

Business Natural Language (BNL)

ThoughtWorks

<http://thoughtworks.com>

Jay Fields

<http://jayfields.com>

wtf?

Domain Specific Language for Domain Experts

distinguishing characteristics

natural language

written by the business

developer

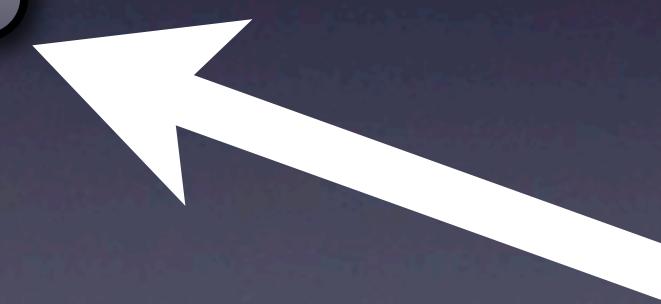
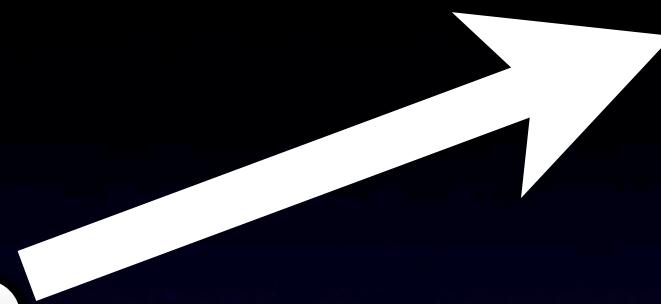
business

qa

business

developer

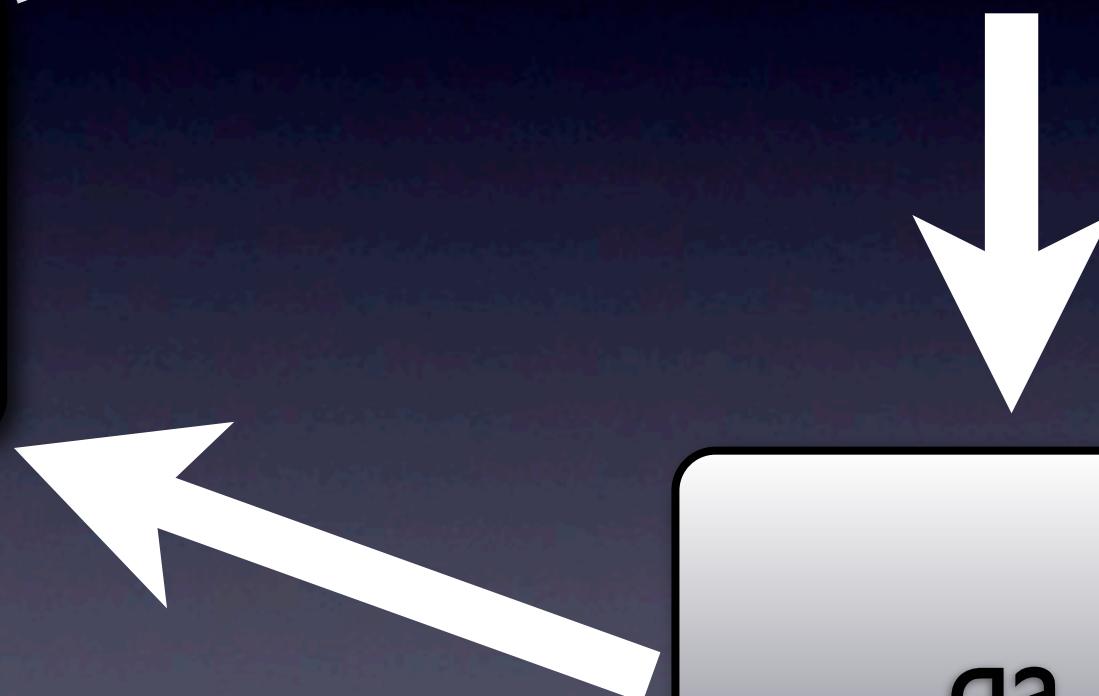
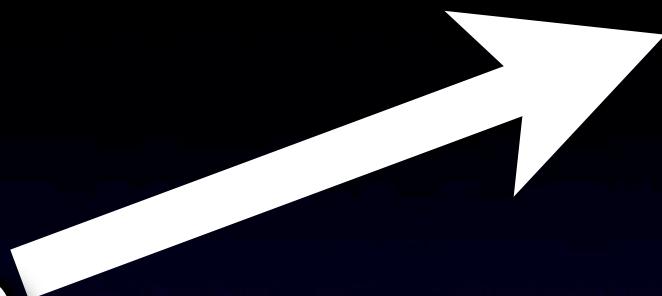
qa



business

developer

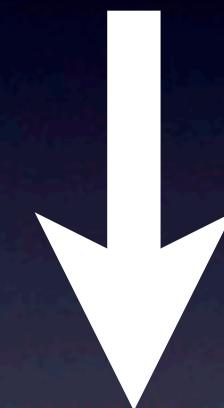
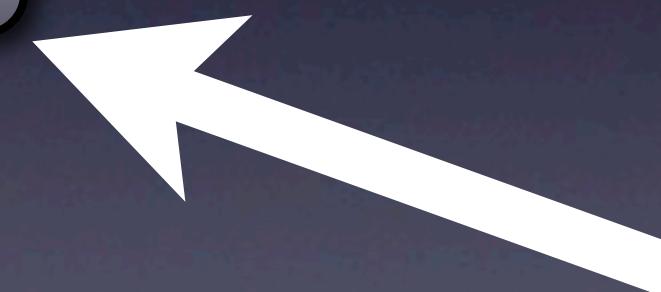
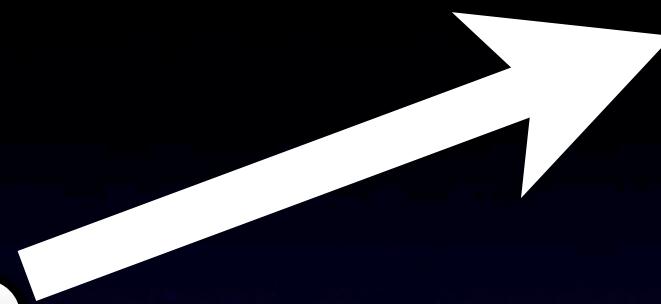
qa



business

developer

qa



business



DSL

Benefits



Managers: Distribute Workload

Managers can distribute workload among team members.

Managers can delegate tasks to team members.

Managers can prioritize tasks and assign them to team members.

Managers can provide guidance and support to team members.

Managers can monitor progress and provide feedback to team members.

Managers can adjust workload distribution based on team member's performance and availability.

Developers: Focus on Technical Issues

Business:Time To Market

htf?

(does it work)

external

**First Class /
Business Class /
BusinessFirst /
Full Fare
Economy Class**

A, C, D, J,
R, Y, H Z

2

Economy Class

K, M, N, B,
O, V, U

I

specification

Award 2 points if
fare class is A, C, D,
J, R, Y, H, Z

code

Award 2 points if
fare class is A, C, D,
J, R, Y, H, Z

```
class FareRule<~  
  def apply(segment)<~  
    %w[a, c, d, j, r, y, h z].include?(segment.fare_class.downcase) ? 2 : 0<  
  end<  
end
```

```
def self.create_rule(line)-
  case line-
    when /^award (\d+) points* if fare class is ([A-Za-z]+, [A-Za-z]+)*$/i-
      create_fare_rule(Object.new, $1, $2)-
      # .. other whens-
    end-
  end-

def self.create_fare_rule(rule, amount, fare_classes)-
  eval "def rule.apply(segment)-
        %w[#{fare_classes.delete(',')}]>.include?(segment.fare_class.downcase) ? #{amount} : 0-
      end"-
  rule-
end-
```

code, specification,

code, specification,
and documentation

readable
documentation

Syntax OK

```
Award 1 point If fare class is B  
Award 1 points if fare class is A  
Award 1 point if flight is to SFO  
Award 5 points if flight is from EWR  
Award 8 points if flight is From las
```

Update

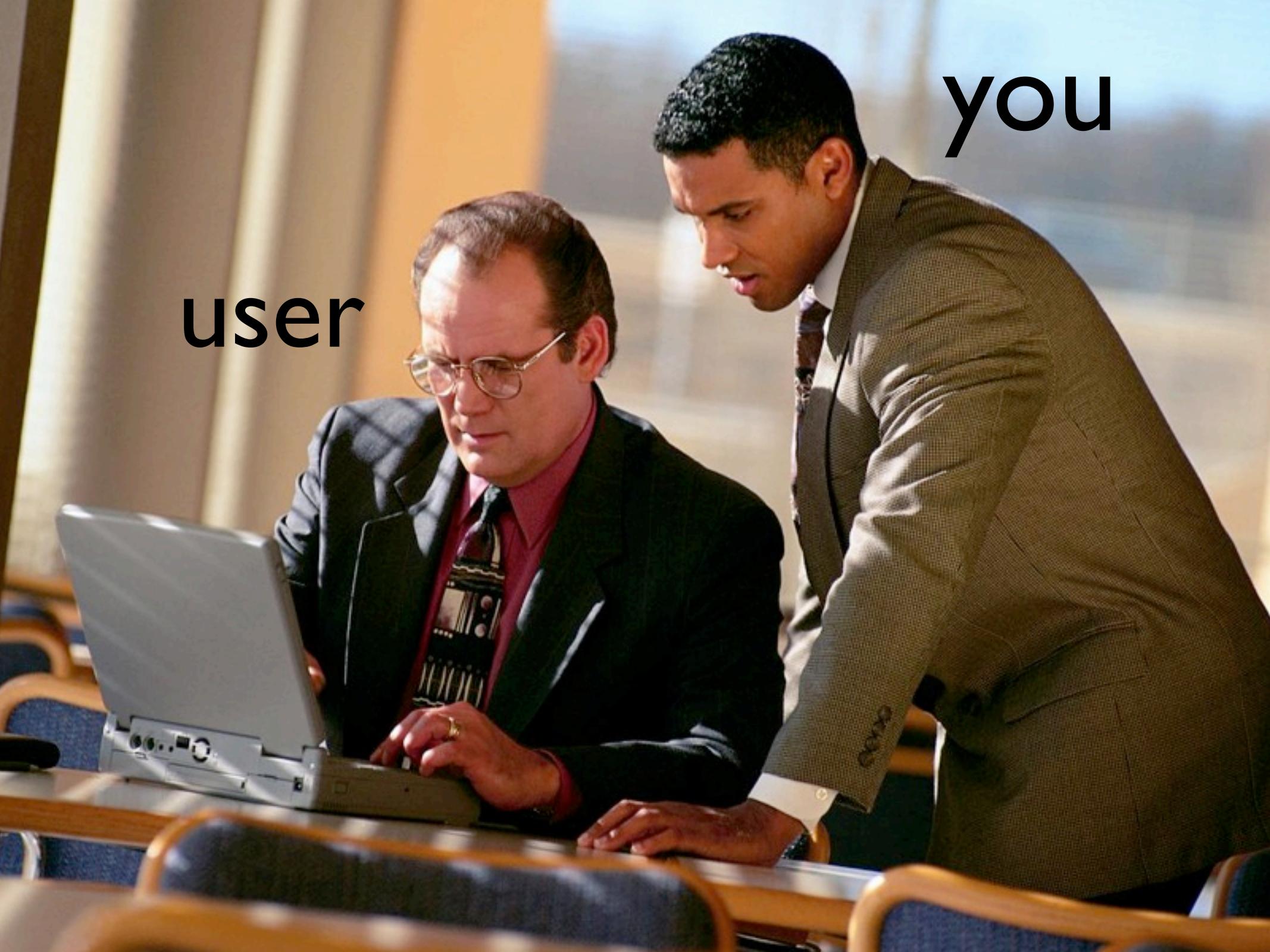
descriptive and
maintainable
phrases (damp)

maintainable

readable by any
subject matter
expert

you

user



Syntax OK

```
Award 1 point If fare class is B
Award 1 points if fare class is A
Award 1 point if flight is to SFO
Award 5 points if flight is from EWR
Award 8 points if flight is From las
```

Update

A stylized illustration depicting a classroom scene. A teacher, shown from the side and wearing a teal dress, points towards the right with their right hand. Several students are seated at desks, facing the teacher. The students are represented by dark silhouettes against a light background.

Training





"A change that used to take 3-4 days now takes about 10 minutes."

Teachers
Talking
French



© 2001 VHS. All rights reserved. Manufactured by
VHS International, Inc., Parsippany, NJ 07054



```
---  
- :to: SF0  
  :frequent_flyer_number: BK1126  
  :from: EWR  
  :fare_class: A  
- :to: EWR  
  :frequent_flyer_number: BK7670  
  :from: LAS  
  :fare_class: B  
- :to: LAS  
  :frequent_flyer_number: BK8783  
  :from: SF0  
  :fare_class: B  
- :to: SF0  
  :frequent_flyer_number: BK7107  
  :from: LAS  
  :fare_class: C  
- :to: SEO
```

Fare class A tickets: 344

Fare class B tickets: 335

Fare class C tickets: 321

Trips to EWR: 329

Trips to SFO: 350

Trips to LAS: 321

Trips from EWR: 316

Trips from SFO: 337

Trips from LAS: 347

Award 1 point If fare class is B : 335
Award 1 points if fare class is A, C : 665
Award 1 point if flight is to SFO : 350
Award 1 points if flight is from EWR : 316

Fare class A tickets: 344
Fare class B tickets: 335
Fare class C tickets: 321

Trips to EWR: 329
Trips to SFO: 350
Trips to LAS: 321

Trips from EWR: 316
Trips from SFO: 337
Trips from LAS: 347

The screenshot shows a web browser window with the following details:

- Address Bar:** http://localhost:3000/awards
- Toolbar:** Includes standard browser icons for back, forward, search, and refresh.
- Title Bar:** A blue header bar with the text "Fake Internal Application" in white, bold, italicized font.
- Content Area:** Displays the following data:
 - Total points earned: 1666
 - Audit:**
 - Fare class A tickets: 344
 - Fare class B tickets: 335
 - Fare class C tickets: 321
 - Trips to EWR:** 329
 - Trips to SFO:** 350
 - Trips to LAS:** 321
 - Trips from EWR:** 316
 - Trips from SFO:** 337
 - Trips from LAS:** 347
 - Award 1 point If fare class is B :** 335
 - Award 1 points if fare class is A, C :** 665
 - Award 1 point if flight is to SFO :** 350
 - Award 1 points if flight is from EWR :** 316
- Bottom Left:** A blue link labeled "Edit Rules".

The screenshot shows a web browser window with the following details:

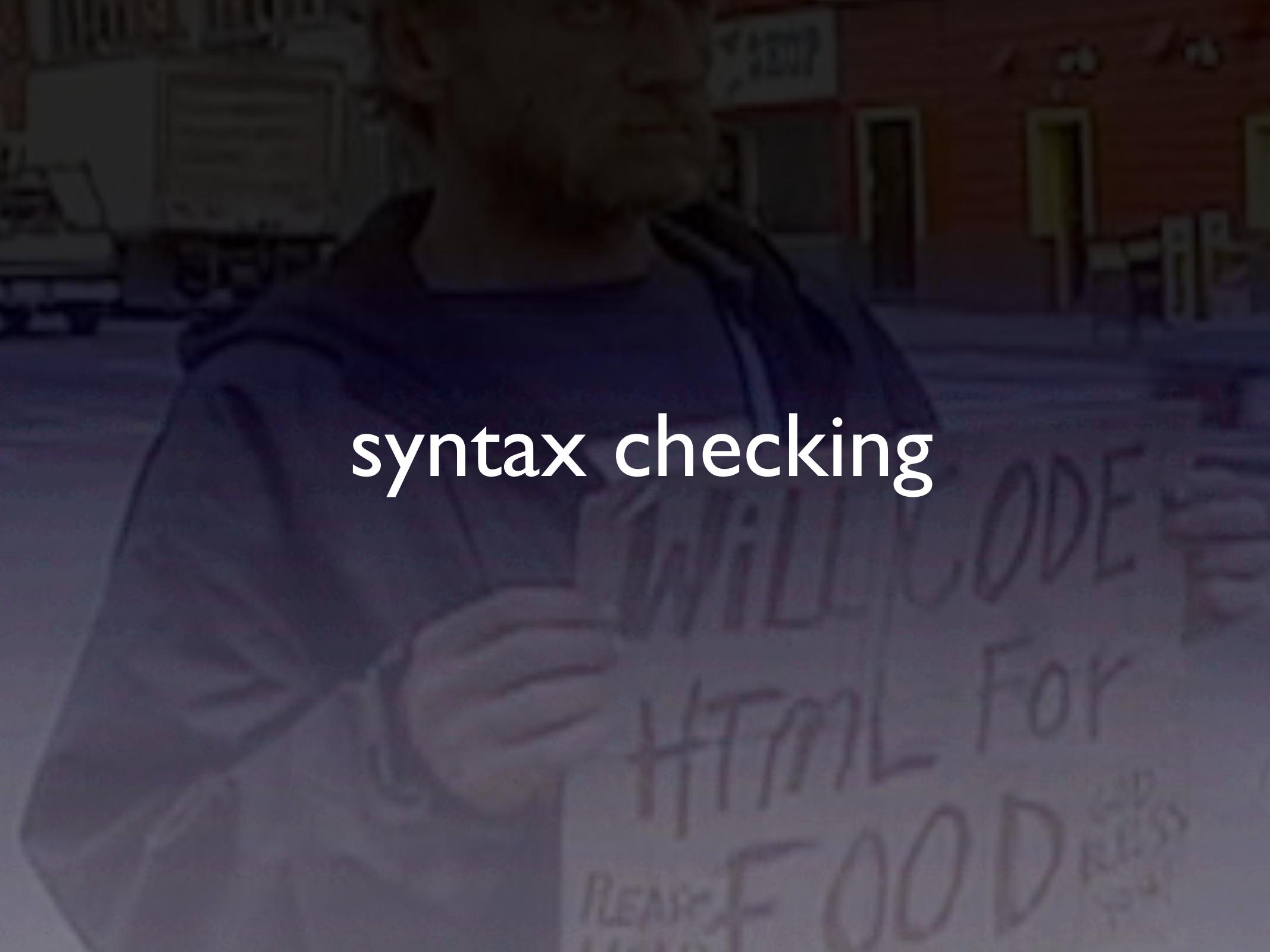
- Address Bar:** http://localhost:3000/awards
- Title Bar:** Fake Internal Application
- Content Area:**
 - Total points earned: 1666
 - Audit:
 - Fare class A tickets: 344
 - Fare class B tickets: 335
 - Fare class C tickets: 321
 - Trips to EWR: 329
 - Trips to SFO: 350
 - Trips to LAS: 321
 - Trips from EWR: 316
 - Trips from SFO: 337
 - Trips from LAS: 347
 - Award 1 point If fare class is B : 335
 - Award 1 points if fare class is A, C : 665
 - Award 1 point if flight is to SFO : 350
 - Award 1 points if flight is from EWR : 316
- Bottom Left:** Edit Rules

A color photograph of a man with short brown hair, wearing a dark purple zip-up hoodie, sitting outdoors and holding a large, light-colored cardboard sign. He is looking slightly upwards and to his right. The background shows a brick building with a red door and a window, and some trees and a fence in the distance.

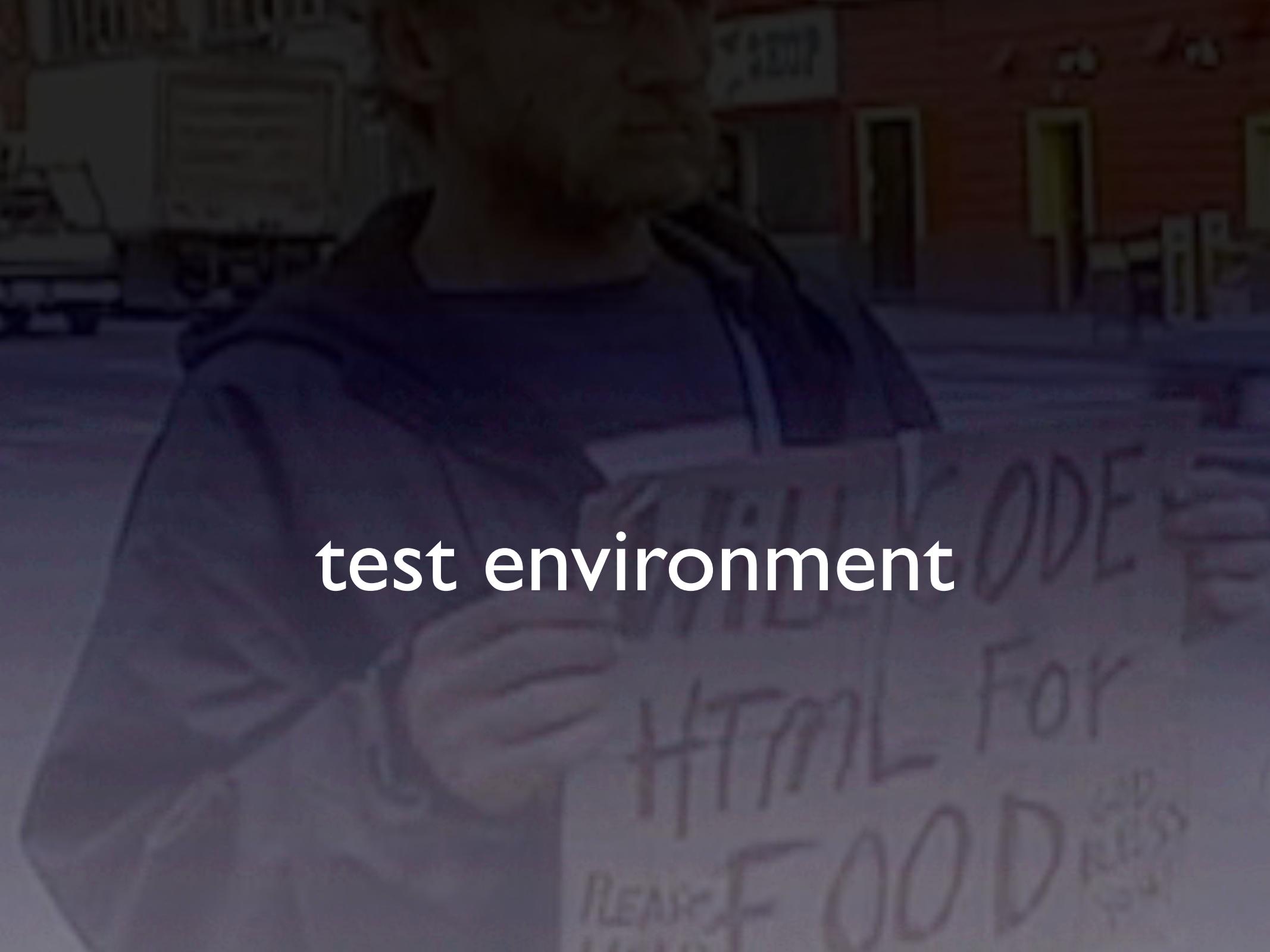
WILL CODE
HTML FOR
RAMA FOOD

language workbench

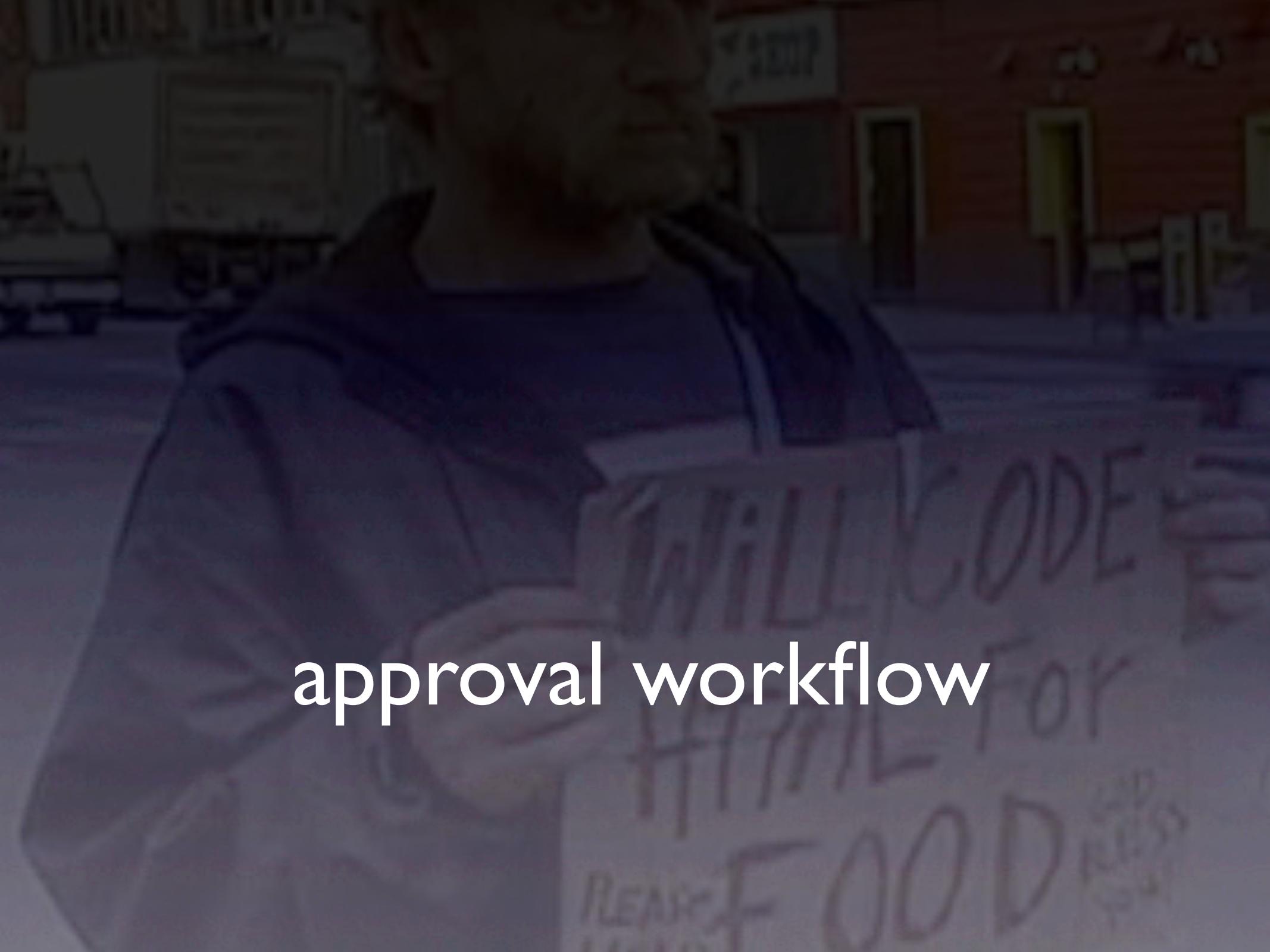
HTML code
REACT components

A person is sitting at a desk in an office environment, looking down at their hands which are clasped together. In the background, a computer monitor displays a large amount of text, likely code or a document, in a monospaced font. The scene is dimly lit, with the screen being the primary light source.

syntax checking

A person is sitting at a desk in a dimly lit room, looking at a computer screen. The screen displays the text "test environment".

test environment

A dark, grainy photograph of a person sitting at a desk in an office. The person is facing away from the camera, looking at a computer screen. The background is blurred, showing office equipment and papers.

approval workflow

business

business



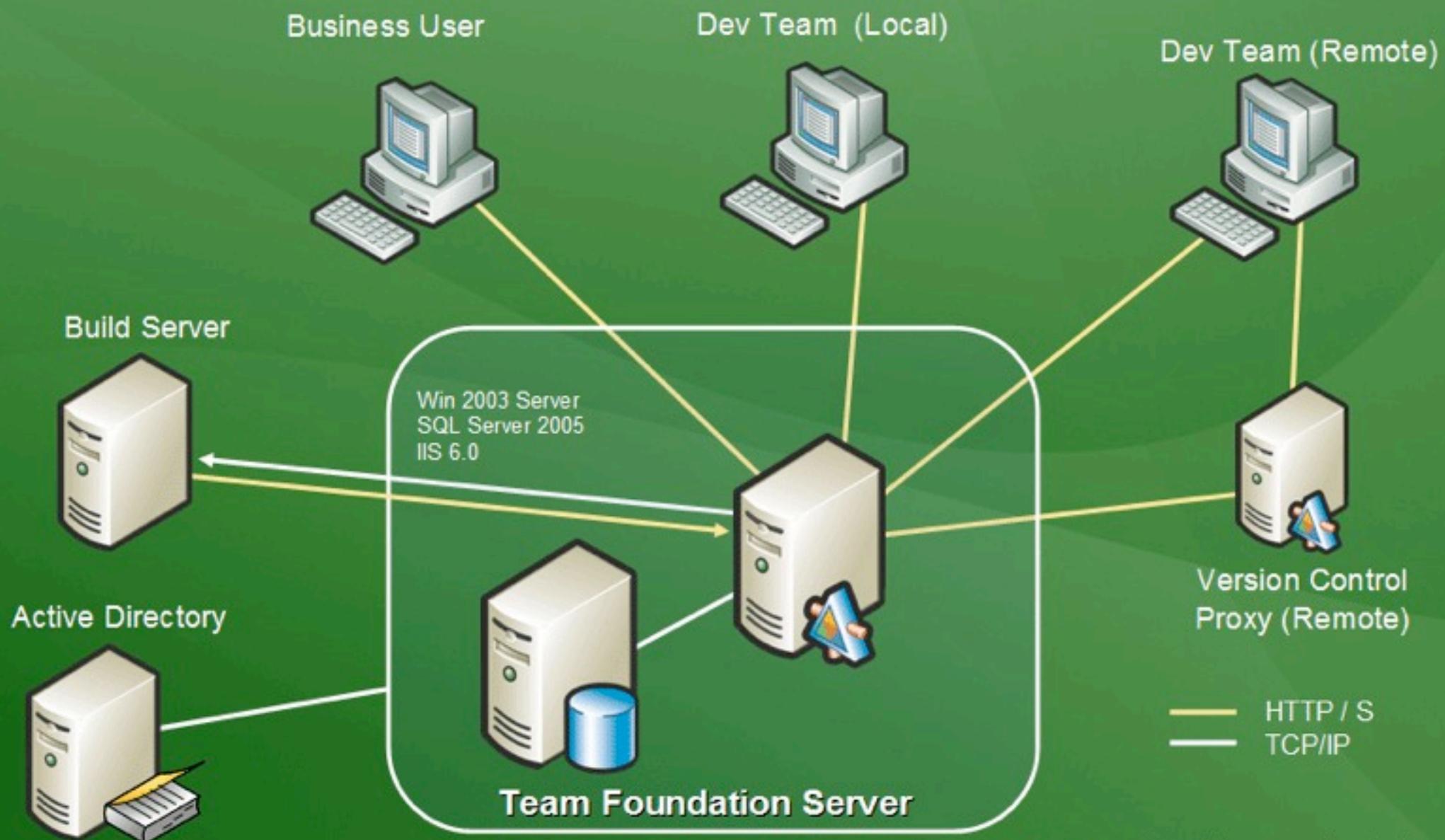
```
graph TD; business[business] --> approver[approver]; approver --> production[production]
```

business

approver

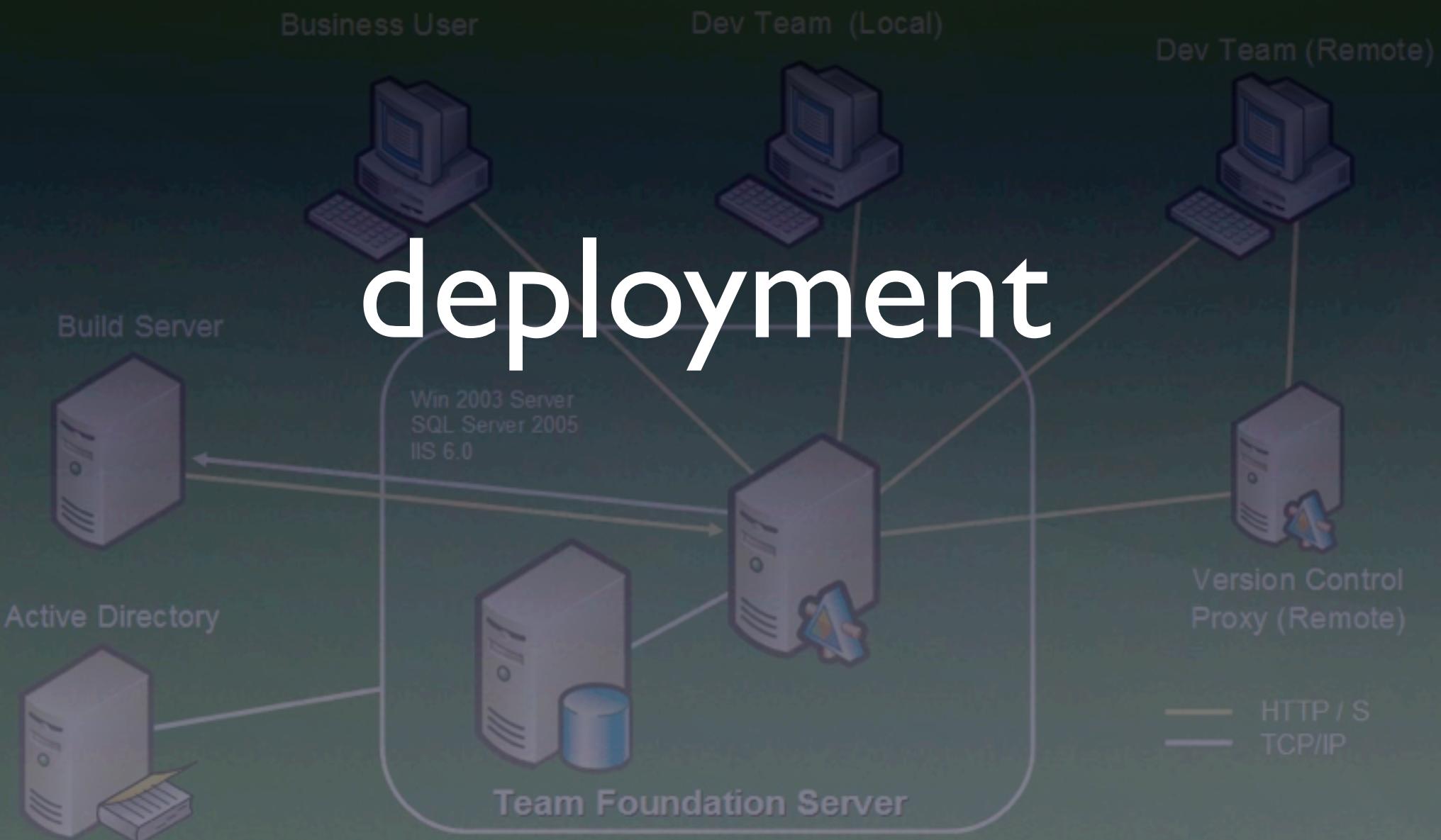
production

Team System Deployment



Team System Deployment

deployment



*map.js X *index.html 2

```
geocoder.getLatLng(address, function (location) {
    map.setCenter(location, 13);
    var marker = new google.maps.Marker(location);
    map.addOverlay(marker);
});
```

Multiple annotations found at this line:

- missing ; before statement
- 1 changed line

latField.value = location.lat();
lngField.value = location.lng();

bad syntax ((({;
indovHtml getAddressIn:

Syntax error checking

Source Preview

Problems X

1 error, 0 warnings, 0 infos (Filter matched 1 of 10 items)

Description	Resource	Path
Errors (1 item)		

when the far
did you mean ‘fare’?

A screenshot of a code editor showing a file named 'map.js'. The code contains several lines of JavaScript related to geocoding and mapping. A red box highlights a syntax error in the line 'latField.value = location.lat();', specifically the closing brace '}'. A tooltip from the IDE indicates multiple annotations: '- missing ; before statement' and '- 1 changed line'. The status bar at the bottom shows '1 error, 0 warnings, 0 infos (Filter matched 1 of 10 items)'.

```
geocoder.getLatLang(address, function (location) {
    map.setCenter(location, 13);
    var marker = new google.maps.Marker(location);
    map.addOverlay(marker);
});
```

Multiple annotations found at this line:

- missing ; before statement
- 1 changed line

bad syntax ({{ {;

latField.value = location.lat();

latField.value = location.lng();

});

Source Preview

Problems

1 error, 0 warnings, 0 infos (Filter matched 1 of 10 items)

Description Resource Path

Errors (1 item)

You entered pint. Award amount should be followed by 'point' or 'points' on line 3

Award 1 point If fare class is B
Award 1 points if fare class is A
Award 1 pint if flight is to SFO
Award 5 points if flight is from EWR
Award 8 points if flight is From las

Update

```
class Syntax-
  def self.parse(rules)
    find_errors(rules).any? ? find_errors(rules).join("\n") : "Syntax OK"
  end

  def self.find_errors(rules)
    rules.split(/\n/).enum_with_index.inject([]) do |errors, (line, index)|
      parse_line(errors, line, index + 1)
      errors
    end
  end

  def self.parse_line(errors, line, index)
    Line.new(line, index).check_line
  rescue Exception => ex
    errors << ex.message
  end
end
```

Syntax OK

```
Award 1 point If fare class is B  
Award 1 points if fare class is A  
Award 1 point if flight is to SFO  
Award 5 points if flight is from EWR  
Award 8 points if flight is From las
```

Update

```
class Line<~  
  attr_accessor :text, :line_number  
  def initialize(text, line_number)  
    self.text = text.squeeze(" ").strip  
    self.line_number = line_number  
  end  
  
  def check_line  
    check_pop(/^award/i, "Rules must start with 'Award'")  
    check_pop(/^\d+/i, "Rules must specify an award amount (e.g. 1, 23, 100)")  
    check_pop(/^points*/i, "Award amount should be followed by 'point' or 'points'")  
    check_pop(/^if /i, "Award point(s) should be followed by if")  
  end  
  
  def check_pop(regex, message_if_no_match)  
    check(regex, message_if_no_match)  
    pop(regex)  
  end  
  
  def pop(regex)  
    text.gsub! regex, ""  
  end  
  
  def check(regex, message_if_no_match)  
    raise "You entered #{text.gsub(/^\S+.*/, "\\\1")}. #{message_if_no_match} -  
          on line #{line_number}" unless text =~ regex  
  end  
end
```

```
class Line<~  
attr_accessor :text, :line_number~  
def initialize(text, line_number)~  
  self.text = text.squeeze(" ").strip~  
  self.line_number = line_number~  
end~  
  
def self.parse_line(errors, line, index)~  
  Line.new(line, index).check_line~  
rescue Exception => ex~  
  errors << ex.message~  
end~
```

```
class Line<~  
  attr_accessor :text, :line_number  
  def initialize(text, line_number)  
    self.text = text.squeeze(" ").strip  
    self.line_number = line_number  
  end  
  
  def check_line  
    check_pop(/^award/i, "Rules must start with 'Award'")  
    check_pop(/^\d+/i, "Rules must specify an award amount (e.g. 1, 23, 100)")  
    check_pop(/^points*/i, "Award amount should be followed by 'point' or 'points'")  
    check_pop(/^if /i, "Award point(s) should be followed by if")  
  end  
  
  def check_pop(regex, message_if_no_match)  
    check(regex, message_if_no_match)  
    pop(regex)  
  end  
  
  def pop(regex)  
    text.gsub! regex, ""  
  end  
  
  def check(regex, message_if_no_match)  
    raise "You entered #{text.gsub(/^\S+.*/, "\\\1")}. #{message_if_no_match} -  
          on line #{line_number}" unless text =~ regex  
  end  
end
```

```
def self.parse_line(errors, line, index)-
  Line.new(line, index).check_line-
rescue Exception => ex-
  errors << ex.message-
end-
-
def check(regex, message_if_no_match)-
  raise "You entered #{text.gsub(/^\S+.*/, "\\\1")}. #{message_if_no_match} -
        on line #{line_number}" unless text =~ regex-
end-
end
```

```
class Line<~  
  attr_accessor :text, :line_number  
  def initialize(text, line_number)  
    self.text = text.squeeze(" ").strip  
    self.line_number = line_number  
  end  
  
  def check_line  
    check_pop(/^award/i, "Rules must start with 'Award'")  
    check_pop(/^\d+/i, "Rules must specify an award amount (e.g. 1, 23, 100)")  
    check_pop(/^points*/i, "Award amount should be followed by 'point' or 'points'")  
    check_pop(/^if /i, "Award point(s) should be followed by if")  
  end  
  
  def check_pop(regex, message_if_no_match)  
    check(regex, message_if_no_match)  
    pop(regex)  
  end  
  
  def pop(regex)  
    text.gsub! regex, ""  
  end  
  
  def check(regex, message_if_no_match)  
    raise "You entered #{text.gsub(/^\S+.*/, "\\\1")}. #{message_if_no_match} -  
          on line #{line_number}" unless text =~ regex  
  end  
end
```

```
class Line-
  def check_line-
    check_pop(/^award /i, "Rules must start with 'Award'")-
    check_pop(/^\d+ /i, "Rules must specify an award amount (e.g. 1, 23, 100)")-
    check_pop(/^points* /i, "Award amount should be followed by 'point' or 'points'")-
    check_pop(/^if /i, "Award point(s) should be followed by if")-
    check(/^(fare|flight) /i, "If should be followed by either 'fare' or 'flight'")-
    text =~ /^fare /i ? check_fare : check_flight-
    raise "You entered #{text.gsub(/^\.*/, "\\\1").strip} at -
          the end of line #{line_number}" unless text =~ /^$/-
  end-
  - def check_fare-
    pop(/^fare /i)-

    check_pop(/^class /i, "Fare should be followed by 'class'")-
    check_pop(/^is /i, "Class should be followed by 'is'")-
    check_pop(/^([A-Za-z], [A-Za-z])* /i, "Is should be followed by valid fare -
          classes list (e.g. A or A, B, C)")-
  end-
  - def check_flight-
    pop(/^flight /i)-

    check_pop(/^is /i, "Flight should be followed by 'is'")-
    check_pop(/^([to|from]) /i, "Is should be followed by either 'to' or 'from'")-
    check_pop(/^([A-Za-z][A-Za-z][A-Za-z])/i, "Please enter a valid airport code")-
  end-
end
```




Applies Where?





Author?

**Change
Frequency**



commercial
language
workbenches

commercial language workbenches

MPS - JetBrains
Intentional





80/20 Rule

Treetop or Racc



**More
complicated rules**



**Contradicting
Rules**



Maturity

Questions

qcon@jayfields.com

ThoughtWorks

<http://thoughtworks.com>

Jay Fields

<http://jayfields.com>