

Emergent Architecture with TDD/BDD

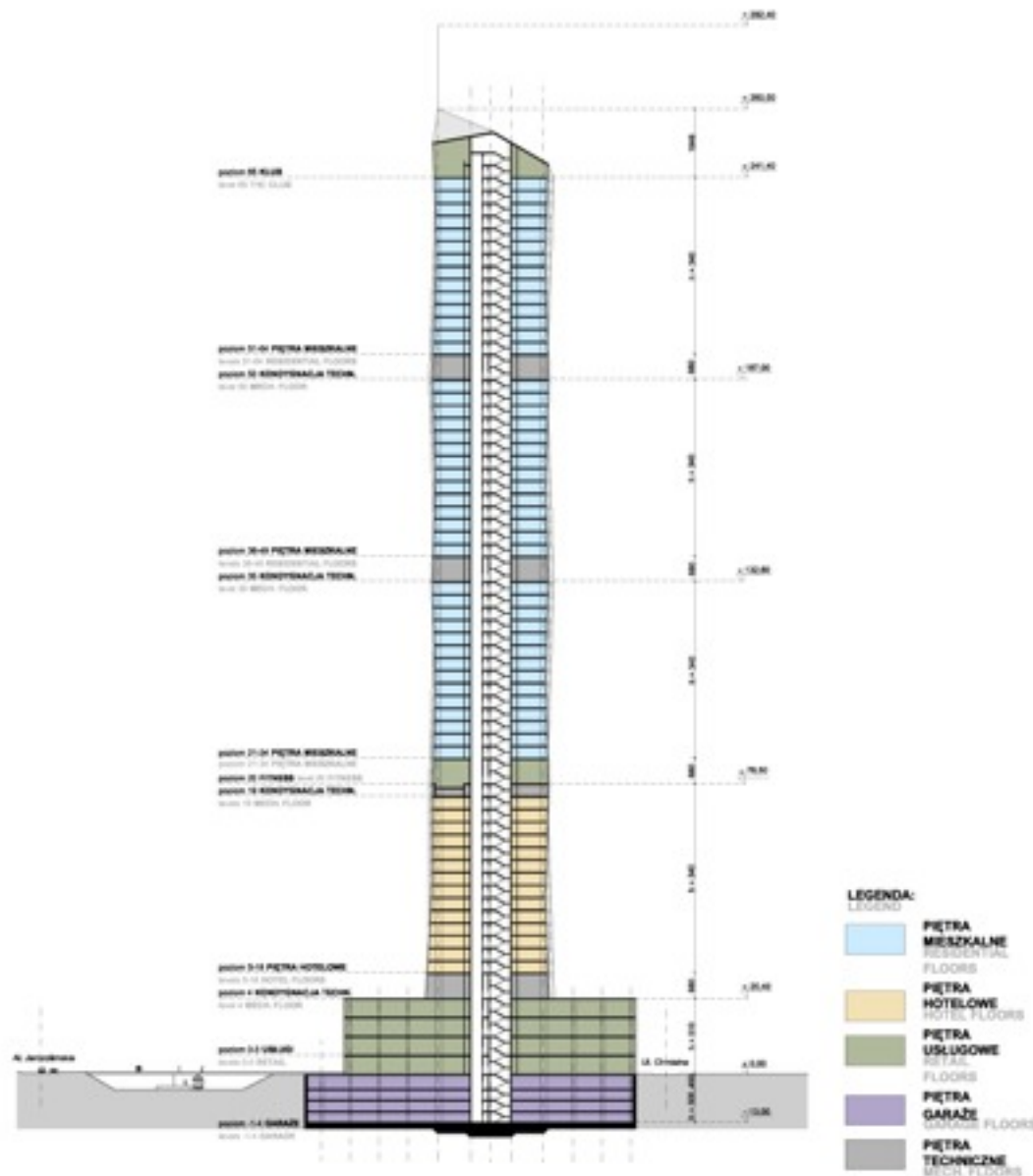
James Kovacs

Technical Evangelist, JetBrains

@jameskovacs | jameskovacs.com

james.kovacs@jetbrains.com





INWESTOR / INVESTOR

CHMIELNA DEVELOPMENT Sp. z o.o.

Warszawa
ul. Chmielna 85/87

OBIEKT / BUILDING

BUDYNEK WYSOKOŚCIOWY
HIGH-RISE BUILDING

Warszawa
ul. Chmielna / Miedziąna

PROJEKTANT / DESIGNER

mwharchitekci
ANDRZEJ WYSZYŃSKI

ul. Młotowska 1, 01-201 Warszawa, tel. 887 88 82, fax 887 88 82
biuro@mwarchitekci.pl
mgr. Dariusz KOPYŃSKI
mgr. Andrzej PŁATOWSKI
mgr. Andrzej WYSZYŃSKI

ul. Młotowska 1, 01-201 Warszawa, tel. 887 88 82, fax 887 88 82
biuro@mwarchitekci.pl
mgr. Marek GIERCZYŃSKI
mgr. Jacek DOŁĘŻYŃSKI
mgr. Maciej ZERNI
stud.arch. Maciej GŁOZDZIŃSKI

ETAP / PHASE

WYKONANIE / EXECUTION

WYKONANIE / EXECUTION

SKALA / SCALE

DATA / DATE

PROJEKT KONCEPCYJNY
CONCEPT DESIGN

WYKONANIE / EXECUTION

WYKONANIE / EXECUTION

SKALA / SCALE

DATA / DATE

ARCHITEKTURA
ARCHITECTURE

PRZEMIANKA A-A
SECTION A-A

1:1000

PAŹDZIERNIK 2007

UNIFIED
MODELING
LANGUAGE



Rational the software development company

Rational Rose®

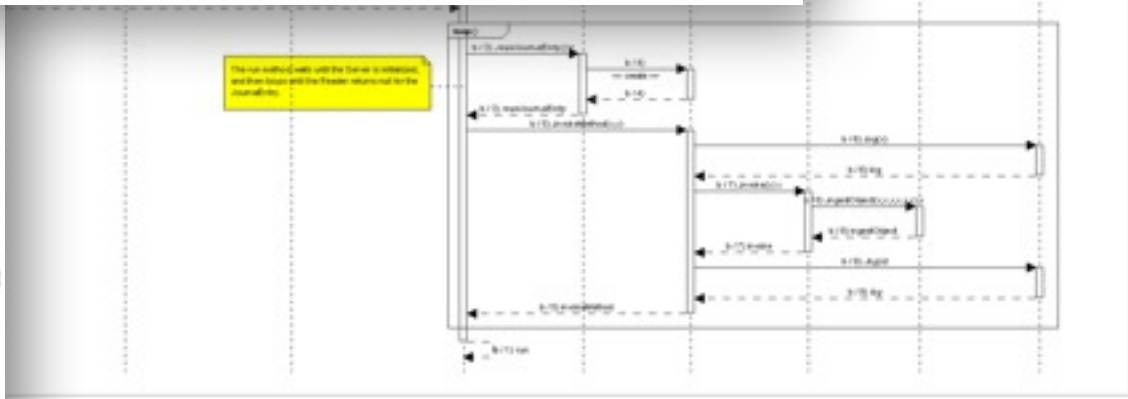
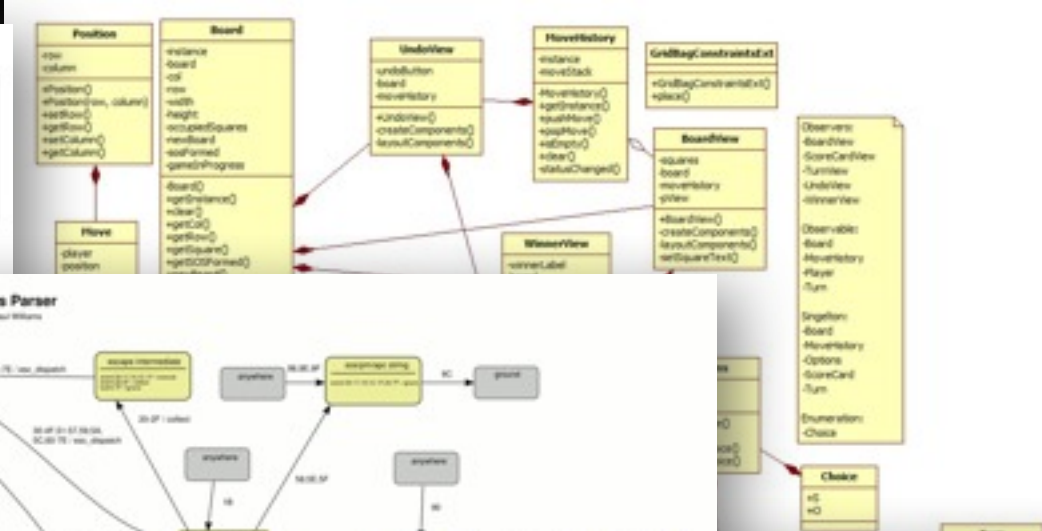
www.rational.com



Rational

the software development company

© Copyright 1991 - 2001. Rational Software Corporation.
ALL RIGHTS RESERVED. See "About Box" for legal notices.





Handwritten text in Odia script, likely a title or description of the diagram.



MONTY PYTHON and the Holy Grail



© 1974 National Film Trustee Company Limited. All Rights Reserved. Python (Monty) Pictures, Ltd.
© 2006 Layout and Design Sony Pictures Home Entertainment Inc. All Rights Reserved.






```
1 using System;
2
3 namespace Machine.Specifications.Example
4 {
5     [Subject(typeof(Account), "Funds transfer")]
6     public class when_transferring_between_two_accounts
7     : AccountSpecs
8     {
9         Because of = () =>
10             fromAccount.Transfer(1m, toAccount);
11
12         It should_debit_the_from_account_by_the_amount_transferred = () =>
13             fromAccount.Balance.ShouldEqual(0m);
14
15         It should_credit_the_to_account_by_the_amount_transferred = () =>
16             toAccount.Balance.ShouldEqual(2m);
17     }
18
19     [Subject(typeof(Account), "Funds transfer"), Tags("failure")]
20     public class when_transferring_an_amount_larger_than_the_balance_of_the_from_account
21     : AccountSpecs
22     {
23         static Exception exception;
24         Because of = () =>
25             exception = Catch.Exception(() => fromAccount.Transfer(2m, toAccount));
26
27         It should_not_allow_the_transfer = () =>
28             exception.ShouldBeOfType<Exception>();
29     }
30
31     public class failure {}
32
33 public abstract class AccountSpecs
34 {
```

Santander

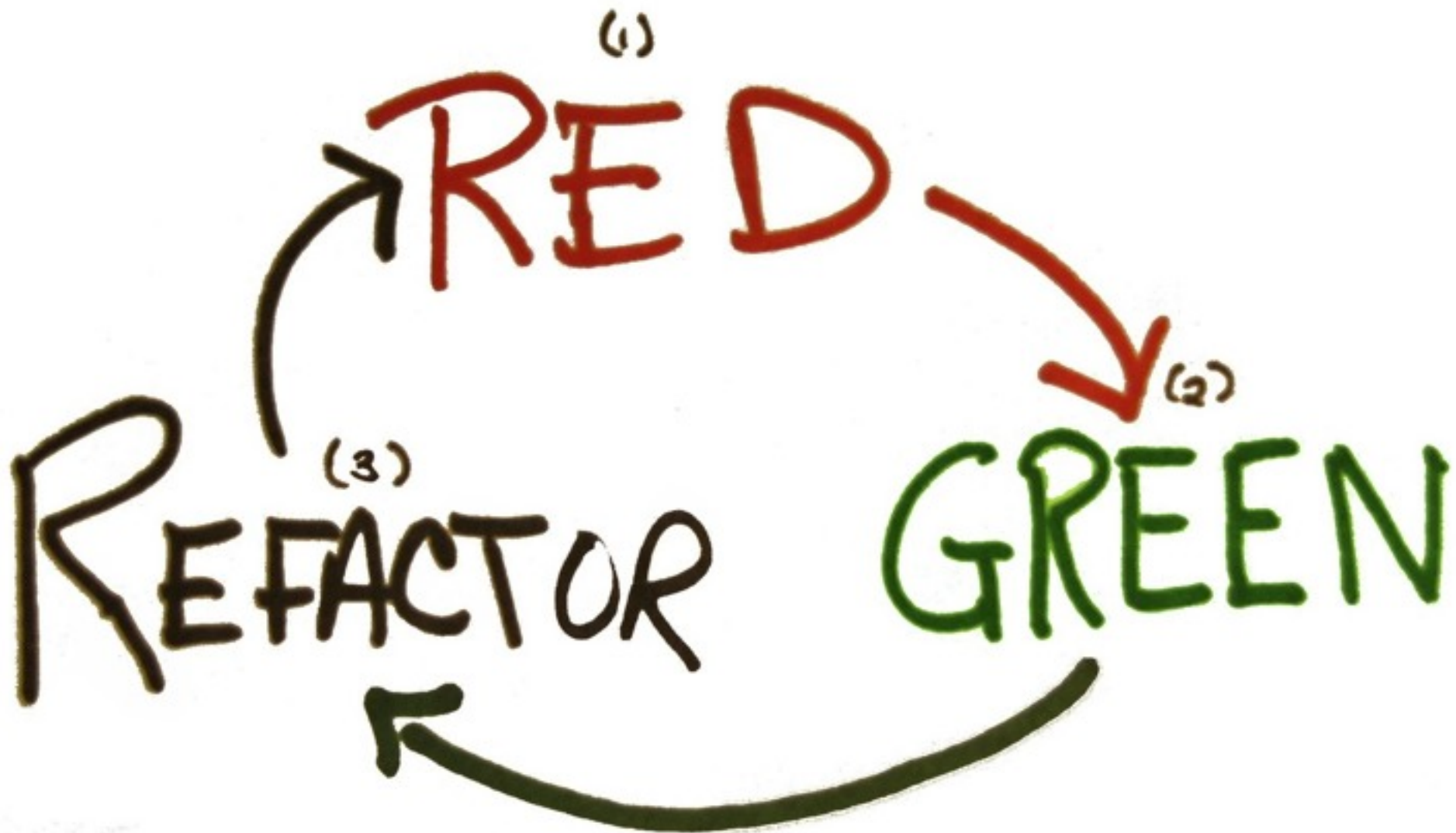














	Abby	Bob	Claire	David	Ellen	Frank
Abby		X		X		
Bob	X				X	
Claire						X
David	X				X	
Ellen		X		X		
Frank			X			

Given

- an empty shopping cart

When

- an item is added

Then

- there should be one item in the cart

Given

- a shopping cart containing a single item

When

- an identical item is added

Then

- there should be one type of item in the cart
- the count of that item should be two

Given

- a shopping cart containing items

When

- the user tries to check out over HTTP

Then

- the user should be redirected to HTTPS

Given

- a shopping cart containing items
- and the user's session is lost

When

- the user views the cart

Then

- the same items should be in the cart
- the cart should be displayed in less than one second
- an error should be logged

Resources

- “Test-Driven Development: By Example” by Kent Beck
 - <http://amzn.to/jHffKp>
- “Introducing BDD” by Dan North
 - <http://dannorth.net/introducing-bdd/>
- Machine Specifications aka MSpec
 - <https://github.com/machine/machine.specifications>
- SpecFlow
 - <http://specflow.org>
- RSpec
 - <http://rspec.info>
- Cucumber
 - <http://cukes.info>