992 is leap year.
- But in century years, a year is a leap only when it is divisible by 400. This part effects century years. Example: 1800, 1900 are not leap years. 1600, 2000 are.

Client non-functional requirements:

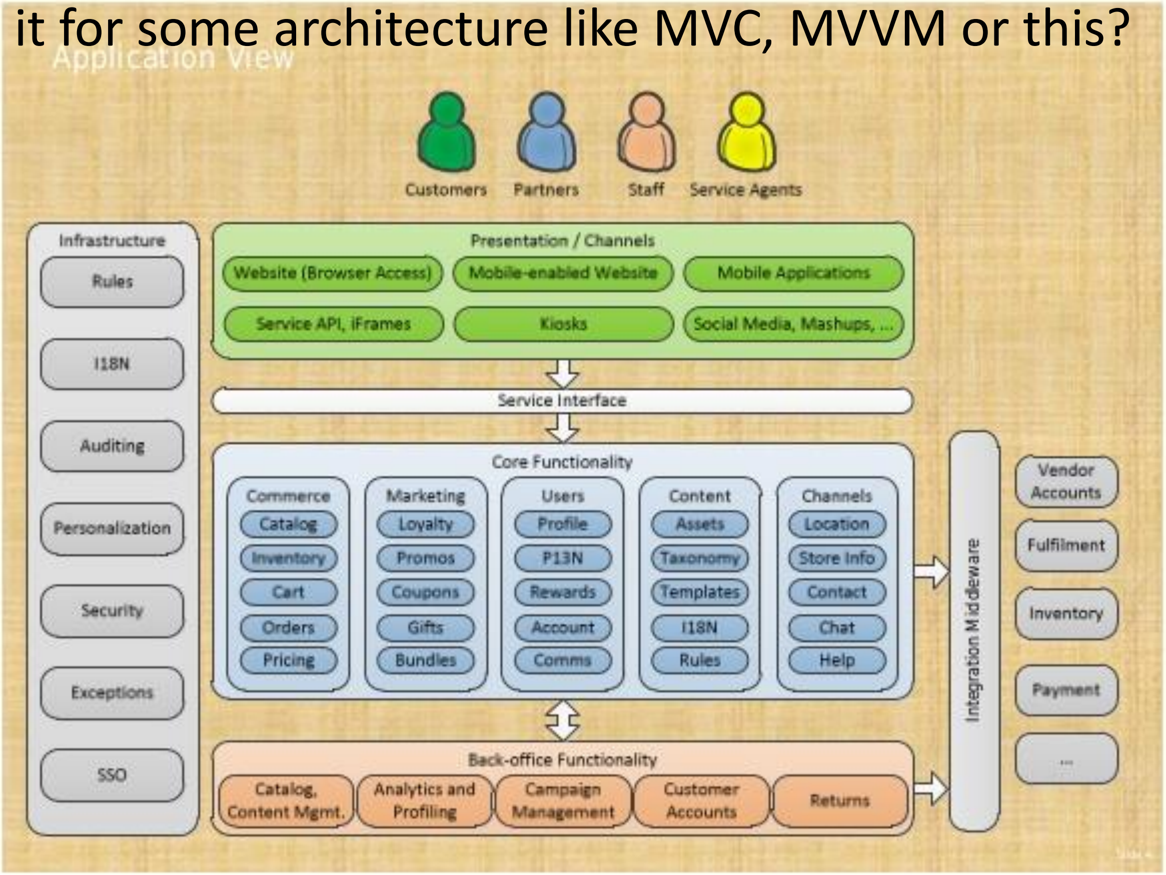
- Code must be tested.
- Code must be simple so that client can read it. (Client only read English)

SCRUMCOP

Sprint 3: Code for client aka ATDD

What do you thing about TDD?

- Can we use it at Sendo?
- Can you it for some architecture like MVC, MVVM or this?



Probably not. Because Classic TDD are for

- Mutable objects encapsulating state.
- Pure functions and immutable value objects.
- Detail, stand alone application: algorithms, logic, conditionals.
- Is this what we work on daily?

What we usually work on

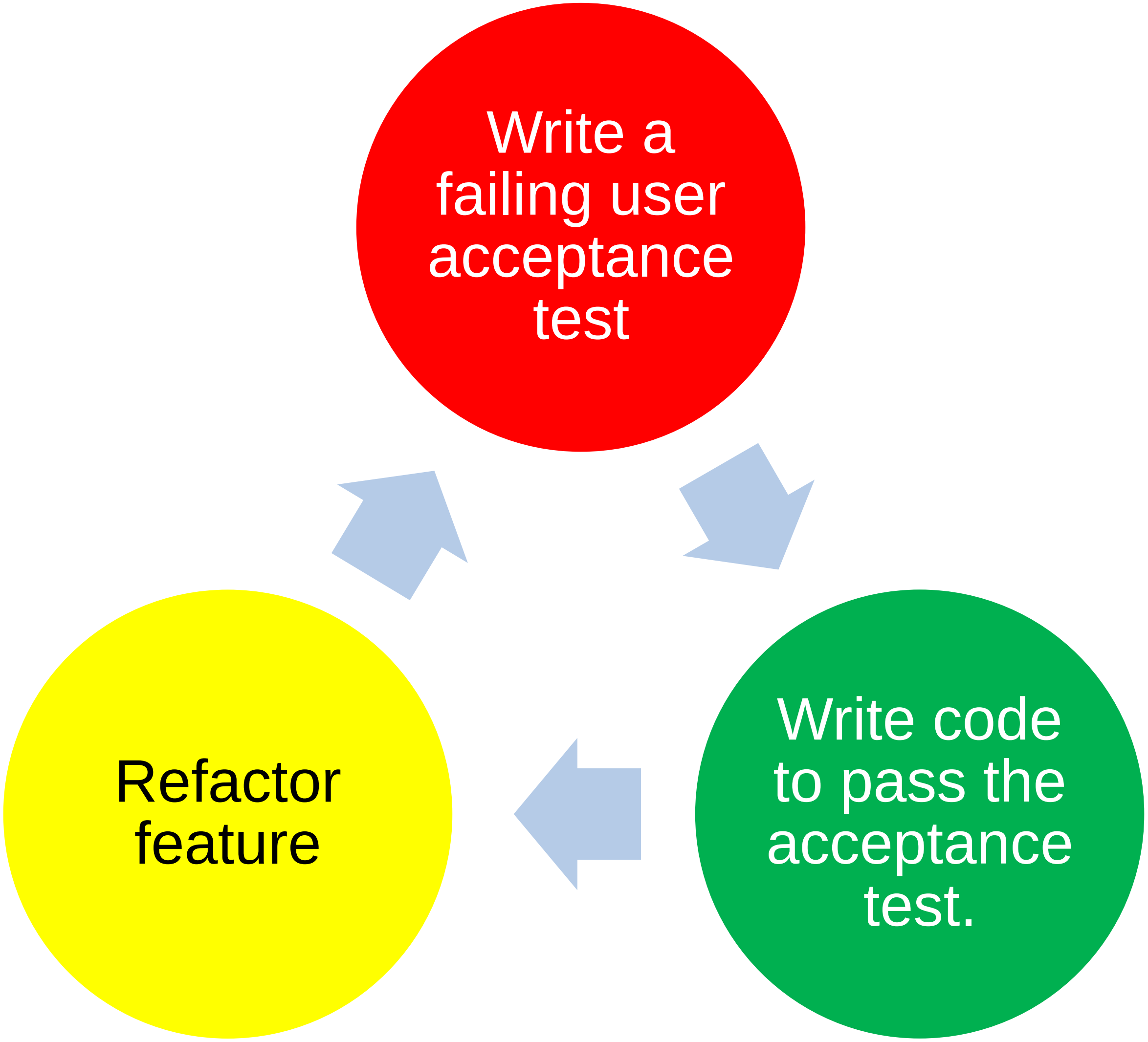
- Interate a new library into current working system.
- Query database and transform data to display it to user.
- Make a web form to store data into database accordingly.
- Etc...

We rarely have the luxury of working on a stand-alone application service.

Mockist/London/Intergration TDD

- Write a user acceptance test.
- Write code to pass the acceptance test by:
  - Write a failing unit test.
  - Write code to pass the unit test.
  - Refactor code.
- Refactor whole feature.

Mockist/London/Intergration TDD

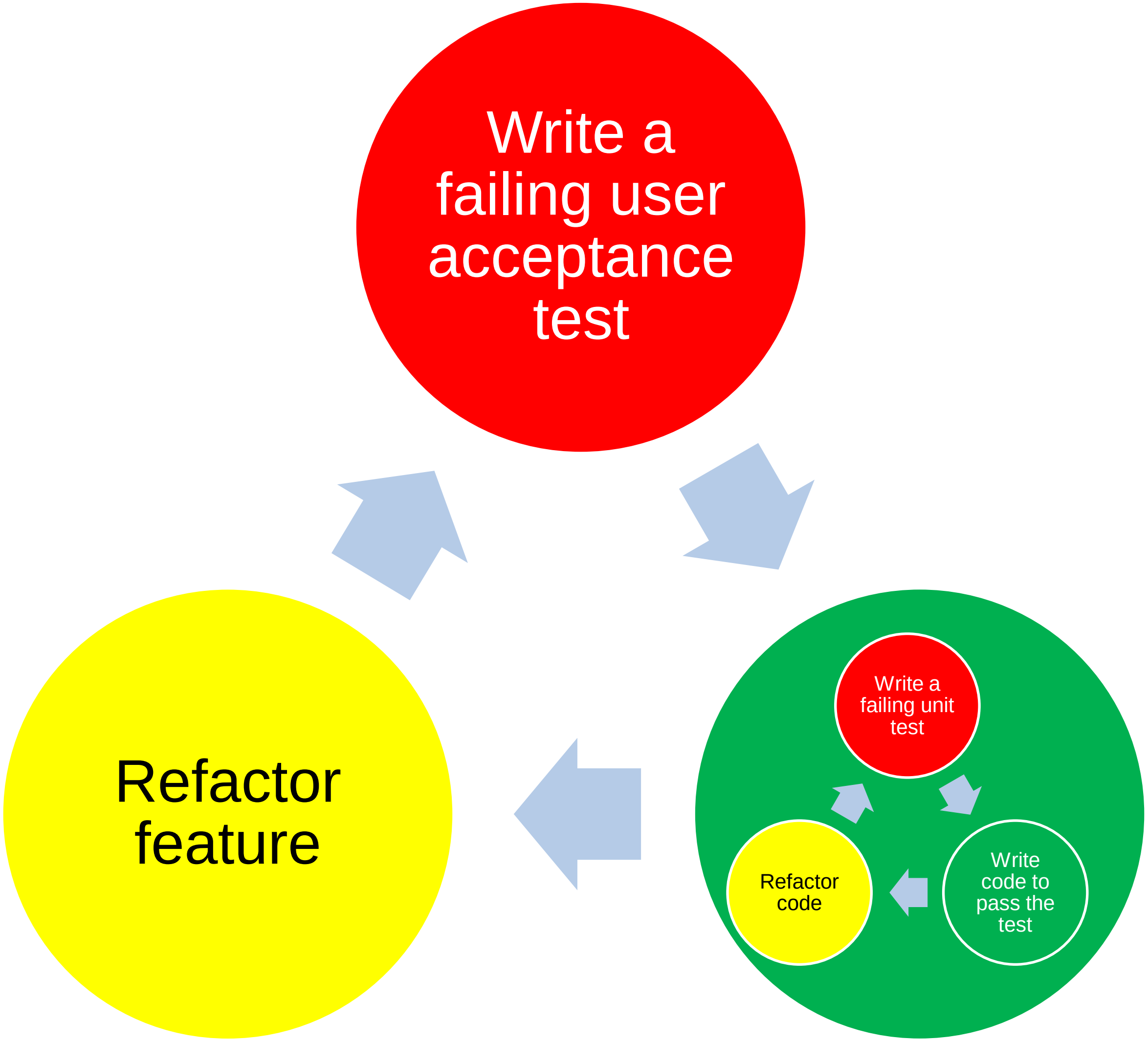


Write a failing user acceptance test

Write code to pass the acceptance test.

Refactor feature

Mockist/London/Intergration TDD



Write a failing user acceptance test

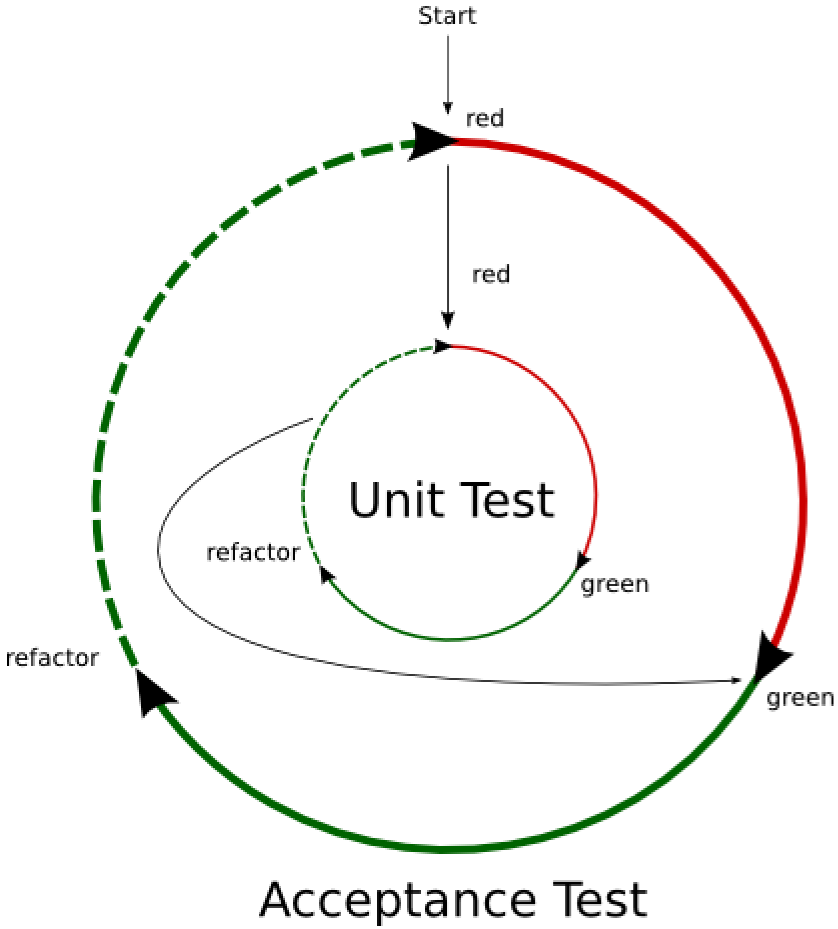
Write a failing unit test

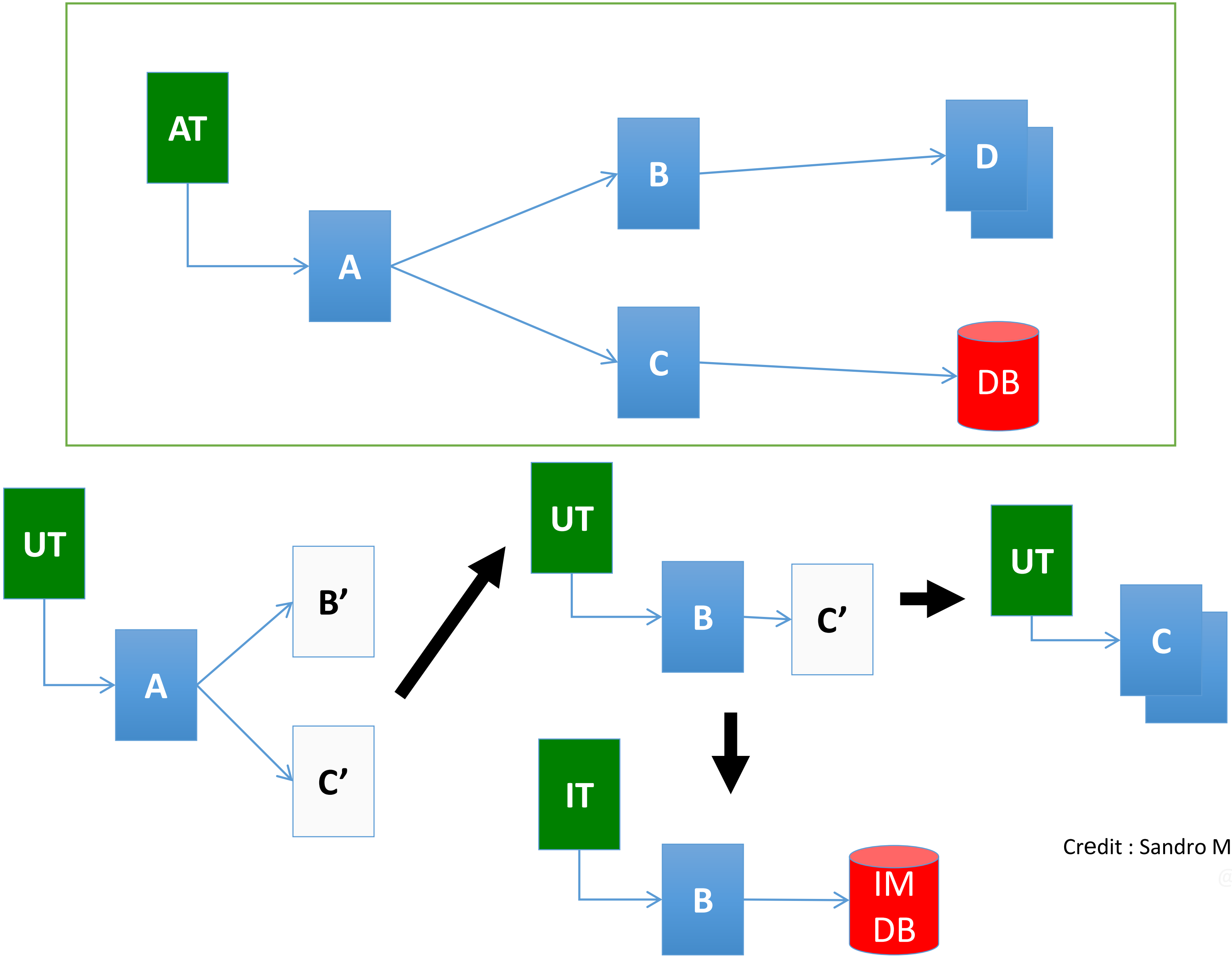
Write code to pass the test

Refactor code

Refactor feature

Mockist/London/Intergration TDD





Credit : Sandro MANCUSO

ATDD Excercise - Bank Kata

Requirement:

Given a client makes a deposit of 1000 on 10-01-2012

And a deposit of 2000 on 13-01-2012

And a withdrawal of 500 on 14-01-2012

When she prints her bank statement

Then she would see

date	credit	debit	balance
14/01/2012	500.00		2500.00
13/01/2012	2000.00		3000.00
10/01/2012	1000.00		1000.00

Account

```
void deposit(int value);

void withdrawal(int value);

void printStatement();
```

SCRUMCOP

Sprint 4: Code for the past aka Legacy Code

Legacy System

A social networking website for travellers

- You need to be logged in to see the content
- You need to be a friend to see someone else’s trips

Trip Service

- Form a team of 2-3.
- Go to https://github.com/hieplenet/ScrumKata
- Download Excercise 2 – Trip Service of your favorite langage.
- Read the solution and test the trip service.

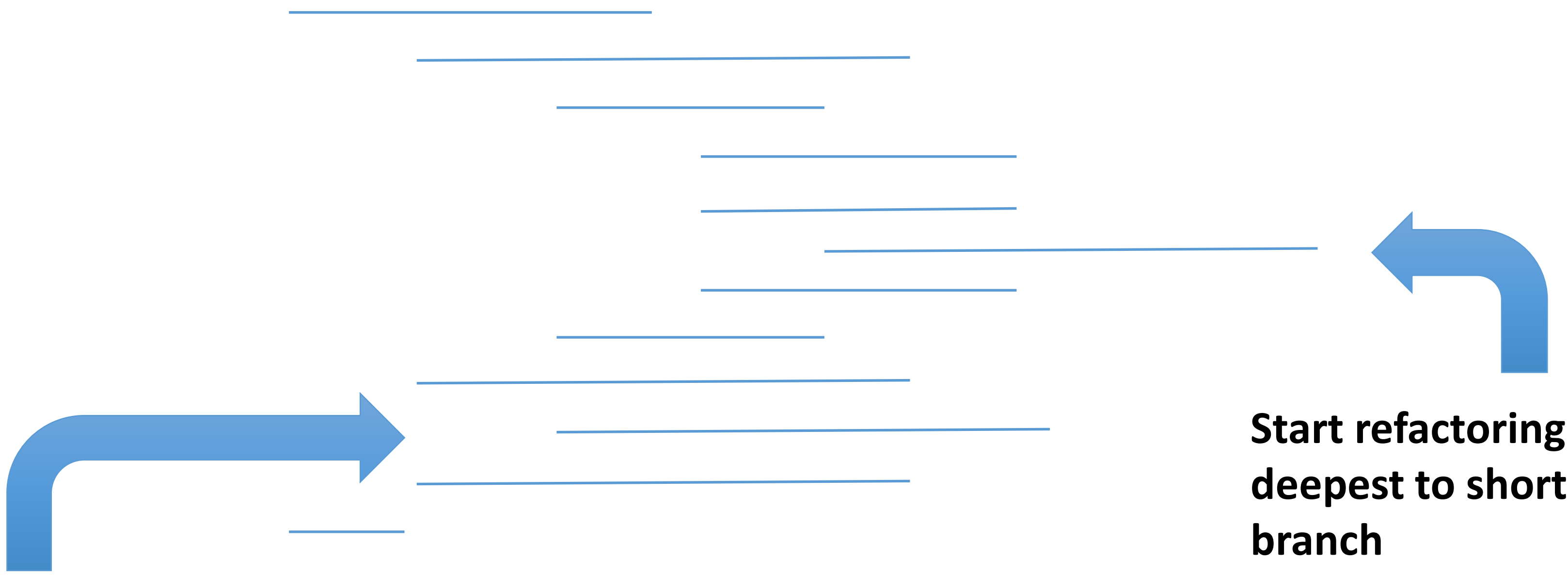
Trip Service

```java
public List<Trip> getTripsByUser(User user) throws UserNotLoggedInException {
    List<Trip> tripList = new ArrayList<Trip>();
    User loggedUser = UserSession.getInstance().getLoggedUser();
    boolean isFriend = false;
    if (loggedUser != null) {
        for (User friend : user.getFriends()) {
            if (friend.equals(loggedUser)) {
            isFriend = true;
            break;
            }
        }
        if (isFriend) {
                tripList = TripDAO.findTripsByUser(user);
        }
        return tripList;
    } else {
        throw new UserNotLoggedInException();
    }
}
```

Pain of working with legacy code

1. You can not change production code if you don’t understand it.
2. You can not understand the code if you don’t test its behavior.
3. You can not test its behavior without changing the code.

Working with Legacy Code Tips



Start refactoring from deepest to shortest branch

Start testing from shortest to deepest branch

Appendix - Law of Code

$$\mathrm{E} = mc^2$$

$$\mathrm{Error} = (More\ code)^2$$

$$\mathrm{Error} = (Machine\ code)^2$$

Appendix - Code for human

```
{
 StringBuffer a = new StringBuffer();
 a.append(loc + "?" + var1 + "&param1" + var2 + "&param2" + var3 + "&param3" +
var4 + "&param4" + var5 + "&param5" + var6 + "&param6");
 return a.toString();
}
```

Appendix - Code for human

```
private bool ValidateUser(string username, string password) {
        if (Validator.Validate(username, password)) {
            FormAuthentication.Login(username);
            return true;
        }
        return false;
    }
```

Appendix - Code for human

```
private void SetSelectedItemAt(int controlId)
    {
        foreach (var checkBox in Items)
        {
            if (checkBox.Id == controlId)
            {
                var isSelect = checkBox.IsSelected;
                checkBox.IsSelected = !isSelect;
                if(!isSelect == false)
                {
                    ClearValue();
                }
                break;
            }
        }
    }
```

Appendix

- Dependency Injection
  - MEF
  - Spring

- BDD
  - Jbehave
  - Cucumber
  - Nspec
  - Specflow

Appendix

- Continuous Test tools
  - Infinitest
  - Ncrunch
  - Karmar
- Continuous Intergration
  - Jenkins
  - TFS
  - Team City

THANK
YOU