

“the times, they are a changin’.”
-Bob Dylan

winston tsang

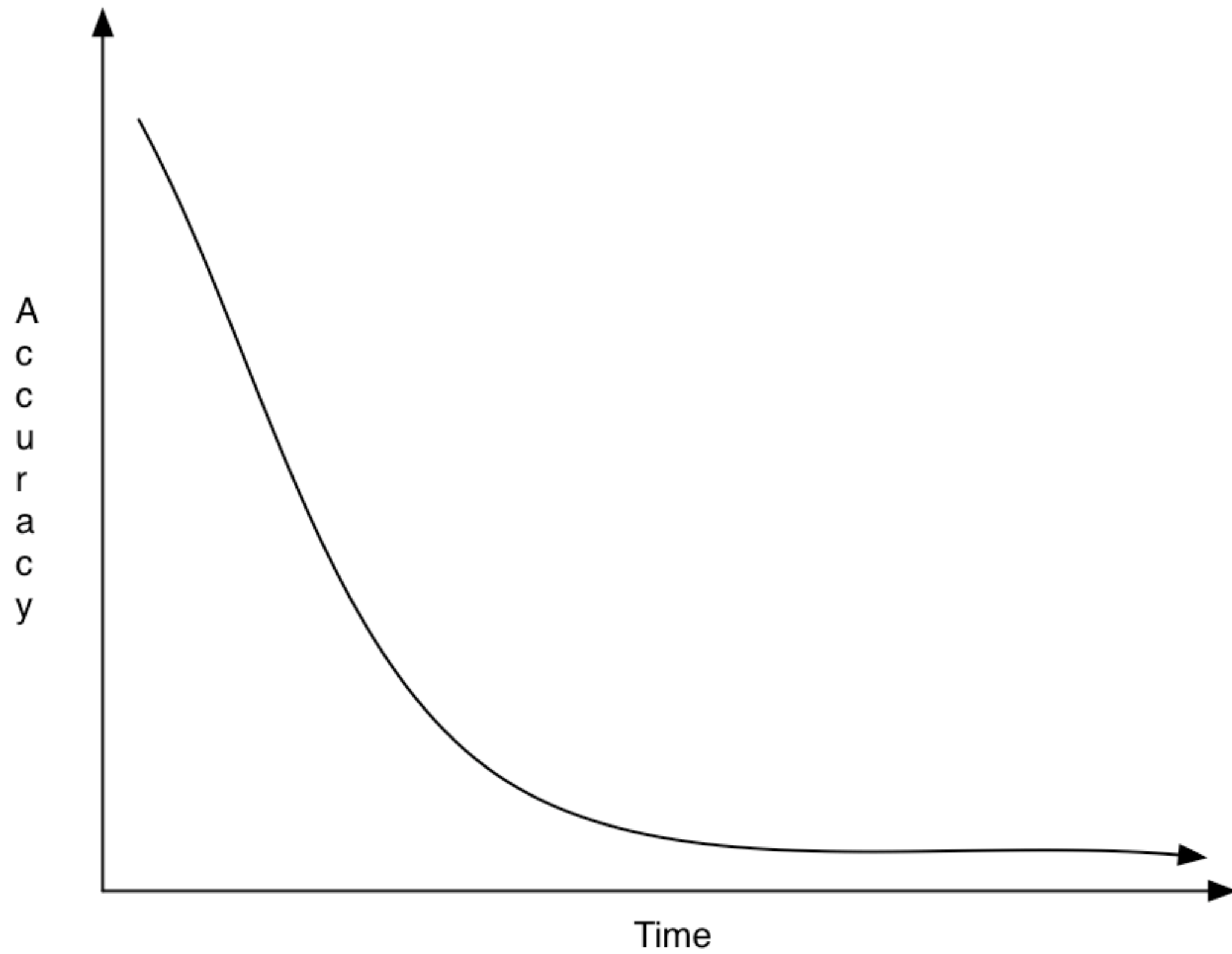
twitter: winstont

email: rubyconsumer@rubyconsumer.com

web: <http://www.rubyconsumer.com/blog>

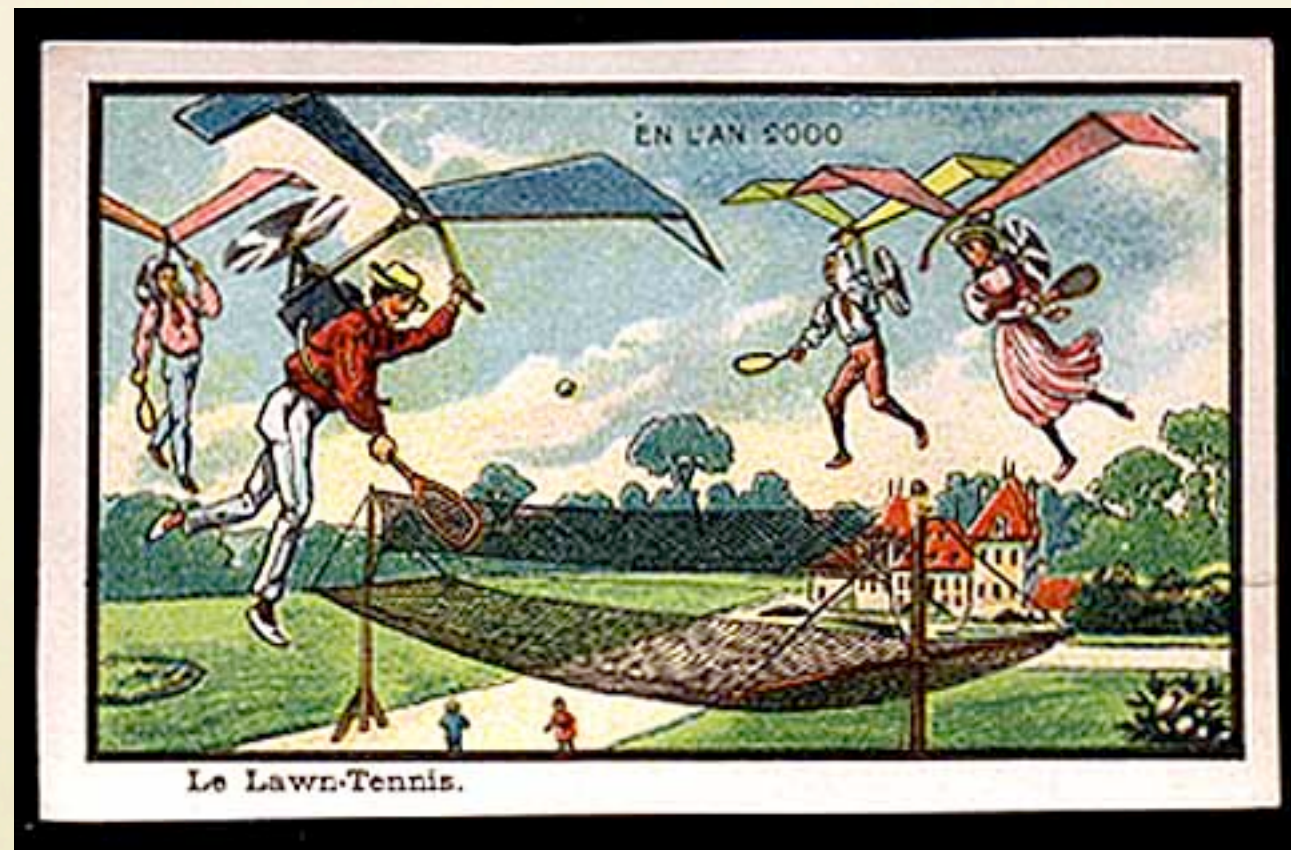
PREDICTING THE FUTURE

sometimes... maybe... if you're lucky...





Une Partie de Croquet.





Steve Wozniak
with Gina Smith

iWoz

How I invented the personal computer,
co-founded Apple, and had fun doing it

COMPUTER GEEK TO CULT ICON



"I knew it was the future. There wasn't a doubt in my mind. It was like a one-way door to the future--and once you went through it, you could never turn back... It's so rare to be able to see the future like that. I can't promise it'll happen to you. But when you see it, you will know it. If this ever happens to you leap at the chance to get involved." -Steve Wozniak

“Look, I can’t predict the future.
But I am sensitive to some areas
from which change is emerging.”

-Laney in Idoru by William Gibson

A Look Into The Past...

Packages

Red Hat's RPM

Debian's Apt

FreeBSD's Ports

Perl's ppm

Java's Jars

Ruby's Gems




```
gem install <gem>
```


gem install rake


```
gem install rspec
```



```
gem install rcov
```



```
gem install gruff
```



```
gem install ferret
```



```
gem install ZenTest
```



```
gem install rails
```


So easy it's like...

Cheating!


```
gem install cheat
```



```
gem list --local
```


`gem list --remote`


```
glrb:/Users/glrb/src/rails> gem list
```

```
*** LOCAL GEMS ***
```

```
actionmailer (1.2.5, 1.0.1)
```

```
Service layer for easy email delivery and testing.
```

```
actionpack (1.12.5, 1.9.1)
```

```
Web-flow and rendering framework putting the VC in MVC.
```

```
actionwebservice (1.1.6, 0.8.1)
```

```
Web service support for Action Pack.
```

```
activerecord (1.14.4, 1.11.1)
```

```
Implements the ActiveRecord pattern for ORM.
```

```
activesupport (1.3.1, 1.1.1)
```

```
Support and utility classes used by the Rails framework.
```

```
BlueCloth (1.0.0)
```

```
BlueCloth is a Ruby implementation of Markdown, a text-to-HTML  
conversion tool for web writers. Markdown allows you to write using  
an easy-to-read, easy-to-write plain text format, then convert it to  
structurally valid XHTML (or HTML).
```

```
builder (2.0.0)
```

```
Builders for Markup.
```

```
capistrano (1.2.0)
```

```
Capistrano is a framework and utility for executing commands in  
parallel on multiple remote machines, via SSH. The primary goal is  
to simplify and automate the deployment of web applications.
```

```
cheat (1.1.0, 1.0.2)
```

```
Cheat is a simple command line utility reference program.
```

```
classifier (1.3.0)
```

```
A general classifier module to allow Bayesian and other types of  
classifications.
```

```
color-tools (1.3.0)
```

```
color-tools provides colour space definition and manipulation as  
well as commonly named RGB colours.
```


gem_server


```
glrb:/Users/glrb> gem_server  
[2006-10-25 17:31:06] INFO WEBrick 1.3.1  
[2006-10-25 17:31:06] INFO ruby 1.8.5 (2006-08-25) [i686-darwin8.8.1]  
[2006-10-25 17:31:06] INFO WEBrick::HTTPServer#start: pid=6161 port=8808
```


RubyGems Documentation Index

Summary

There are 70 gems installed:

[actionmailer](#), [actionmailer](#), [actionpack](#), [actionpack](#), [actionwebservice](#), [actionwebservice](#), [activerecord](#), [activerecord](#), [activesupport](#), [activesupport](#), [BlueCloth](#), [builder](#), [capistrano](#), [cheat](#), [cheat](#), [classifier](#), [color-tools](#), [daemons](#), [dr_nic_magic_models](#), [extensions](#), [fastercsv](#), [fcgi](#), [ferret](#), [ferret](#), [flexmock](#), [flexmock](#), [gem_plugin](#), [gruff](#), [hoe](#), [hoe](#), [hoe](#), [instiki](#), [irb-history](#), [libxml-ruby](#), [login_generator](#), [madeleine](#), [madeleine](#), [mongrel](#), [mongrel_cluster](#), [needle](#), [net-sftp](#), [net-ssh](#), [payment](#), [pdf-writer](#), [radiant](#), [radius](#), [rails](#), [rails](#), [rails-app-installer](#), [rake](#), [rcov](#), [RedCloth](#), [RedCloth](#), [rspec](#), [Rubilicious](#), [rubyforge](#), [rubyzip](#), [rubyzip](#), [shipping](#), [sources](#), [sqlite3-ruby](#), [stemmer](#), [transaction-simple](#), [typo](#), [tinfo](#), [tinfo](#), [unicode](#), [wirble](#), [ZenTest](#), [ZenTest](#).

Gems

actionmailer 1.0.1 [[rdoc](#)] [[www](#)] - depends on [actionpack](#).

Service layer for easy email delivery and testing.

actionmailer 1.2.5 [[rdoc](#)] [[www](#)] - depends on [actionpack](#).

Service layer for easy email delivery and testing.

actionpack 1.12.5 [[rdoc](#)] [[www](#)] - depends on [activesupport](#).

Web-flow and rendering framework putting the VC in MVC.

actionpack 1.9.1 [[rdoc](#)] [[www](#)] - depends on [activesupport](#).

Web-flow and rendering framework putting the VC in MVC.

actionwebservice 0.8.1 [[rdoc](#)] [[www](#)] - depends on [actionpack](#), [activerecord](#), [activesupport](#).

Web service support for Action Pack.

Files	Classes	Methods
README lib/gem_plugin.rb	GemPlugin GemPlugin::Base GemPlugin::Manager GemPlugin::PluginNotLoaded	Plugin (GemPlugin) category= (GemPlugin::Base) config (GemPlugin::Manager) create (GemPlugin::Manager) inherited (GemPlugin::Base) load (GemPlugin::Manager) loaded? (GemPlugin::Manager) new (GemPlugin::Manager)

README

Path: [README](#)

Last Update: Sun Oct 08 15:23:34 -0400 2006

GemPlugin: Gem Based Plugin System

[GemPlugin](#) is a system that lets your users install gems and lets you load them as additional features to use in your software. It originated from the Mongrel (mongrel.rubyforge.org) project but proved useful enough to break out into a separate project.

[GemPlugin](#) works by listing the gems installed, and doing a `require_gem` on any that have the right dependencies. For example, if a gem depends on "gem_plugin" and "mongrel" then it'll load as a Mongrel plugin. This makes it so that users of the plugins only need to gem install (and maybe config a bit), and plugin authors only need to make gems.

Implementers

To use [GemPlugin](#) in your system you only have to require 'gem_plugin' and then use the `GemPlugin::Manager.create`, `GemPlugin::Manager.load`, and `GemPlugin::Manager.available` methods to work with them.

- `GemPlugin::Manager.load` — Takes a "depend include/exclude map" and loads plugins based on it.
- `GemPlugin::Manager.create` — Takes a URI style name and some options then creates one for you.
- `GemPlugin::Manager.available` — Lets you inspect and mess with the internal plugin registry.

Loading Plugins

As an example from Mongrel it's necessary to load plugins that depend on rails after the Rails system is configured, but load other plugins right when Mongrel is ready. To do this we very first do:


```
] gem_server  
`gem_server` has been deprecated. Use `gem server` instead.
```


Development

script/console


```
] script/console  
Loading development environment.  
>>
```



```
] script/console  
Loading development environment.  
>> @george = Person.find_by_name('George')  
>> @bob = Person.find_by_name('Bob')  
>> @bob.friends << @george
```


Testing

Testing Java....

Configuration

Configuration

Server

Configuration

Server

XML

Application Server

Configuration

Server

XML

Application Server

Configuration

Server

Web Server

XML

Application Server

Configuration

Server

Database

Web Server

XML

Application Server

Ant

Configuration

Server

Database

Web Server

XML

Application Server

Tomcat

Ant

Configuration

Server

Database

Web Server

XML

Application Server

Tomcat

Ant

Cruisecontrol

Configuration

Server

Database

Web Server

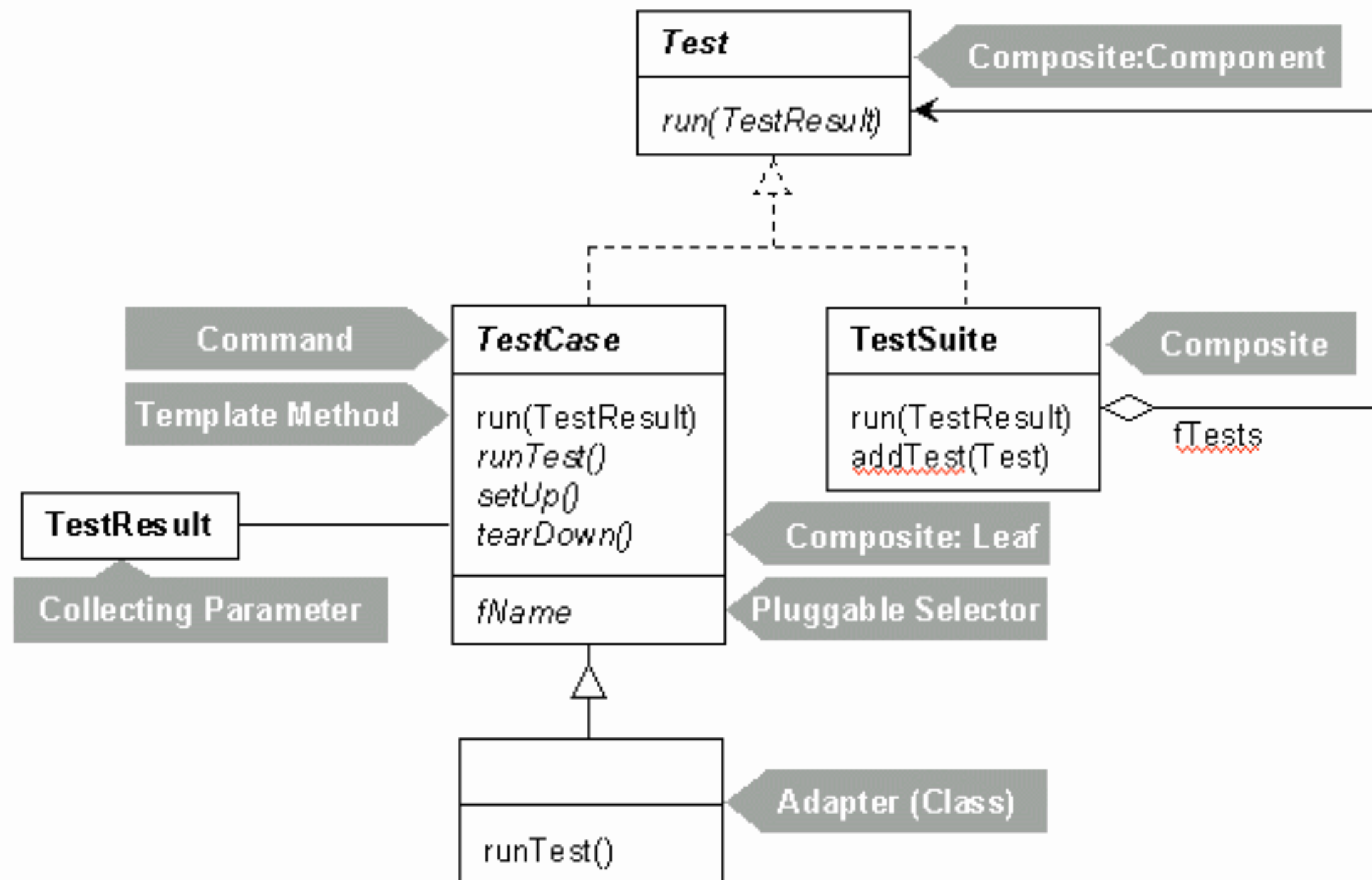
XML

Later...

Much Later...



JUnit



huh?

Testing Rails....



Fixtures

Test Case

Test

Fixtures

Fixtures

Albert:

id: 1

title: Albert's Home Theater Meet 2006

datetime: 2006-05-01 14:00:00

location: The Abyss

multiple_sessions: Yes

Bob:

id: 2

title: Bob's Home Theater 2006

datetime: 2006-06-01 14:00:00

location: Bottom Of The Sea

multiple_sessions: Yes

Test Cases

Unit Test Case

```
class EventTest < Test::Unit::TestCase
  fixtures :events

  def test_retrieve_event
    e = Event.find(:first)

    assert_equal events(:Albert).id, 1
  end
end
```


Fuctional Test Case

```
class EventsControllerTest < Test::Unit::TestCase
  fixtures :events

  def test_index
    get :index
    assert_response :success
    assert_template 'list'
  end
end
```


Test

rake test

```
glrb:/Users/glrb/src/rails/HEMI> rake test
```

```
/opt/local/bin/ruby -Ilib:test "/opt/local/lib/ruby/gems/1.8/gems/rake-0.7.1/lib/rake/rake_test_loader.rb" "test/unit/attendee_test.rb" "test/unit/event_test.rb" "test/unit/user_test.rb"
```

```
Loaded suite /opt/local/lib/ruby/gems/1.8/gems/rake-0.7.1/lib/rake/rake_test_loader
```

```
Started
```

```
.....
```

```
Finished in 0.598629 seconds.
```

```
17 tests, 30 assertions, 0 failures, 0 errors
```

```
/opt/local/bin/ruby -Ilib:test "/opt/local/lib/ruby/gems/1.8/gems/rake-0.7.1/lib/rake/rake_test_loader.rb" "test/functional/account_controller_test.rb" "test/functional/attendees_controller_test.rb" "test/functional/events_controller_test.rb"
```

```
Loaded suite /opt/local/lib/ruby/gems/1.8/gems/rake-0.7.1/lib/rake/rake_test_loader
```

```
Started
```

```
.....
```

```
Finished in 0.330386 seconds.
```

```
32 tests, 84 assertions, 0 failures, 0 errors
```


Review...

Fixtures

Test Case

Test

Easy!

Simple!

Easier?

Simpler?

YES!

autotest


```
gem install ZenTest
```


autotest


```
glrb:/Users/glrb/src/rails/HEMI> autotest
/opt/local/bin/ruby -I.:lib:test -rtest/unit -e "%w[test/functional/events_controller_test.rb test/functional/
attendees_controller_test.rb test/unit/attendee_test.rb test/unit/user_test.rb test/unit/event_test.rb test/functional/
account_controller_test.rb].each { |f| load f }" | unit_diff -u
Loaded suite -e
Started
```

```
.....
Finished in 0.44881 seconds.
```

```
=====
49 tests, 114 assertions, 0 failures, 0 errors
```

```
# waiting...Thu Oct 26 11:03:04 -0400 2006
```


Deployment

Easy?

Simple?

YES!

capistrano

cap --apply-to <project>


```
glrb:/Users/glrb/src/rails/demo> cap --apply-to .  
exists  config  
create  config/deploy.rb  
exists  lib/tasks  
create  lib/tasks/capistrano.rake
```


capistrano.rake

```
desc "A macro-task that updates the code, fixes the symlink, and restarts the application servers."
task(:deploy) { cap :deploy }

desc "Similar to deploy, but it runs the migrate task on the new release before updating the symlink."
task(:deploy_with_migrations) { cap :deploy_with_migrations }

desc "Displays the diff between HEAD and what was last deployed."
task(:diff_from_last_deploy) { cap :diff_from_last_deploy }

desc "Disable the web server by writing a \"maintenance.html\" file to the web servers."
task(:disable_web) { cap :disable_web }

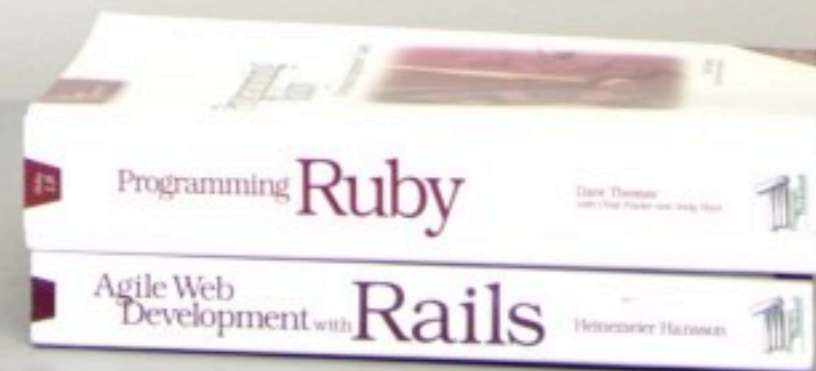
desc "Re-enable the web server by deleting any \"maintenance.html\" file."
task(:enable_web) { cap :enable_web }
```


deploy.rb

```
set :application, "application"
set :repository, "http://svn.yourhost.com/#{application}/trunk"

# =====
# ROLES
# =====
# You can define any number of roles, each of which contains any number of
# machines. Roles might include such things as :web, or :app, or :db, defining
# what the purpose of each machine is. You can also specify options that can
# be used to single out a specific subset of boxes in a particular role, like
# :primary => true.

role :web, "www01.example.com", "www02.example.com"
role :app, "app01.example.com", "app02.example.com", "app03.example.com"
role :db,  "db01.example.com", :primary => true
role :db,  "db02.example.com", "db03.example.com"
```



<http://rubyconsumer.com>

“The future is here. It’s just
not evenly distributed yet.”
-William Gibson