

Stop Testing.

Start Storytelling.

Mike Schutte

RailsConf 2018

1 / 30

@tmikeschu

Time

5

Topic

About Me

1

Goals

4

Testing Paradigms

20

Storytelling

8

Demo

2

Wrap-up

2/30

@tmikeschu

```
[5,1,4,20,8,2].reduce(0, &:+) # => 40
```

3/30

@tmikeschu

```
[5,1,4,20,8,2].reduce(0, &:+) # => 40
```

```
[5,1,4,20,8,2].sum # => 40
```

4/30

@tmikeschu

About Me

Detroit

Quikly (Rails, React, Node.js, Go)

environmental science -> sociology -> software development
University of Denver -> Indiana University -> Turing -> Quikly

runner, musician, reader, writer

5/30

@tmikeschu

Goals

6/30

@tmikeschu

Goals perspective

7/30

@tmikeschu

Goals
perspective
confidence

8/30

@tmikeschu

Testing Paradigms

bonus word: *para-dig-MAT-ic (adj)*

¹ <http://www.dictionary.com/browse/paradigm>

Testing Paradigms

bonus word: *para-dig-MAT-ic (adj)*

- a framework containing the basic assumptions, ways of thinking, and methodolog[ies] [for discovering new ideas] that are commonly accepted by members of a scientific community.¹

¹ <http://www.dictionary.com/browse/paradigm>

Test last

10/30

@tmikeschu

Test last

- The thing that's supposed to happen is the the thing that happens.

10/30

@tmikeschu

Test last

- The thing that's supposed to happen is the the thing that happens.
- ~> It's blue because it's blue 🤔.

10/30

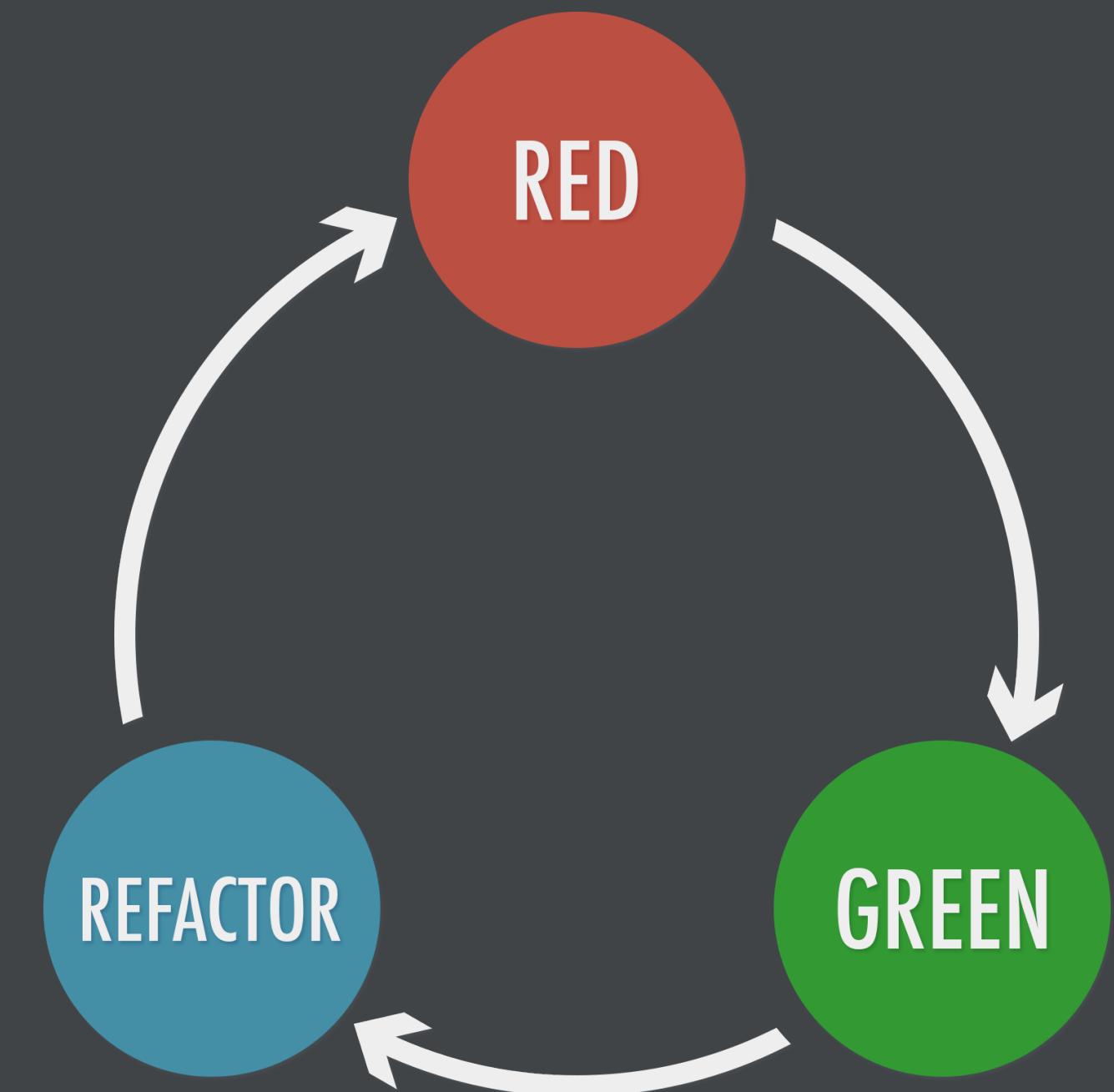
@tmikeschu

Test last



sheepfilms.co.uk

11/30
@tmikeschu



12 / 30

@tmikeschu

Test first

13 / 30

@tmikeschu

Test first

- Writing a test to fit an implementation

13 / 30

@tmikeschu

Test driven

Behavior driven

14 /30

@tmikeschu

Test driven Behavior driven

- Implementing code to fit a test

14 / 30

@tmikeschu

Test driven

Behavior driven

- Implementing code to fit a test
- *'Program to an interface, not an implementation'*
 - GOF

Storytelling and Software

*Software is **always** means to an end*

15 / 30

@tmikeschu

Storytelling and Software

*Software is **always** means to an end*

- Function to Feature

Storytelling and Software

*Software is **always** means to an end*

- Function to Feature
- (HINT: the end is to help users.)

Storytelling and Software

(minutes)

16 / 30

@tmikeschu

Storytelling and Software

(minutes)

- communication (1)

16 / 30

@tmikeschu

Storytelling and Software

(minutes)

- communication (1)
- context (3)

16 / 30

@tmikeschu

Storytelling and Software

(minutes)

- communication (1)
- context (3)
- abstraction (1)

16 / 30

@tmikeschu

Storytelling and Software

(minutes)

- communication (1)
- context (3)
- abstraction (1)
- encapsulation (3)

16 / 30

@tmikeschu

Communication

Users communicate with software.

Let's make good listeners.

17 /30

@tmikeschu

Context

18/30

@tmikeschu

Context

- :cause && :effect

18/30

@tmikeschu

Context

- `:cause && :effect`
- More context = more understanding

18/30

@tmikeschu

Context

- **:cause && :effect**
- More context = more understanding
- More understanding = more predictive power

18/30

@tmikeschu

Context

- **:cause && :effect**
- More context = more understanding
- More understanding = more predictive power
- More predictive power = more confident decision-making

Context

- :cause && :effect
- More context = more understanding
- More understanding = more predictive power
- More predictive power = more confident decision-making
- More confident decision-making = more happiness!

18/30

@tmikeschu

Context

When we tell stories about our software, we write better software.

19 / 30

@tmikeschu

Abstraction

20 / 30

@tmikeschu

Abstraction

- Touching a stove

20 / 30

@tmikeschu

Abstraction

- Touching a stove
- Breathing

20 / 30

@tmikeschu

Abstraction

- Touching a stove
- Breathing
- Muscle memory

20 / 30

@tmikeschu

Abstraction

- Touching a stove
- Breathing
- Muscle memory
- Literally every language

20 / 30

@tmikeschu

Encapsulation

We encapsulate our code, why not our thoughts?

21 / 30

@tmikeschu

Encapsulation

We encapsulate our code, why not our thoughts?

- Too many jobs makes you bad at all of them.

Encapsulation

**Code constrains clarity and
imagination.**

22 / 30

@tmikeschu

Encapsulation

Code constrains clarity and imagination.

- Tests can be your compass home

22 / 30

@tmikeschu

Encapsulation

Meta alert!

23 / 30

@tmikeschu

Encapsulation

Meta alert!

- Programs solve problems

23 / 30

@tmikeschu

Encapsulation

Meta alert!

- Programs solve problems
- Programming is a problem-solving process

23 / 30

@tmikeschu

Encapsulation

Meta alert!

- Programs solve problems
- Programming is a problem-solving process
- Maybe the principles that make better *programs* can also result in better *programming*?

23 / 30

@tmikeschu

User <-----> Programmer

24 / 30

@tmikeschu

Implementing

User <-----X-----> Programmer

25 / 30

@tmikeschu

Storytelling/testing

User <X-----> Programmer

26 / 30

@tmikeschu

Crowdsource Demo!



27 /30

@tmikeschu

Crowdsource Demo!



- What's something you have to implement in the near future? Could be anything from function to feature.

27 /30

@tmikeschu

Wrap-up

28/30

@tmikeschu

Wrap-up

- Testing is a *paradigmatic* process, not a product.

28/30

@tmikeschu

Wrap-up

- Testing is a *paradigmatic* process, not a product.
- Communication: build good listeners

28/30

@tmikeschu

Wrap-up

- Testing is a *paradigmatic* process, not a product.
- Communication: build good listeners
- Context: enable confident decision making

28/30

@tmikeschu

Wrap-up

- Testing is a *paradigmatic* process, not a product.
- Communication: build good listeners
- Context: enable confident decision making
- Abstraction: you're good at it, it let's you do cool stuff.

28/30

@tmikeschu

Wrap-up

- Testing is a *paradigmatic* process, not a product.
- Communication: build good listeners
- Context: enable confident decision making
- Abstraction: you're good at it, it let's you do cool stuff.
- Encapsulation: apply the same discipline to your process as you do to your code.

28/30

@tmikeschu

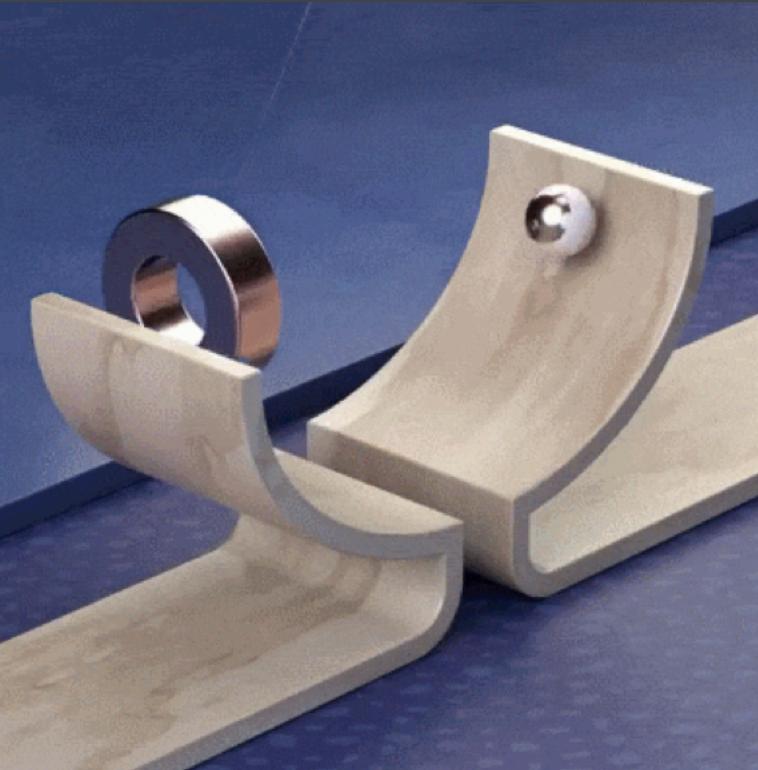
Wrap-up

- Testing is a *paradigmatic* process, not a product.
- Communication: build good listeners
- Context: enable confident decision making
- Abstraction: you're good at it, it let's you do cool stuff.
- Encapsulation: apply the same discipline to your process as you do to your code.
- Storytelling keeps you closer to users

28/30

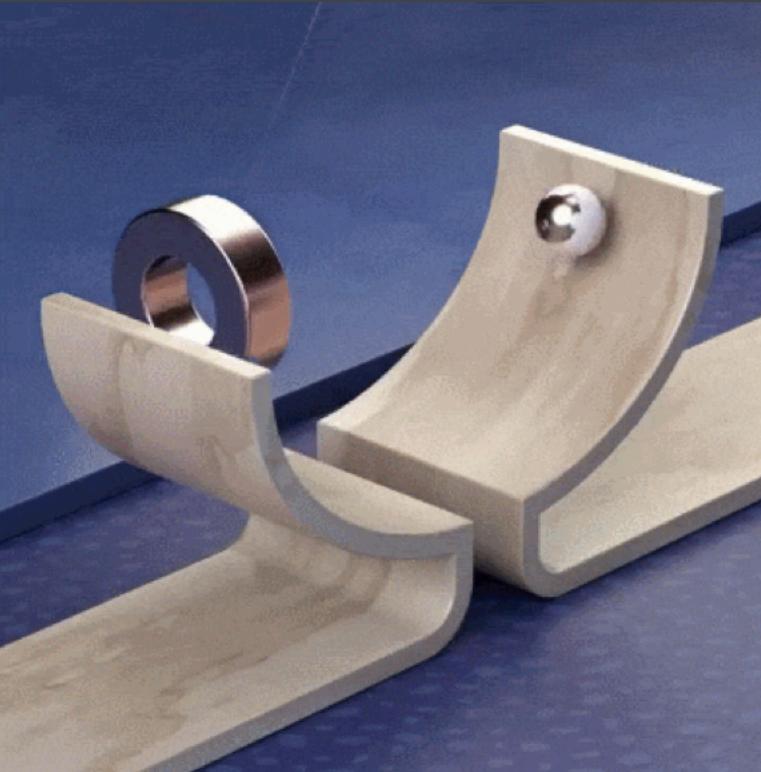
@tmikeschu

Action Plan



29 / 30
@tmikeschu

Action Plan



- Develop a pendulum workflow between storyteller and developer.

29 / 30

@tmikeschu

thank you!

30 / 30

@tmikeschu