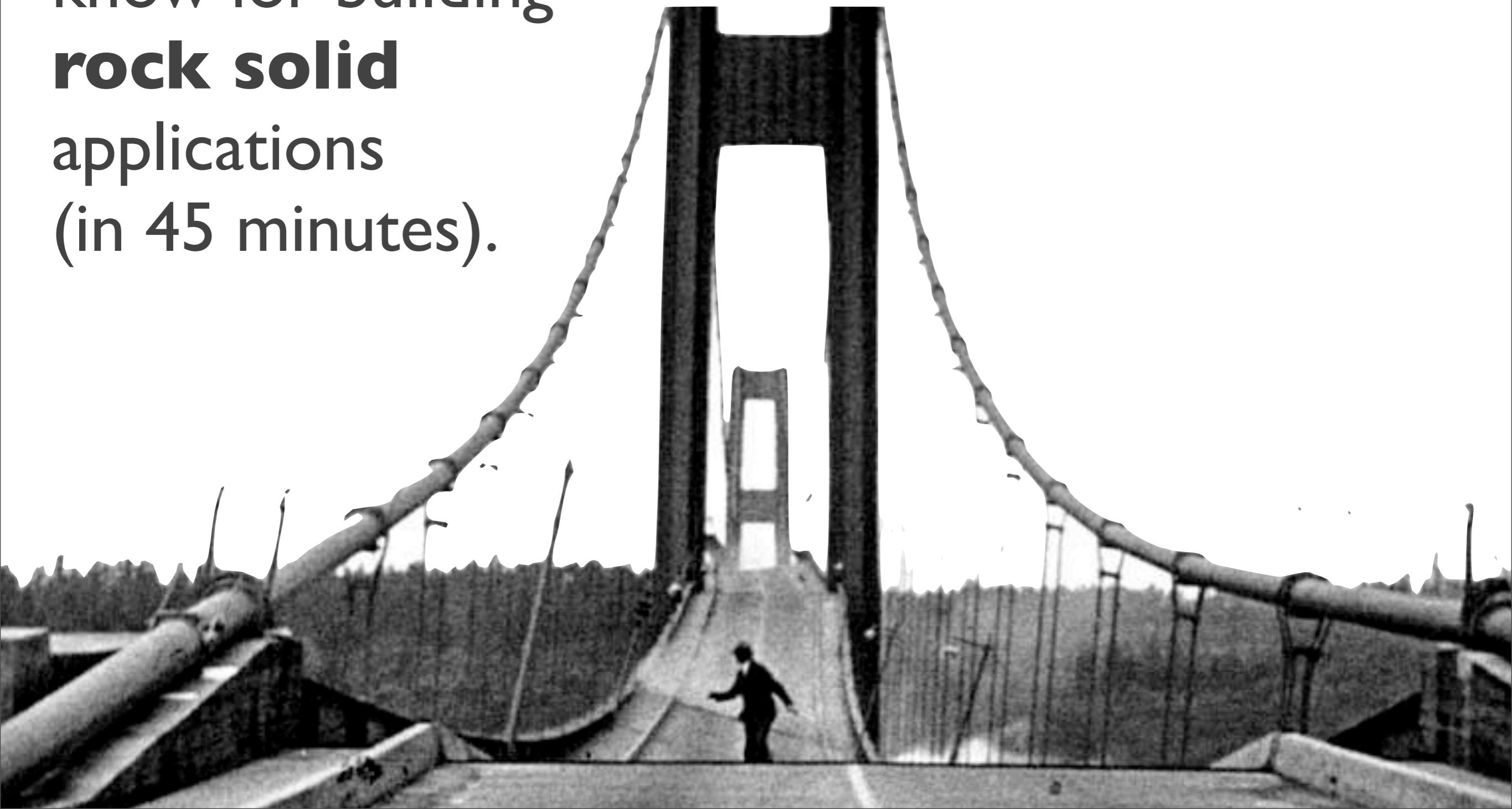


# Coding for Failure

All you need to  
know for building  
**rock solid**  
applications  
(in 45 minutes).



**why?**

**When?**

**How?**

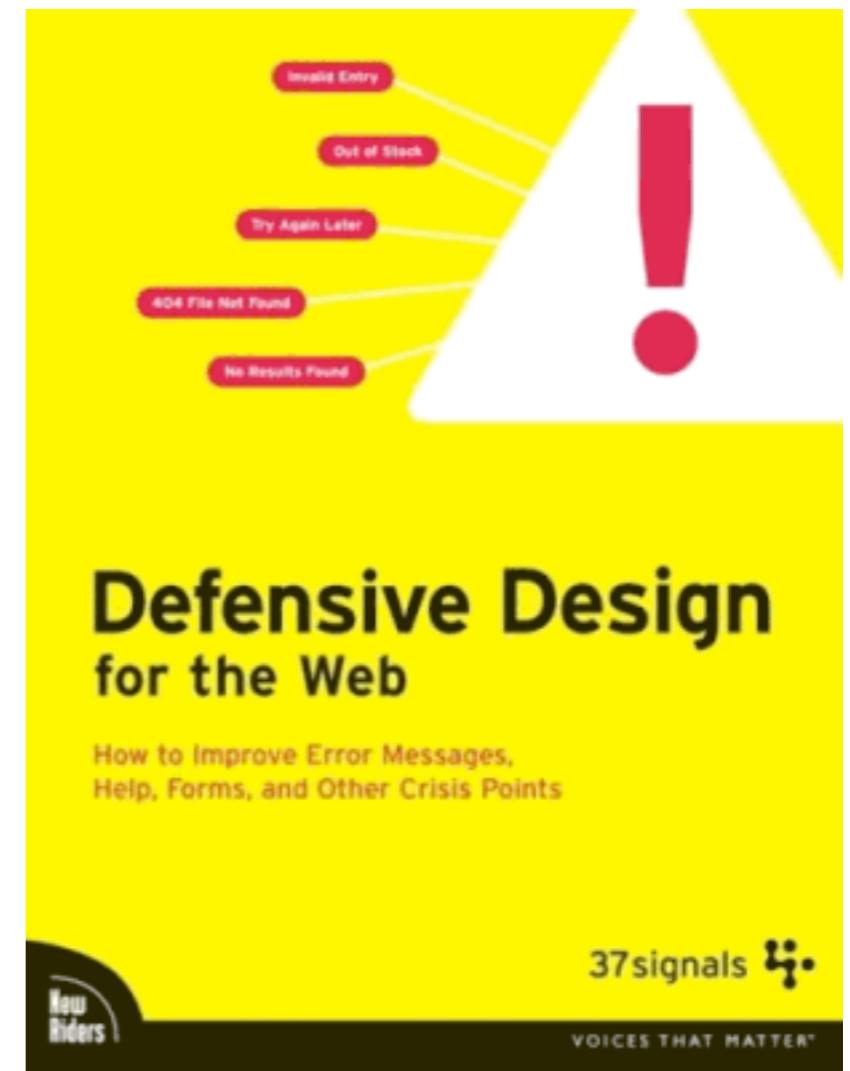
# For the People



# Defensive Design for the Web

How to improve error messages,  
help, forms, and other crisis points

37signals



Error Pages

Failure  
Emails

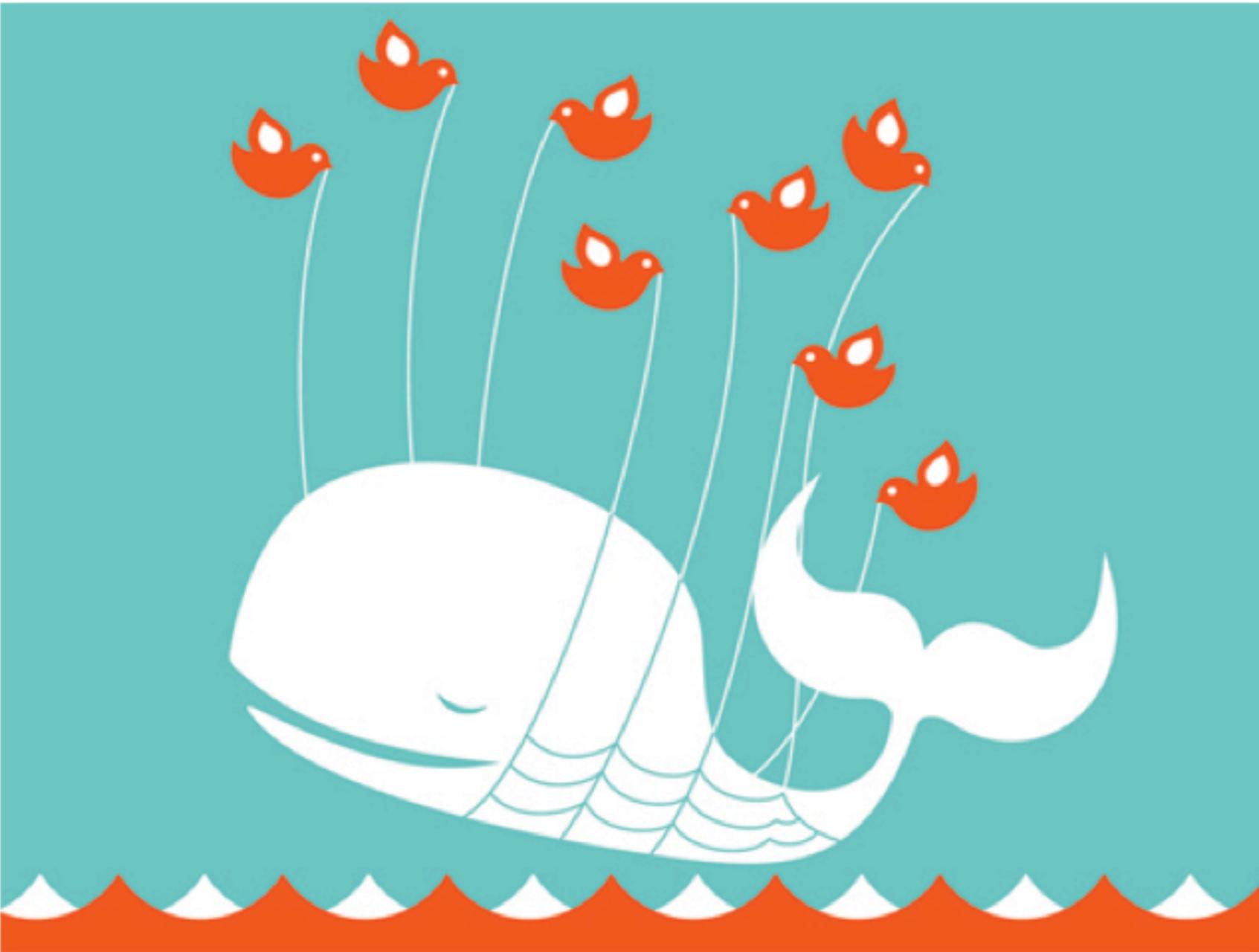
Flash  
messages

Validations

XML  
responses

## Twitter is over capacity.

Too many tweets! Please wait a moment and try again.





For you

Help yourself sleep at  
night.

# NetworkAppliance



Why?

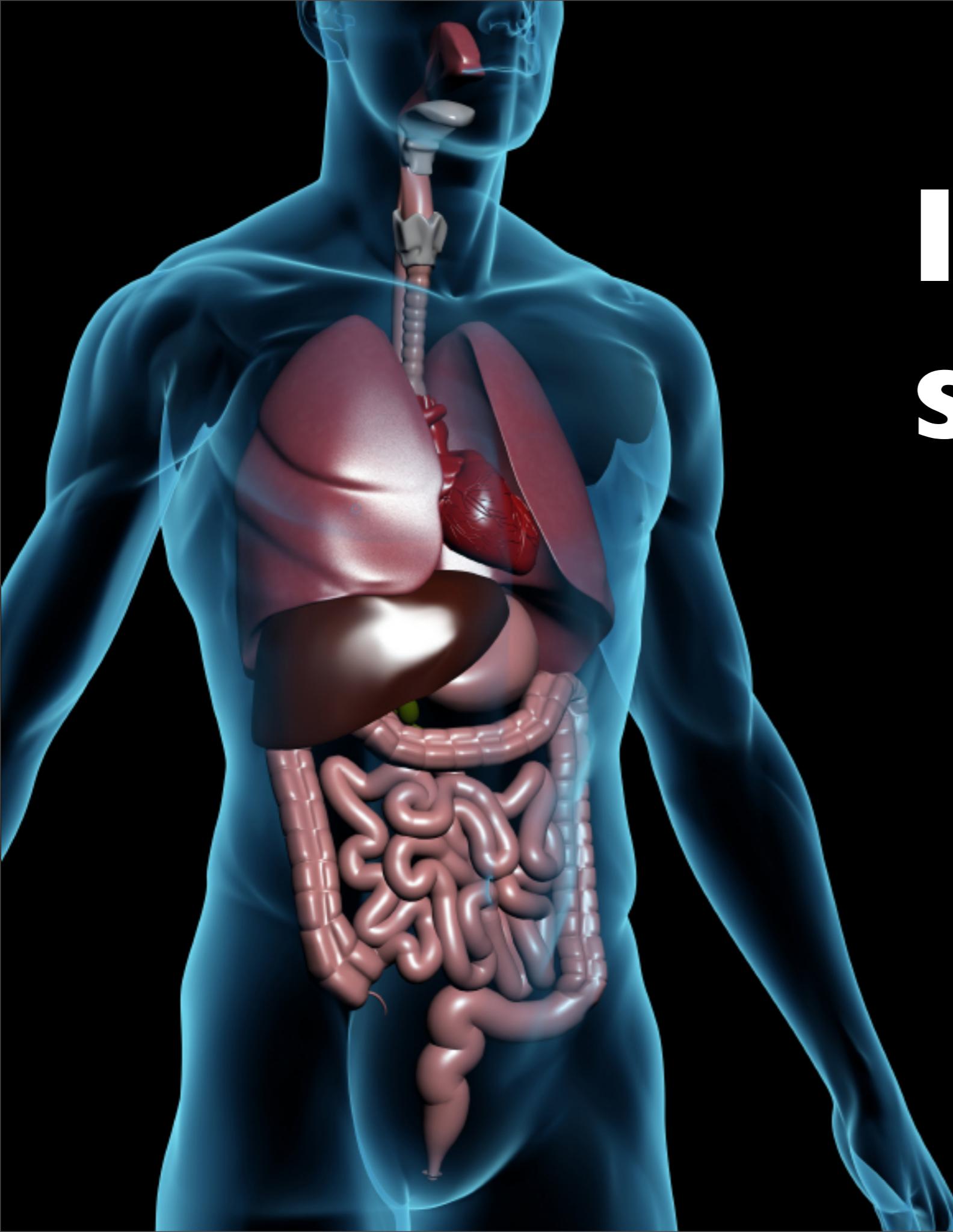
**When?**

How?

# **Think about failure points**



# External services

A detailed 3D anatomical model of the human torso and head. The skin is rendered in a translucent blue, allowing a clear view of the internal organs. The lungs are large and reddish-pink, situated in the chest. Below them is the heart, also reddish-pink. To the right of the heart is the liver, a large reddish-brown organ. Further down and to the left is the stomach, a large, dark reddish-brown sac. Below the stomach is the small intestine, which is coiled and pinkish-red. To the right of the small intestine is the large intestine, which is also coiled and has a slightly darker color. The head is shown at the top, with the brain visible in a cross-section. The neck contains the trachea and esophagus.

# Internal services

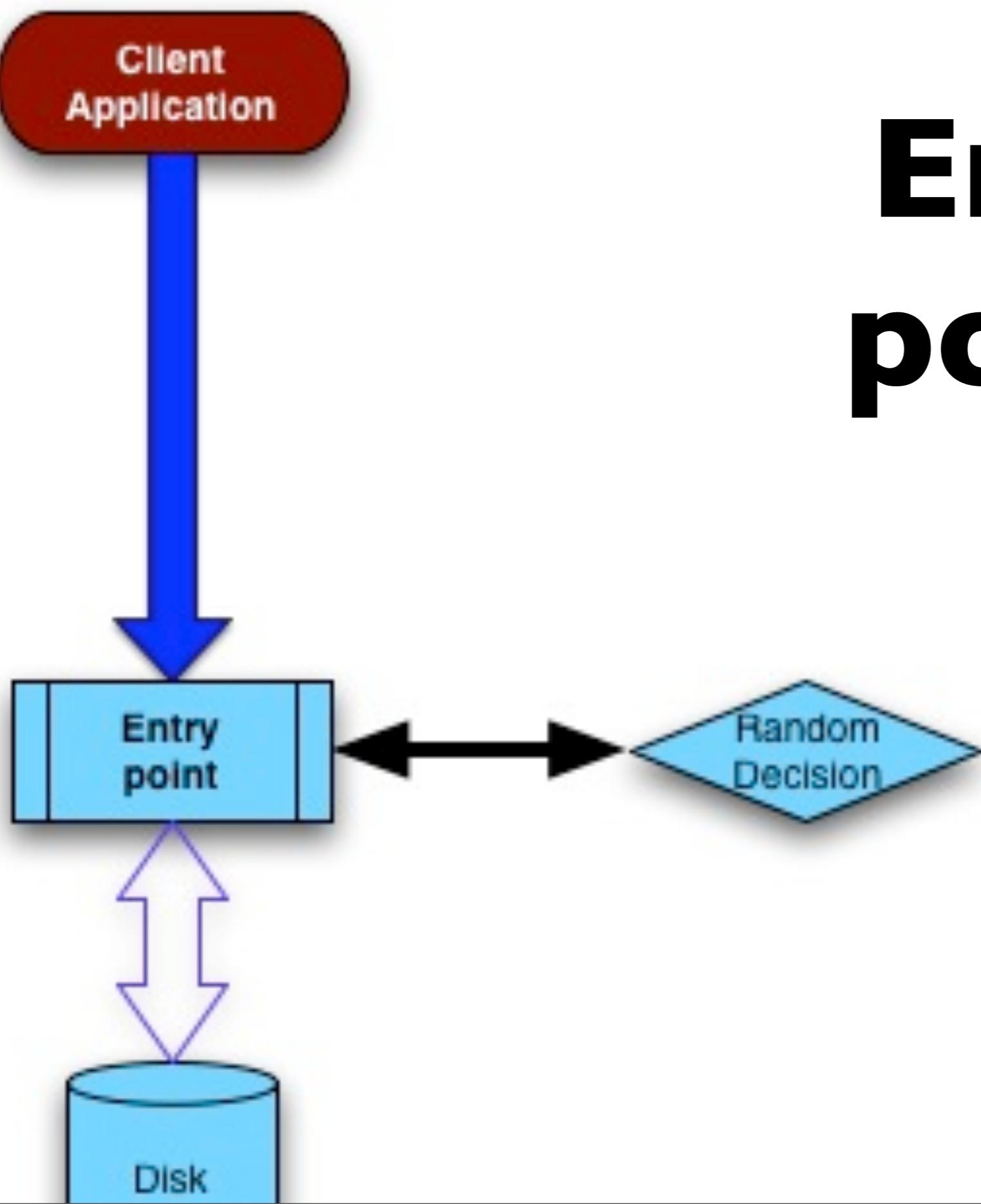


# Complex Operations



Performance  
bottle necks

# Entry points



When to  
**WORRY**  
and when to  
*Relax*

A photograph of a space shuttle launching from a launch pad. The shuttle is positioned vertically in the center of the frame, moving upwards. A massive, billowing plume of white smoke and fire erupts from its base, filling the lower half of the image. The background consists of a dark, cloudy sky. In the foreground, there is a dark, flat landscape, likely the ground or a nearby body of water.

You Aren't NASA.

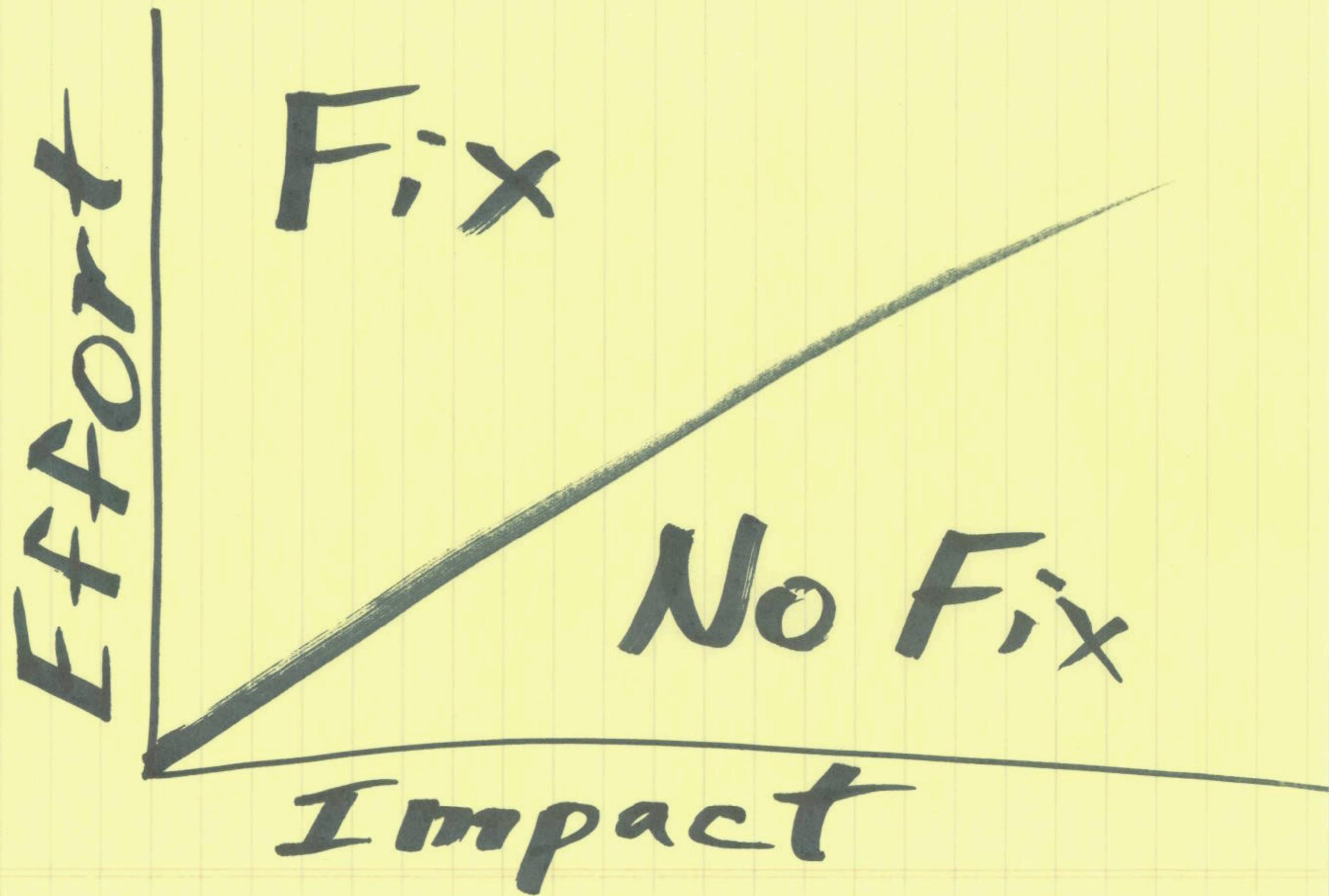
You're a website.



**EXAMPLE:**

**#destroy actions**

```
def destroy
  @project = Project.find(params[:id])
  @project.destroy
  flash[:notice] = "Project #{@project.id}
                    has been removed."
  redirect_to projects_path
end
```



Why?

When?

**How?**

# Guidelines

1. 2. 3

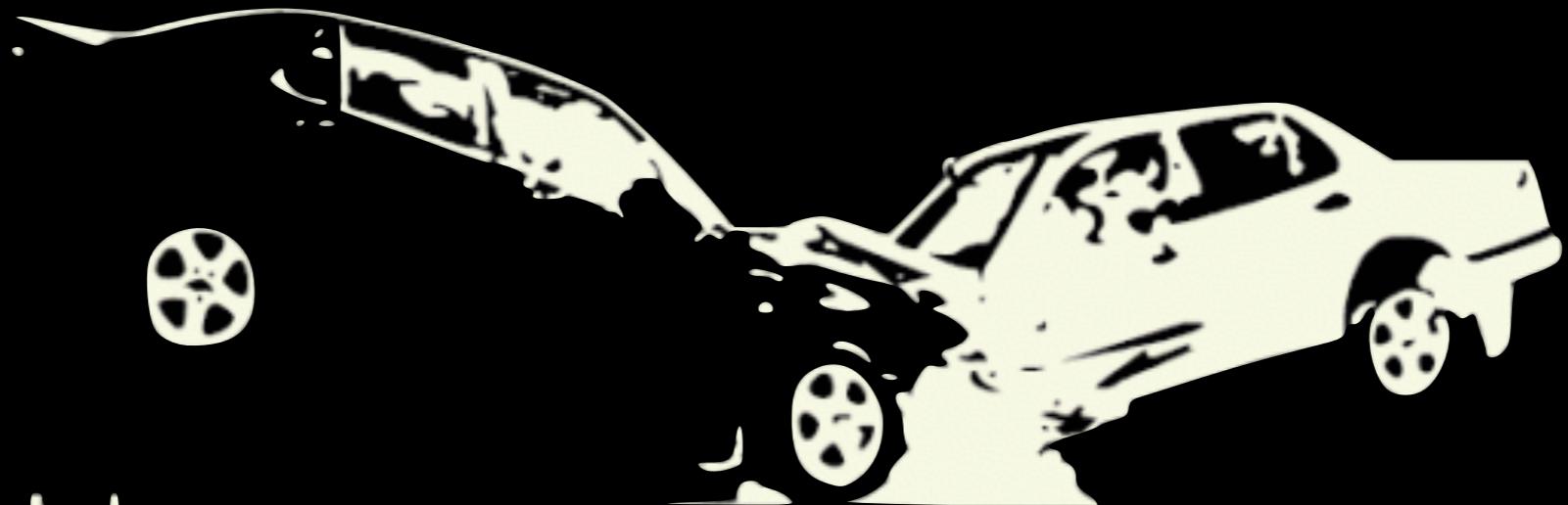
# ONE

# **Fail Fast**



"Think about what kinds of defects are possible and how they occur. Place your assertions so that the software fails earlier - close to the original problem - making the problem easy to find."

Jim Shore "*Fail Fast*"



ctrl + Z

Catch bugs as soon as  
they happen

# Sanity checks

```
def api_down(msg)
  log("API is down: #{msg.inspect}")
  HoptoadNotifier.notify(
    :error_message => "API down: #{msg.inspect}")
  false
end
```

```
def service_available?
  log("Pinging API to see if it's alive...")
  r = submit_xml(status_url, ping_xml)
  return api_down(r["error"]) if r["error"]
  return api_down(r) unless r["Response"]
  return api_down(r) unless r["Response"]["Error"]
  return true
end
```

```
svn commit -m path/to/file
```

TWO

Fail

Loud



```
users.each do |user|
  user.activated = true
  user.save
end
```

```
users.each do |user|
  user.activated = true
  user.save!
end
```

```
message = ""
users.each do |user|
  user.activated = true
  unless user.save
    message += "#{user.name}: " +
      "#{user.errors.full_messages.join('; ')}"
  end
end

unless message.blank?
  flash[:warning] = "Unable to update some " +
    "of the users: #{message}"
end
```

logs

Rails.logger.error("...")

email  
(code?)

Pager  
(no code)

exception  
notifier  
(spam)

# Hoptoad

[ERRORS](#)[PROJECTS](#)[USERS](#)[MY PROFILE](#)[LOGOUT](#)

## widgetfinger

[Edit this project](#)[Show resolved errors](#)

MESSAGE	LOCATION	LATEST	COUNT
<a href="#">URI::InvalidURIError: bad URI(is not URI?): http://www.go...</a>	hits#create	1 day ago	6762
<a href="#">ActionController::UnknownAction: No action responded to show</a>	reminder#show	3 days ago	1
<a href="#">ActionController::UnknownAction: No action responded to i...</a>	sessions#index	3 days ago	1
<a href="#">ActionController::UnknownAction: No action responded to i...</a>	accounts#index	3 days ago	4
<a href="#">ActionView::TemplateError: divided by 0</a>	stats/visits#show	4 days ago	13
<a href="#">Liquid::SyntaxError: Unknown tag 'endsection'</a>	pages#edit	12 days ago	6
<a href="#">HoptoadTestingException: Testing hoptoad via "rake hoptoa..."</a>	hoptoad_verification#verify	about 1 month ago	1
<a href="#">Liquid::SyntaxError: 'end' is not a valid delimiter for ...</a>	pages#edit	about 1 month ago	2
<a href="#">ActionView::TemplateError: divided by 0</a>	stats/visits/month#show	about 1 month ago	1
<a href="#">ActionController::UnknownAction: No action responded to s...</a>	account#subscription	about 1 month ago	1
<a href="#">ActionController::UnknownAction: No action responded to s...</a>	account#subscription	about 1 month ago	1
<a href="#">ActionController::UnknownAction: No action responded to i...</a>	accounts#index	about 1 month ago	2

# Hoptoad

```
rescue => e
    notify_hoptoad(e)
    flash[:failure] = "Failure rerouting
                        encryptions. We've
                        been notified about
                        the problem."
end
```

# Hoptoad

```
begin
  my_unpredictable_method(params)
rescue => e
  HoptoadNotifier.notify(
    :error_class    => "Special Error",
    :error_message => "Special Error: #{e.message}"
  )
end
```

# THREE

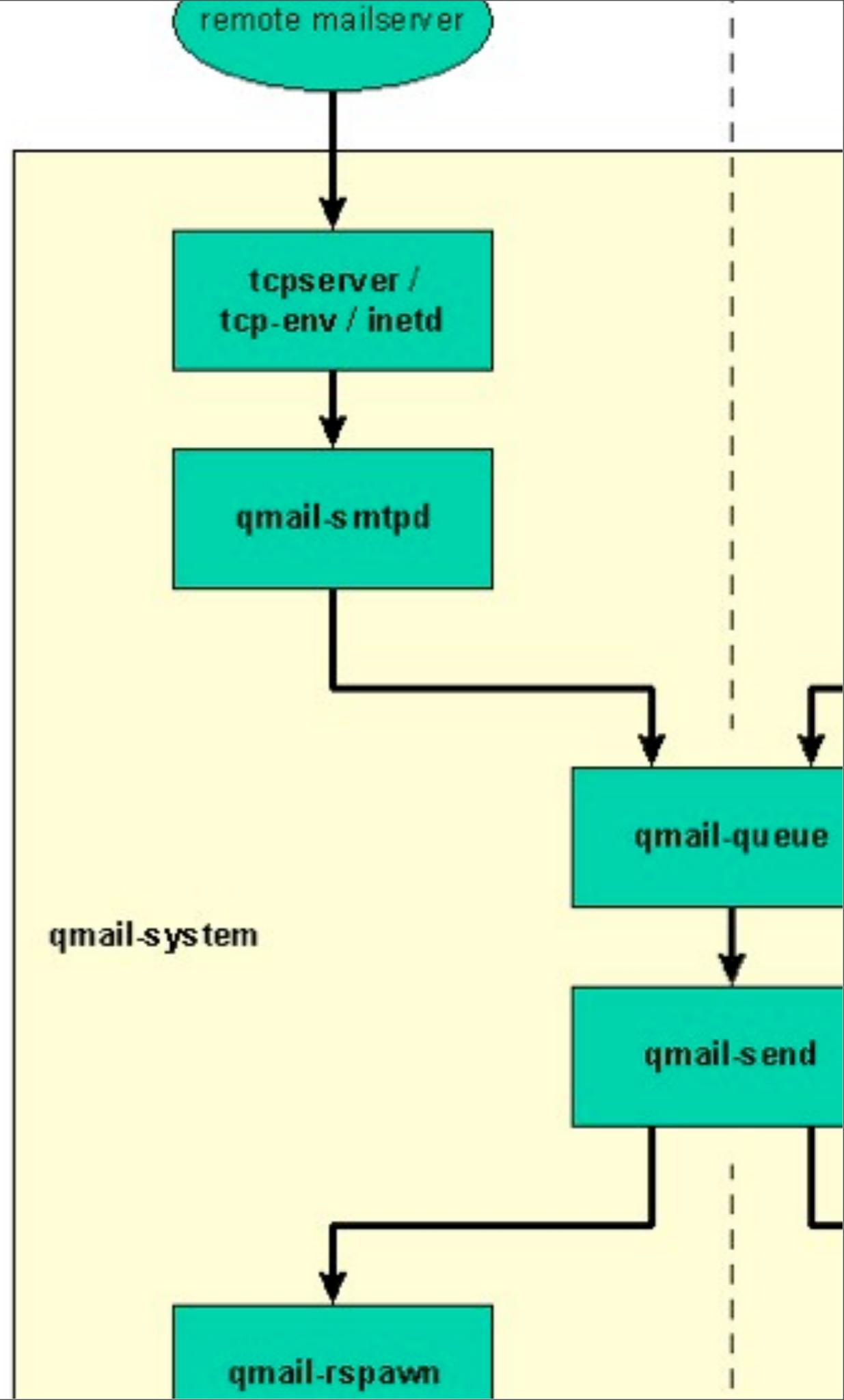
# zero Trust





Components  
should *never*  
trust inputs

# The Security Architecture of qmail



All right...  
Let's see some code.

# External Services

# Net::HTTP exceptions

```
begin
  response = Net::HTTP.post_form("...")
rescue Timeout::Error,
  Errno::EINVAL,
  Errno::ECONNRESET,
  EOFError,
  Net::HTTPBadResponse,
  Net::HTTPHeaderSyntaxError,
  Net::ProtocolError => e
# ...
end
```

# SMTP exceptions

# initializer

```
SERVER_EXCEPTIONS = [TimeoutError,  
                     IOError,  
                     Net::SMTPUnknownError,  
                     Net::SMTPServerBusy,  
                     Net::SMTPAuthenticationError]  
  
CLIENT_EXCEPTIONS = [Net::SMTPFatalError,  
                     Net::SMTPSyntaxError]  
  
EXCEPTIONS = SERVER_EXCEPTIONS + CLIENT_EXCEPTIONS
```

# controller

```
begin
  Mailer.deliver_message
rescue *CLIENT_EXCEPTIONS
  flash[:warning] = 'The email address supplied
                     is invalid. Please check
                     for spelling mistakes.'
  redirect_to #...
rescue *SERVER_EXCEPTIONS
  flash[:warning] = 'We encountered an internal
                     issue while attempting to
                     deliver this email. Please
                     try again in a few minutes.'
  redirect_to #...
end
```

rake task

```
begin
  Mailer.deliver_message
rescue *EXCEPTIONS => e
  notify_hoptoad(e)
end
```

# Command line

**Always check  
response codes**

```
cd directory; rm -rf
```

cd directory **&&** rm -rf

```
files = `cd /path && ls`  
unless $? .success?  
  fail "Command failed: #{$?}"  
end
```

# cron



crontab -e

MAILTO=support@thoughtbot.com

0 0 \* \* \* /home/username/current/script/cron-nightly

Small As Possible

# script/cron\_nightly

```
#!/bin/bash
. ~/.bash_profile
cd ~/current

echo "Starting Nightly Tasks on $(date)"      >> log/cron 2>&1
./script/runner 'Task.nightly_task!'          >> log/cron 2>&1
if [ -f ~/primary ]; then
  echo "Running primary tasks"                >> log/cron 2>&1
  ./script/runner "Mailer.deliver_nightly"    >> log/cron 2>&1
fi
echo "Done With Nightly Tasks on $(date)"    >> log/cron 2>&1
```

# script/cron\_nightly

```
#!/bin/bash
. ~/.bash_profile
cd ~/current
```

```
echo "Starting Nightly Tasks on $(date)"    >> log/cron 2>&1
./script/runner 'Task.nightly_task!'        >> log/cron 2>&1
if [ -f ~/primary ]; then
  echo "Running primary tasks"                >> log/cron 2>&1
  ./script/runner "Mailer.deliver_nightly" >> log/cron 2>&1
fi
echo "Done With Nightly Tasks on $(date)"  >> log/cron 2>&1
```

Standardize your  
environment

# script/cron\_nightly

```
#!/bin/bash
. ~/.bash_profile
cd ~/current

echo "Starting Nightly Tasks on $(date)" >> log/cron 2>&1
./script/runner 'Task.nightly_task!' >> log/cron 2>&1
if [ -f ~/primary ]; then
  echo "Running primary tasks" >> log/cron 2>&1
  ./script/runner "Mailer.deliver_nightly" >> log/cron 2>&1
fi
echo "Done With Nightly Tasks on $(date)" >> log/cron 2>&1
```

Some tasks  
happen on all servers

# script/cron\_nightly

```
#!/bin/bash
. ~/.bash_profile
cd ~/current

echo "Starting Nightly Tasks on $(date)"    >> log/cron 2>&1
./script/runner 'Task.nightly_task!'        >> log/cron 2>&1
if [ -f ~/primary ]; then
  echo "Running primary tasks"              >> log/cron 2>&1
  ./script/runner "Mailer.deliver_nightly" >> log/cron 2>&1
fi
echo "Done With Nightly Tasks on $(date)"  >> log/cron 2>&1
```

Some tasks happen once  
for the whole cluster

# script/cron\_nightly

```
#!/bin/bash
. ~/.bash_profile
cd ~/current

echo "Starting Nightly Tasks on $(date)"
./script/runner 'Task.nightly_task!'
if [ -f ~/primary ]; then
  echo "Running primary tasks"
  ./script/runner "Mailer.deliver_nightly"
fi
echo "Done With Nightly Tasks on $(date)"
```

```
>> log/cron 2>&1
```

# Log Everything

# script/cron\_nightly

```
module Task
  def nightly_task!
    my_unpredictable_method(params)
  rescue => e
    HoptoadNotifier.notify(
      :error_class    => "Special Error",
      :error_message => "Special Error: #{e.message}"
    )
  end
end
```

## Notify on failures

# Exceptions



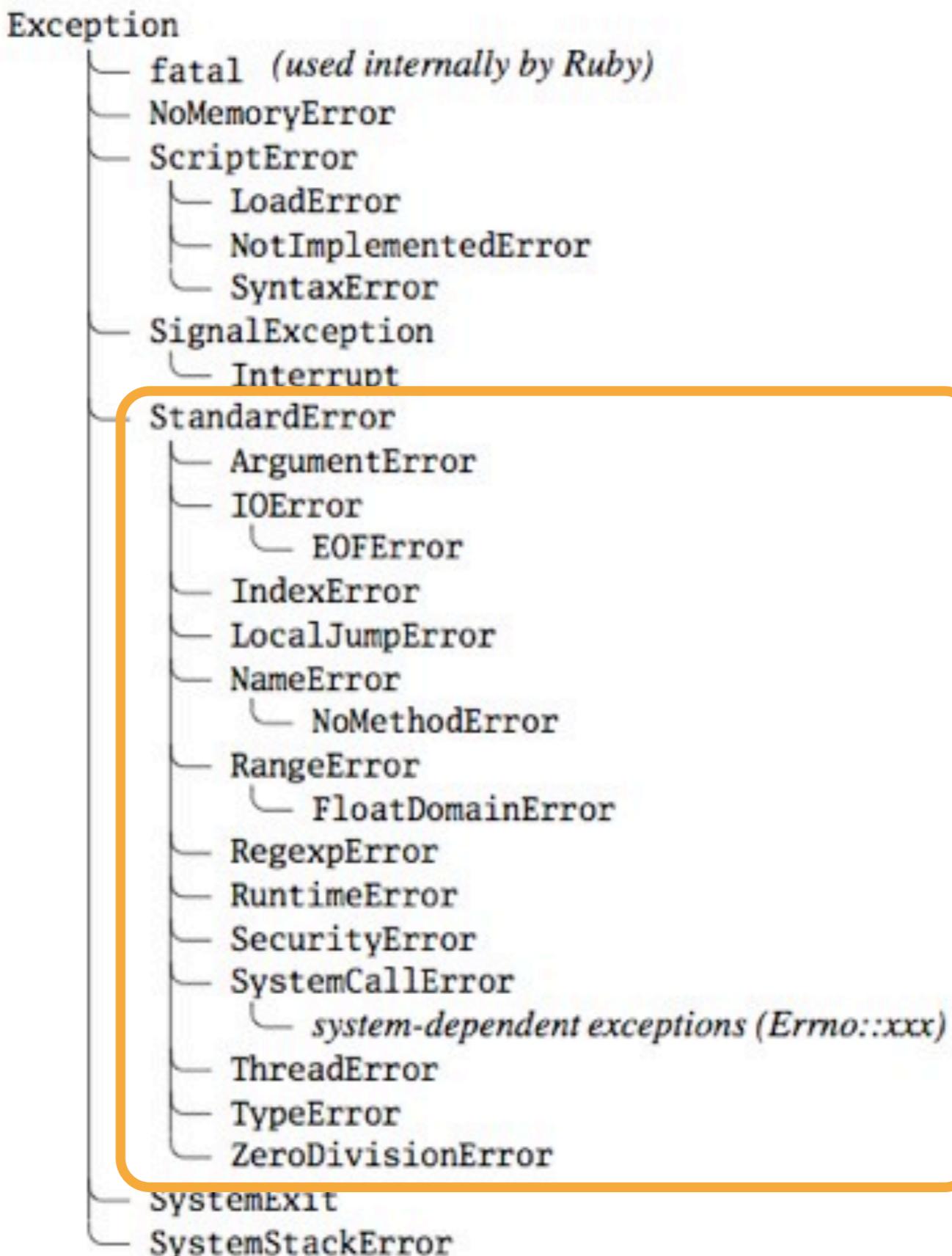
```
x = Foo.bar.baz rescue nil
```

A rescue clause with no parameter is treated as if it had a parameter of StandardError. This means that some lower-level exceptions will not be caught by a parameterless rescue class. If you want to rescue every exception, use

**rescue Exception => e**

Dave Thomas "*Programming Ruby*"

Figure 27.1. Standard exception hierarchy



begin

foo

rescue

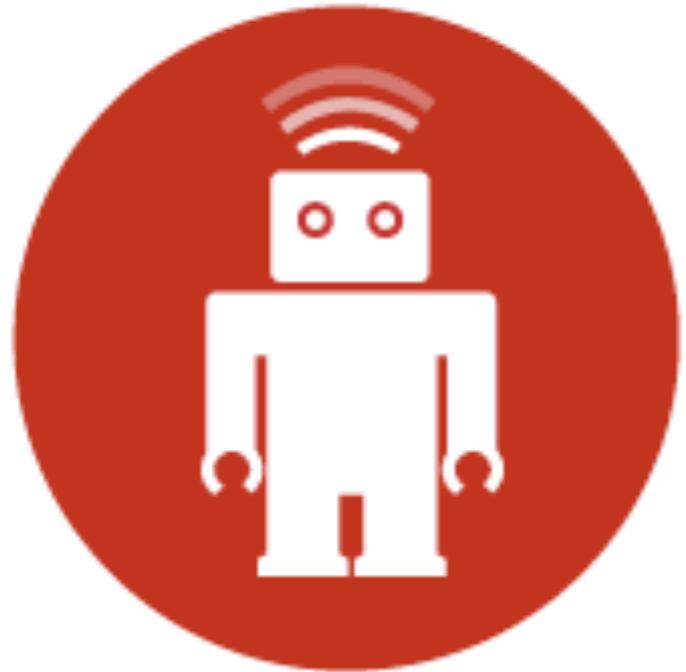
# ...

end

```
begin
  foo
rescue Exception => e
  #
  # ...
end
```

# Exceptions...

*...should be exceptional*



**thoughtbot**

<http://thoughtbot.com>

<http://giantrobots.thoughtbot.com>

**Tammer Saleh**

[tsaleh@thoughtbot.com](mailto:tsaleh@thoughtbot.com)

<http://tammersaleh.com>

# References

**Fail Fast**

<http://martinfowler.com/ieeeSoftware/failFast.pdf>

**qmail Architecture**

[http://hillside.net/plop/2004/papers/mhafizI/PLoP2004\\_mhafizI\\_0.pdf](http://hillside.net/plop/2004/papers/mhafizI/PLoP2004_mhafizI_0.pdf)

**Net::HTTP exceptions**

<http://tammersaleh.com/posts/rescuing-net-http-exceptions>

**Timeout.rb bug**

<http://blog.headius.com/2008/02/rubys-threadraise-threadkill-timeoutrb.html>

**Programming Ruby**

<http://pragprog.com/titles/ruby/programming-ruby>