Code Refactoring
Live Coding Demo

Peter Kofler, 'Code Cop'
@codecopkofler
www.code-cop.org

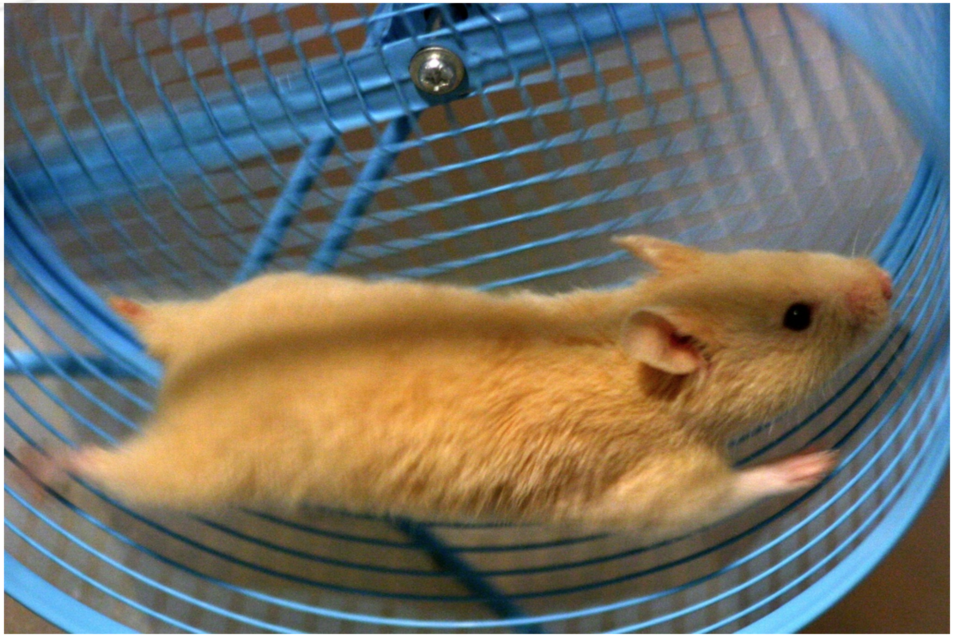
Peter Kofler

- Ph.D. (Appl. Math.)
- Professional Software Developer for 15 years
- Specialised in Re-Engineering of Legacy Code
- “fanatic about code quality”

Training on the Job?

Yes, some but...

- only what is already there
- Trial & Error not popular in production
- no practice - only production
- time pressure

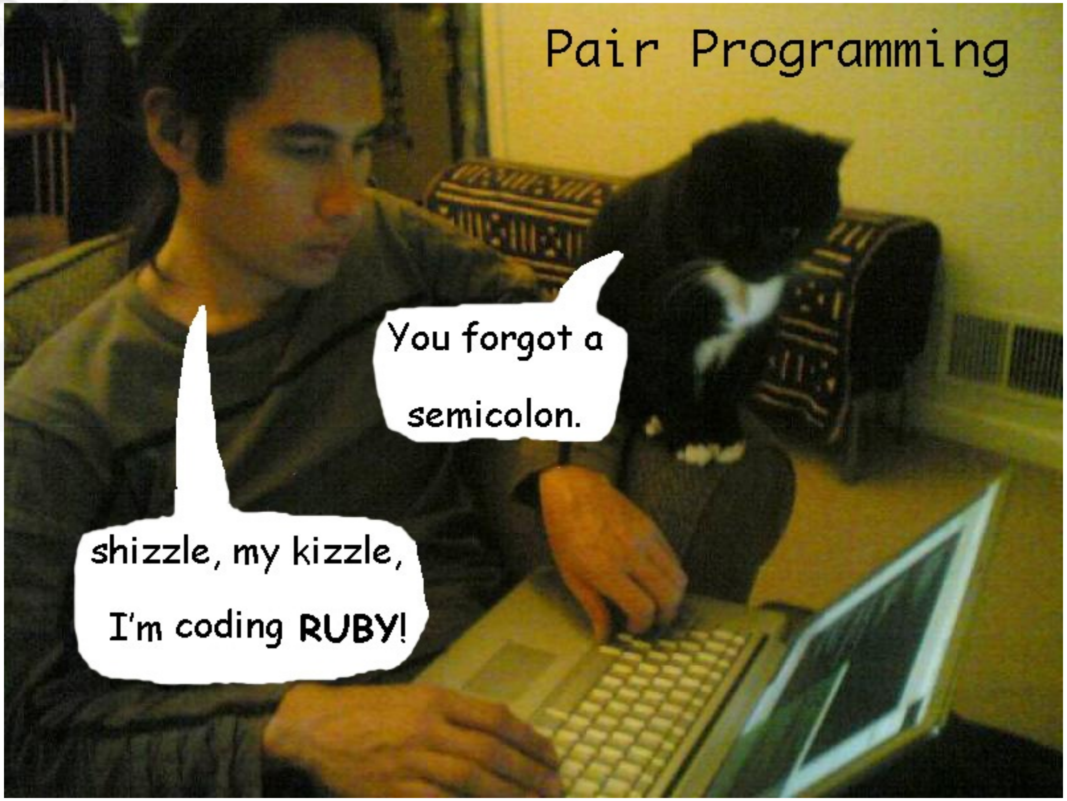


I help development teams with

- Professionalism
- Quality and Productivity
- Continuous Improvement

Mentoring

- Pair Programming
- Programming Workshops
- Deliberate Practice, e.g. Coding Dojos



Developing Quality Software Developers

Refactoring is a technique
for **restructuring**
an existing body of code,
altering its internal structure
without changing
its external behaviour.

(Martin Fowler)

Welcome to the Gilded Rose

The existing inventory system

- We have **items** to sell. Items degrade in quality the older they get.
- All items have a **SellIn value** which denotes the number of days we have to sell the item.
- All items have a **Quality value** which denotes how valuable the item is.

Requirements

- At the end of each day our system lowers both values for every item.
- Once the sell by date has passed, Quality degrades twice as fast.
- The Quality of an item is never negative.
- The Quality of an item is never more than 50.

Special Item: Brie

- *Aged Brie* actually increases in Quality the older it gets.

Backstage Passes

- *Backstage passes*, increases in Quality as it's SellIn value approaches (by a complex formula)

- but Quality drops to 0 after the concert.

Special Item

- *Sulfuras*, a legendary item, never has to be sold or decreases in Quality.

New Requirement

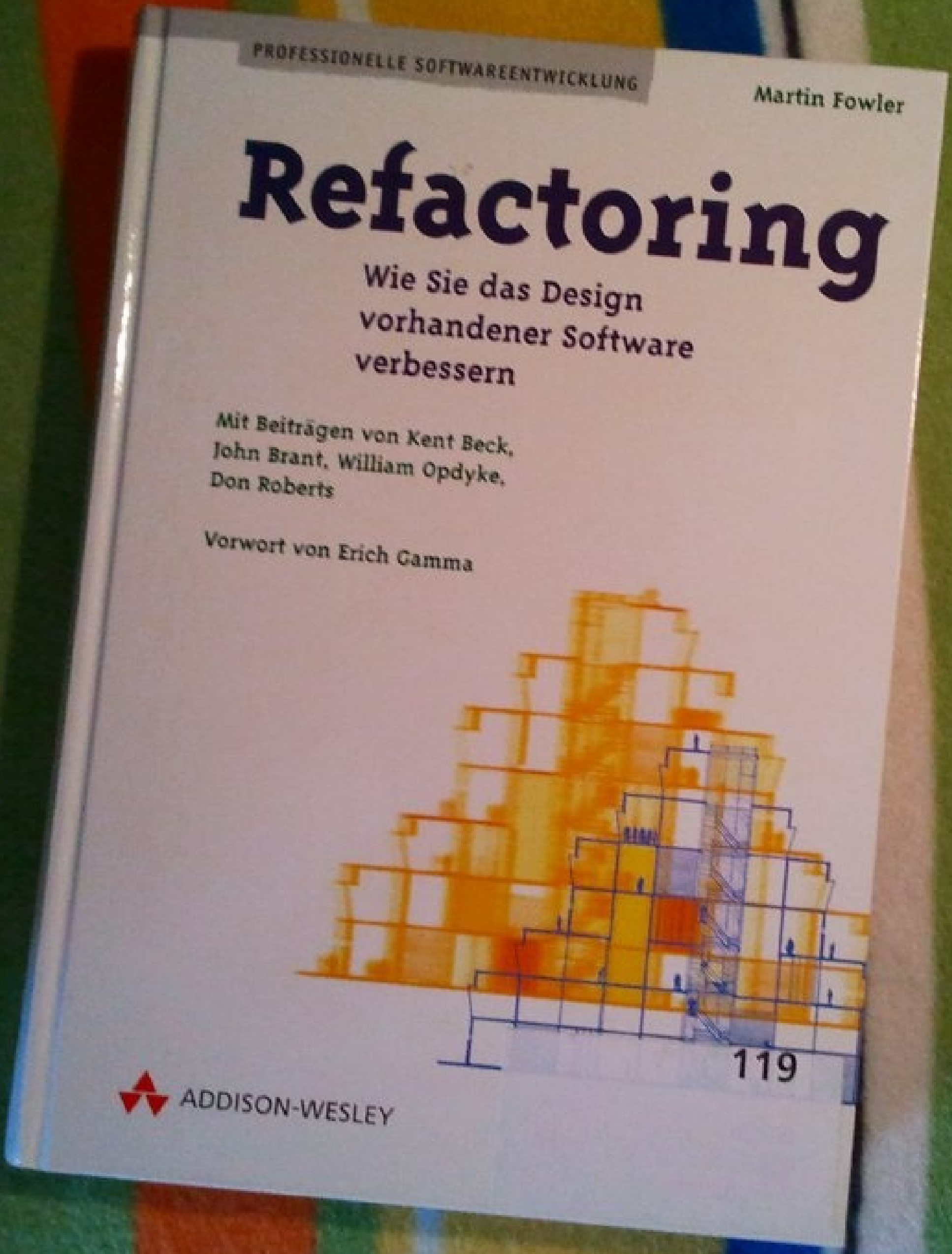
- We have recently signed a supplier of conjured items. This requires an update to our system:
  - *Conjured* items degrade in Quality twice as fast as normal items.

What Will We Do Now?

- I will implement the new requirement.
- We will change the code together.
- I will (need) to refactor the existing code.

Demo

Step by step: *https://bitbucket.org/pkofler/gilded-rose-refactoring/branch/refactor-7*



PROFESSIONELLE SOFTWAREENTWICKLUNG

Martin Fowler

Refactoring

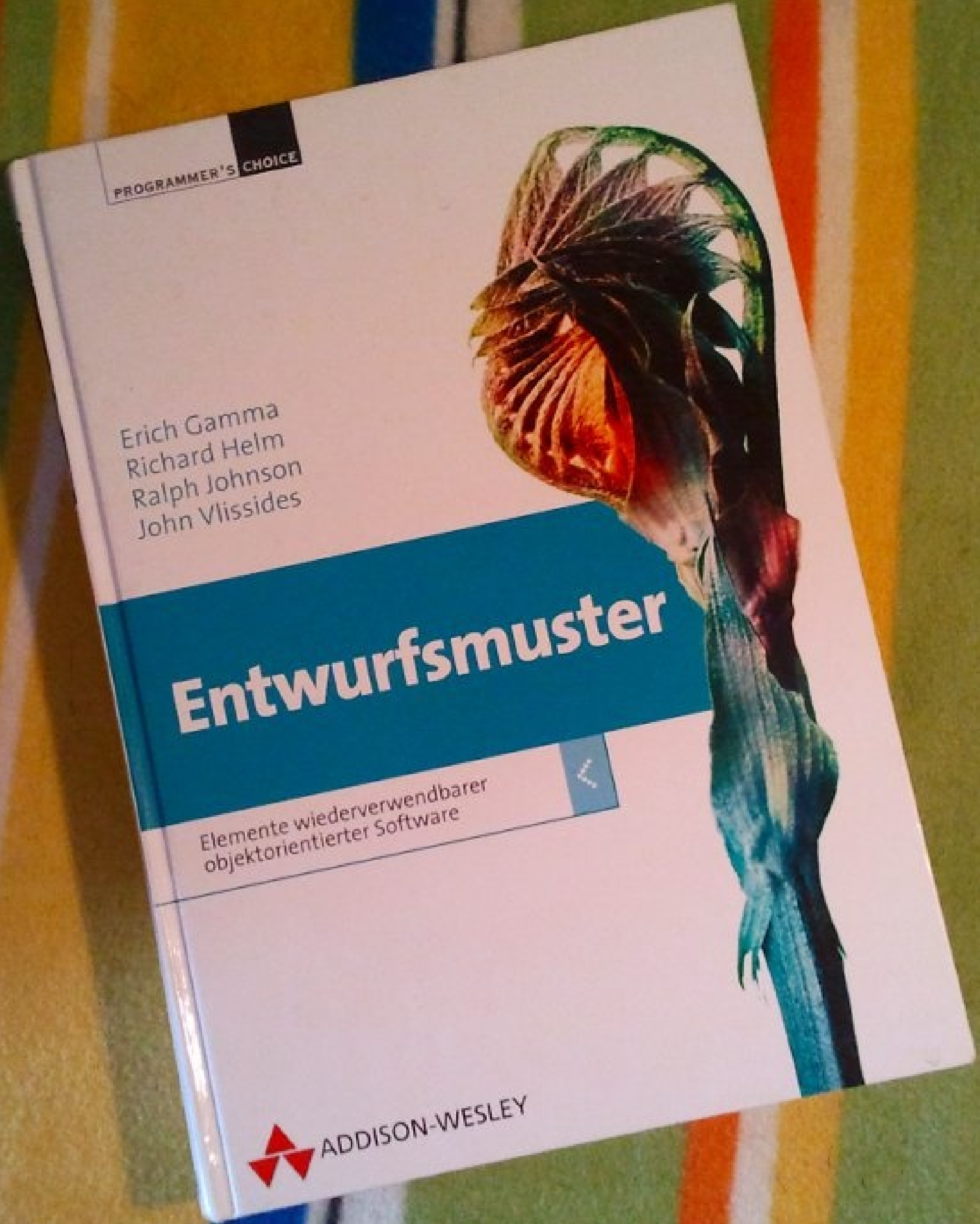
Wie Sie das Design
vorhandener Software
verbessern

Mit Beiträgen von Kent Beck,
John Brant, William Opdyke,
Don Roberts

Vorwort von Erich Gamma

119

ADDISON-WESLEY



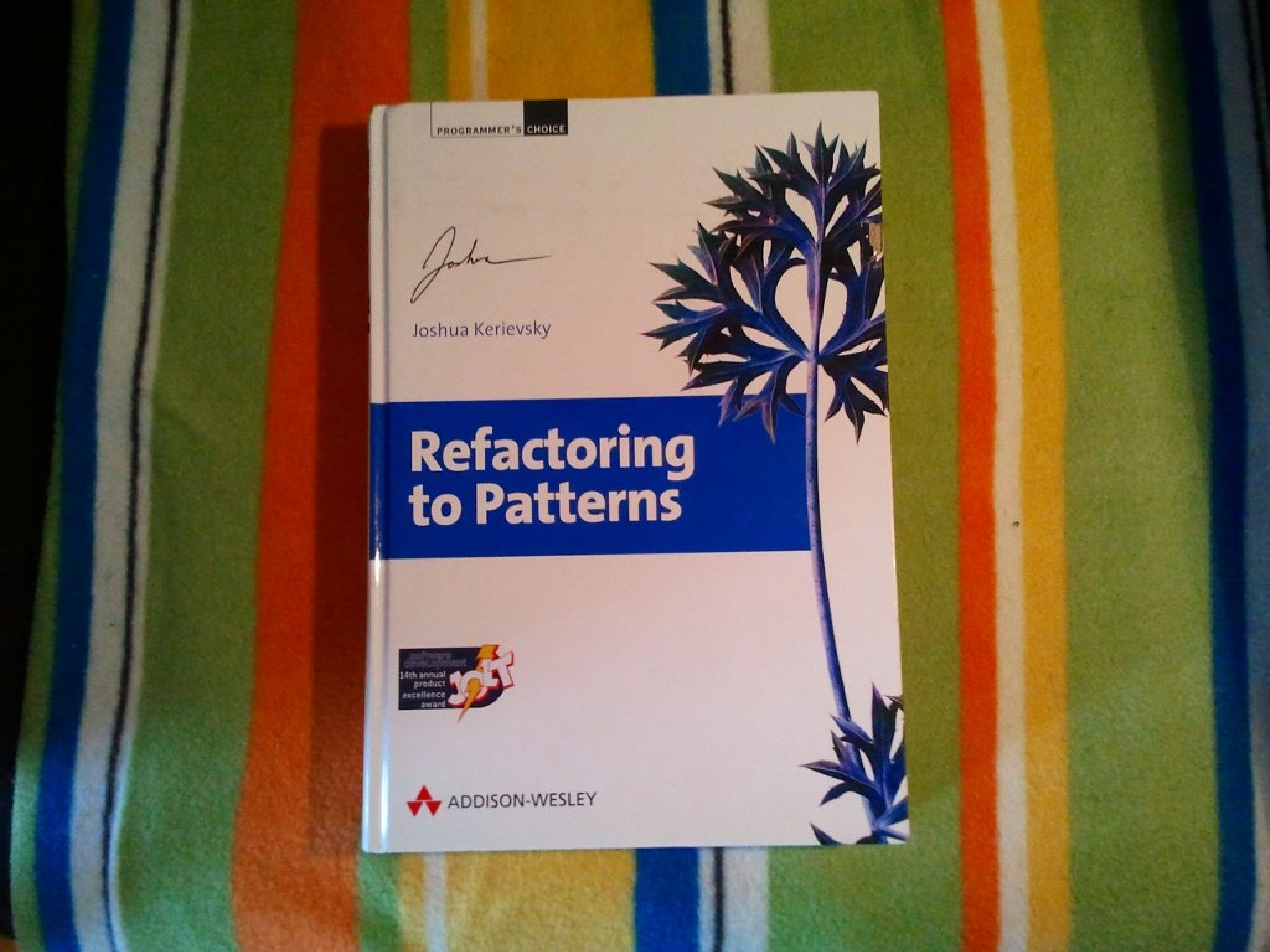
PROGRAMMER'S CHOICE

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Entwurfsmuster

Elemente wiederverwendbarer
objektorientierter Software

ADDISON-WESLEY



PROGRAMMER'S CHOICE

Joshua

Joshua Kerievsky

Refactoring to Patterns

ADDISON-WESLEY

Now the New Requirement!

- We have recently signed a supplier of conjured items. This requires an update to our system:
  - *Conjured* items degrade in Quality twice as fast as normal items.

What I did

- could not implement the requirement easily, had to refactor first.
- made it more readable.
- removed duplication (extracted duplicates).
- split method into logically coherent blocks.

What I did

- simplified complex boolean conditions.
- replaced duplicated if-statements with polymorphy (extracted Strategy pattern).
- moved common code into the strategy.
- finally implemented the new requirement ;-)

Read these books now!
Explore the Refactorings!
Practice them in your IDE!

Peter Kofler

@codecopkofler

www.code-cop.org

Gilded Rose Kata by

Emily Bache

@emilybache

http://coding-is-like-cooking.info/2013/03/writing-good-tests-for-the-gilded-rose-kata/

CC Images

- Hamster http://www.flickr.com/photos/zebrapares/4529836138
- master http://www.flickr.com/photos/sherpas428/4350620602/
- pairing http://www.flickr.com/photos/dav/94735395/
- inn http://www.flickr.com/photos/danielleblue/170496395/
- Brie http://www.flickr.com/photos/chez_loulou/2767503201
- pass http://www.flickr.com/photos/frf_kmeron/5556518514
- Sulfuras https://www.flickr.com/photos/sharelabs/11195626116