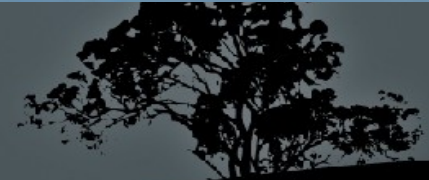


Refactoring to Patterns

How Mephisto went from a single engine Lada to a multi-engine jet fighter



The Story

Behind a multi-engine
Mephisto



The Story

Comments awaiting your approval

(27) Subversion Primer for Rails projects

(16) Piston 2.0 Progress: Piston can import Subversion repositories into Subversion working copies



The Story

237 spams



The Story

Over a weekend



The Story

Comments for all articles

"[b]Real Amateur Porn Stolen from users PC[/b] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/wmf/s.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/ox6u.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/em83l.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/vib9x2.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/3ow3g.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/n58k3.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/bn1cv.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/52a8i.jpg[/img]/url] [url=http://mixtube.net/homemade/][img]http://mixtube.net/homemade/c92qa.jpg[/img]/url] Hey, I have a lot of respect for all you guys who like to eat pussy because there are too few of you out there. And I'm not the only woman who says this. Furthermore, some of you guys who are giving it the old college try are not doing too well, so maybe this little lesson will help you out. When a woman finds a man who gives good head, she's found a treasure she's not going to let go of too quickly. This is one rare customer and she knows it. She won't even tell her girlfriends about it or that guy will become the most popular man in town. So, remember, most guys can fuck, and those who can usually do it satisfactorily, but the guy who gives good head, he's got it made. Most women are shy about their bodies. Even if you've got the world's most gorgeous woman in bed with you, she's going to worry about how you like her body. Tell her it's beautiful, tell her which parts you like best, tell her anything, but get her to trust you enough to let you down between her legs. Now stop and look at what you see. Beautiful, isn't it? There is nothing that makes a woman more unique than her pussy. I know. I've seen plenty of them. They come in all different sizes, colors and shapes; some are tucked inside like a little girl's cuntie and some have thick luscious lips that come out to greet you. Some are nested in bushes of fur and others are covered with transparent fuzz. Appreciate your woman's unique qualities and tell her what makes her special."

— [jenixmixbe](#) ([jenixmixbe@mail.ru](#)) said 3 minutes ago [Edit](#) | [Approve](#) | [Delete](#)

The Story

- Long



The Story

- Long
- Arduous



The Story

- Long
- Arduous
- Hazardous

Comments for all articles

Ooops!



~~Cred: peopl I love Git, and I want to learn more. Thanks for showing me.
— I canjames@gmail.com, said 2 min. ago —~~



The Story - Solution

Akismet

[About Akismet](#)

[Download](#)

[FAQ](#)

[Commercial Use](#)

[Development](#)

[Blog](#)

[Contact Us](#)

Remember the days
before spam?

Download
Akismet



The Story - Solution

[WHAT IS IT?](#)[SIGN UP](#)[DOWNLOADS](#)[API](#)[COMMERCIAL USE](#)[FAQ](#)[COMMUNITY](#)

End annoying comment spam. (we mean it!)

Tired of wasting time managing comment spam on your blog? Try Defensio. You'll soon wonder how you ever lived without it.

Sign up Now

Free!



Stellar Accuracy

You'll deal with less spam on your blog, period.

Easier Spam Management

Managing your spambox will become dead simple.

Personalized Filtering

Your own habits will help make the filter better over time.

Transparency Through Statistics

Meaningful stats so that you'll really know what's going on.

[Learn more about Defensio »](#)

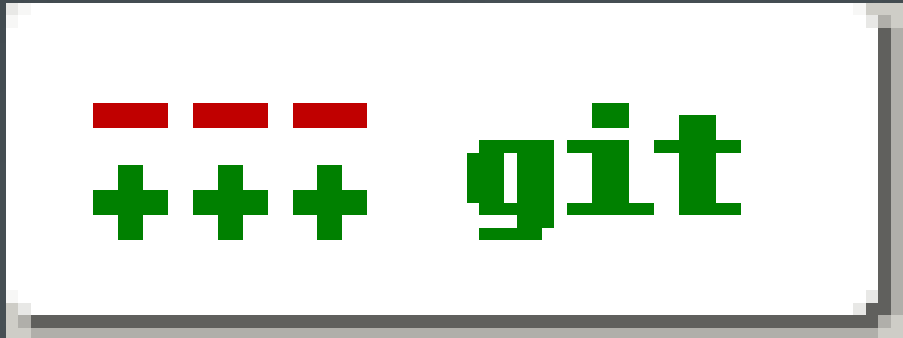
The Story - Solution

- Carl kindly asked for a Defensio plugin on Mephisto



The Story - Solution

- I wanted to try Git



```
+++ git
```



The Story - Solution

- Mephisto arrived on GitHub

Mephisto hits 0.8, moves to GitHub

Posted by [courtenay](#) | [22 comments](#)



The Story - Solution

**Time to Refactor
to Design Patterns**



Refactoring

What is refactoring ?



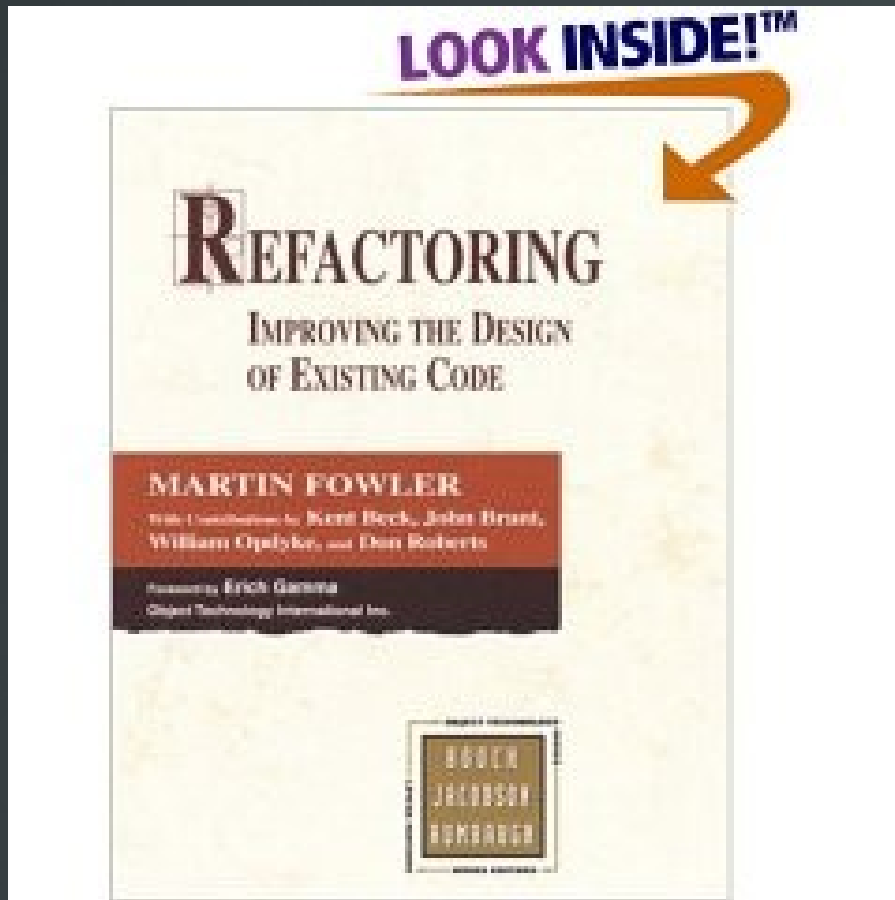
Refactoring

Refactoring (noun): a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior.

Martin Fowler, *Refactoring: Improving the Design of Existing Code*, page 53



Refactoring



- Refactoring:
Improving the Design
of Existing Code



Refactoring

Without changing its external behavior



Refactoring - Example

```
class Comment
  def check_approval
    # ...
  end
end
```



Refactoring - Example

```
class Comment
  def ham?
    # ...
  end
end
```



Design Patterns

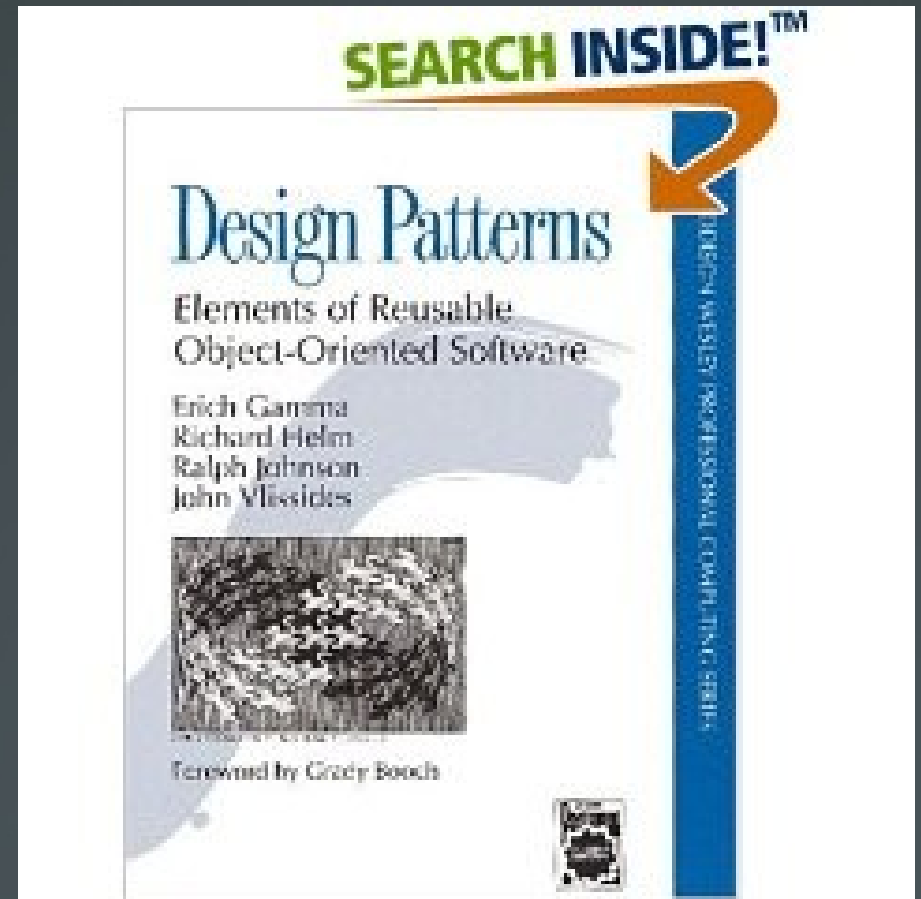
Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice

Christopher Alexander, *A Pattern Language*, page x



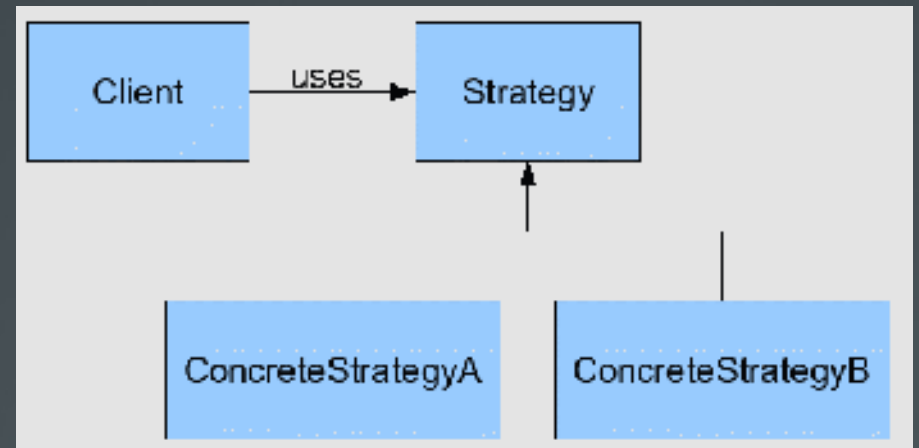
Design Patterns

- Design Patterns:
Elements of Reusable
Object-Oriented
Software



Design Patterns - Example

- Strategy: Define a family of algorithms, encapsulate each one, and make them interchangeable.



Refactoring to Patterns

Refactoring, with the intention
of instantiating a design pattern



Mephisto 0.8 *Drax*

```
class Comment
  def check_approval(site, request)
    self.approved = site.approve_comments?
    if valid_comment_system?(site)
      akismet = Akismet.new(site.akismet_key, site.akismet_url)
      self.approved = !akismet.comment_check(
        comment_spam_options(site, request))
    end
  end
end
```



Mephisto 0.8 *Drax* + Defensio

```
class Comment
  def check_approval(site, request)
    self.approved = site.approve_comments?
    if valid_akismet_comment_system?(site)
      akismet = Akismet.new(site.akismet_key, site.akismet_url)
      self.approved = !akismet.comment_check(
        akismet_comment_spam_options(site, request))
    elsif valid_defensio_comment_system?(site)
      defensio = Defensio.new(site.defensio_key,
site.defensio_url)
      self.approved = !defensio.comment_check(
        defensio_comment_spam_options(site, request))
    end
  end
end
end
```



Mephisto 0.8 *Drax* + Defensio + Chummy

```
class Comment
  def check_approval(site, request)
    self.approved = site.approve_comments?
    case
    when valid_akismet_comment_system?(site)
      akismet_validation(site, request)
    when valid_defensio_comment_system?(site)
      defensio_validation(site, request)
    when valid_chummy_comment_system?(site)
      chummy_validation(site, request)
    # and so on...
  end
end
end
```



Core of the problem ?

Be able to support multiple
algorithms to detect comment spam

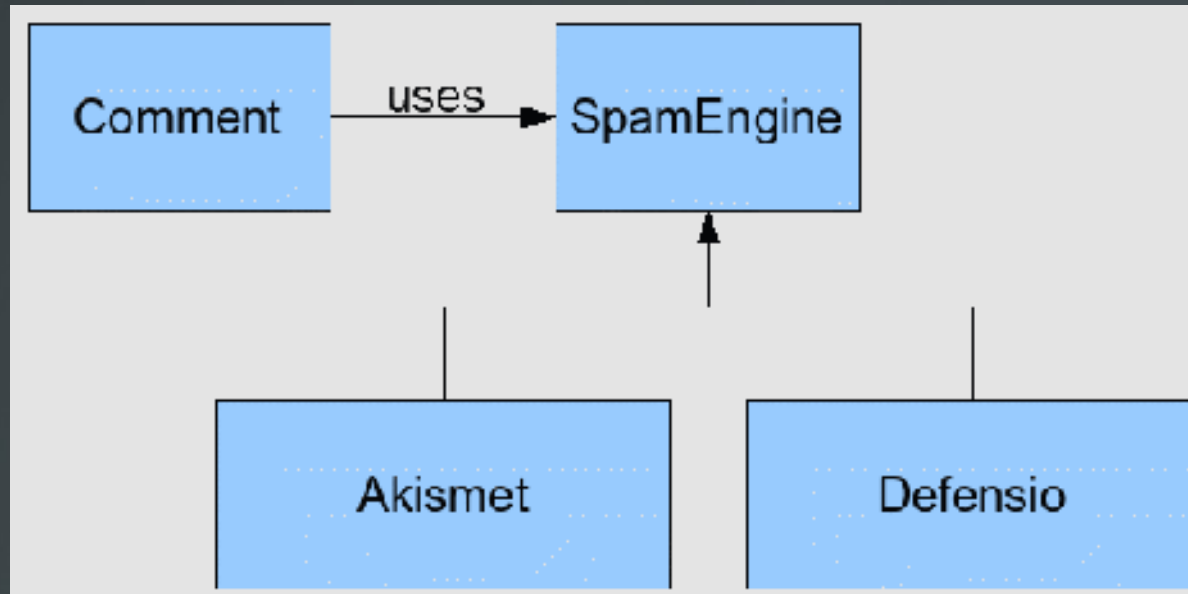


How ?

Strategy Design Pattern



Refactoring to Strategy Pattern



Refactoring to Strategy Pattern

Introduce Class

```
class Mephisto::SpamDetectionEngines::AkismetEngine  
end
```

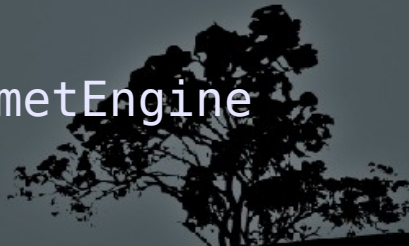


Refactoring to Strategy Pattern

Move Method

```
class Comment
  def check_approval(site, request)
    self.approved = site.approve_comments?
    if valid_comment_system?(site)
      akismet = Akismet.new(
        site.akismet_key, site.akismet_url)
      self.approved = !akismet.comment_check(
        comment_spam_options(site, request))
    end
  end
end
```

```
class Mephisto::SpamDetectionEngines::AkismetEngine
end
```

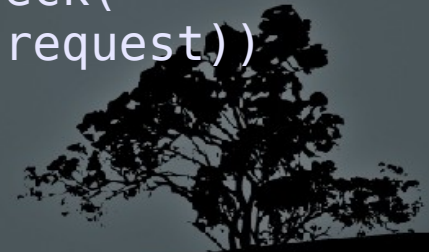


Refactoring to Strategy Pattern

Move Method

```
class Comment
  def check_approval(site, request)
    Mephisto::SpamDetectionEngines::AkismetEngine.new(
      site.akismet_key, site.akismet_url).check_approval(
        self, site, request)
  end
end
```

```
class Mephisto::SpamDetectionEngines::AkismetEngine
  def check_approval(comment, site, request)
    comment.approved = !akismet.comment_check(
      comment_spam_options(comment, site, request))
  end
end
```



Refactoring to Strategy Pattern

Structural Changes



Refactoring to Strategy Pattern

Introduce Subclass

```
class Mephisto::SpamDetectionEngines::DefensioEngine <  
  Mephisto::SpamDetectionEngines::AkismetEngine  
end
```



Refactoring to Strategy Pattern

Add Behavior

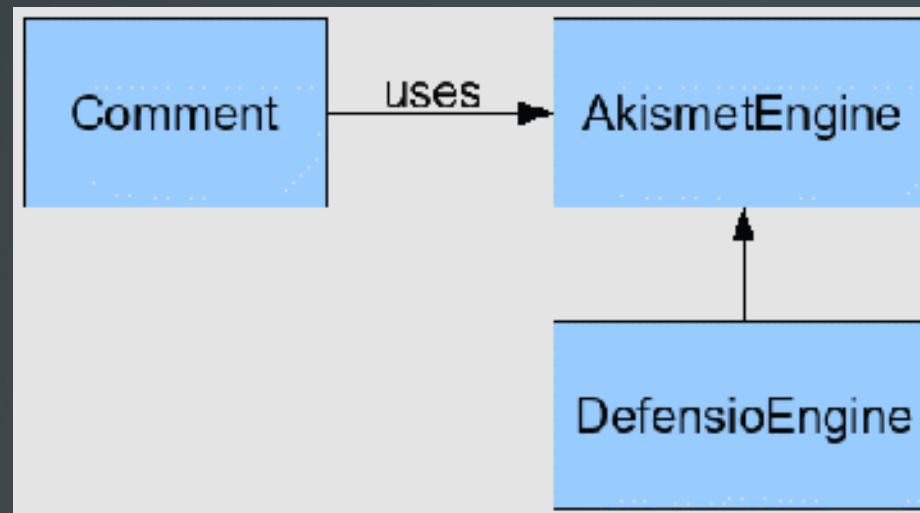
```
class Mephisto::SpamDetectionEngines::DefensioEngine <
  Mephisto::SpamDetectionEngines::AkismetEngine
  def check_approval(comment, site, request)
    !defensio.check_comment(
      comment_spam_options(comment, site, request))
  end

  def comment_spam_options(comment, site, request)
    # Defensio-specific spam options
  end
end
```



Refactoring to Strategy Pattern

Structural Changes



Refactoring to Strategy Pattern

Extract Superclass

```
module Mephisto::SpamDetectionEngines
  class Base
  end

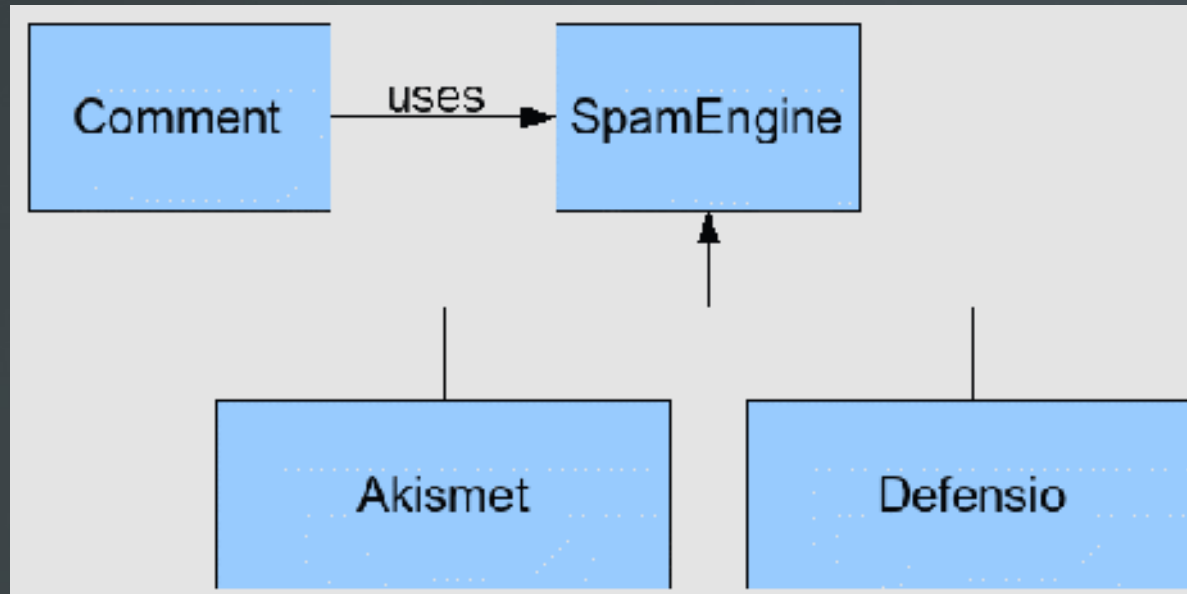
  class DefensioEngine < Base
  end

  class AkismetEngine < Base
  end
end
```



Refactoring to Strategy Pattern

Structural Changes



Creational Patterns

Abstract Factory Pattern



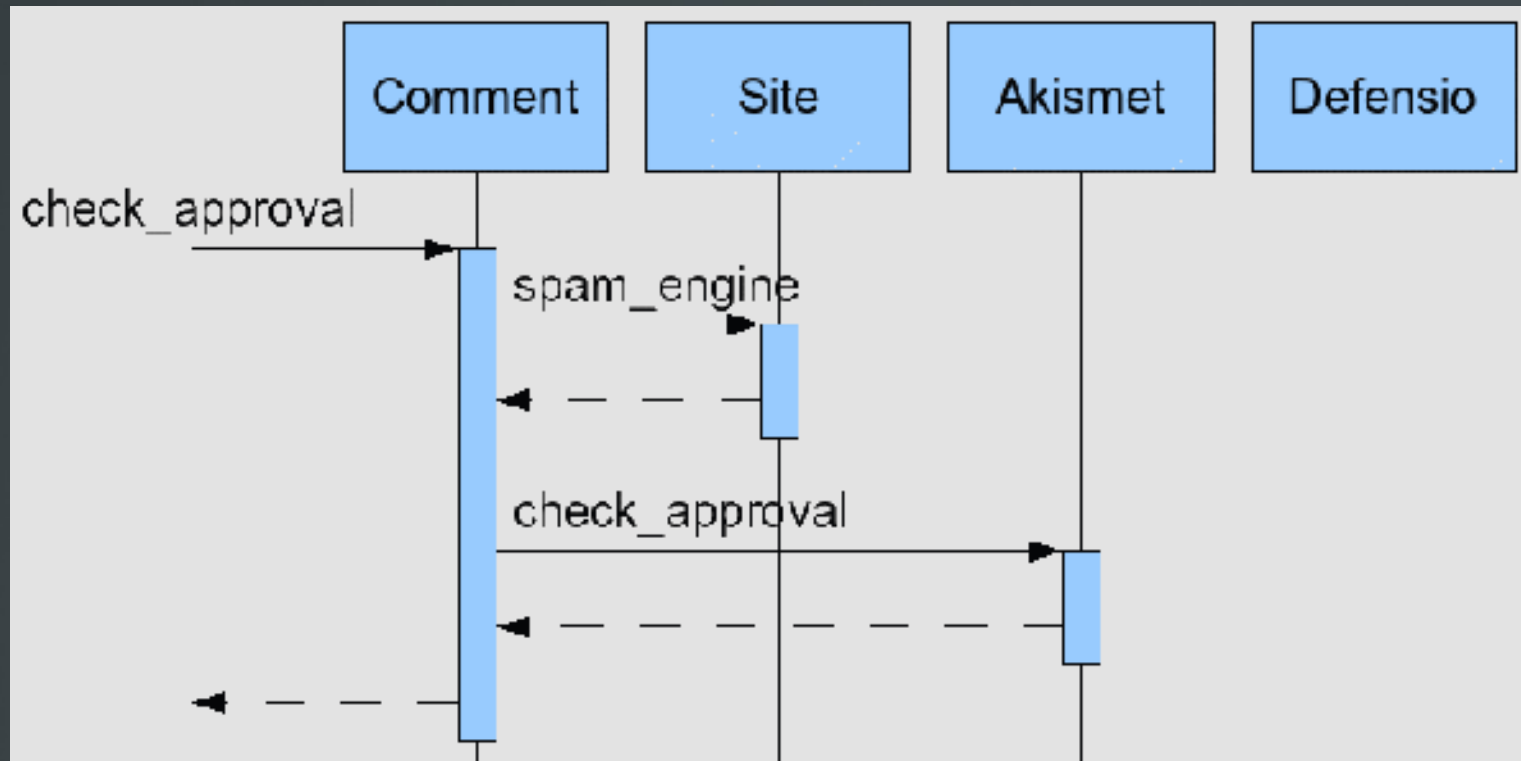
Creational Patterns

```
class Site
  def spam_engine
    class_name = read_attribute(:spam_detection_engine)
    if class_name.blank?
      Mephisto::SpamDetectionEngines::
        NullEngine.new(self)
    else
      class_name.constantize.new(self)
    end
  end
end
```



Refactoring to Strategy Pattern

Interaction Diagram



Adapter Pattern

Convert the interface of a class into another interface clients expect.



Adapter Pattern

```
def check_approval  
  # implementation  
end
```



Adapter Pattern

```
def ham?  
  # implementation  
end
```

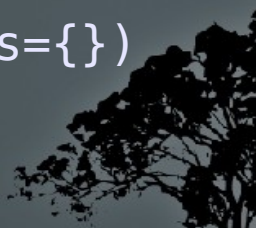


Refactoring to Adapter Pattern

Before

```
class CommentsController
  def create
    if @comments.check_approval(site, request) then
      # thanks
    else
      # thanks, but moderated
    end
  end
end

class Comment
  def check_approval(site, request, options={})
    # implementation
  end
end
```

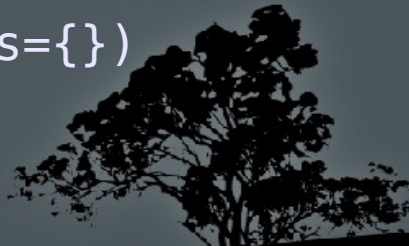
A silhouette of a tree on a hill, located in the bottom right corner of the slide.

Refactoring to Adapter Pattern

Rename Method

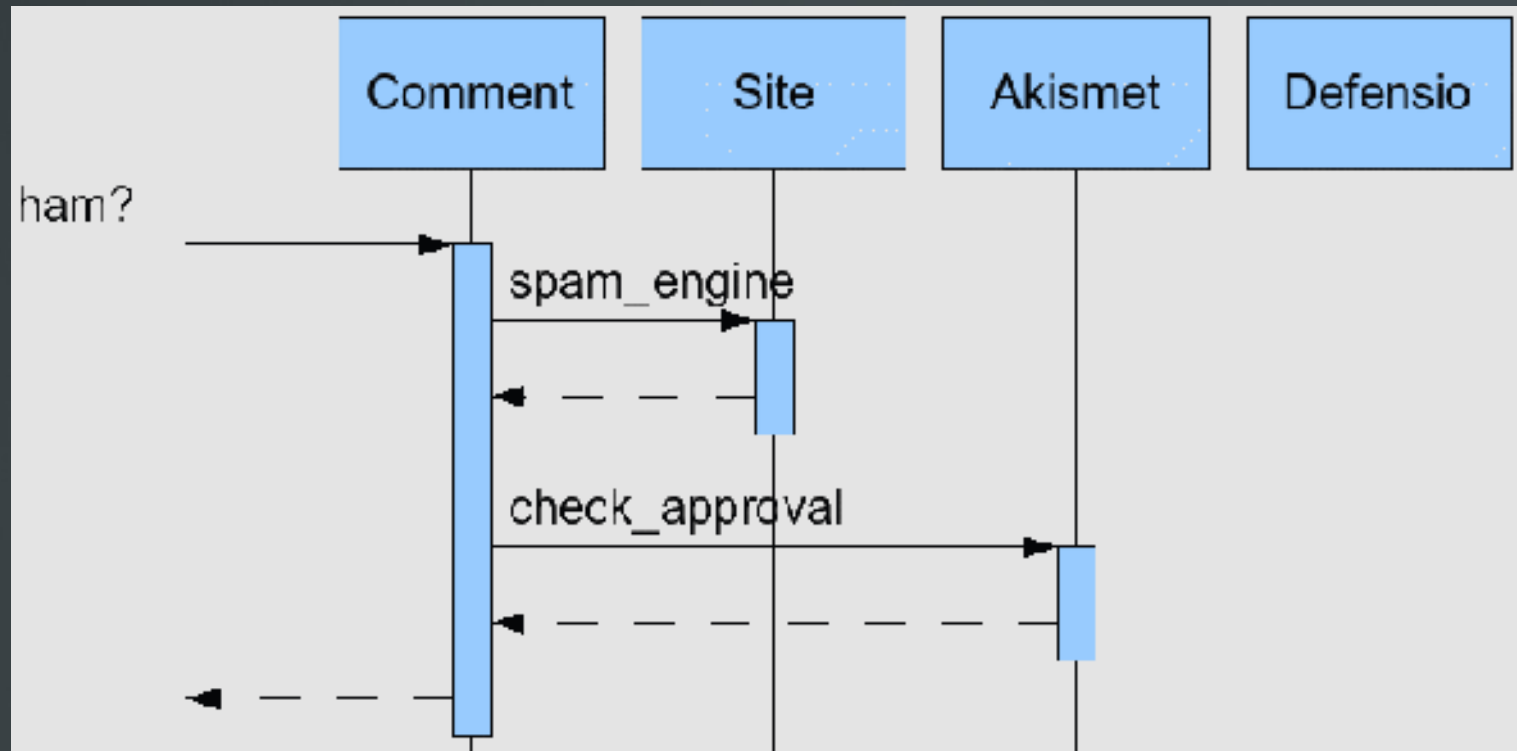
```
class CommentsController
  def create
    if @comments.ham?(site, request) then
      # thanks
    else
      # thanks, but moderated
    end
  end
end

class Comment
  def check_approval(site, request, options={})
    # implementation
  end
end
```

A silhouette of a tree on a hill, located in the bottom right corner of the slide.

Refactoring to Adapter Pattern

Interaction Diagram



Who am I ?



- François Beausoleil



Who am I ?



- François Beausoleil
- A Single Programmer's Blog



Who am I ?



- François Beausoleil
- A Single Programmer's Blog
- XLsuite



Creative Commons Attributions



dave_7

<http://flickr.com/photos/daveseven/204585649/>



DanieVDM

<http://flickr.com/photos/dvdmerwe/251163984/>

