AFS
Agile Full Stack
Developer Bootcamp
Thoughtworks ©

Refactoring & Practice

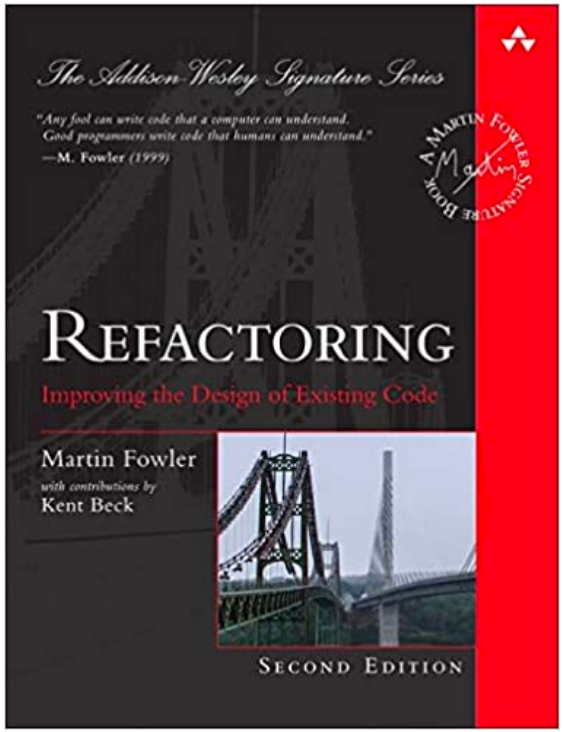
thoughtworks

© 2022 Thoughtworks Commercial in Confidence

Refactor Overview

© 2022 Thoughtworks Commercial in Confidence

What's Refactoring?



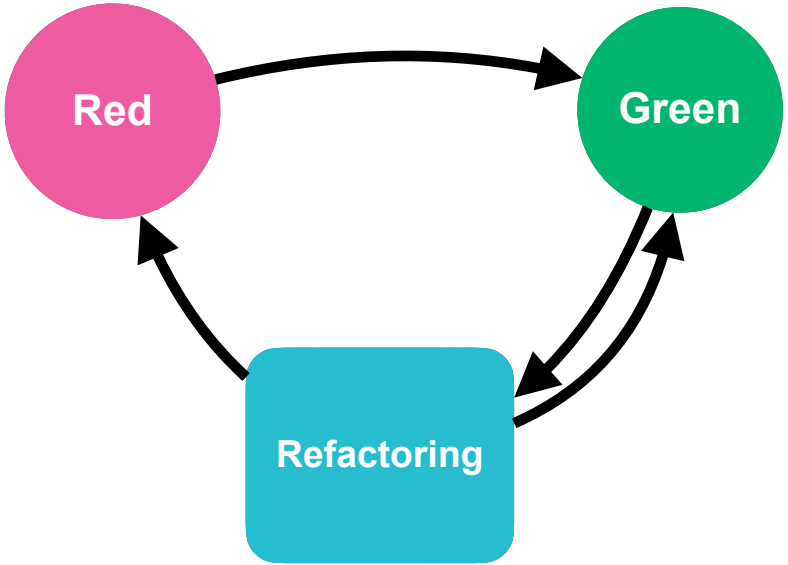
*A **change** made to the internal structure of software **without changing its observable behavior**.*

- Martin Fowler, author of Refactoring

© 2022 Thoughtworks Commercial in Confidence

What's Refactoring?

*A **change** made to the internal structure of software **without changing its observable behavior**.*



© 2022 Thoughtworks Commercial in Confidence

Why Refactoring?

make software ***easier to understand*** *and* ***cheaper to modify***

- Martin Fowler, author of Refactoring

© 2022 Thoughtworks Commercial in Confidence

When do we refactor?

Rule of Three	**New Features**
Bug Fix	**Code Review**

© 2022 Thoughtworks Commercial in Confidence

How to refactor?

1. Test Protection

© 2022 Thoughtworks Commercial in Confidence

Code Smell

thoughtworks

© 2022 Thoughtworks Commercial in Confidence

Code Smell

In computer programming, a **code smell** is any characteristic in the source code of a program that possibly indicates a deeper problem.

© 2022 Thoughtworks Commercial in Confidence

https://en.m.wikipedia.org/wiki/Code_smell
https://martinfowler.com/bliki/CodeSmell.html

Code Smell

...NOT bugs

...NOT technically incorrect

© 2022 Thoughtworks Commercial in Confidence

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

Code Smell

1. Mysterious Name
2. Duplicated Code
3. Long Function
4. Long Parameter List
5. Global data
6. Mutable data
7. Divergent Change
8. Shotgun Surgery
9. Feature Envy
10. Data Clumps
11. Primitive Obsession
12. Repeated Switches
13. Loops
14. Lazy Element
15. Speculative Generality
16. Temporary Field
17. Message Chains
18. Middle Man
19. Insider Trading
20. Large Class
21. Alternative Classes with Different Interfaces
22. Data Class
23. Refused Bequest
24. Comments

Book: Refactoring - Improving the Design of Existing Code

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
public int cmp(List<Integer> ps) {
    int tp = 0;

    for (Integer p : ps) {
        tp += p;
    }

    return tp;
}
```

© 2022 Thoughtworks Commercial in Confidence

Mysterious Name

```
public int cmp(List<Integer> ps) {
    int tp = 0;

    for (Integer p : ps) {
        tp += p;
    }

    return tp;
}
```

```
public int computeTotalPrice(List<Integer> prices) {
    int totalPrice = 0;

    for (Integer price : prices) {
        totalPrice += price;
    }

    return totalPrice;
}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
public class Person {


    public void party() {
        // confirm about time
        // confirm about place
        // confirm about the participants
        // dress up for the party
    }

    public void meeting() {
        // confirm about time
        // confirm about place
        // confirm about the participants
        // prepare the meeting materials
    }

}
```

© 2022 Thoughtworks Commercial in Confidence

Duplicated Code

```
public class Person {


    public void party() {
        // confirm about time
        // confirm about place
        // confirm about the participants
        // dress up for the party
    }

    public void meeting() {
        // confirm about time
        // confirm about place
        // confirm about the participants
        // prepare the meeting materials
    }

}
```

```
public class Person {


    public void party() {
        confirmSchedule();
        // dress up for the party
    }

    public void meeting() {
        confirmSchedule();
        // prepare the meeting materials
    }
    private void confirmSchedule() {
        // confirm about time
        // confirm about place
        // confirm about the participants
    }


}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
public void UpdateQuality(){
  for (var i = 0; i < Items.Count; i++) {
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a TAFKAL80ETC concert") {
      if (Items[i].Quality > 0) {
        if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
          Items[i].Quality = Items[i].Quality - 1;
        }
      }
    } else {
      if (Items[i].Quality < 50) {
        Items[i].Quality = Items[i].Quality + 1;

        if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert") {
          if (Items[i].SellIn < 11) {
            if (Items[i].Quality < 50) {
              Items[i].Quality = Items[i].Quality + 1;
            }
          }

          if (Items[i].SellIn < 6) {
            if (Items[i].Quality < 50){
              Items[i].Quality = Items[i].Quality + 1;
            }
          }
        }
      }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
      Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0) {
      if (Items[i].Name != "Aged Brie") {
        if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert") {
          if (Items[i].Quality > 0) {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
              Items[i].Quality = Items[i].Quality - 1;
            }
          }
        }
        else {
          Items[i].Quality = Items[i].Quality - Items[i].Quality;
        }
      }
      else {
```

Long Function

```
public void UpdateQuality(){
  for (var i = 0; i < Items.Count; i++) {
    if (Items[i].Name != "Aged Brie" && Items[i].Name != "Backstage passes to a
TAFKAL80ETC concert") {
      if (Items[i].Quality > 0) {
        if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
          Items[i].Quality = Items[i].Quality - 1;
        }
      }
    } else {
      if (Items[i].Quality < 50) {
        Items[i].Quality = Items[i].Quality + 1;

        if (Items[i].Name == "Backstage passes to a TAFKAL80ETC concert") {
          if (Items[i].SellIn < 11) {
            if (Items[i].Quality < 50) {
              Items[i].Quality = Items[i].Quality + 1;
            }
          }

          if (Items[i].SellIn < 6) {
            if (Items[i].Quality < 50){
              Items[i].Quality = Items[i].Quality + 1;
            }
          }
        }
      }
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
      Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0) {
      if (Items[i].Name != "Aged Brie") {
        if (Items[i].Name != "Backstage passes to a TAFKAL80ETC concert") {
          if (Items[i].Quality > 0) {
            if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
              Items[i].Quality = Items[i].Quality - 1;
            }
          }
        }
        else {
```

```
public void UpdateQuality() {
  for (var i = 0; i < Items.Count; i++) {
    if (Items[i].Name != "Aged Brie" &&
Items[i].Name != "Backstage passes to a TAFKAL80ETC concert") {
      DecrementQualityForNormalItems(i);
    } else {
      UpdateQualityForItemsThatAgeWell(i);
    }

    if (Items[i].Name != "Sulfuras, Hand of Ragnaros") {
      Items[i].SellIn = Items[i].SellIn - 1;
    }

    if (Items[i].SellIn < 0) {
      UpdateQualityForExpiredItems(i);
    }
  }
}
```

thoughtworks Commercial in Confidence

What's wrong with this code?

```
public void order(int orderNumber, String userName,
String userPhoneNumber, String userAddress,
String userGender) {

    // to do something
}
```

© 2022 Thoughtworks Commercial in Confidence

Long Parameter List

```
public void order(int orderNumber, String userName,
String userPhoneNumber, String userAddress,
String userGender) {

    // to do something
}
```

```
public void order(int orderNumber, User user) {
    // to do something
}

public class User {
    private final String name;
    private final String phoneNumber;
    private final String address;
    private final String gender;

    public User(String name, String phoneNumber,
                String address, String gender) {
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.address = address;
        this.gender = gender;
    }

    // getter and setter
    ……
}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
public List<Integer> filterEven() {
    List<Integer>  evens = new ArrayList<>();

    for (Integer item: array) {
        if(item % 2 == 0) evens.add(item);
    }

    return evens;
}
```

© 2022 Thoughtworks Commercial in Confidence

Loops

```
public List<Integer> filterEven() {
    List<Integer>  evens = new ArrayList<>();

    for (Integer item: array) {
        if(item % 2 == 0) evens.add(item);
    }

    return evens;
}
```

```
public List<Integer> filterEven() {
    return array.stream()
            .filter(item -> item % 2 == 0)
            .collect(Collectors.toList());
}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
# add 10 to x
x += 10

# Calculates the premium
public int calculatePremium(policy)
{
  ...
}

# This class compares premiums
public class PremiumComparator {
  ...
}
```

© 2022 Thoughtworks Commercial in Confidence

Comments

```
# add 10 to x
x += 10

# Calculates the premium
public int calculatePremium(policy)
{
  ...
}

# This class compares premiums
public class PremiumComparator {
  ...
}
```

```
x += 10


public int calculatePremium(policy)
{
  ...
}


public class PremiumComparator {
  ...
}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
if (user.getStatus() == 2) {
    // do something
}

if ("ACC".equals(application.getStatus())) {
    // do something
}
```

© 2022 Thoughtworks Commercial in Confidence

Magic numbers/strings

```
if (user.getStatus() == 2) {
    // do something
}

if ("ACC".equals(application.getStatus())) {
    // do something
}
```

```
public static final int SINGLE = 2;

if (user.getStatus() == SINGLE) {
    // do something
}


public static final String ACCEPTED = “ACC";

if (ACCEPTED.equals(application.getStatus())) {
    // do something
}
```

© 2022 Thoughtworks Commercial in Confidence

What's wrong with this code?

```
public boolean isEligibleForDiscount(User user) {
    boolean eligible = false;

    if (user.getEnrollmentStatus() == EnrollmentStatus.ENROLLED) {
        if (user.getAge() == ELIGIBILITY_AGE) {
            if (user.getSchool() == University.ASTRO_UNIVERSITY) {
                eligible = true;
            }
        }
    }
    else {
        eligible = false;
    }

    return eligible;
}

public boolean isEligibleForDiscountNew(User user) {

    if (isScrolled(user)) {
        return false;
    }
    if (isOfAge(user)) {
        return isInAstroUniversity(user);
    }

    return false;
}
```

© 2022 Thoughtworks Commercial in Confidence

Unused code

```
public boolean isEligibleForDiscount(User user) {
    boolean eligible = false;

    if (user.getEnrollmentStatus() == EnrollmentStatus.ENROLLED) {
        if (user.getAge() == ELIGIBILITY_AGE) {
            if (user.getSchool() == University.ASTRO_UNIVERSITY) {
                eligible = true;
            }
        }
    }
    else {
        eligible = false;
    }

    return eligible;
}

public boolean isEligibleForDiscountNew(User user) {

    if (isScrolled(user)) {
        return false;
    }
    if (isOfAge(user)) {
        return isInAstroUniversity(user);
    }

    return false;
}
```

```
public boolean isEligibleForDiscount(User user) {

    if (isScrolled(user)) {
        return false;
    }
    if (isOfAge(user)) {
        return isInAstroUniversity(user);
    }

    return false;
}
```

© 2022 Thoughtworks Commercial in Confidence

Primitive obsession

```
public class Yatzy {

   public int getOnes(int d1, int d2, int d3) {
    int sum = 0;
    if (d1 == 1) sum++;
    if (d2 == 1) sum++;
    if (d3 == 1) sum++;

    return sum;
  }

  public int getLargeStraight(int d1, int d2, int d3) {
    int[] tallies;
    tallies = new int[4];
    tallies[d1 - 1] += 1;
    tallies[d2 - 1] += 1;
    tallies[d3 - 1] += 1;
    if (tallies[1] == 1 &&
            tallies[2] == 1 &&
            tallies[3] == 1) {
        return 20;
    }

    return 0;
  }
}
```

```
public class Yatzy {

  public int getOnes(Dice dice) {
    return dice.countWithValue(DieValue.One);
  }

  public int getLargeStraight(Dice dice) {
    if (dice.countWithValue(DieValue.Two) == 1 &&
            dice.countWithValue(DieValue.Three) == 1 &&
            dice.countWithValue(DieValue.Four) == 1)
        return 20;
    return 0;
  }
}
```

© 2022 Thoughtworks Commercial in Confidence

Feature Envy

```
public class Customer {
    private Phone mobilePhone;

    public Customer(Phone mobilePhone) {
        this.mobilePhone = mobilePhone;
    }

    public String getMobilePhoneNumber() {
        return "(" +
                mobilePhone.getAreaCode() + ") " +
                mobilePhone.getPrefix() + "-" +
                mobilePhone.getNumber();
    }
}
```

```
public class Phone {
    private final String unformattedNumber;

    public Phone(String unformattedNumber) {
        this.unformattedNumber = unformattedNumber;
    }
    public String getAreaCode() {
        return unformattedNumber.substring(0,3);
    }
    public String getPrefix() {
        return unformattedNumber.substring(3,6);
    }
    public String getNumber() {
        return unformattedNumber.substring(6,10);
    }

    public String getMobilePhoneNumber() {
        return "(" + getAreaCode() +
                ") " + getPrefix() +
                "-" + getNumber();
    }
}
```

© 2022 Thoughtworks Commercial in Confidence

Temporary Field

```
public class PhoneAccount {

    private double excessMinutesCharge;
    private static final double RATE = 8.0;

    public double computeBill(int minutesUsed,
      int includedMinutes) {
        excessMinutesCharge = 0.0;
        int excessMinutes = minutesUsed - includedMinutes;
        if (excessMinutes >= 1) {
            excessMinutesCharge = excessMinutes * RATE;
        }
        return excessMinutesCharge;
    }
}
```

```
public class PhoneAccount {
    private static final double RATE = 8.0;

    public double computeBill(int minutesUsed,
      int includedMinutes) {
        int excessMinutes = minutesUsed - includedMinutes;
        if (excessMinutes >= 1) {
            return excessMinutes * RATE;
        }
        return 0.0;
    }

}
```

© 2022 Thoughtworks Commercial in Confidence

Refactoring Practice

© 2022 Thoughtworks Commercial in Confidence

Refactoring Demo

1. **Test Protection**
    - Run test, ensure all tests pass
    - If no test, create test first
2. **Find code smell**
3. **Refactor and make sure all tests pass**
4. **Baby step**
    - Do refactoring from easy to complex
    - Refactor one by one
    - One refactor with one commit

© 2022 Thoughtworks Commercial in Confidence

Refactoring Practice

https://github.com/afs-public-202211/practice-refactoring-step-by-step

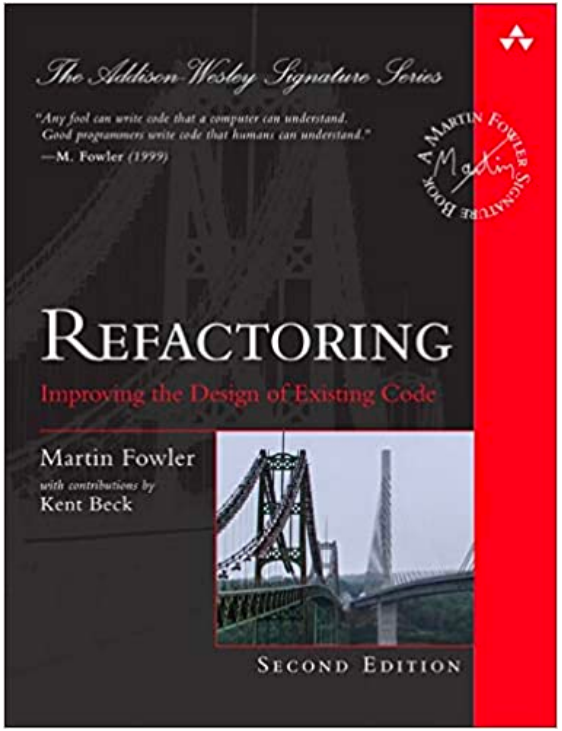
© 2022 Thoughtworks Commercial in Confidence

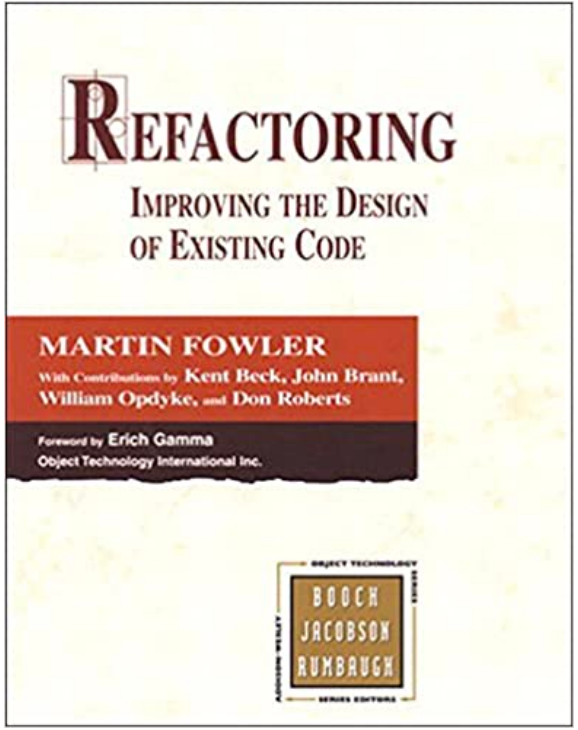
Refactoring Practice

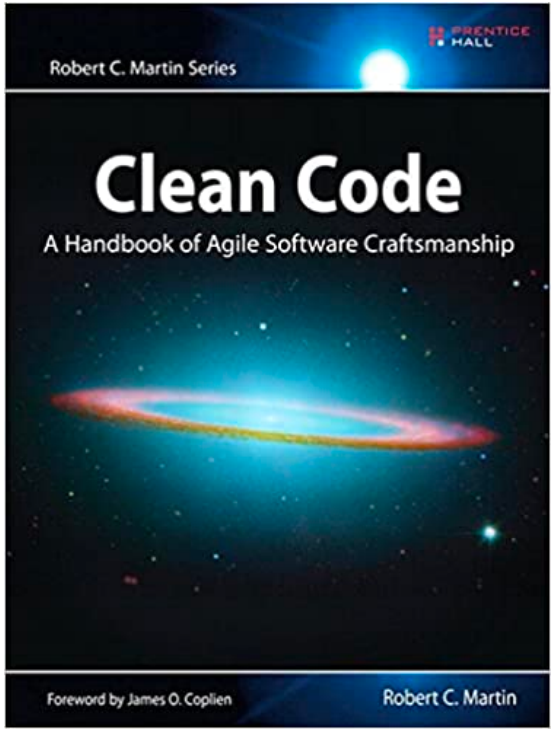
https://github.com/afs-public-202211/practice-refactoring-techniques

© 2022 Thoughtworks Commercial in Confidence

Books







© 2022 Thoughtworks Commercial in Confidence

Thank You !

thoughtworks

© 2022 Thoughtworks Commercial in Confidence