# How Code Gets to Prod

“an engineer’s favorite place to be”

# What is this presentation?


BRANDING
& IDENTITY

```java

int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9};

int sum;
for (int i = 0; i < numbers.length; i++) {
    sum = sum + number[i];
}

return sum;
```

## For Loop

```java

  int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

  int sum = 0;
  int i = 0;

  while (i < numbers.length) {
      sum = sum + numbers[i];
      i++;
  }
```

## While Loop

```java
int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int sum = Arrays.asList(numbers)
    .stream()
    .reduce(0, (subtotal, element) -> subtotal + element);
```

## Java Stream

# Product Manager and Designer Survey

1. On a scale from one to ten, how well do you feel you could describe what it is a software engineer at FordLabs does?
2. What does a software engineer do?
3. What are the kinds of things a software engineer knows?
4. Is there anything you about what a software engineer does that you would like to know more about?

# Software Engineer Survey

1. On a scale from one to ten, how well do you feel the other job families could describe what it is a software engineer at FordLabs does?
2. What does a software engineer do?
3. What are the kinds of things a software engineer knows?
4. Is there anything you about what a software engineer does that you you think is important for the other job families to know?

What non software engineers think

What software engineers think

software engineers think do

software engineers do

OCULI

## Set filters

Create visualizations based on the filters that you set below

**Target vehicle:**

Stallion

**Competitors:**

- Volnda F5-0
- Gavin Solstice
- Maxy Apache
- Maize Rover
- Tappa Tower
- Stadium Life-35
- Mex Vrooms
- Ubuntu Ranger
- Zamboni Ruffagus
- Coral H-621

**Time range (Model years):**

*Data available from 2000 - 2019. For best accuracy to calculate Attribute Ranking, use a minimum of 10 years' worth of data.*

2000 to 2019

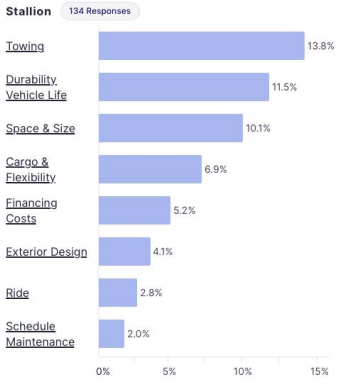
GENERATE VISUALIZATIONS

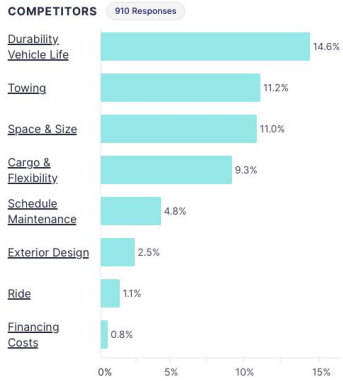
Purchase Reasons | Attribute Ranking | Satisfaction vs Purchase Reason

## Top 8 purchase reasons

Methodology

Percent of customers who listed a given attribute as their main purchase reason




## All purchase reasons

Methodology

View and sort for any attribute, seeing the difference compared to competitors

| ATTRIBUTE | STALLION: PUR. REASON % | COMPETITORS: PUR. REASON % | DIFFERENCE |
|---|---|---|---|
| Towing | 13.8% | 11.2% | 2.6% |
| Durability Vehicle Life | 11.5% | 14.6% | -3.1% |
| Space & Size | 10.1% | 11.0% | -0.9% |
| Cargo & Flexibility | 6.9% | 9.3% | -3.3% |
| Finacing Costs | 5.2% | 0.8% | 4.4% |
| Exterior Design | 4.1% | 2.5% | 1.6% |
| Ride | 2.8% | 1.1% | 1.7% |
| Schedule Maintenance | 2.0% | 4.8% | -2.8% |

OCULI

Set filters

Create visualizations
the filters that you s

Target vehicle:

Stallion

Competitors:

Volnda F5-0
Gavin Solstice
Maxy Apache
Maize Rover
Tappa Tower
Stadium Life-35
Mex Vrooms
Ubuntu Ranger
Zamboni Ruffagu
Coral H-621

Time range (Model yea

Data available from 2000 -
accuracy to calculate Attrib
a minimum of 10 years' wort

2000 to

GENERATE VISUA

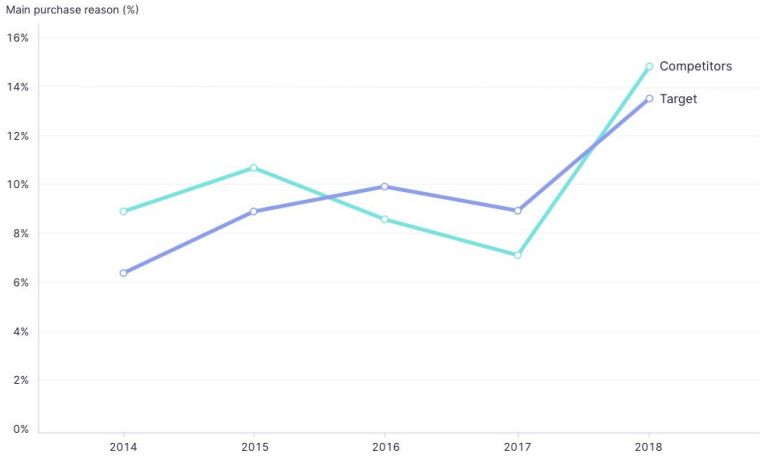
Purchase Reason over Time | Satisfaction over Time

## 'Towing' as the main purchase reason over time

Methodology

Percent of customers who listed 'Towing' as their main purchase reason from 2014 - 2018

**Competitors:** Volnda F5-0, Gavin Solstice, Maxy Apache, Maize Rover, Tappa Tower, Stadium Life-35, Mex Vrooms, Ubuntu Ranger, Zamboni Ruffagus, Coral H-621




| | | | |
|---|---|---|---|
| Towing | 13.8% | 11.2% | 2.6% |
| Durability Vehicle Life | 11.5% | 14.6% | -3.1% |
| Space & Size | 10.1% | 11.0% | -0.9% |
| Cargo & Flexibility | 6.9% | 9.3% | -3.3% |
| Financing Costs | 5.2% | 0.8% | 4.4% |
| Exterior Design | 4.1% | 2.5% | 1.6% |
| Ride | 2.8% | 1.1% | 1.7% |
| Schedule Maintenance | 2.0% | 4.8% | -2.8% |

1. Identify old code that can be reused and what new code needs to be written

- The Language
- The Framework
- The Ends

# Programming Languages

# Programming Language Defined

“a formal language, which comprises a set of instructions that produce various kinds of output”

```java
public int sum(int[] numbers) {
    int sum = 0;
    for (int i = 0; i < numbers.length; i++) {
        sum = sum + numbers[0];
    }
    return sum;
}
```

Java

```python
def sum(numbers) -> int:
    sum = 0
    for number in numbers:
        sum = sum + number
    return sum
```

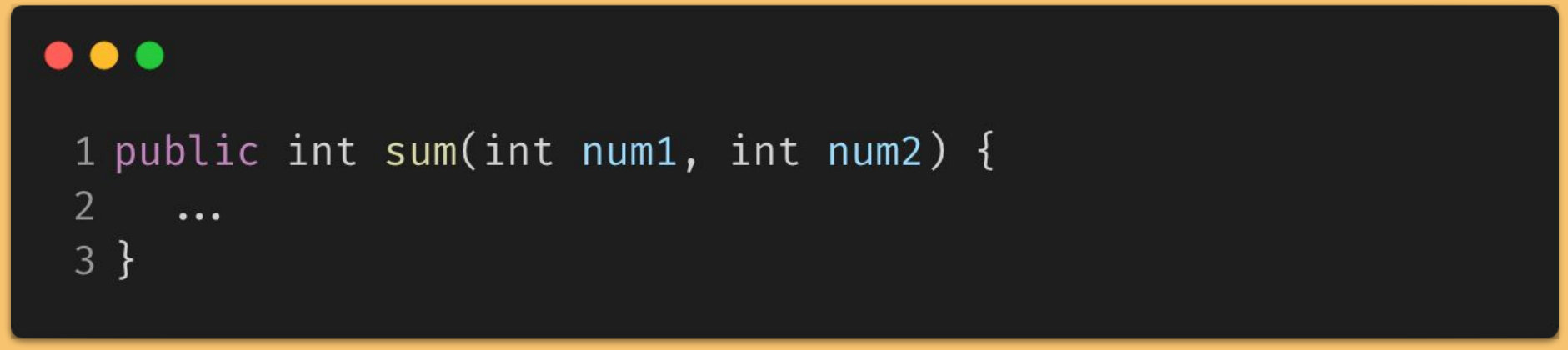
Python

```javascript
function sum(numbers) {
    let sum = 0
    for (let i = 0; i < numbers.length; i++) {
        sum = sum + numbers[i]
    }
    return sum;
}
```

JavaScript

Typed

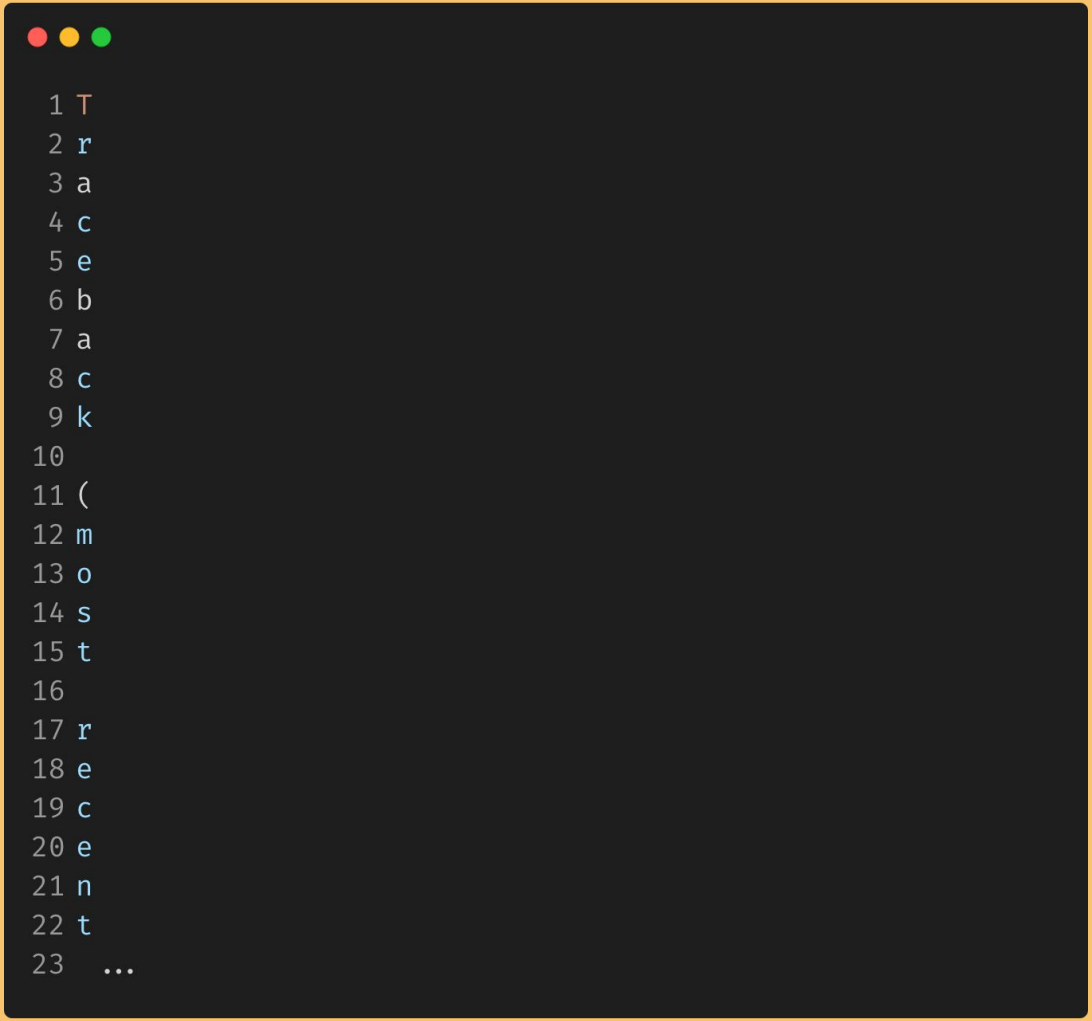
Untyped

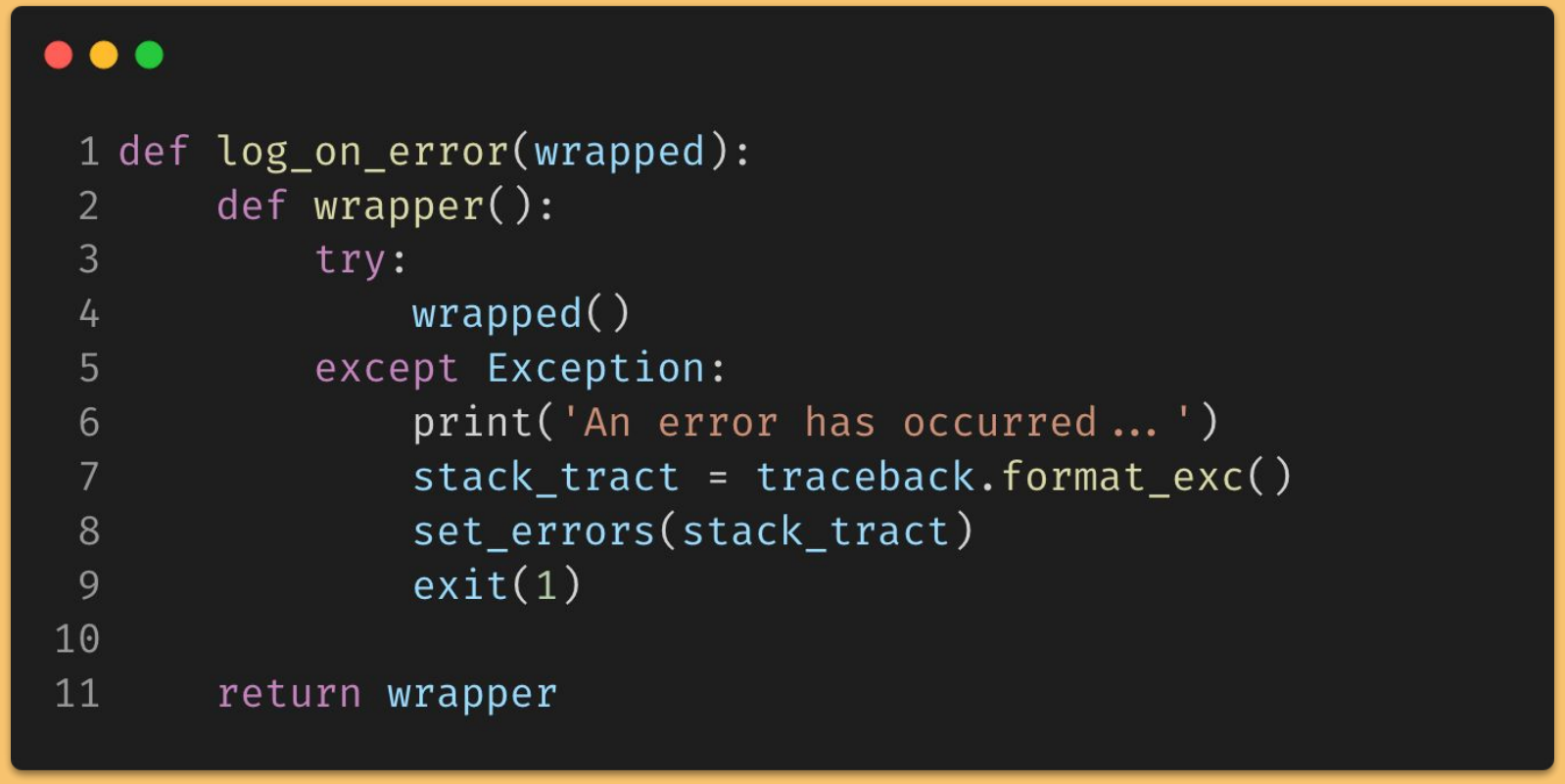


```
1 public int sum(int num1, int num2) {
2    ...
3 }
```

```
1 function sum(num1, num2) {
2    ...
3 }
```

```
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/util/errors.py", line 9, in wrapper
    wrapped()
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/main.py", line 46, in main
    command = command_factory.create(sys.argv[1:])
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/factory.py", line 22, in create
    return self._create_with_settings(args, get_config())
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/factory.py", line 29, in _create_with_settings
    command_factory: CommandFactoryMethod = self.command_builder_map[command_arg]
KeyError: 'errror'
```


T
r
a
c
e
b
a
c
k
(
m
o
s
t
r
e
c
e
n
t
...



```python
def log_on_error(wrapped):
    def wrapper():
        try:
            wrapped()
        except Exception:
            print('An error has occurred ... ')
            stack_tract = traceback.format_exc()
            set_errors(stack_tract)
            exit(1)

    return wrapper
```

```
stack_tract = traceback.format_exc()
set_errors(stack_tract)
```

```
stack_tract = traceback.format_exc()
set_errors(stack_tract)
```

```
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/util/errors.py", line 9, in wrapper
    wrapped()
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/main.py", line 46, in main
    command = command_factory.create(sys.argv[1:])
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/factory.py", line 22, in create
    return self._create_with_settings(args, get_config())
  File "/usr/local/lib/python3.7/site-packages/guet-2.3.2-py3.7.egg/guet/factory.py", line 29, in _create_with_settings
    command_factory: CommandFactoryMethod = self.command_builder_map[command_arg]
KeyError: 'errror'
```

Engineers pick a framework or a programming language for learning, access, or comfort

**Katerina Borodina**
@kathyra_

just heard someone say "at least if i die and go to hell i won't have to look at javascript again". lmao where do you think javascript came from

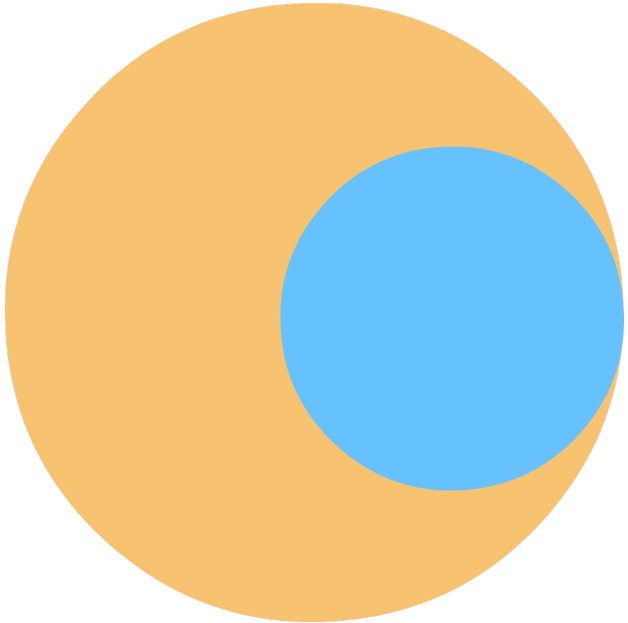
7:09 AM · Nov 8, 2019 · Twitter for Android

# Frameworks

# Software Framework

“an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software”

Framework Code



Engineer Code

Flask

# The Ends

# Front-End Defined

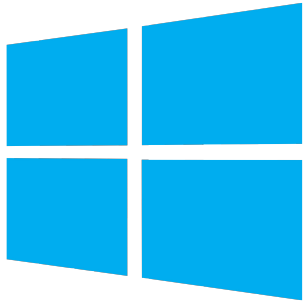
“the lowest layer at which application programmers consider data structure and presentation, instead of simply sending data in the form of datagrams or packets between hosts.”

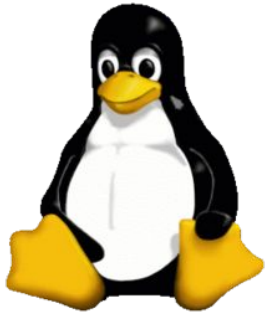
# Front-End Redefined

“The part the user has on their device.”

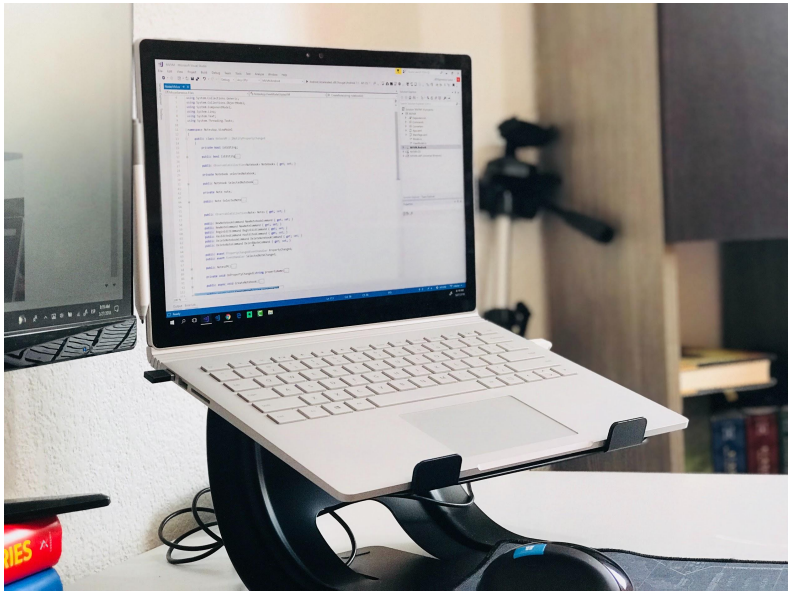
Web

Mobile

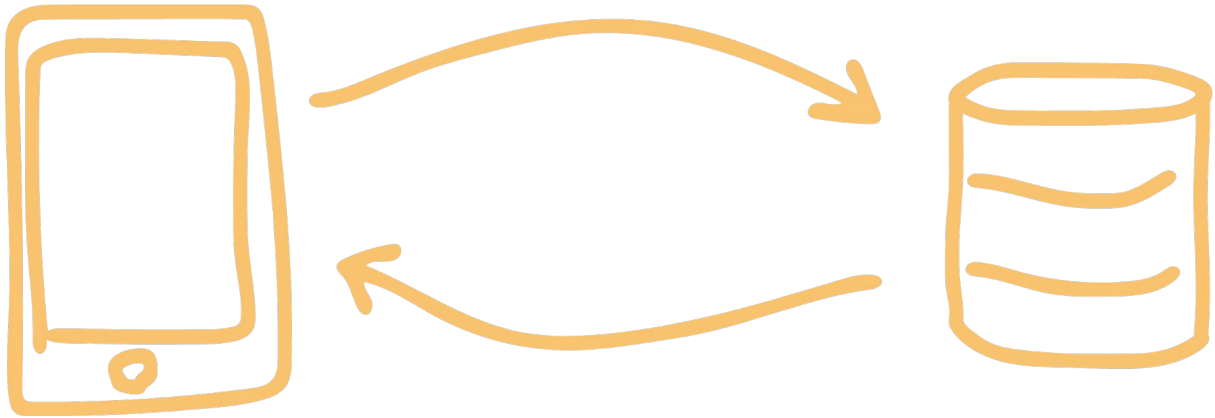




Desktop



Desktop and mobile front-ends live on users’ devices



```
<html>
  <body>
    <h1>This is a title</h1>
    <div>
      <p>This is the first paragraph of the page.</p>
      <p>This is the second paragraph of the page.</p>
    </div>
    <div>
      <h2>This is a funny picture</h2>
      <img src="./kennel.jpg" />
    </div>
  </body>
</html>

```

# This is a title

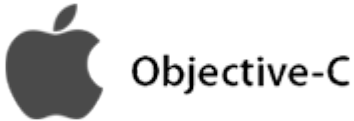
This is the first paragraph of the page.

This is the second paragraph of the page.

## This is a funny picture








Web

Mobile

# The Back-End Defined

“a layer of a computer program which provides simplified access to data stored in persistent storage of some kind”

# The Back-End Redefined

“A collection of endpoints that create, read, update, or delete data stored somewhere”

## 2.13 Endpoint

### 2.13.1 The Endpoint Component

An Endpoint component defines the particulars of a specific endpoint at which a given service is available.

Endpoint components are local to a given Service component (see **A.2 Fragment Identifiers**).

The Binding component specified by the {binding} property of an Endpoint component is said to be *applied* to the Interface component which is the value of the {interface} property of the parent Service component of the Endpoint. According to the constraints given below, if this Binding component has an {interface} property, its value must be the Interface component the Binding component is applied to.

The {address} property is optional to allow for means other than IRIs to be used, e.g. a WS-Addressing Endpoint Reference [*WSA 1.0 Core*]. It is also possible that, in certain scenarios, an address will not be required, in which case this property may be absent.

The properties of the Endpoint component are as follows:

- {name} REQUIRED. An *xs:NCName*.
- {binding} REQUIRED. A Binding component.
- {address} OPTIONAL. An *xs:anyURI*. This *xs:anyURI* MUST be an absolute IRI as defined by [*IETF RFC 3987*].† If present, the value of this attribute represents the network address at which the service indicated by the parent Service component's {interface} property is offered via the binding referred to by the {binding} property. **Note** that the presence in this property of the characters "?" and "#" can conflict with those potentially added by the query string serialization mechanism, as defined in Serialization as "application/x-www-form-urlencoded" ([*WSDL 2.0 Adjuncts*], section 6.8.2).
- {parent} REQUIRED. The Service component that contains this component in its {endpoints} property.

For each Endpoint component in the {endpoints} property of a Service component, the {name} property MUST be unique. Note that this constraint is enforced by the normative WSDL 2.0 XML schema.

For each Endpoint component in the {endpoints} property of a Service component, the {binding} property MUST either be a Binding component with an unspecified {interface} property or a Binding component with an {interface} property equal to the {interface} property of the Service component.†

### 2.13.2 XML Representation of Endpoint Component

```
<description>
  <service>
    <endpoint
          name="xs:NCName"
          binding="xs:QName"
          address="xs:anyURI"? >
      <documentation />*
    </endpoint>+
  </service>
</description>
```

“

Hours of trial and error can save you minutes of reading the documentation.

**Someone on the internet, probably**

”

# Endpoint Redefined

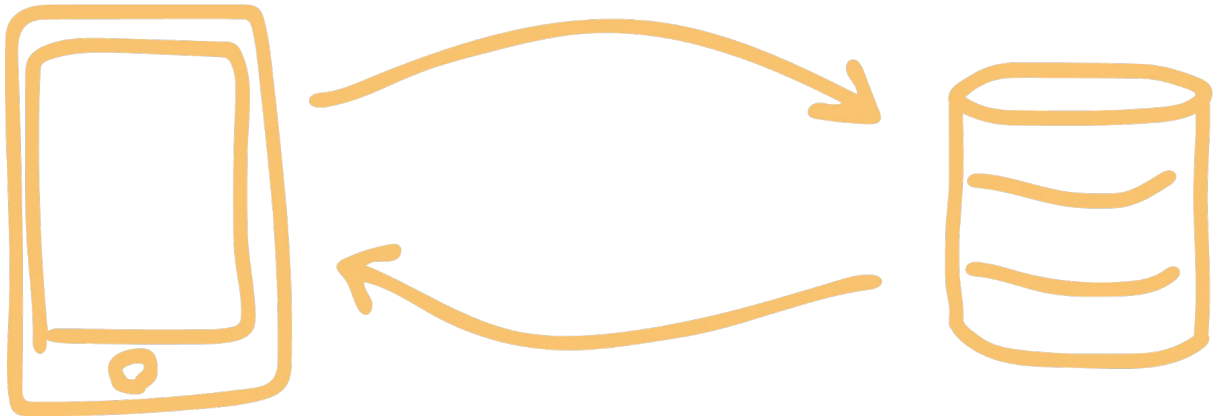
“The entry place to particular services in your backend application”

# Back-End Endpoints

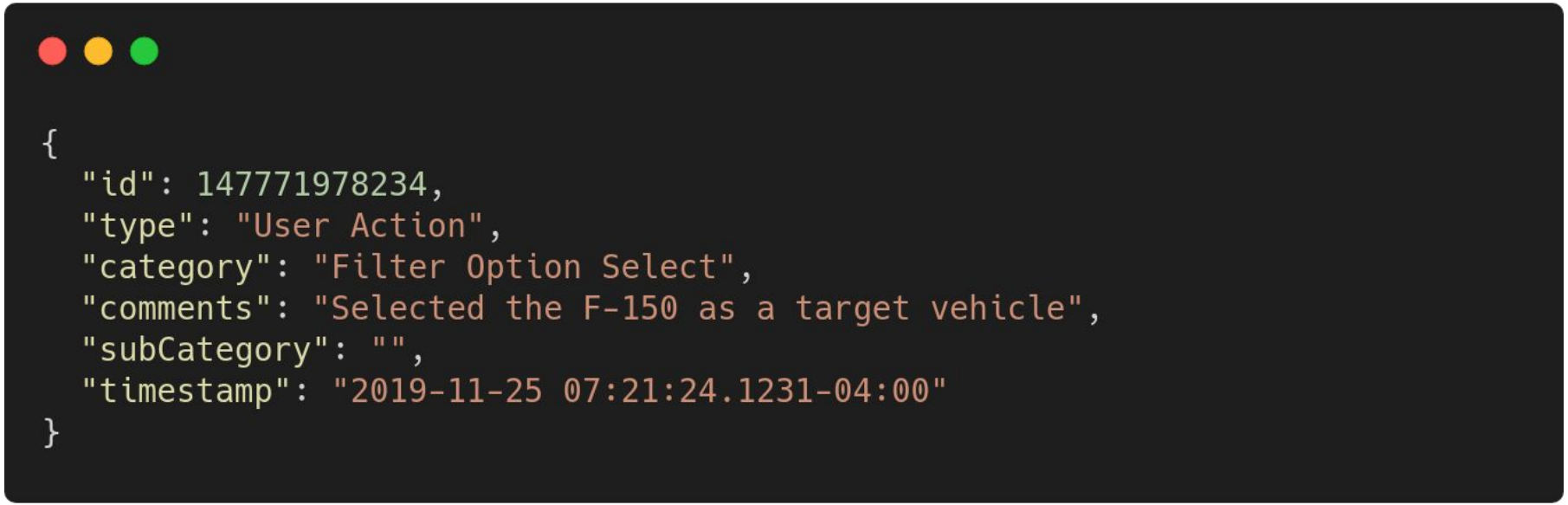
```
GET      /api/nameplates

GET      /api/purchasereason/main?nameplates=STALLION
                                  &startYear=2010
                                  &endYear=2019
POST     /api/events

GET      /api/events
```



# JavaScript Object Notation (JSON)



```
{
  "id": 147771978234,
  "type": "User Action",
  "category": "Filter Option Select",
  "comments": "Selected the F-150 as a target vehicle",
  "subCategory": "",
  "timestamp": "2019-11-25 07:21:24.1231-04:00"
}
```

```
{
  "type": "User Action",
  "category": "Filter Option Select",
  "comments": "Selected the F-150 as a target vehicle"
}
```

POST

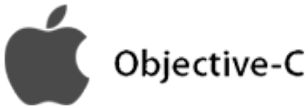
```
{
  "id": 147771978234,
  "type": "User Action",
  "category": "Filter Option Select",
  "comments": "Selected the F-150 as a target vehicle",
  "subCategory": "",
  "timestamp": "2019-11-25 07:21:24.1231-04:00"
}
```

GET


TS

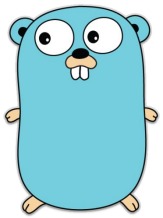




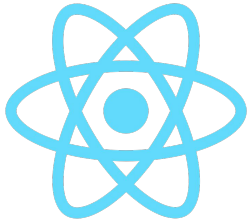
JS

Objective-C


C#

C++

php


1. Identify old code that can be reused and what new code needs to be written



Flask


Front-End

Backend-End

OCULI

## Set filters

Create visualizations based on the filters that you set below

**Target vehicle:**

Stallion

**Competitors:**

- Volnda F5-0
- Gavin Solstice
- Maxy Apache
- Maize Rover
- Tappa Tower
- Stadium Life-35
- Mex Vrooms
- Ubuntu Ranger
- Zamboni Ruffagus
- Coral H-621

**Time range (Model years):**

*Data available from 2000 - 2019. For best accuracy to calculate Attribute Ranking, use a minimum of 10 years' worth of data.*

2000 to 2019

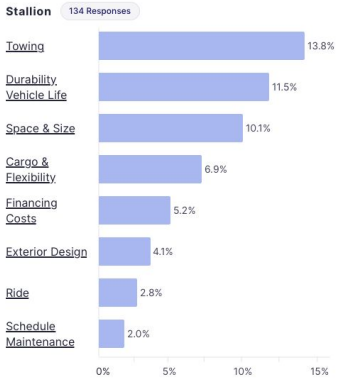
GENERATE VISUALIZATIONS

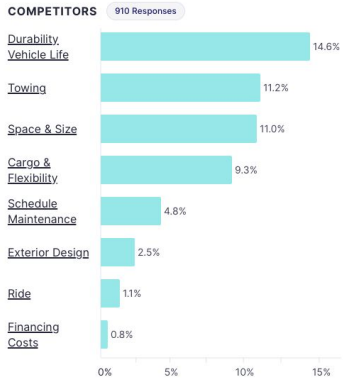
Purchase Reasons | Attribute Ranking | Satisfaction vs Purchase Reason

## Top 8 purchase reasons

Methodology

Percent of customers who listed a given attribute as their main purchase reason




## All purchase reasons

Methodology

View and sort for any attribute, seeing the difference compared to competitors

| ATTRIBUTE | STALLION: PUR. REASON % | COMPETITORS: PUR. REASON % | DIFFERENCE |
|---|---|---|---|
| Towing | 13.8% | 11.2% | 2.6% |
| Durability Vehicle Life | 11.5% | 14.6% | -3.1% |
| Space & Size | 10.1% | 11.0% | -0.9% |
| Cargo & Flexibility | 6.9% | 9.3% | -3.3% |
| Finacing Costs | 5.2% | 0.8% | 4.4% |
| Exterior Design | 4.1% | 2.5% | 1.6% |
| Ride | 2.8% | 1.1% | 1.7% |
| Schedule Maintenance | 2.0% | 4.8% | -2.8% |

```
/api/purchasereason/ranking?startYear=2010
                           &endYear=2019
                           &target=STALLION
                           &competitors=UBUNTU_RANGER
                           &competitors=ZAMBONI_RUFFAGUS
```

```
[
    {
        "competitorPercentage": 229,
        "targetPercentage": .331,
        "attribute": "fun_to_drive"
    }
    ...
]
```

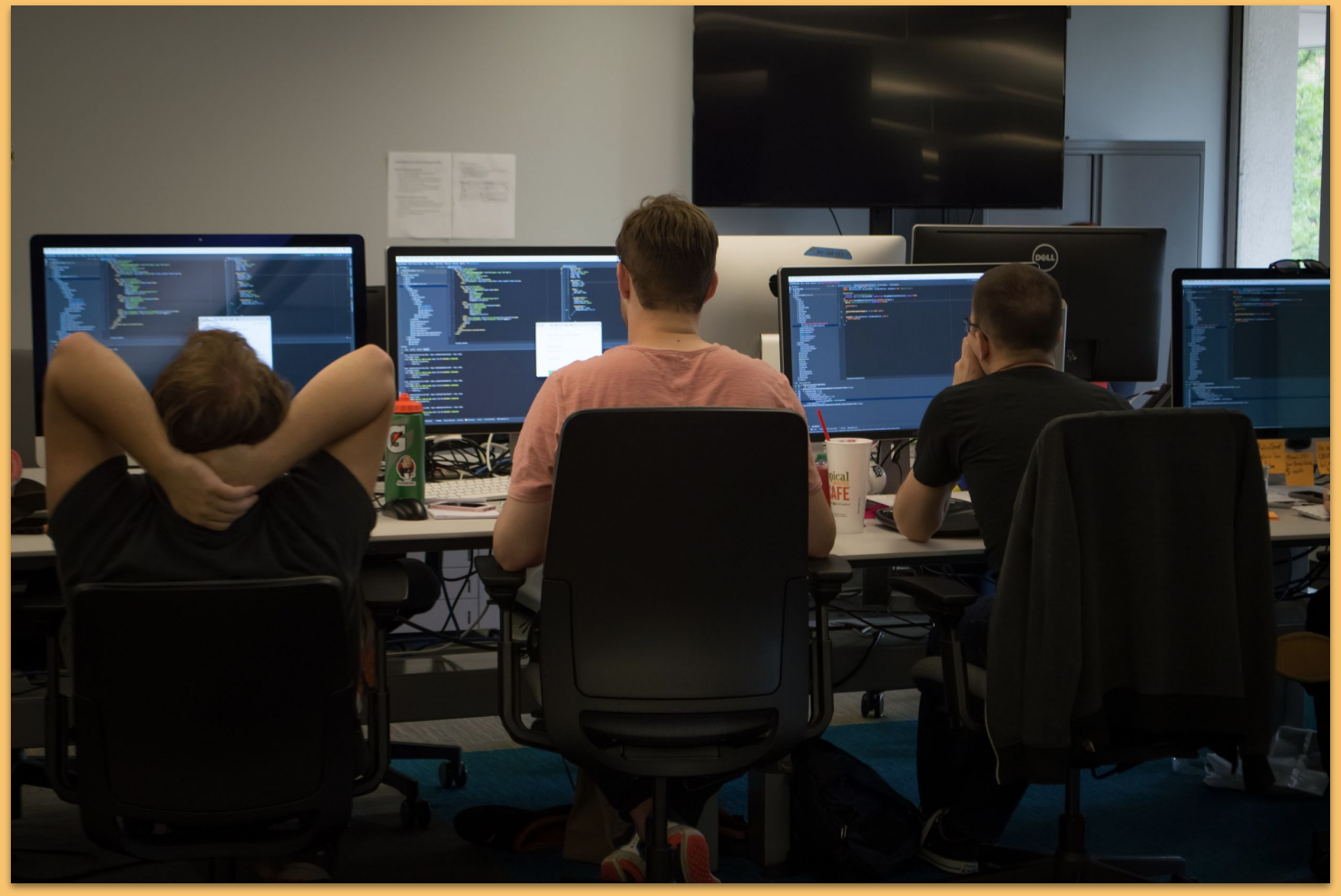
2. Think about how the new code should work

/api/purchasereason/main?nameplates=STALLION
&nameplates=ZAMBONI_RUFFAGUS
&startModelYear=2018
&endModelYear=2019
&collapse=true

```
[
  {
    "modelYears": [
      2018
    ],
    "nameplates": [
      "STALLION"
    ],
    "values": [
      {
        "value": 204,
        "name": "fun_to_drive"
      },
      ...
    ]
  },
  ...
]
```

3. Do the implementation

# Paired Programming



# +15%

Cost of Development

# -15%

Bugs Present

*The Costs and Benefits of Pair Programming*. Alistair Cockburn and Laurie Williams

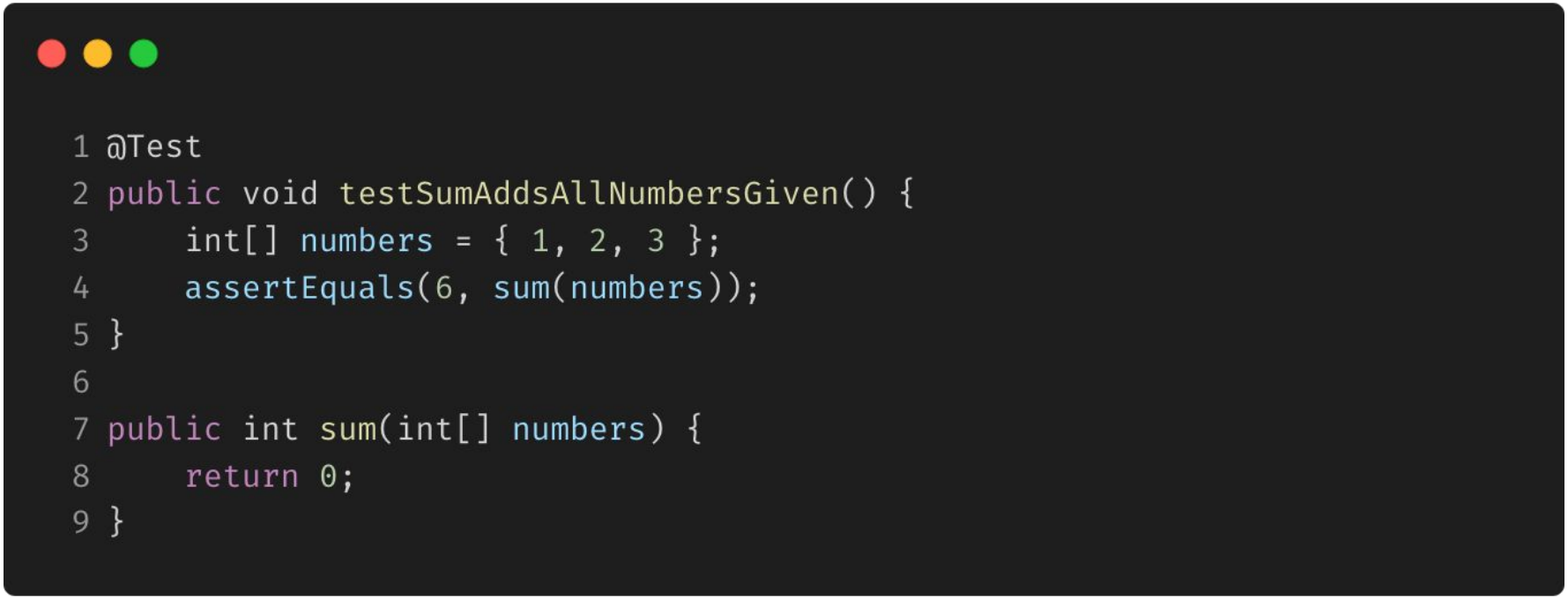
# Test-Driven Development

Red

Green

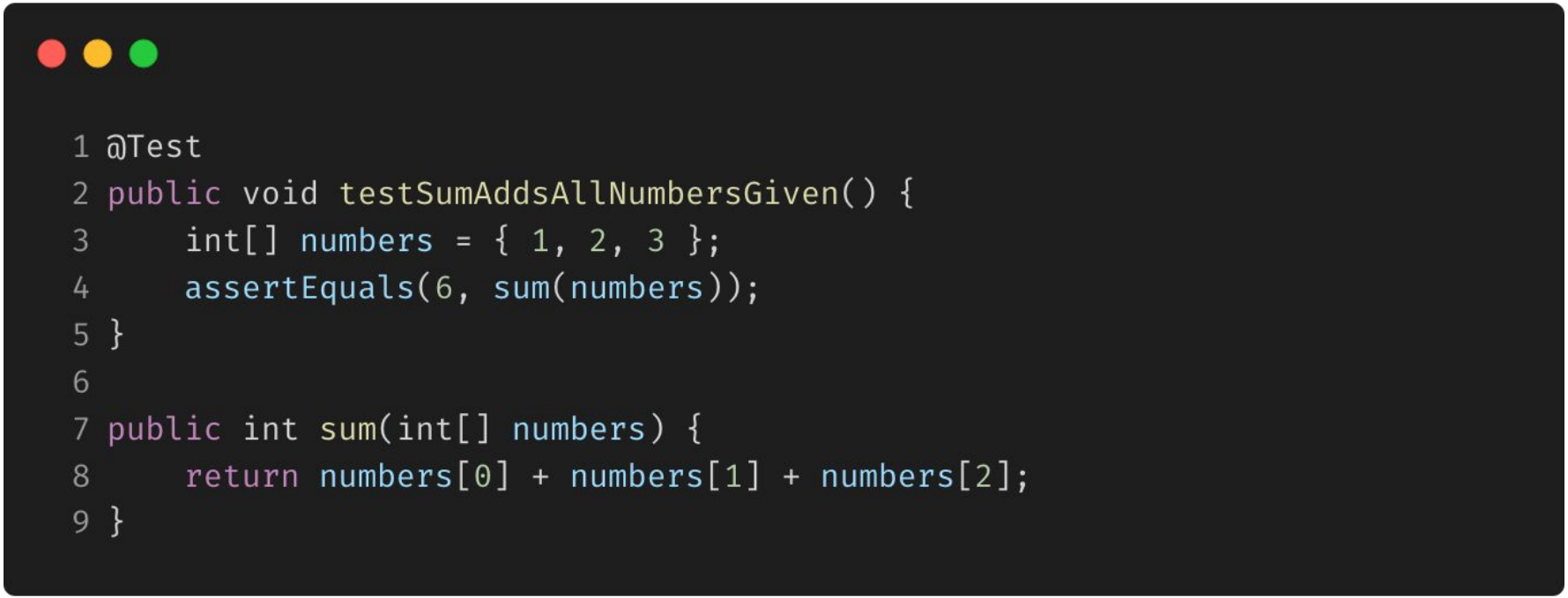
Refactor

# Red



```
1 @Test
2 public void testSumAddsAllNumbersGiven() {
3     int[] numbers = { 1, 2, 3 };
4     assertEquals(6, sum(numbers));
5 }
6
7 public int sum(int[] numbers) {
8     return 0;
9 }
```

# Green

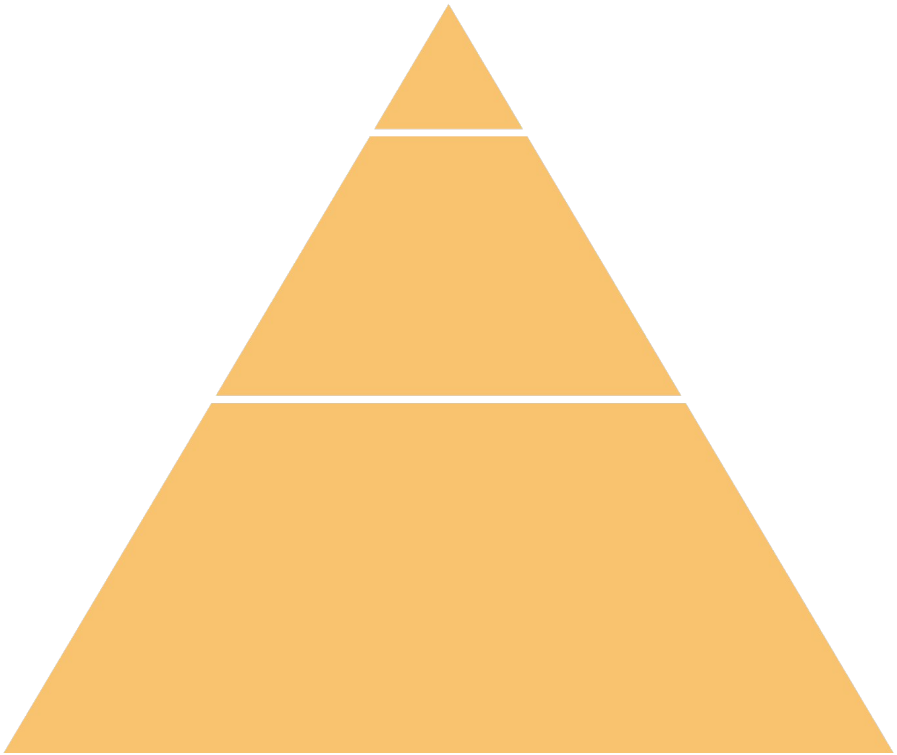


```
@Test
public void testSumAddsAllNumbersGiven() {
    int[] numbers = { 1, 2, 3 };
    assertEquals(6, sum(numbers));
}

public int sum(int[] numbers) {
    return numbers[0] + numbers[1] + numbers[2];
}
```

# Refactor

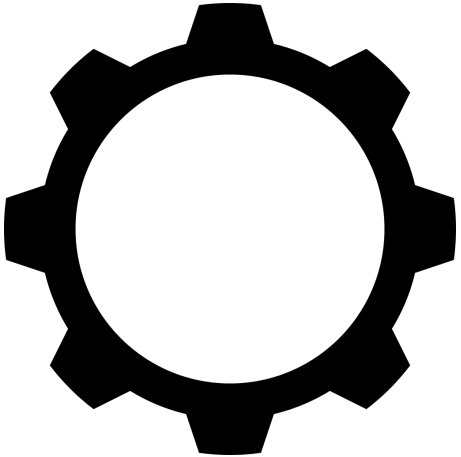
```
1 @Test
2 public void testSumAddsAllNumbersGiven() {
3     int[] numbers = { 1, 2, 3 };
4     assertEquals(6, sum(numbers));
5 }
6
7 public int sum(int[] numbers) {
8     int sum = 0;
9     for (int i = 0; i < numbers.length; i++) {
10        sum = sum + numbers[i]
11    }
12    return sum;
13 }
```



End-to-end tests

Integration tests

Unit tests



Reduced Complexity

Explicit Documentation

Regression Safety

“

“If it’s too hard to think of how to test an interface, your interface is too complex”

”

```java
/**
 * Calculate the summation of a given list of integers
 * @param numbers values to be summed
 * @return the summation of the given numbers
 */
public int sum(int[] numbers) {
    int sum = 0;
    for (int i = 0; i < numbers.length; i++) {
        sum = sum + numbers[i]
    }
    return sum;
}
```

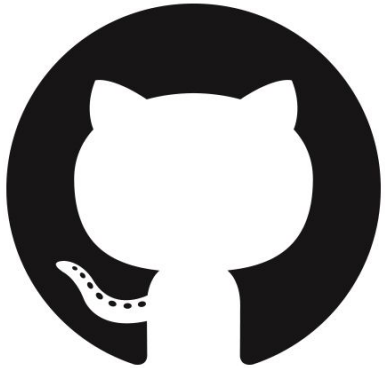
```java
/**
 * Calculate the summation of a given list of integers
 * @param numbers values to be summed
 * @return the summation of the given numbers
 */
public int sum(int[] numbers) {
    int sum = 0;
    for (int i = 0; i < numbers.length; i++) {
        if (numbers[i] > 0) {
            sum = sum + numbers[i]
        }
    }
    return sum;
}
```

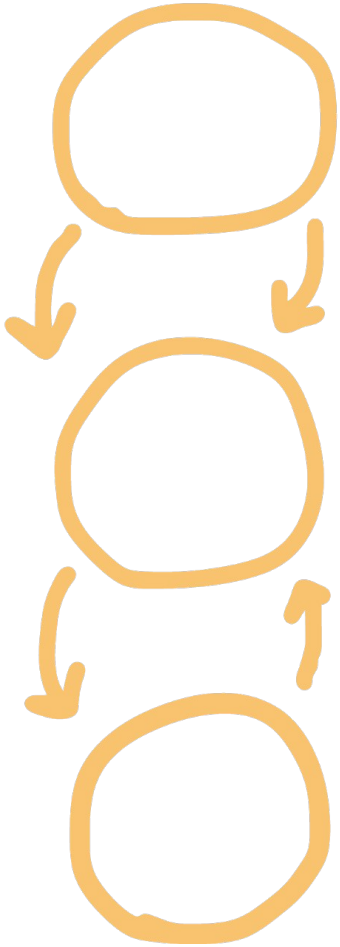
13 e2e/test_errors.py

Viewed

```
@@ -0,0 +1,13 @@
+ from e2e import DockerTest
+
+
+ class TestError(DockerTest):
+
+     def test_uncaught_errors_are_written_to_log_file(self):
+         self.guet_init()
+         self.add_command('guet notacommand')
+         self.save_file_content('.guet/errors')
+
+         self.execute()
+
+         self.assertEqual(self.get_file_text('.guet/errors')[0], 'Traceback (most recent call last):')
```

# Committing Code



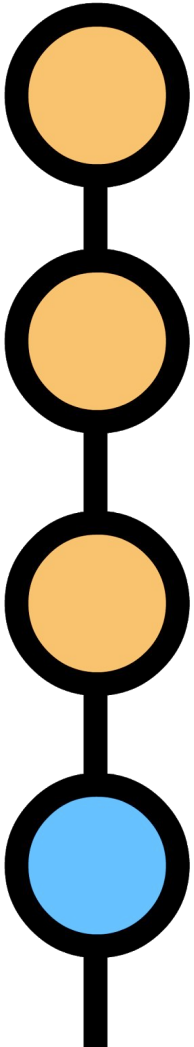


Remote Repository

Local Repository

Working Changes

# It’s a best practice to commit early and often



# Continuous Integration & Continuous Deployment

- Back to Dashboard
- **Status**
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Rename
- Pipeline Syntax
- Git Polling Log

**Build History** trend

find x

- #42 Nov 26, 2019, 8:16 AM
- #41 Nov 22, 2019, 11:04 AM
- #40 Nov 22, 2019, 9:19 AM
- #39 Nov 21, 2019, 4:19 PM
- #38 Nov 21, 2019, 3:59 PM
- #37 Nov 20, 2019, 4:42 PM
- #36 Nov 20, 2019, 4:30 PM
- #35 Nov 20, 2019, 3:48 PM
- #34 Nov 20, 2019, 3:46 PM
- #33 Nov 20, 2019, 3:44 PM
- #32 Nov 20, 2019, 2:55 PM
- #31 Nov 20, 2019, 10:33 AM
- #30 Nov 19, 2019, 4:56 PM
- #29 Nov 18, 2019, 11:01 AM

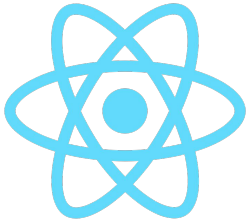
# Pipeline dev

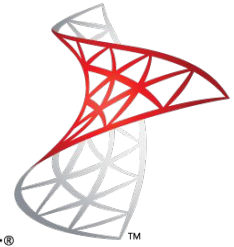
Recent Changes

## Stage View

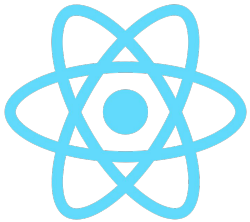
| | Declarative: Checkout SCM | Lint | Test | Build | Deploy |
|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~4min 17s) | 581ms | 33s | 12s | 9s | 2min 56s |
| #42 Nov 26 08:16 — No Changes | 526ms | | | | |
| | 3min 52s | | | | |
| #41 Nov 22 11:04 — 1 commit | 561ms | 35s | 14s | 8s | 3min 45s |
| #40 Nov 22 09:19 — 1 commit | 557ms | 33s | 13s | 9s | 5min 12s |
| #39 Nov 21 16:19 — 1 commit | 550ms | 37s | 13s | 8s | 3min 48s |
| #38 Nov 21 15:59 — 1 commit | 648ms | 33s | 14s | 9s | 2min 22s |

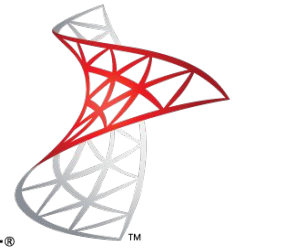
“As the general stress level rises, manual builds tend to be done less often and less well, resulting in more errors and more stress.”




Development




Production

# 4. Go through review

# Design Review

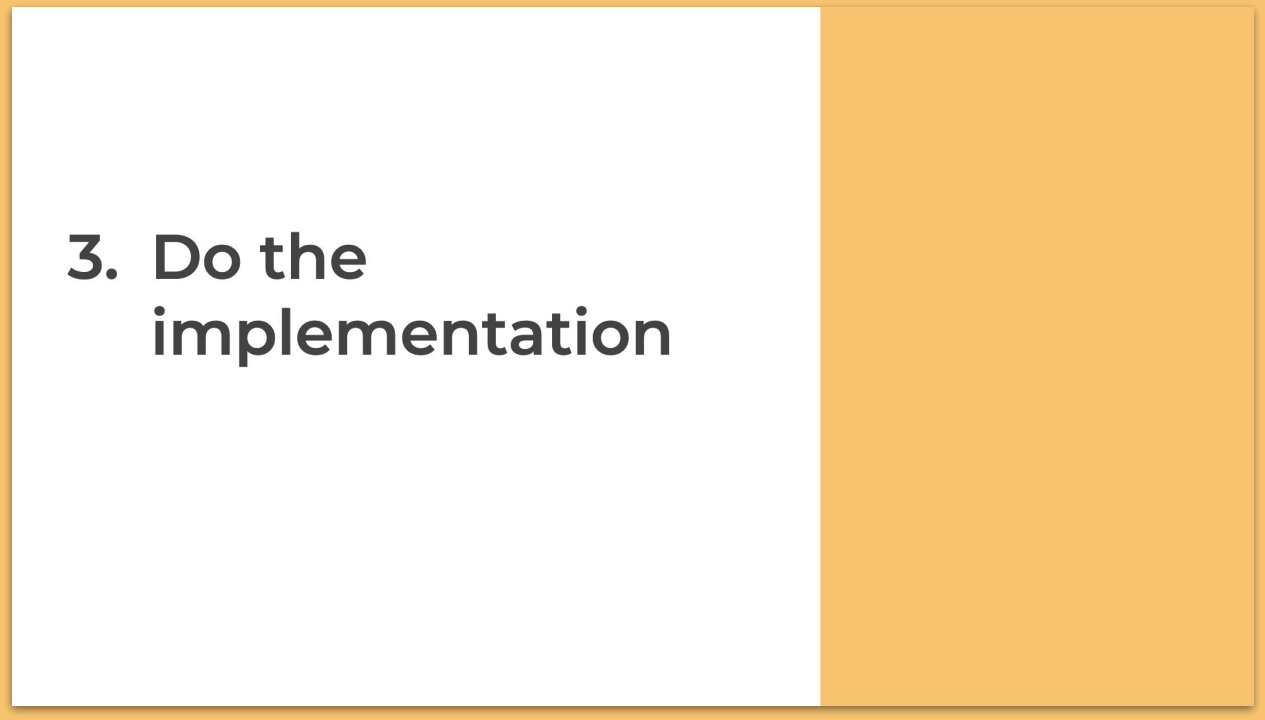
Gets the OK from Justin

Justin Finds a Problem

We didn’t understand the story

We can’t do the story as described

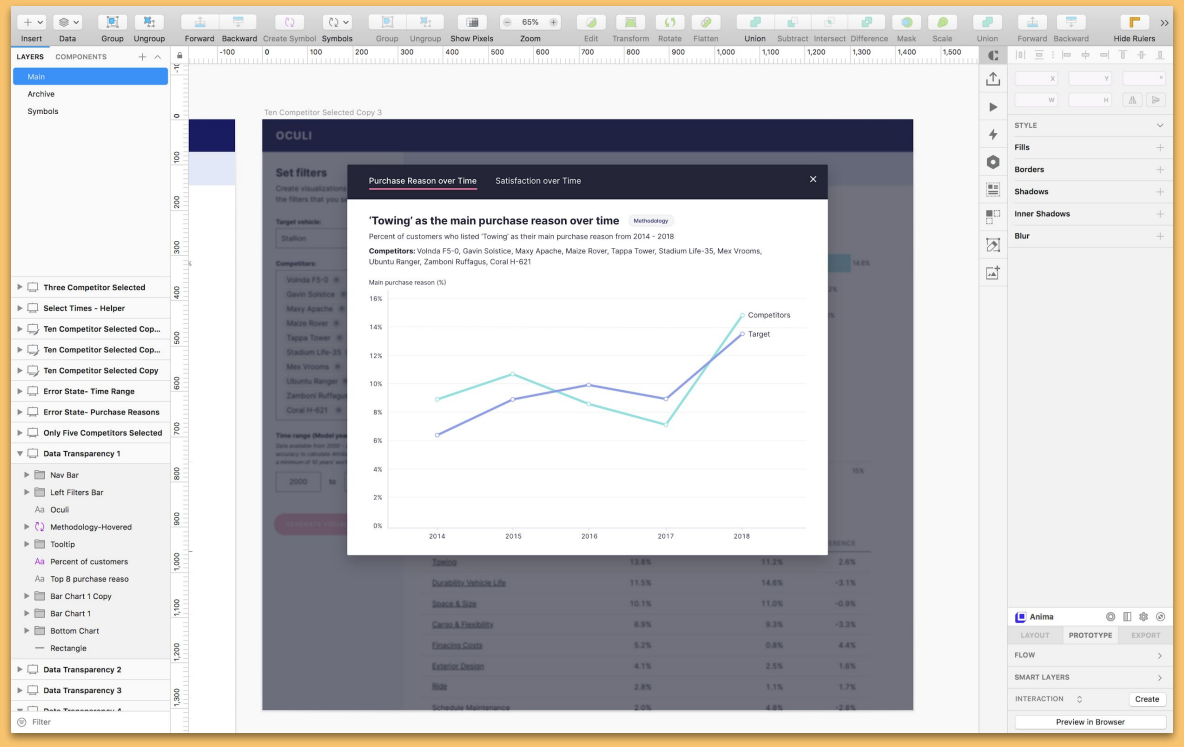
1. It’s trivial to fix, so we go back and do it



2. It’s a non-negotiable requirement to the story, so we go back and implement it

3. Do the implementation

# 3. It wasn’t what we wanted to begin with, so we change the design



OCULI

## Set filters

Create visualizations based on the filters that you set below

**Target vehicle:**

Stallion

**Competitors:**

- Volnda F5-0
- Gavin Solstice
- Maxy Apache
- Maize Rover
- Tappa Tower
- Stadium Life-35
- Mex Vrooms
- Ubuntu Ranger
- Zamboni Ruffagus
- Coral H-621

**Time range (Model years):**

*Data available from 2000 - 2019. For best accuracy to calculate Attribute Ranking, use a minimum of 10 years' worth of data.*

2000 to 2019

GENERATE VISUALIZATIONS

Purchase Reasons | Attribute Ranking | **Satisfaction vs Purchase Reason**

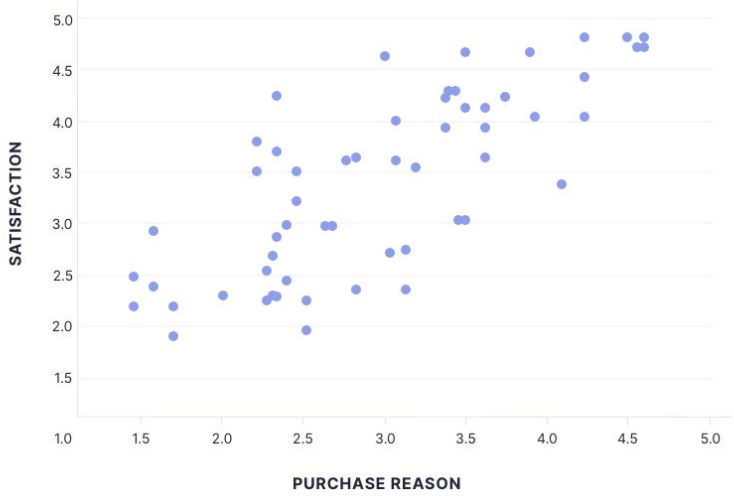
## Comparing satisfaction vs purchase reason data

Methodology

These charts show satisfaction and purchase reason on a scale of 1 (worst) to 5 (best) for each attribute from the selected target vehicle and year.

**Data for:** 2017 Stallion

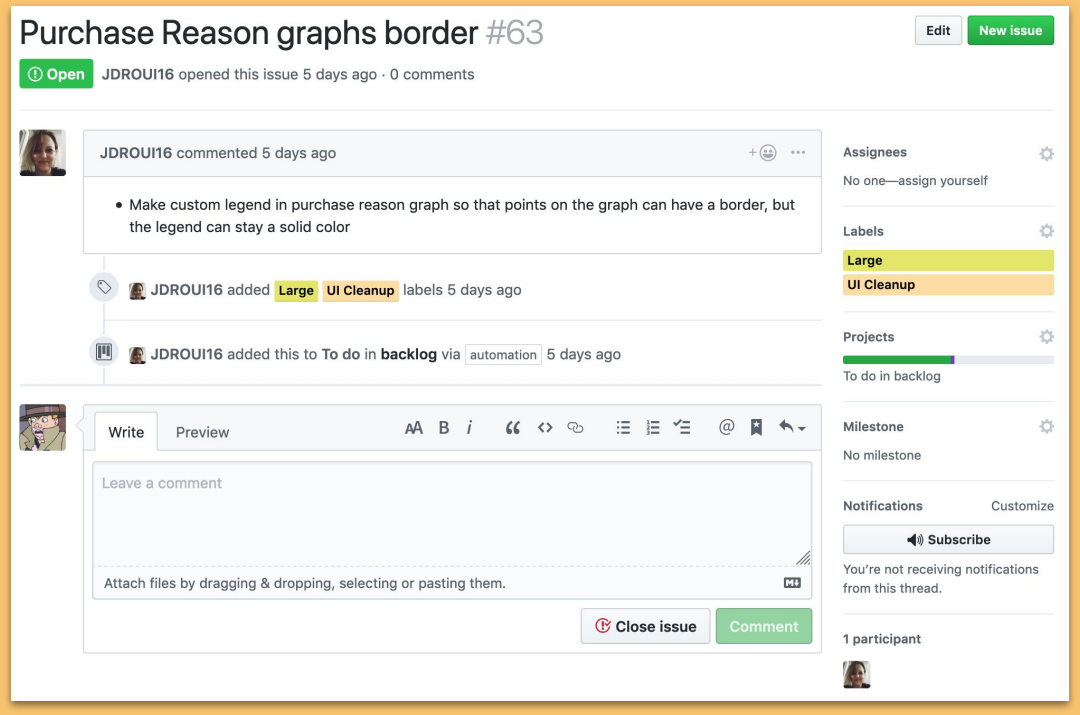
*Based on your filters from the left panel.*




**HIGH SATISFACTION/HIGH PURCHASE REASON**


**LOW SATISFACTION/HIGH PURCHASE REASON**

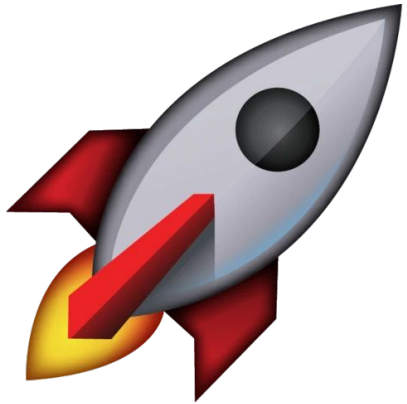
# 4. We take the feature as is, and make new stories to include the missing pieces



# PM Review

# PM Review with Jenai is exactly like Design Review, but with more roasting

# 5. Take the Rocket Ship to Prod



- Back to Dashboard
- **Status**
- Changes
- Build Now
- Delete Pipeline
- Configure
- Full Stage View
- Rename
- Pipeline Syntax
- Git Polling Log

**Build History** trend

find x

- #42 Nov 26, 2019, 8:16 AM
- #41 Nov 22, 2019, 11:04 AM
- #40 Nov 22, 2019, 9:19 AM
- #39 Nov 21, 2019, 4:19 PM
- #38 Nov 21, 2019, 3:59 PM
- #37 Nov 20, 2019, 4:42 PM
- #36 Nov 20, 2019, 4:30 PM
- #35 Nov 20, 2019, 3:48 PM
- #34 Nov 20, 2019, 3:46 PM
- #33 Nov 20, 2019, 3:44 PM
- #32 Nov 20, 2019, 2:55 PM
- #31 Nov 20, 2019, 10:33 AM
- #30 Nov 19, 2019, 4:56 PM
- #29 Nov 18, 2019, 11:01 AM

# Pipeline dev

Recent Changes

## Stage View

| | Declarative: Checkout SCM | Lint | Test | Build | Deploy |
|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~4min 17s) | 581ms | 33s | 12s | 9s | 2min 56s |
| #42 Nov 26 08:16 — No Changes | 526ms | | | | |
| | 3min 52s | | | | |
| #41 Nov 22 11:04 — 1 commit | 561ms | 35s | 14s | 8s | 3min 45s |
| #40 Nov 22 09:19 — 1 commit | 557ms | 33s | 13s | 9s | 5min 12s |
| #39 Nov 21 16:19 — 1 commit | 550ms | 37s | 13s | 8s | 3min 48s |
| #38 Nov 21 15:59 — 1 commit | 648ms | 33s | 14s | 9s | 2min 22s |

# Questions?

# Thank you for attending. Goodbye.