

I'll take Potpourri for 200, Alex

Curt Clifton

Rose-Hulman
Institute of
Technology



http://www.eonline.com/uberblog/celebs/c108631_Alex_Trebek.html

Q1

Plan for Today

- ✦ Schedule Notes
- ✦ Test-Driven Development
- ✦ Refactoring
- ✦ Transition to Iteration 2
 - ✦ Analysis Refresh
 - ✦ Discuss Milestone 4

Test-Driven Development

- Key Ideas:
 - 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

Advantages of TDD

- Unit tests actually get written
- Programmer satisfaction is increased
- 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

Refactoring

- ✦ Structured, disciplined method to rewrite/restructure existing code *without changing its external behavior*
- ✦ Typically combined with test-driven development
 - ✦ Tests ensure that behavior didn't change
- ✦ Rewriting is achieved by a series of very small 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

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
...	...

Cartoon of the Day



Used by permission. <http://notinventedhe.re/on/2010-1-18>

From Iteration 1 to Iteration 2

- Iteration 2 corresponds to Milestone 4 in the class
- Take a few minutes to review Milestone 4
- Answer quiz question

Some Typical Iteration 2 Activities

Though not necessarily for our projects,
since we took smaller bites in iteration 1.

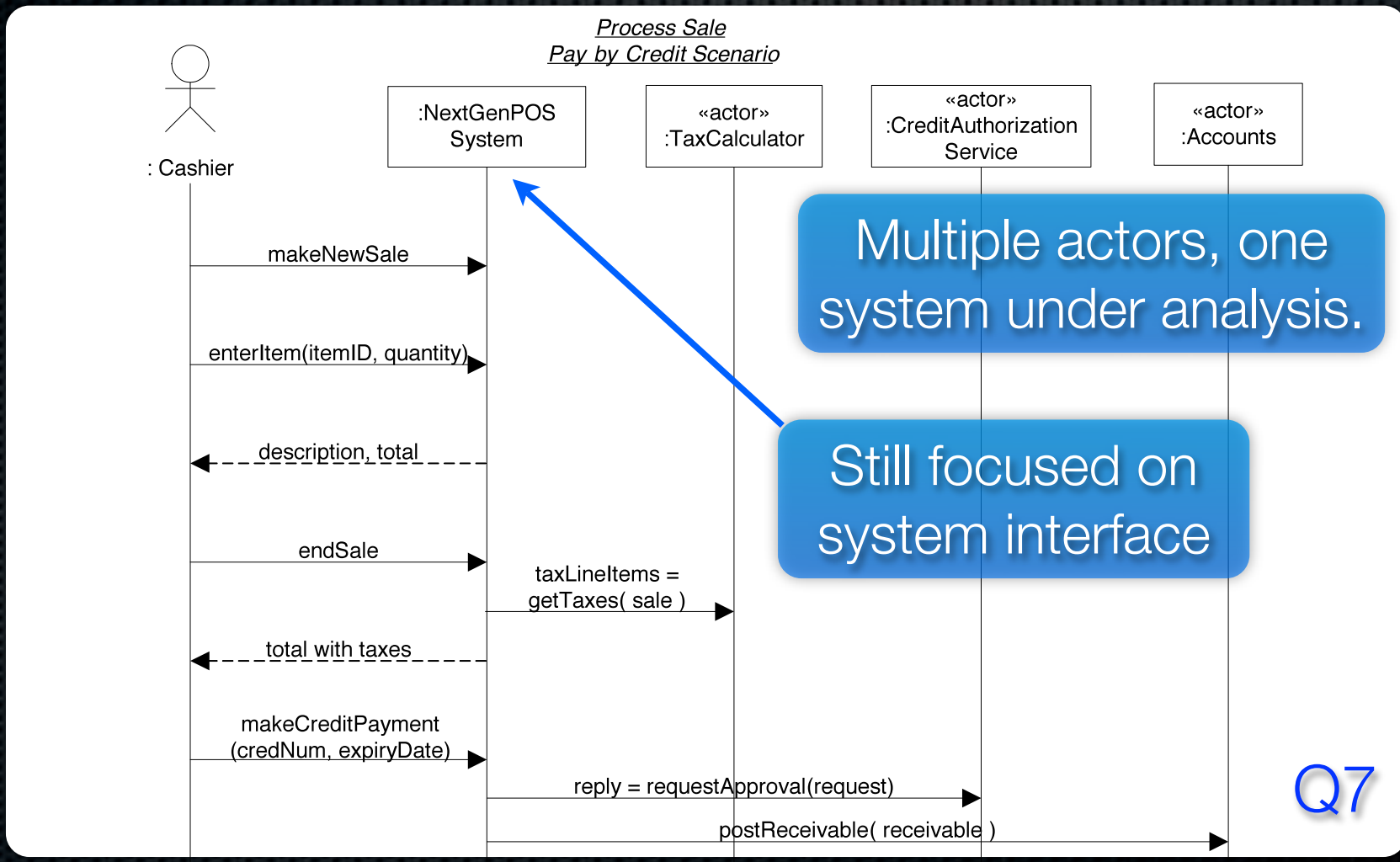
Second Iterations

- Would typically add a few lower risk use cases
 - First iteration would focus on greatest risks
- Would typically do analysis for a significant portion of the system's features—maybe 80%
 - Wouldn't implement all of them yet
- Might implement some alternative scenarios for use cases where we only did the main scenario in iteration 1

SSDs in Second Iterations

- Often updated to show some intersystem collaboration
- Update other analysis artifacts as needed...
 - Domain model: might introduce subclasses to deal with variability
 - Operation contracts: if new system operations warrant detailed post-conditions

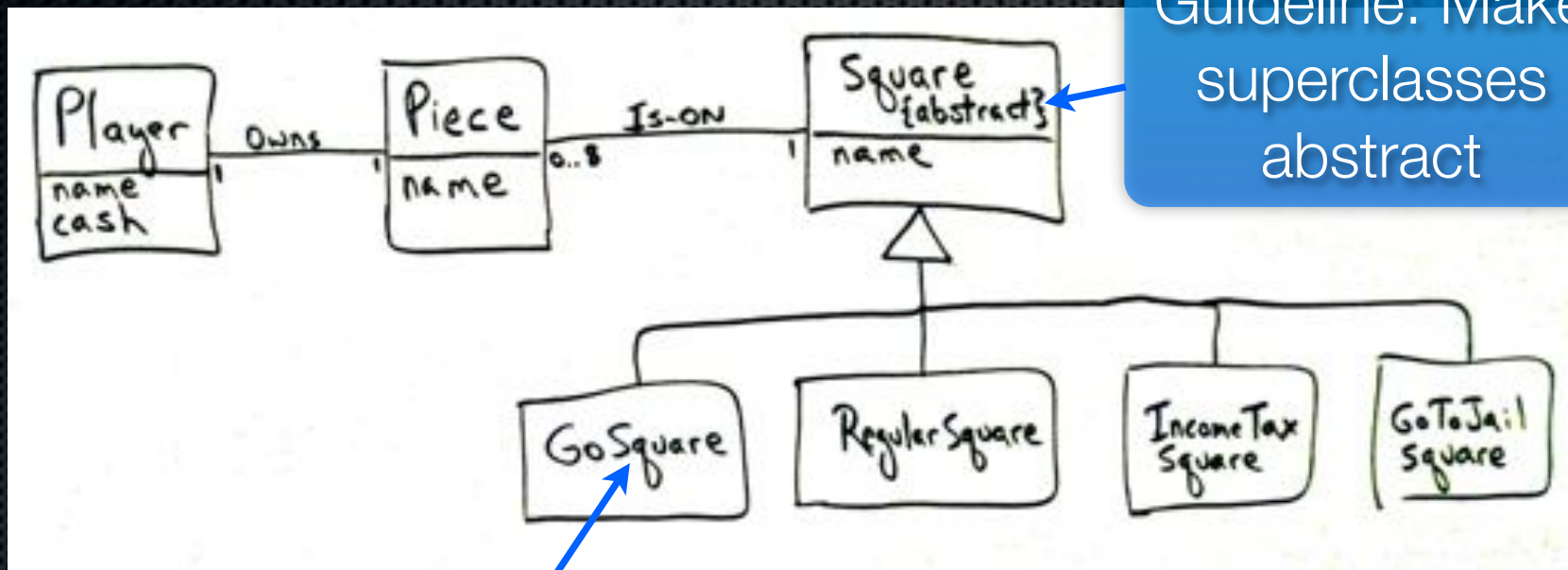
Example SSD Showing Intersystem Collaboration



Conceptual Subclasses in Domain Models

- ✦ Create a conceptual subclass 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?