G'day!

I'm Lindsay

I'm Lindsay

http://holmwood.id.au/~lindsay/

@auxesis

Behaviour driven monitoring with

cucumber-nagios


IBM

```
Feature: google.co.uk
  It should be up
  And I should be able to search for things

  Scenario: Searching for things
    When I visit "http://google.co.uk/"
And I fill in "q" with "wikipedia"
And I press "Google Search"
Then I should see "www.wikipedia.org"
```

```
When /^I visit (.*)$/ do |path|
  pending
end
```

```
When /^I follow "(.*)"$/ do |link|
  pending
end
```

Mechanize

The Mechanize library is used for automating interaction with websites.

Mechanize automatically stores and sends cookies, follows redirects, can follow links, and submit forms. Form fields can be populated and submitted.

Mechanize also keeps track of the sites that you have visited as a history.

Mechanize

The Mechanize library is used for automating interaction with websites.

Mechanize automatically stores and sends cookies, follows redirects, can follow links, and submit forms. Form fields can be populated and submitted.

Mechanize also keeps track of the sites that you have visited as a history.

Webrat

Webrat lets you quickly write expressive and robust acceptance tests for a Ruby web application.

Features:

- **Browser Simulator for expressive, high level acceptance testing**
- **Supports multiple Ruby web frameworks: Rails, Merb and Sinatra**

Webrat

Webrat lets you quickly write expressive and robust acceptance tests for a Ruby web application.

Features:

- Browser Simulator for expressive, high level acceptance testing
- Supports multiple Ruby web frameworks: Rails, Merb and Sinatra

Webrat::MechanizeAdapter

```
World do
  Webrat::Session.new(
    Webrat::MechanizeAdapter.new
  )
end

# features/support/env.rb
```

```
# features/google.co.uk/search.feature


Feature: google.co.uk
  It should be up
  And I should be able to search for things

  Scenario: Searching for things
    When I visit "http://google.co.uk/"
And I fill in "q" with "wikipedia"
And I press "Google Search"
Then I should see "www.wikipedia.org"
```

```
$ cucumber --require features/ \
           features/google.co.uk/search.feature


Feature: google.co.uk
  It should be up
  And I should be able to search for things

  Scenario: Searching for things
    When I visit "http://google.co.uk/"
    And I fill in "q" with "wikipedia"
    And I press "Google Search"
    Then I should see "www.wikipedia.org"

1 scenario (1 passed)
3 steps (3 passed)
0m1.481s
```

```
$ cucumber --require features/ \
           features/google.co.uk/search.feature

Feature: google.co.uk
  It should be up
  And I should be able to search for things

  Scenario: Searching for things
    When I visit "http://google.co.uk/"
    And I fill in "q" with "wikipedia"
    And I press "Google Search"
    Then I should see "www.wikipedia.org"

1 scenario (1 passed)
3 steps (3 passed)
0m1.481s
```

--format pretty

cucumber

cucumber
webrat

cucumber

webrat

mechanize

cucumber
webrat
mechanize
nagios

```
$ cucumber-nagios \
    features/google.co.uk/search.feature

Critical: 0, Warning: 0, 4 okay
```

cucumber-nagios

how it works

```
$ gem install cucumber-nagios

$ cucumber-nagios-gen project devopsdays

$ cd devopsdays

$ gem bundle
```

```
$ cucumber-nagios \
          features/google.co.uk/search.feature
```

```
$ cucumber-nagios \
          features/google.co.uk/search.feature

$ cucumber --require features \
          --formatter Cucumber::Formatter::Nagios \
          features/google.co.uk/search.feature
```

```
$ cucumber-nagios \
          features/google.co.uk/search.feature

$ cucumber --require features \
          --formatter Cucumber::Formatter::Nagios \
          features/google.co.uk/search.feature
```

```ruby
require 'cucumber/formatter/console'

module Cucumber
  module Formatter
    class Nagios


    end
  end
end
```

```ruby
def initialize(step_mother, io, options={})
  @failed  = []
  @passed  = []
  @warning = []
  @io = io
end

def after_step_result(... status ...)
  case status
  when :passed
    @passed << step_match
  when :failed
     @failed << step_match
  when :undefined
    @warning << step_match
  end
end
```

```ruby
def after_features(steps)
  print_summary
end

private
def print_summary
  @total = @failed.size + @passed.size + @warning.size
  message = ""
  message += "Critical: #{@failed.size}, "
  message += "Warning: #{@warning.size}, "
  message += "#{@passed.size} okay"
  # nagios performance data
  message += " | passed=#{@passed.size}"
  message += ", failed=#{@failed.size}"
  message += ", nosteps=#{@warning.size}"
  message += ", total=#{@total}\n"

  @io.print(message)
  @io.flush
end
```

```ruby
#!/usr/bin/env ruby

__DIR__ = File.expand_path(File.dirname(__FILE__))
feature = ARGV[0]

command = "#{__DIR__}/cucumber"
command += " --require #{__DIR__}/common.rb"
command += " --require features/"
command += " --format Nagios::NagiosFormatter"
command += " #{feature}"

system(command) ? exit(0) : exit(2)
```

```ruby
#!/usr/bin/env ruby

__DIR__ = File.expand_path(File.dirname(__FILE__))
feature = ARGV[0]

command = "#{__DIR__}/cucumber"
command += " --require #{__DIR__}/common.rb"
command += " --require features/"
command += " --format Nagios::NagiosFormatter"
command += " #{feature}"

system(command) ? exit(0) : exit(2)
```

caveats

AJAX

Progressive Enhancement / Graceful Degradation

existing features + steps

Mechanize + Nokogiri

multiple scenarios

```
(Critical: 0, Warning: 0, 2 okay
Critical: 0, Warning: 0, 4 okay)
```

but why?

asking the wrong questions

ping

tcp connect

is my server up?

can I see my app?

dead vm?

is my server up?

can I see my app?

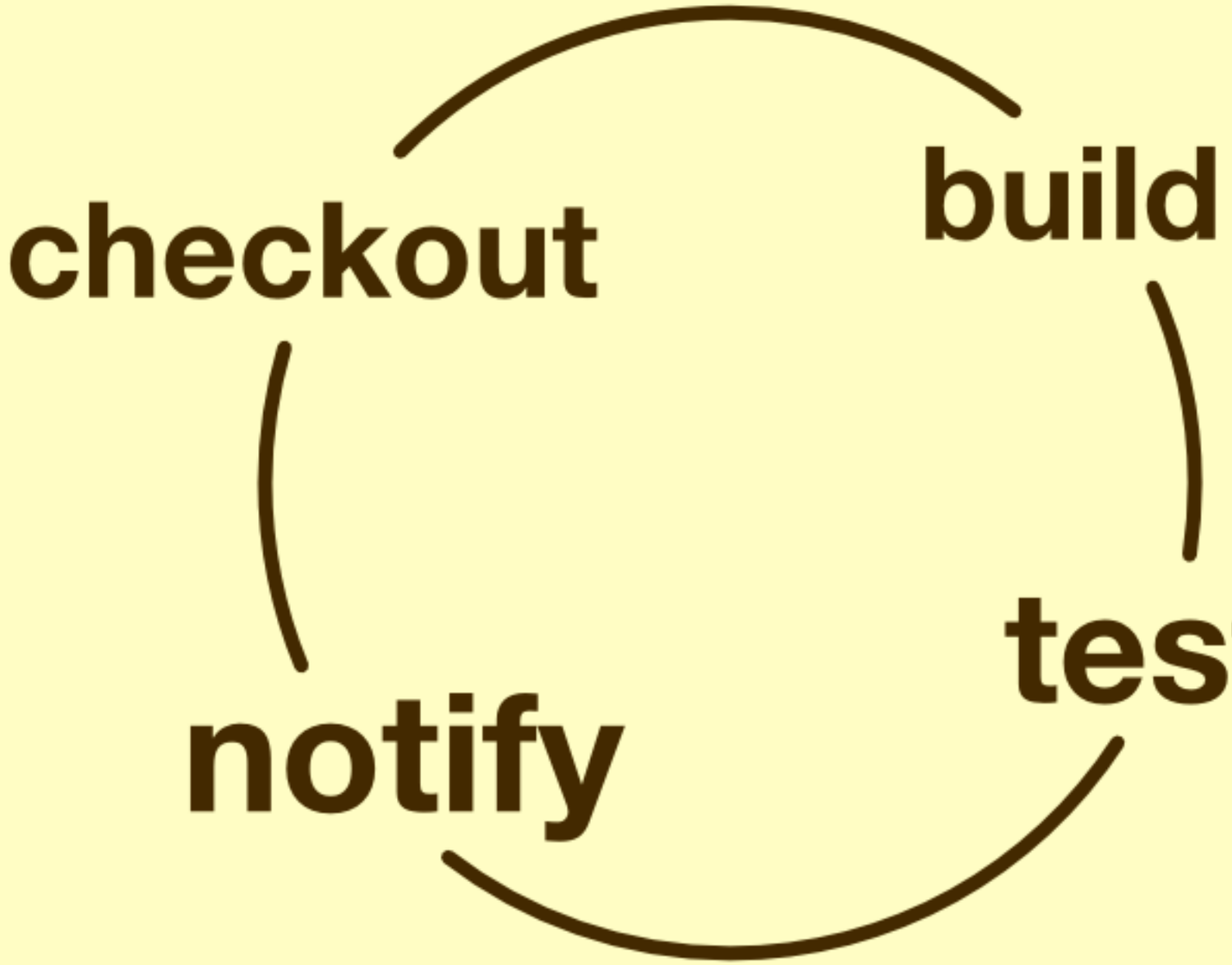
serving content?

the right questions

is my app behaving?

can I navigate?
can I sign in?
can I place an order?
can I ...?

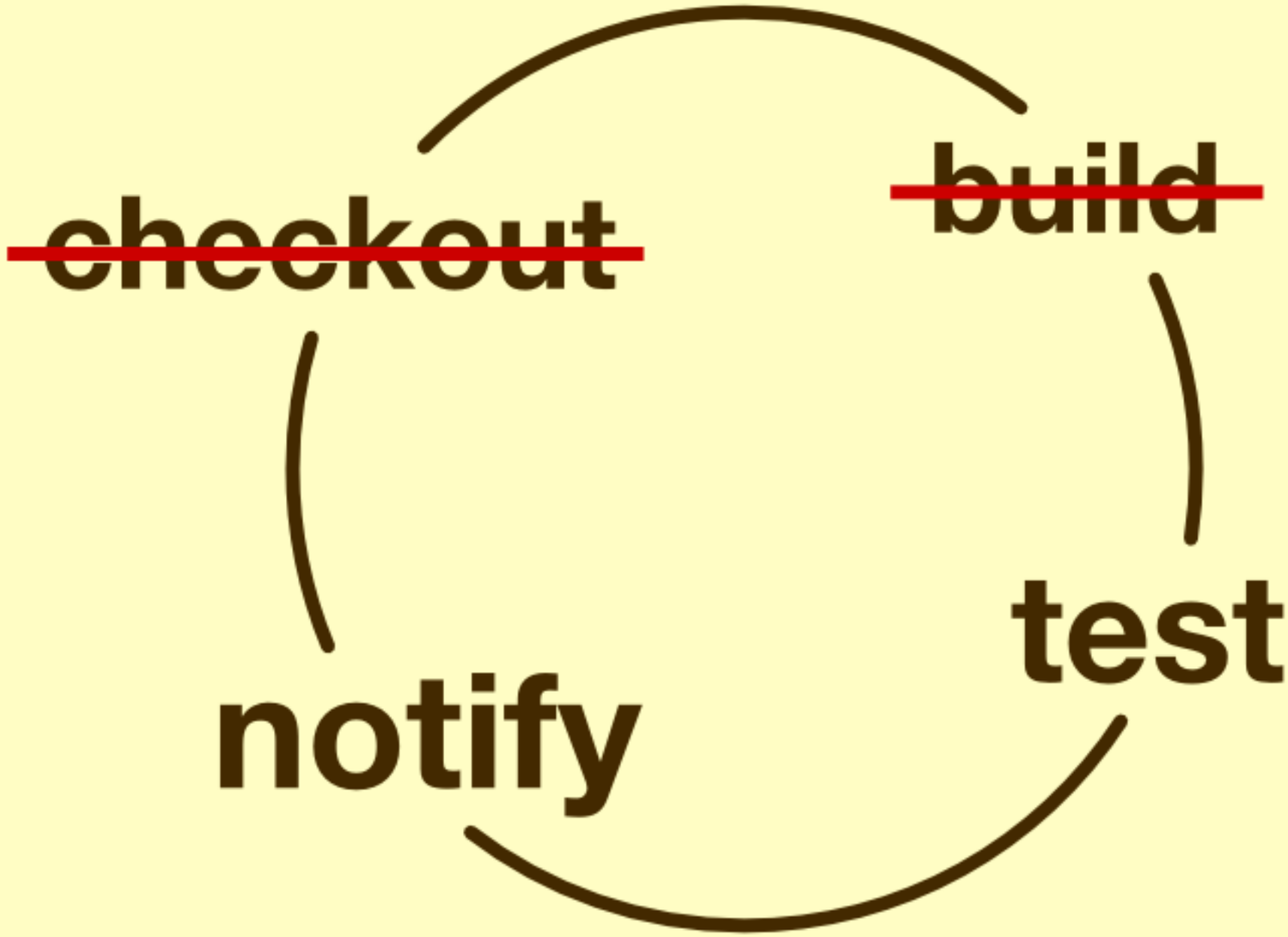
continuous integration



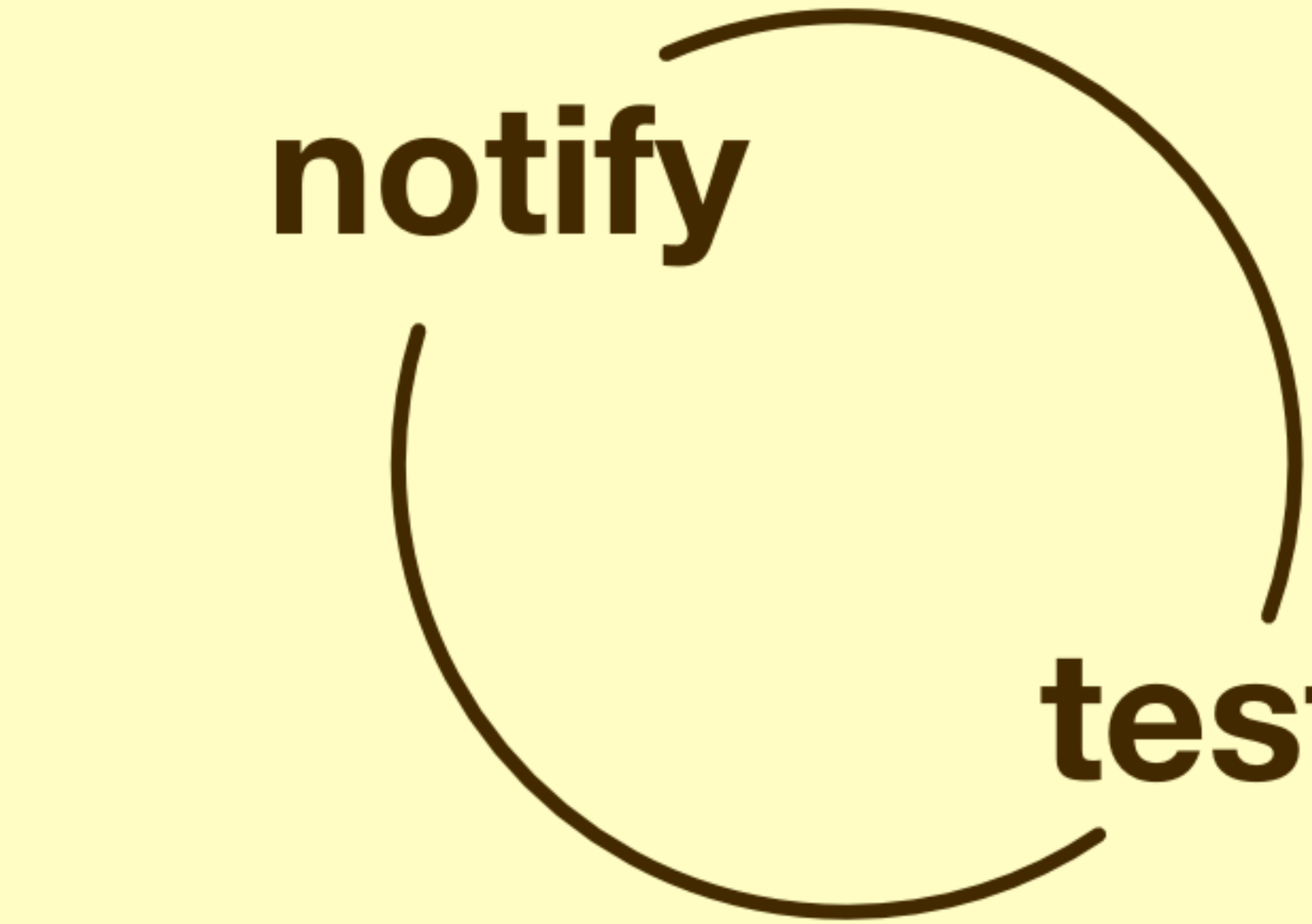
ci lifecycle

monitoring:

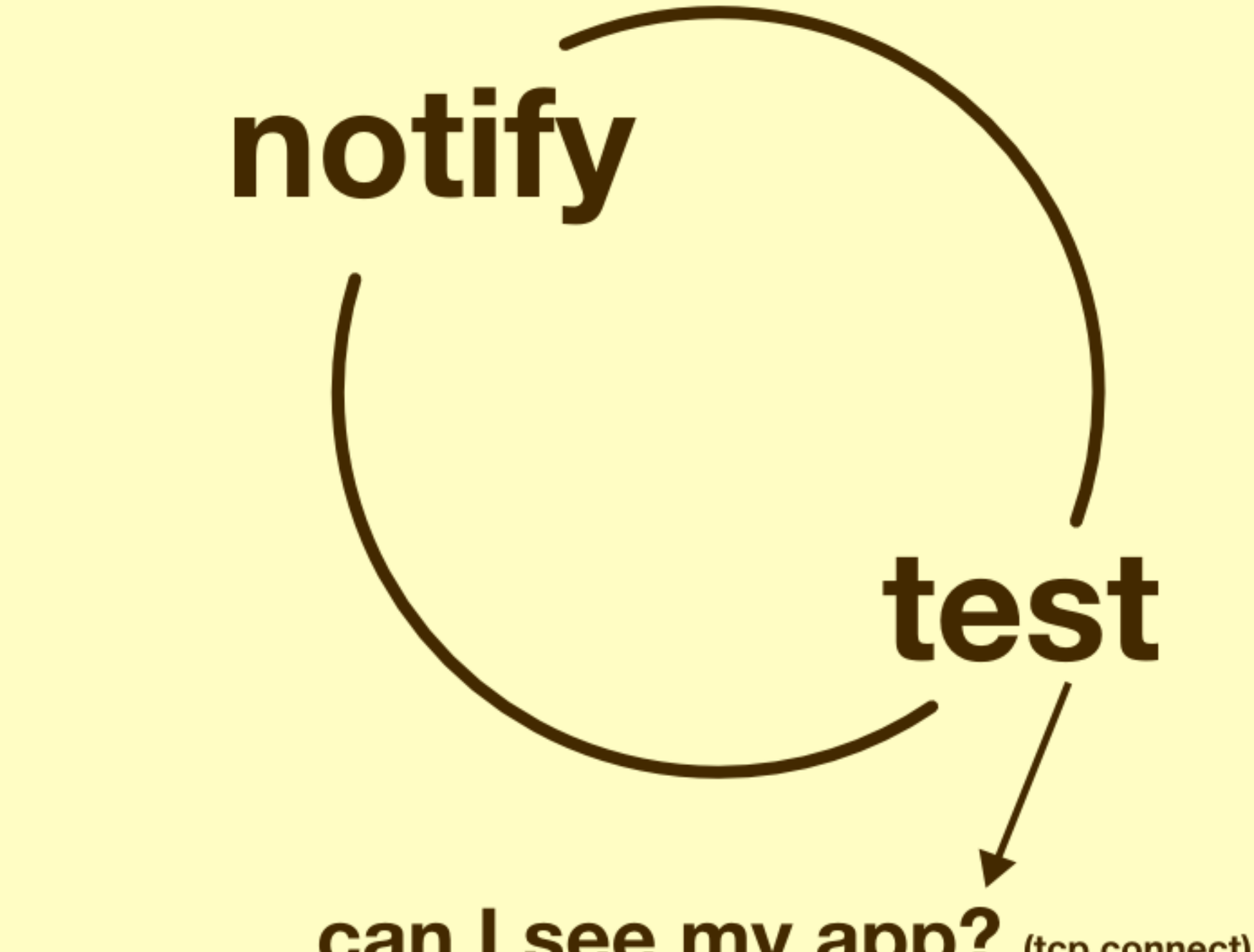
continuous integration for production apps

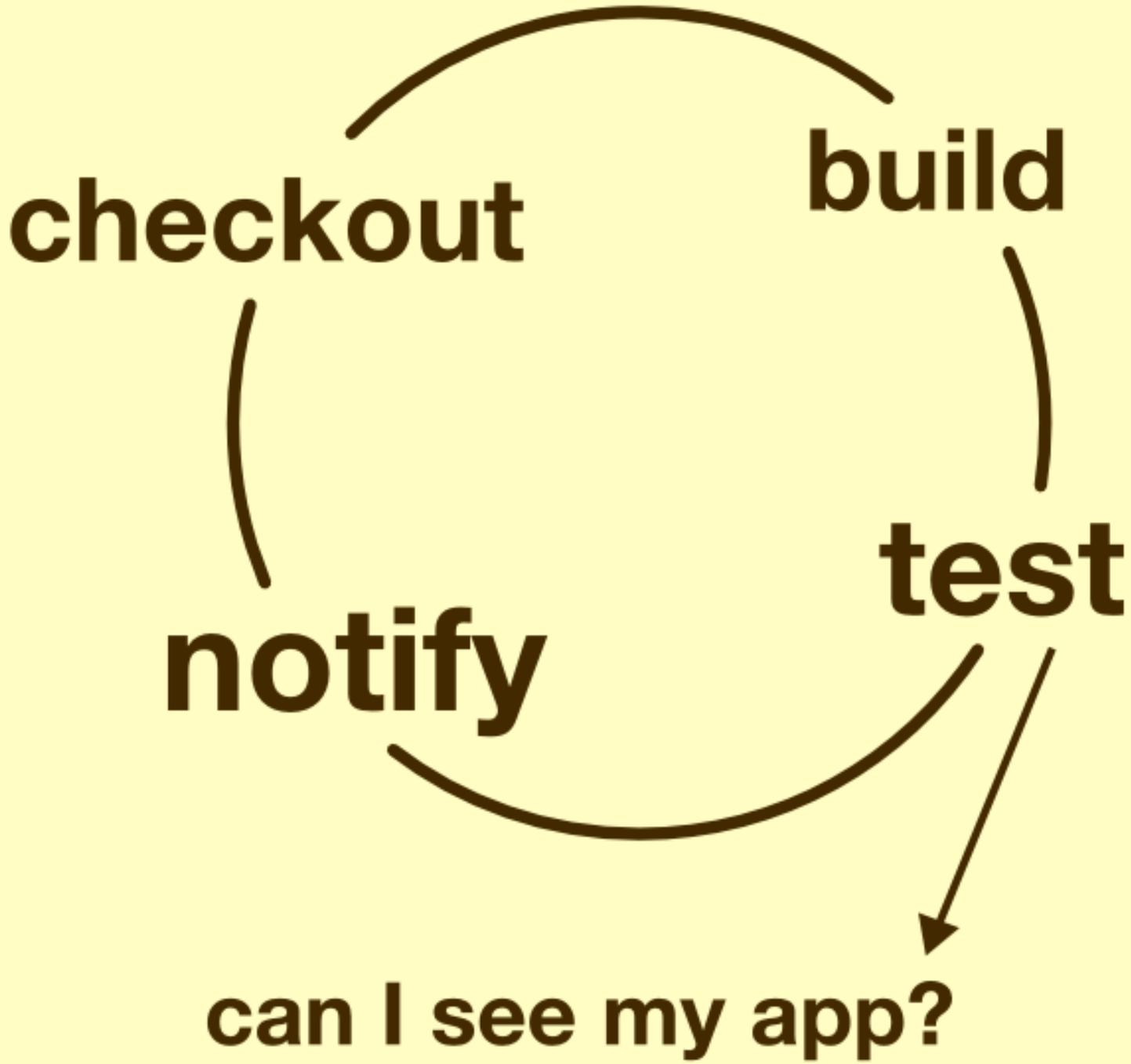


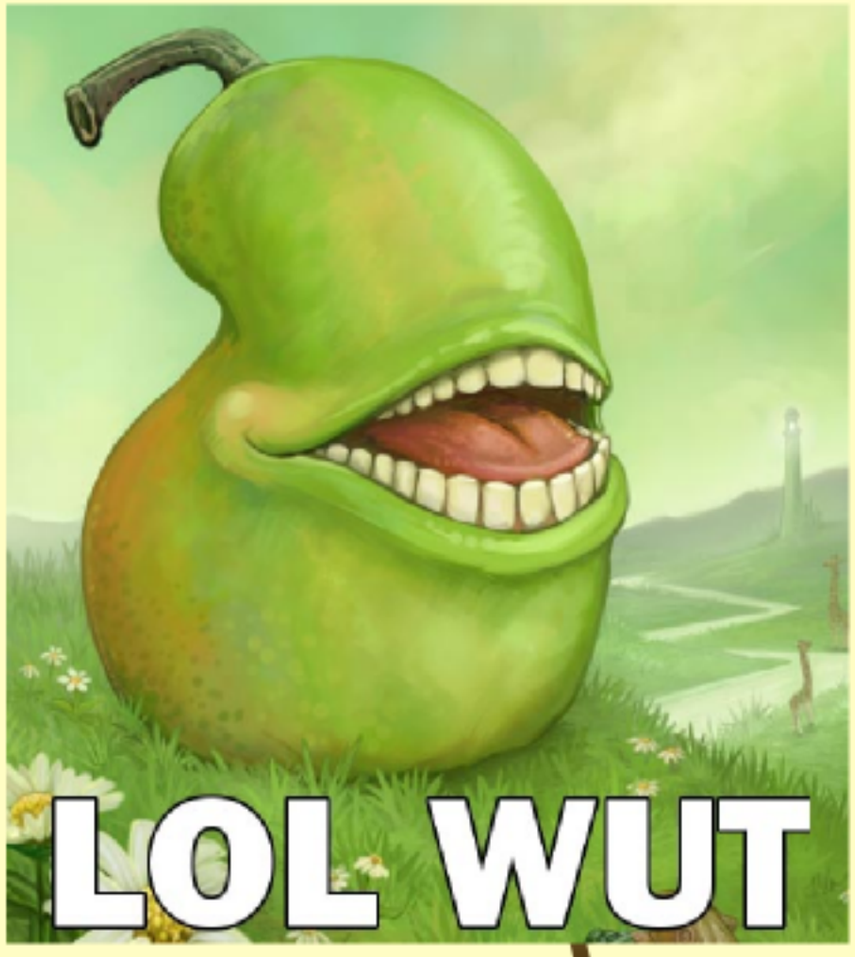
ci lifecycle



monitoring system


notify
test
can I see my app? (tcp connect)


checkout
build
test
notify
can I see my app?



notify

test

can I see my app?

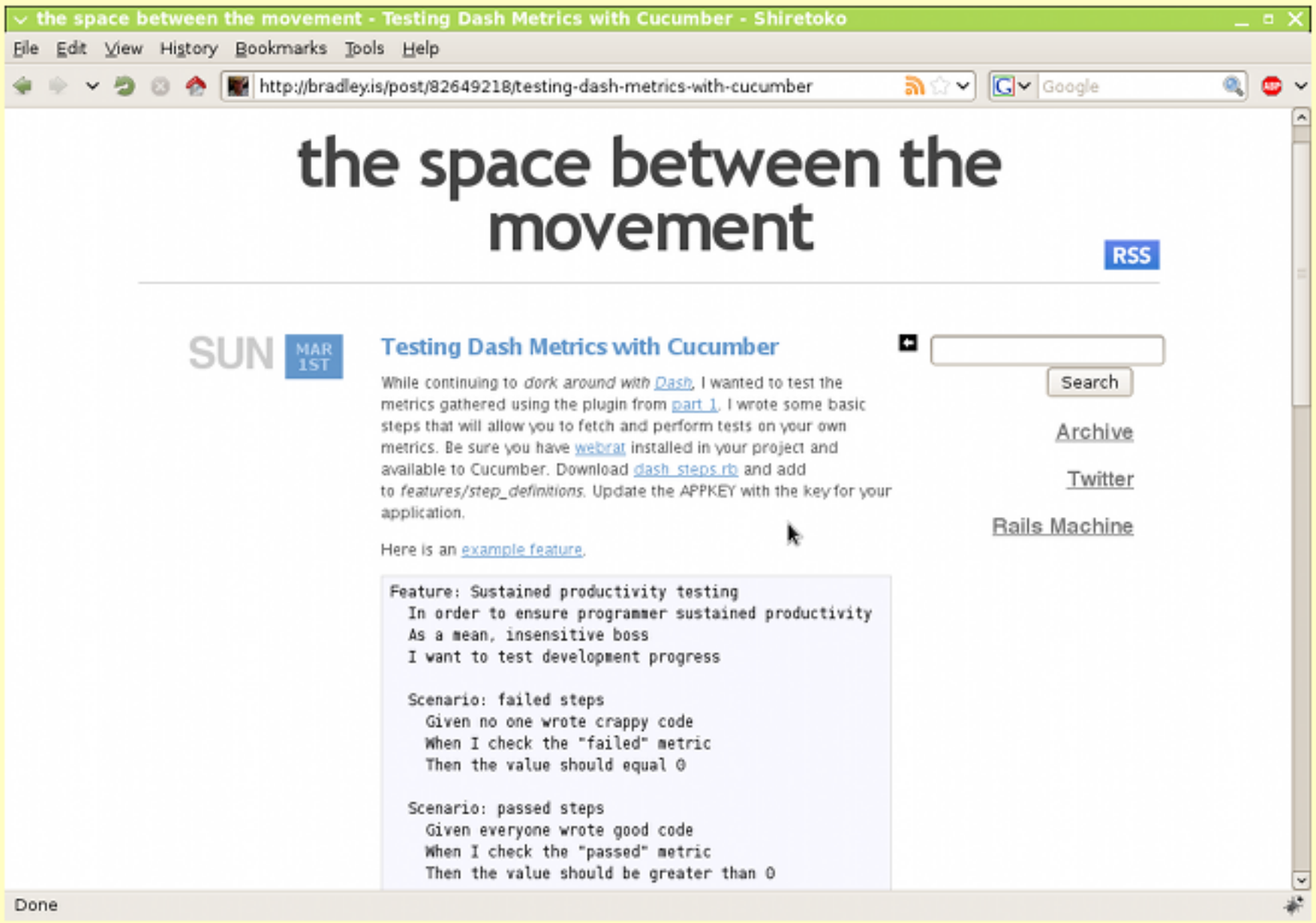
cool uses

(Behaviour Driven Monitoring)

Telephony BDM

with telephony-system-tests (by @jsgoecke)

Dash metrics



cucumber + dash **(by @bradleyktaylor)**

Instead of writing boring monitoring plugins from scratch, you can now do behaviour driven ops.

Transform from a grumpy, misanthropic sysadmin to a hipster, agile developer instantly.

- Bradley Taylor

Thank you!

http://auxesis.github.com/cucumber-nagios

Dash metrics - http://bit.ly/lWWfE

Telephony BDM - http://bit.ly/94TAn