CSSE 374: Test Driven Development & Refactoring (plus an eclectic flyover ☺)

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Learning Outcomes: Problems and Solutions

Recognize differences between problems and solutions and deal with their interactions.

- Apply Design Studio to project design task
- TDD for quality software
- Bad Code Smells
- Introduce Refactoring
- Continuous Analysis

http://www.geekologie.com/2007/02/rubiks_cube_for_the_lazy_perso.php

Design Studios

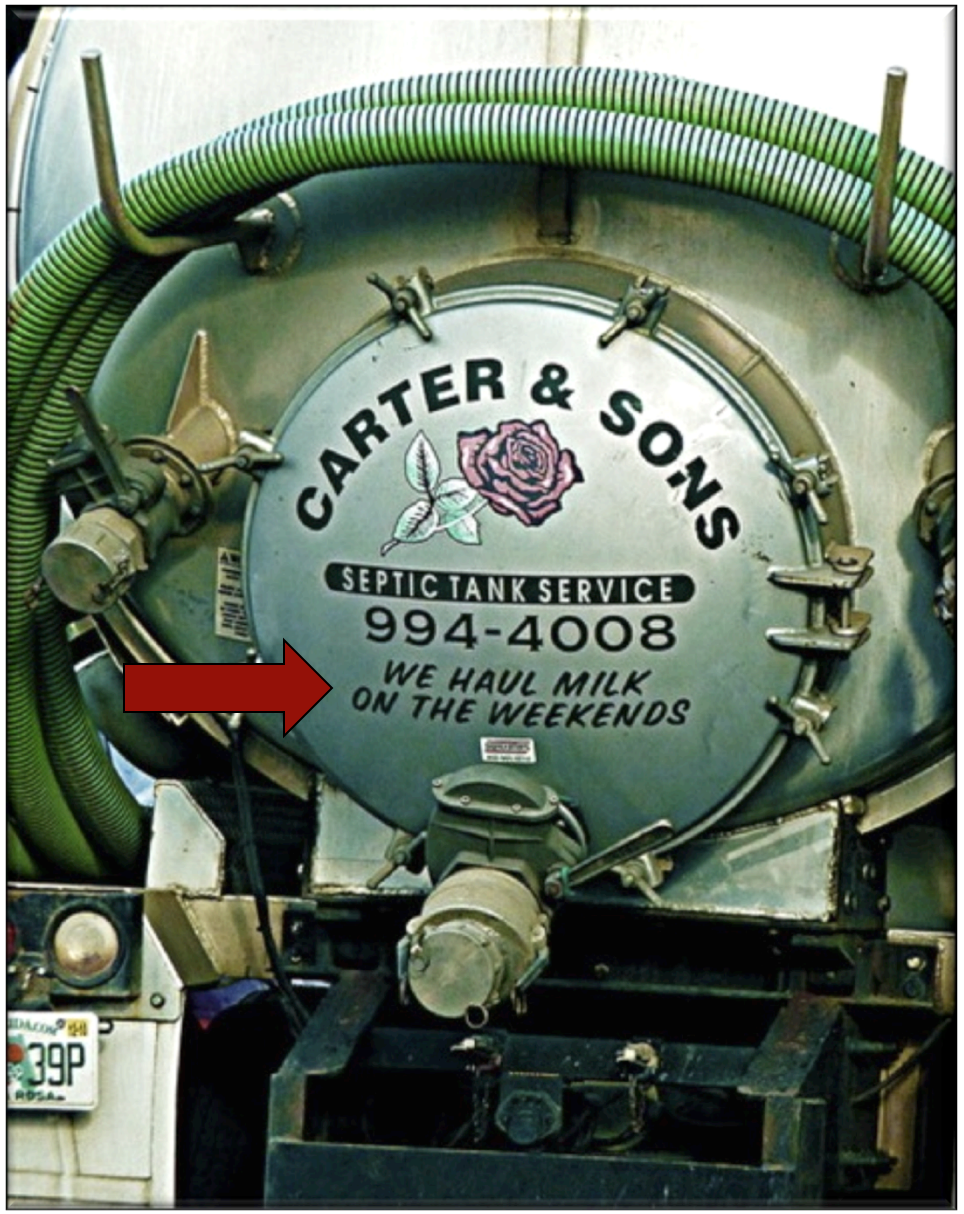
Objective is to share your design with others to communicate the approach or to leverage more eyes on a problem.

- Minute or so to set up…
- 5-6 minute discussion
- 1-2 minute answering questions

1. Team 2.2 – Rovio
2. Team 2.3 – GUI Evaluation Tool
3. Team 2.5 – Academic Paper Cataloging

A little testing goes a long way…

Perhaps a test first strategy could help! ☺



Test-Driven Development: Key Ideas

- Tests get written first before code to ensure that the software behaves as specified
- E.g., Stub in method, then write tests for method before writing the actual method
- Quickly alternate between testing and implementation (i.e., one method at a time)

- Build up a library of test cases (regression)

One advantage claimed for TDD is that it increases programmer satisfaction (versus test-last or testless development). Why might this be the case?

- Think for 14.3 seconds…
- Turn to a neighbor and discuss it for a minute

Advantages of TDD

- *Increased programmer satisfaction*
- Unit tests actually get written
- Tests serve to clarify the interface and document behavior
- As test suite grows, it serves as an automated verification
- Gives developers confidence to make changes

Bad Code Smells

- Duplicated code
- Long methods
- Class with many instance variables
- Class with many methods
- Little or no use of interfaces
- …

Not every bad smell indicates a problem

Bad Smells in Code

- Duplicated Code
- Long Method
- Large Class
- Long Parameter List
- Divergent Change
- Shotgun Surgery
- Feature Envy
- Data Clumps
- Primitive Obsession
- Switch Statements
- Parallel Interface Hierarchies
- Lazy Class
- Speculative Generality
- Temporary Field
- Message Chains
- Middle Man
- Inappropriate Intimacy
- Incomplete Library Class
- Data Class
- Refused Bequest

These are Refactoring Indicators!

Refactoring

Structured, disciplined method to rewrite/ restructure existing code *without changing its external behavior*

- **Typically combined with TDD**
  - **Tests ensure that behavior didn't change**
- ***Martin Fowler's* book on Refactoring is a must read…One of the texts for CSSE 375**

Refactorings, …Code Deodorant?

Refactoring	Description
Extract Method	Transform a long method into a shorter one by factoring out a portion into a private helper method
Extract Constant	Replace a literal constant with a constant variable
Introduce Explaining Variable	Put the result of the expression, or parts of the expression, in a temporary variable with a name that explains its purpose
…	…

Some Example Refactorings

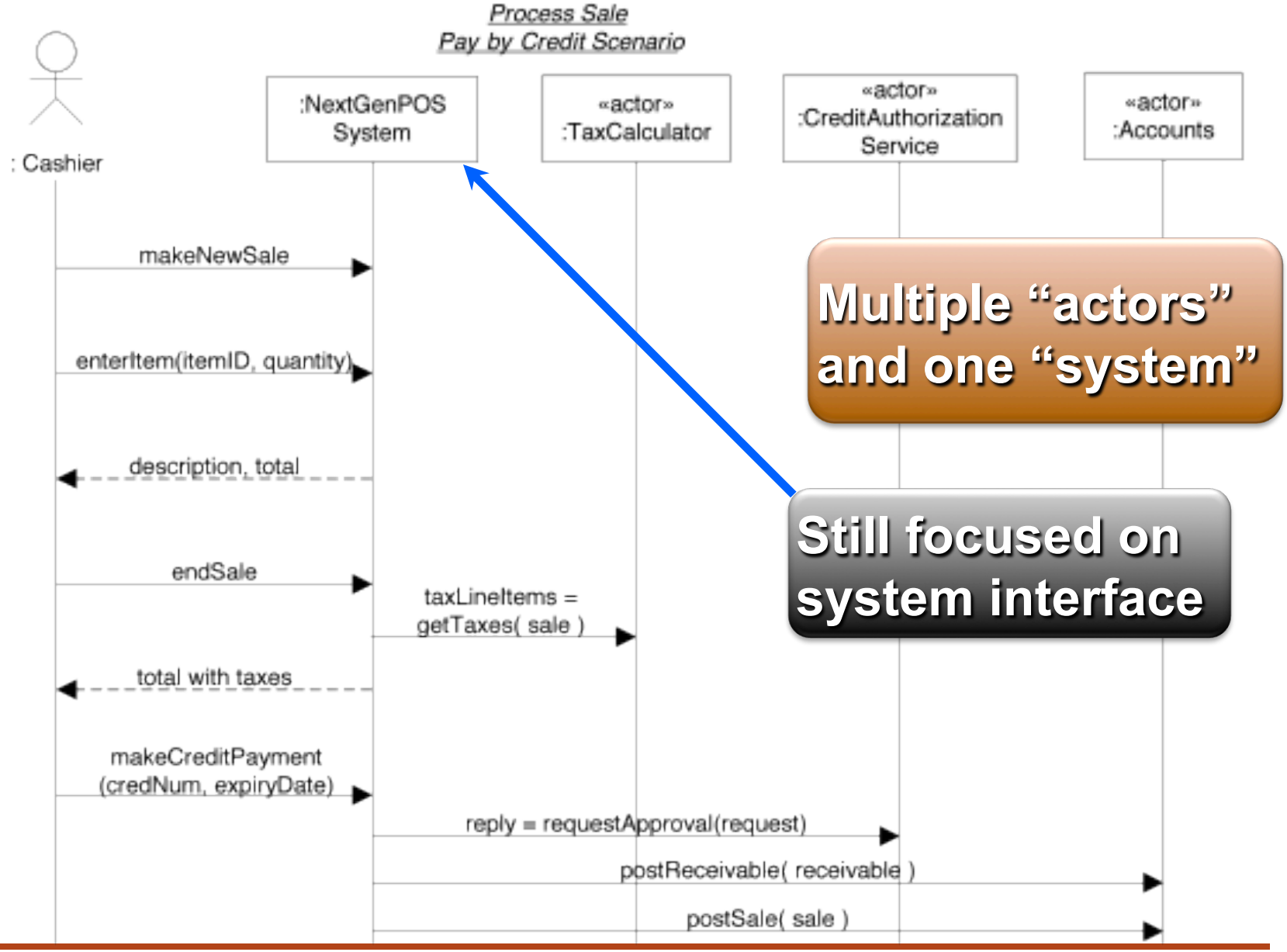
- Add Parameter
- Change Association
- Reference to value
- Value to reference
- Collapse hierarchy
- Consolidate conditionals
- Procedures to objects
- Decompose conditional
- Encapsulate collection
- Encapsulate downcast
- Encapsulate field
- Extract class
- Extract Interface
- Extract method
- Extract subclass
- Extract superclass
- Form template method
- Hide delegate
- Hide method
- Inline class
- Inline temp
- Introduce assertion
- Introduce explain variable
- Introduce foreign method

Satisfaction Guaranteed or get 110% of your product back!

From Iteration 1 to Iteration 2

- Our Iteration 2 corresponds to Milestone 4 in class
- Consider Milestone 4 and Answer quiz question
- Consider some more Analysis in 2nd Iteration

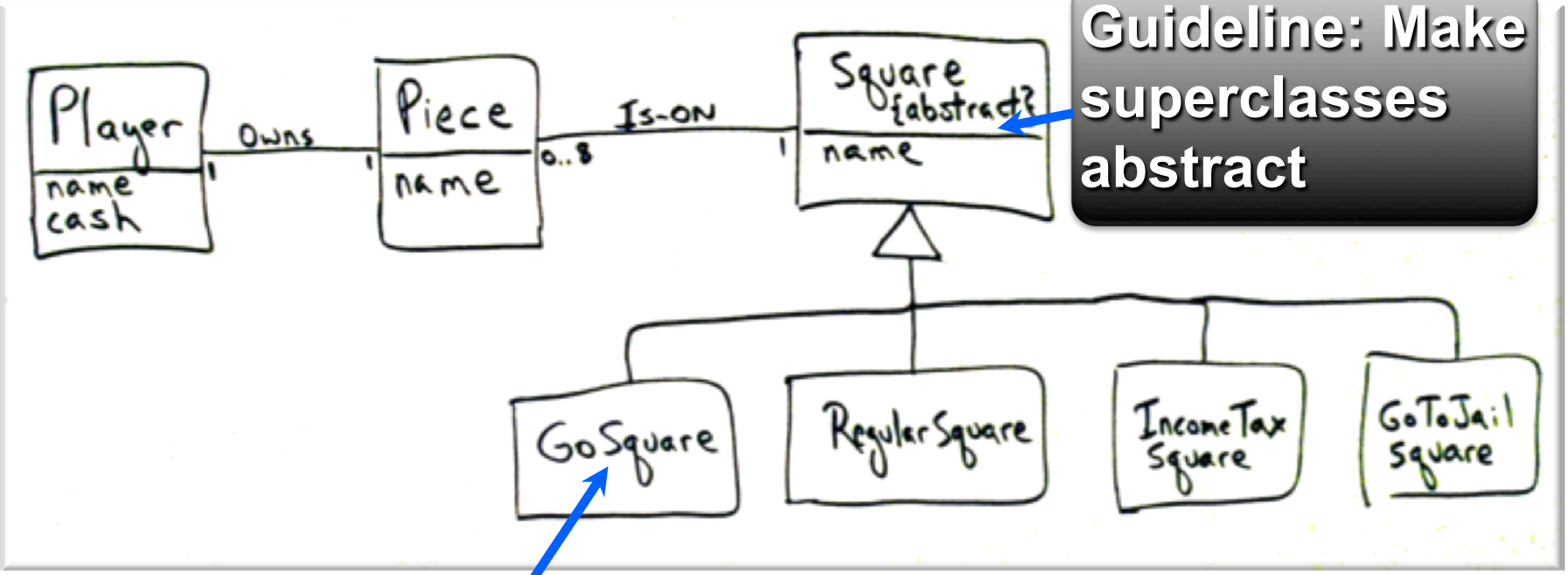
Example SSD with Intersystem Collaboration



Create Conceptual Subclasses in DM when:

- Subclass has additional attributes
- Subclass has additional associations
- Subclass concept “behaves” differently than superclass or other subclasses

Example of Conceptual Subclasses



Guideline: Make superclasses abstract

Guideline: Append superclass name to subclass

Which reason(s) for creating subclasses apply here?

Homework and Milestone Reminders

- Read Chapter 25 on More GRASP
- Milestone 4 – Junior Project Design with More GRASP'ing
  - Due by 11:59pm on Friday, January 28th, 2011
- Coming Homework 5 – BBVS Design using more GRASP Principles
  - Due by 11:59pm Tuesday, January 25th, 2011

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY