

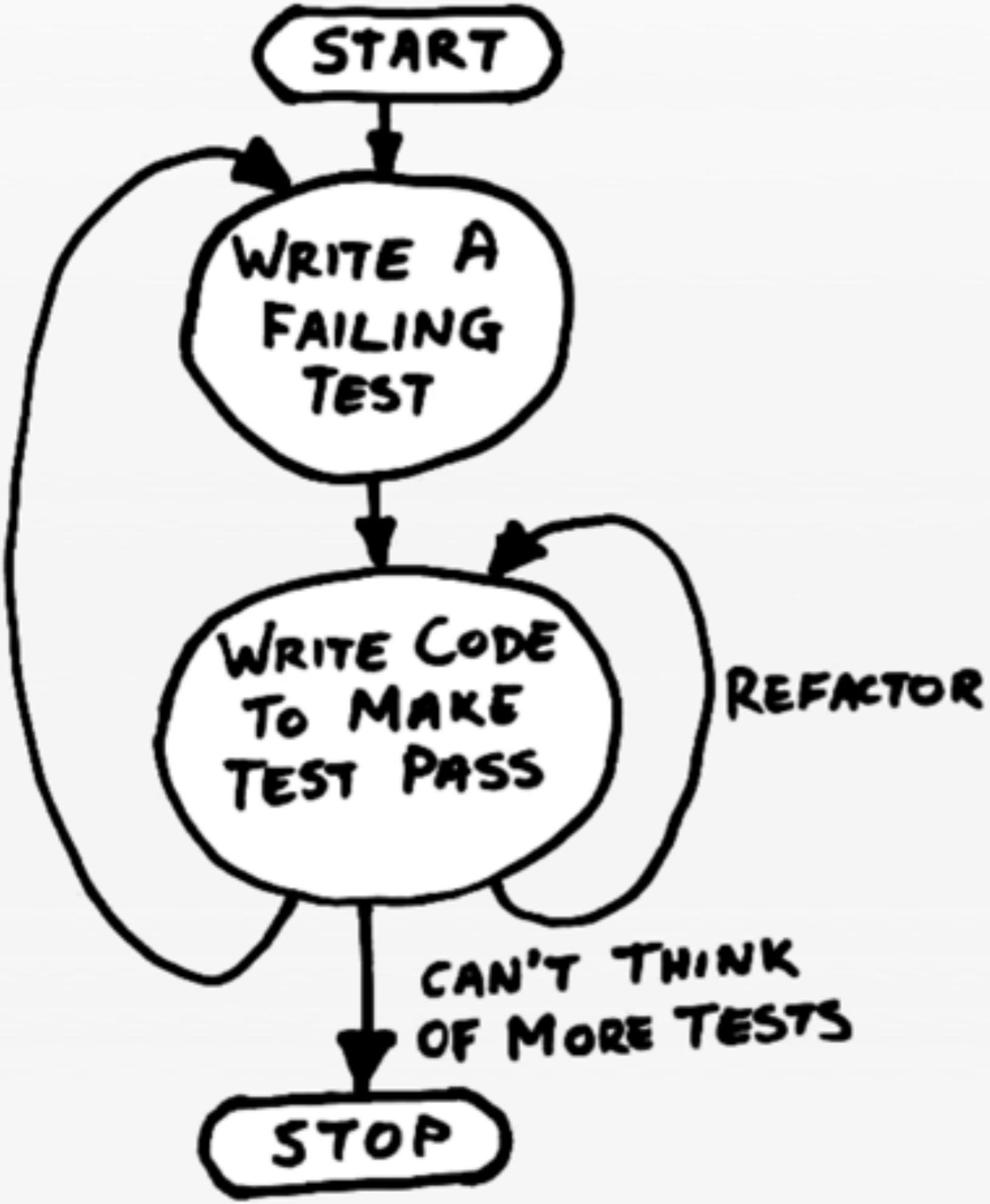
LEEEZE TEIL 2

BDD / TDD

WARUM TDD / BDD?

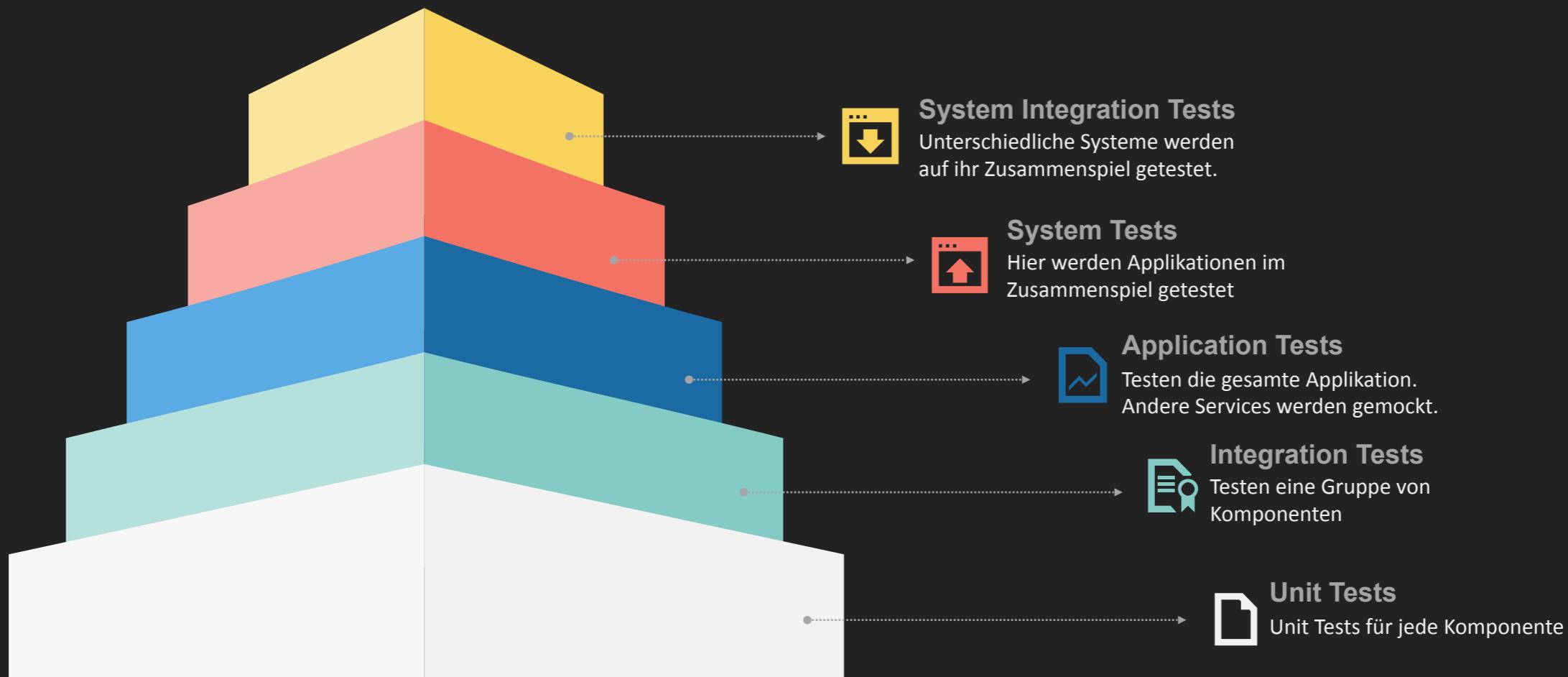
- ▶ Wasserfallmodell: Testphase erst nach Umsetzungsphase
- ▶ Probleme
 - ▶ Testabdeckung oft mangelhaft
 - ▶ Entwickelte Komponenten sind schwierig zu testen
 - ▶ Entwicklung neuer Features führt oft zu Regressionsfehlern
- ▶ Testgetriebene Entwicklung: Gemeinsame Entwicklung von Tests und Features



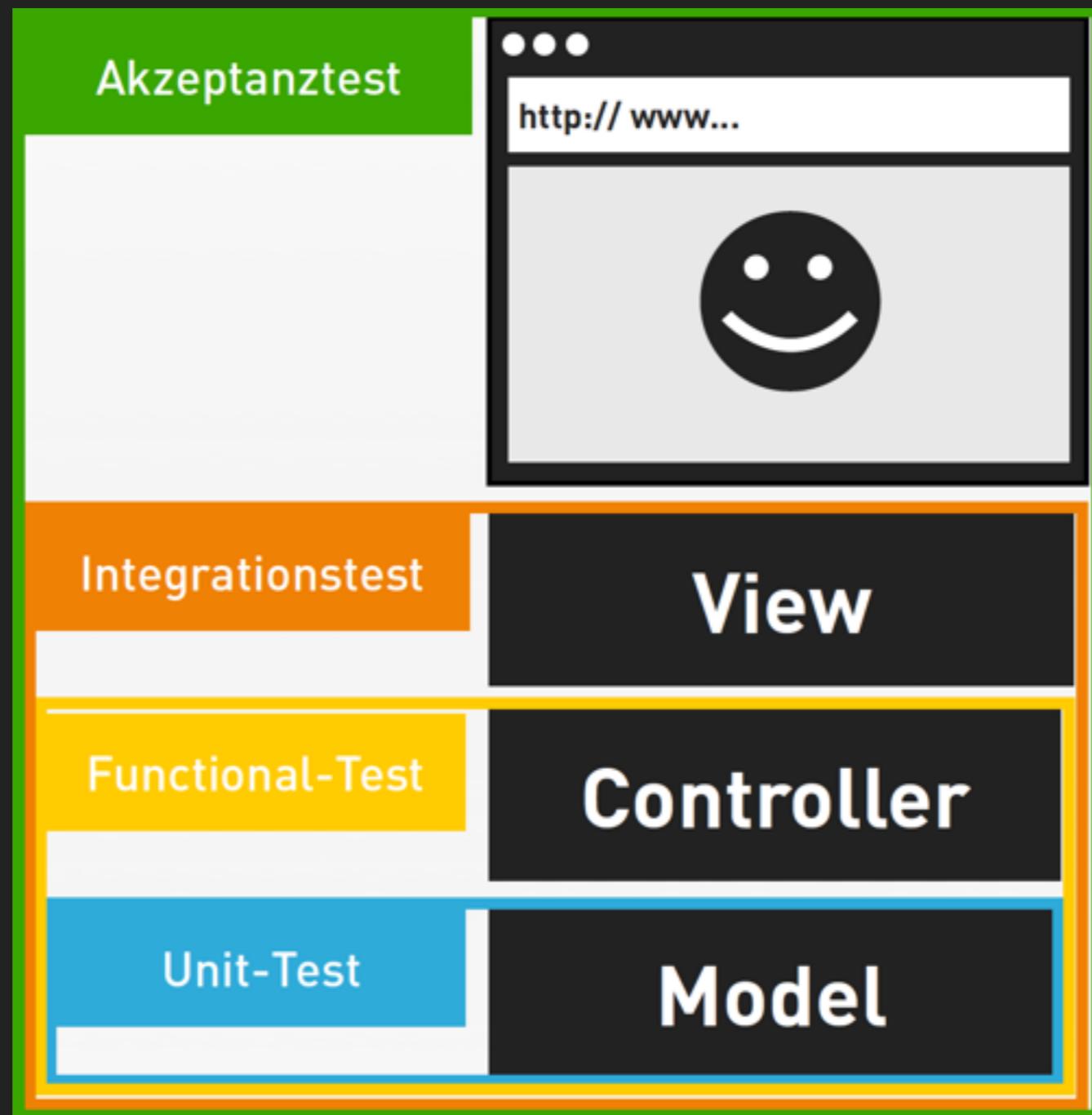


TDD ZYKLUS

Test Pyramide



Test einer Ruby on Rails Anwendung



RSpec Expectations

```
1  require 'spec_helper'  
2  
3  class CalculatorSpec  
4      def plus(n1, n2)  
5          n1 + n2  
6      end  
7  end  
8  
9  RSpec.describe CalculatorSpec do  
10     describe '#plus' do  
11         it 'adds two numbers and returns the sum' do  
12             calc = CalculatorSpec.new  
13             expect(calc.plus(19, 23)).to eql 42  
14         end  
15     end  
16 end
```

RSpec Mocks

```
1  require 'spec_helper'  
2  
3  class CalculatorSpec  
4    def plus(n1, n2)  
5      n1 + n2  
6    end  
7  end  
8  
9  RSpec.describe CalculatorSpec do  
10  describe '#plus' do  
11    it 'adds two numbers and returns the sum' do  
12      calc = CalculatorSpec.new  
13      allow(calc).to receive(:plus).and_return(42)  
14      expect(calc.plus(19, 23)).to eql 42  
15    end  
16  end  
17 end
```

User Story 2 - Spec anlegen

```
1  require 'rails_helper'  
2  
3  RSpec.describe Tour, type: :model do  
4    it 'needs a category' do  
5      category = Category.new  
6      tour = Tour.new title: 'Test Tour', teaser: 'My Teaser',  
7                        description: 'My description',  
8                        url: 'https://www.google.com/maps/'  
9  
10     expect(tour).to_not be_valid  
11  
12     tour.categories << category  
13  
14     expect(tour).to be_valid  
15   end  
16 end
```

User Story 2 - Spec ausführen

```
1. [RSpec results] 13 examples, 1 failures (2 pending) in 0.1413 seconds (ruby)

1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb
   # Not yet implemented
   # ./spec/models/category_spec.rb:4

2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb
   # Not yet implemented
   # ./spec/models/tour_category_spec.rb:4
```

Failures:

1) Tour needs a category

```
Failure/Error: expect(tour).to_not be_valid
  expected #<Tour id: nil, title: "Test Tour", teaser: "My Teaser", description: "My description", url: "https://www.google.com/maps/", created_at: nil, updated_at: nil> not to be valid
  # ./spec/models/tour_spec.rb:10:in `block (2 levels) in <top (required)>'
```

Finished in 0.14132 seconds (files took 1.29 seconds to load)

13 examples, 1 failure, 2 pending

Failed examples:

```
rspec ./spec/models/tour_spec.rb:4 # Tour needs a category
```

```
[1] guard(main)>
```

User Story 2 - Code schreiben

```
1 class Tour < ActiveRecord::Base
2   has_many :tour_categories
3   has_many :categories, through: :tour_categories
4
5   validates :title, :teaser, :description, :url, presence: true
6   validates :url, format: %r|^(http(s?)://www.google.com/maps/|)
7   validates :categories, presence: true
8 end
```

User Story 2 - Spec ausführen



1. [RSpec results] 13 examples, 0 failures (2 pending) in 0.3614 seconds (ruby)

```
[1] guard(main)>  
08:46:08 - INFO - Run all  
08:46:08 - INFO - Running all specs  
.**.....
```

Pending: (Failures listed here are expected and do not affect your suite's status)

```
1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb  
# Not yet implemented  
# ./spec/models/category_spec.rb:4
```

```
2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb  
# Not yet implemented  
# ./spec/models/tour_category_spec.rb:4
```

Finished in 0.3614 seconds (files took 1.3 seconds to load)

13 examples, 0 failures, 2 pending

```
[1] guard(main)>
```

)

Ergänzung - app/controllers/tours_controller.rb

```
1 class ToursController < ApplicationController
2   ...
3   private
4   # Use callbacks to share common setup or constraints between actions.
5   def set_tour ... end
6
7
8
9   # Never trust parameters from the scary internet,
10  # only allow the white list through.
11  def tour_params
12    params.require(:tour).permit(:title, :teaser,
13                                 :description, :url, category_ids: [])
14  end
15 end
```

User Story 2 - Spec Refactoring

```
1  require 'rails_helper'  
2  
3  RSpec.describe Tour, type: :model do  
4      let(:category) { Category.new }  
5      let(:tour) { Tour.new title: 'Test Tour', teaser: 'My Teaser',  
6                             description: 'My description',  
7                             url: 'https://www.google.com/maps/' }  
8  
9      describe 'needs a category' do  
10         it 'is not valid without a category' do  
11             expect(tour).to_not be_valid  
12         end  
13     end  
14  
15     it 'is valid with a category' do  
16         tour.categories << category  
17         expect(tour).to be_valid  
18     end  
19   end
```

TASKBOARD

Inbox

Ein Anwendung soll
über ein einfaches
Styling verfügen

In Progress

Ein Besucher soll
Touren nach Kategorie
filtern können, ...

In Review

Ein Besucher soll
Touren anlegen
können, ...

Complete

USER STORY 2

Ein Besucher soll Touren nach Kategorie filtern können, um schneller die passende Tour zu finden.

Integration-Test:

Ein Besucher soll alle Touren auf der Startseite sehen

VORAUSSETZUNGEN: CAPYBARA

```
35   group :development, :test do
36     gem 'byebug'
37     gem 'rspec-rails'
38     gem 'factory_girl_rails'
39     gem 'guard-rspec'
40     gem 'faker'
41     gem 'capybara'
42   end
```

\$ bundle install

CAPYBARA - EXKURS

- ▶ Capybara bietet eine domänenspezifische Sprache (DSL) zur Steuerung eines virtuellen Browsers
- ▶ RSpec startet zum Testen einen Testserver (~ rails server)
- ▶ Capybara greift auf diesen Testserver zu

CAPYBARA - EXKURS

- ▶ Steuerbefehle:
 - ▶ Seite '/tours' aufrufen:
 - ▶ visit '/tours'
 - ▶ Link klicken:
 - ▶ click_link 'New Tour'
 - ▶ Form ausfüllen und abschicken:
 - ▶ fill_in 'Name', with: 'Münsterland Giro'
 - ▶ click_button 'Create tour'
- ▶ Expectations:
 - ▶ Seite soll bestimmten Inhalt haben:
 - ▶ page.should have_content 'Created!'

1. FEATURE-SPEC SCHREIBEN

```
1  require 'rails_helper'  
2  
3  RSpec.feature 'Show Tours',  
4      'As a "user",  
5      I want to see tours on the start page  
6      so that I can choose between them' do  
7      let(:category) { Category.new }  
8      let(:tour) { Tour.new title: 'Test Tour', teaser: 'My Teaser',  
9          description: 'My description',  
10         url: 'https://www.google.com/maps/' }  
11  
12     before do  
13         tour.categories << category  
14         tour.save  
15     end  
16  
17     context 'start page' do  
18         scenario 'shows all tours on the start page' do  
19             visit root_path  
20             expect(page).to have_content(tour.title)  
21         end  
22     end  
23 end
```

2. FEATURE-SPEC AUSFÜHREN

● ● ●

1. [RSpec results] 15 examples, 1 failures (2 pending) in 0.3771 seconds (ruby)

```
2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeeze.ms/spec/models/to
ur_category_spec.rb
# Not yet implemented
# ./spec/models/tour_category_spec.rb:4
```

Failures:

```
1) Show Tours As a "user",
   I want to see tours on the start page
     so that I can choose between them start page shows all tours on the start page
```

Failure/Error: visit root_path

NameError:

```
undefined local variable or method `root_path' for #<RSpec::ExampleGroups::ShowToursAsAUserIWantToSeeTo
ursOnTheStartPageSoThatICanChooseBetweenThem::StartPage:0x007f99651cf1c8>
# ./spec/features/tour_spec.rb:18:in `block (3 levels) in <top (required)>'
```

Finished in 0.37713 seconds (files took 1.31 seconds to load)

15 examples, 1 failure, 2 pending

Failed examples:

```
rspec ./spec/features/tour_spec.rb:17 # Show Tours As a "user",
   I want to see tours on the start page
     so that I can choose between them start page shows all tours on the start page
```

[1] guard(main)>

3. CODE SCHREIBEN

```
1   Rails.application.routes.draw do
2     resources :tours
3
4     |   root 'tours#index'
5
```

4. FEATURE-SPEC AUSFÜHREN

```
[1] guard(main)> [1] guard(main)> 09:39:49 - INFO - Run all
09:39:49 - INFO - Running all specs
..**.....
Pending: (Failures listed here are expected and do not affect your suite's status)

  1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb
     # Not yet implemented
     # ./spec/models/category_spec.rb:4

  2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb
     # Not yet implemented
     # ./spec/models/tour_category_spec.rb:4

Finished in 0.43616 seconds (files took 1.33 seconds to load)
15 examples, 0 failures, 2 pending

[1] guard(main)> █
```

5. REFACTORING

- ▶ Wir haben zwei Tests, in denen eine Tour und eine Category erstellt werden
 - ▶ NOT DRY!
- ▶ Auch Tests sollen refactored werden.
- ▶ Einführung von Factories

FACTORY_GIRL

- ▶ factory_girl erlaubt das Erstellen von Factories (Fabriken) für Models
- ▶ Eine Factory erzeugt eine Instanz des Models mit sinnvollen Standardwerten
- ▶ Factories werden in spec/factories/ erstellt

FACTORY_GIRL

- ▶ Factory definieren:

```
1  FactoryGirl.define do
2    factory :category do
3      name 'My Category'
4    end
5  end
6
```

- ▶ Factory nutzen:

```
1  FactoryGirl.build(:category)
2  # ~> Category.new name: "My Category"
3
4  FactoryGirl.create(:category)
5  # ~> Category.create name: "My Category"
6
7  FactoryGirl.create(:category, name: 'Some other name')
8  # ~> Category.new name: "Some other name"
```

5. REFACTORING

- ▶ Factory definieren:

```
1   FactoryGirl.define do
2     factory :category do
3       sequence(:name) { |n| "My Category #{n}" }
4     end
5   end
```

5. REFACTORING

- ▶ Factory definieren:

```
1   FactoryGirl.define do
2     factory :tour do
3       title 'My Tour'
4       teaser 'My Teaser'
5       description 'My Description'
6       url 'https://www.google.com/maps/'
7     end
8   end
```

5. REFACTORING

- ▶ Factory einbauen:

```
1  require 'rails_helper'  
2  
3  RSpec.describe Tour, type: :model do  
4      let(:category) { FactoryGirl.build(:category) }  
5      let(:tour) { FactoryGirl.build(:tour) }  
6  
7      describe 'needs a category' do  
8          it 'is not valid without a category' do  
9              expect(tour).to_not be_valid  
10         end  
11     end  
12  
13     it 'is valid with a category' do  
14         tour.categories << category  
15         expect(tour).to be_valid  
16     end  
17   end
```

5. REFACTORING

► Factory einbauen:

```
1  require 'rails_helper'  
2  
3  RSpec.feature 'Show Tours',  
4      'As a "user",  
5          I want to see tours on the start page  
6          so that I can choose between them' do  
7      context 'start page' do  
8          let(:category) { FactoryGirl.create(:category) }  
9          let(:tour) { FactoryGirl.build(:tour, categories: [category]) }  
10  
11     before do  
12         tour.save  
13     end  
14  
15     scenario 'shows all tours on the start page' do  
16         visit root_path  
17         expect(page).to have_content(tour.title)  
18     end  
19  end  
20 end
```

5. REFACTORING

► Tests ausführen:

```
[1] guard(main)>
10:00:25 - INFO - Run all
10:00:25 - INFO - Running all specs
..**.....
Pending: (Failures listed here are expected and do not affect your suite's status)
  1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb
     # Not yet implemented
     # ./spec/models/category_spec.rb:4
  2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb
     # Not yet implemented
     # ./spec/models/tour_category_spec.rb:4
I
Finished in 0.50417 seconds (files took 2.22 seconds to load)
15 examples, 0 failures, 2 pending

[1] guard(main)>
```

USER STORY 2

Ein Besucher soll Touren nach Kategorie filtern können, um schneller die passende Tour zu finden.

1. Specs schreiben

```
1  require 'rails_helper'  
2  
3  RSpec.feature 'Show Tours',  
4      'As a "user",  
5      I want to see tours on the start page  
6      so that I can choose between them' do  
7      context 'start page' do ... end  
20  
21      context 'filter' do  
22          let(:category1) { FactoryGirl.create(:category) }  
23          let(:tour1) { FactoryGirl.build(:tour, title: 'My Tour 1', categories: [category1]) }  
24  
25          let(:category2) { FactoryGirl.create(:category) }  
26          let(:tour2) { FactoryGirl.build(:tour, title: 'My Tour 2', categories: [category2]) }  
27  
28          before do  
29              tour1.save  
30              tour2.save  
31          end  
32  
33          scenario 'allows to select categories' do  
34              visit root_path  
35  
36              click_link category1.name  
37              expect(page).to have_content(tour1.title)  
38              expect(page).to_not have_content(tour2.title)  
39          end  
40      end  
41  end
```

2. Specs ausführen

```
1. [RSpec results] 16 examples, 1 failures (2 pending) in 0.5043 seconds (ruby)
# ./spec/models/category_spec.rb:4
2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/to
ur_category_spec.rb
# Not yet implemented
# ./spec/models/tour_category_spec.rb:4
```

Failures:

```
1) Show Tours As a "user",
   I want to see tours on the start page
   so that I can choose between them filter allows to select categories
Failure/Error: click_link category1.name
Capybara::ElementNotFound:
  Unable to find link "My Category 2"
# ./spec/features/tour_spec.rb:36:in `block (3 levels) in <top (required)>'
```

Finished in 0.50428 seconds (files took 1.54 seconds to load)

16 examples, 1 failure, 2 pending

Failed examples:

```
rspec ./spec/features/tour_spec.rb:33 # Show Tours As a "user",
   I want to see tours on the start page
   so that I can choose between them filter allows to select categories
```

```
[1] guard(main)>
```

3. Code schreiben

```
1   Rails.application.routes.draw do
2     resources :tours
3
4     get ':category_id', to: 'tours#index', as: 'category_tours'
5
6     root 'tours#index'
```

Prefix	Verb	URI Pattern	Controller#Action
	tour	/tours(.:format)	tours#index
	POST	/tours(.:format)	tours#create
new_tour	GET	/tours/new(.:format)	tours#new
edit_tour	GET	/tours/:id/edit(.:format)	tours#edit
	tour	/tours/:id(.:format)	tours#show
	PATCH	/tours/:id(.:format)	tours#update
	PUT	/tours/:id(.:format)	tours#update
	DELETE	/tours/:id(.:format)	tours#destroy
category_tours	GET	/:category_id(.:format)	tours#index
root	GET	/	tours#index

3. Code schreiben

```
1  class ToursController < ApplicationController
2    before_action :set_tour, only: [:show, :edit, :update, :destroy]
3
4    # GET /tours
5    # GET /tours.json
6    def index
7      @tours = if params[:category_id].present?
8        Category.find(params[:category_id]).tours
9      else
10        Tour.all
11      end
12    end
```

3. Code schreiben - tours/index.html.erb

```
1 ⏷  <p id="notice"><%= notice %></p>
2
3  <h1>Listing Tours</h1>
4
5  <ul>
6    <% Category.all.each do |category| %>
7      <li><%= link_to category.name, category_tours_path(category) %></li>
8    <% end %>
9  </ul>
10
11 <ul>
12   <% @tours.each do |tour| %>
13     <li><%= link_to tour.title, tour %></li>
14   <% end %>
15 </ul>
16 <br>
17
18 <%= link_to 'New Tour', new_tour_path %>
19
```

4. Tests ausführen

```
● ● ●
[1] guard(main)>
11:12:26 - INFO - Run all
11:12:26 - INFO - Running all specs
...**.....
Pending: (Failures listed here are expected and do not affect your suite's status)

1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb
   # Not yet implemented
   # ./spec/models/category_spec.rb:4

2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb
   # Not yet implemented
   # ./spec/models/tour_category_spec.rb:4

Finished in 0.53437 seconds (files took 2.18 seconds to load)
16 examples, 0 failures, 2 pending

[1] guard(main)>
```

5. REFACTORING

- ▶ Problem: Die generierten URLs sind hässlich
 - ▶ `http://localhost:3000/1`
- ▶ Lösung: Name der Kategorie an die ID anhängen
 - ▶ `http://localhost:3000/1-radtour`

5. REFACTORING

```
1 • ⚑ class Category < ActiveRecord::Base
2   has_many :tour_categories
3   has_many :tours, through: :tour_categories
4
5 ⚑ def to_param
6   "#{id}-#{name.parameterize}"
7 end
8 end
```

5. REFACTORING - TESTS AUSFÜHREN

```
[1] guard(main)> 1. [RSpec results] 16 examples, 0 failures (2 pending) in 0.5491 seconds (ruby)
[1] guard(main)> 11:16:59 - INFO - Run all
11:16:59 - INFO - Running all specs
...**.....
Pending: (Failures listed here are expected and do not affect your suite's status)

  1) Category add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/category_spec.rb
     # Not yet implemented
     # ./spec/models/category_spec.rb:4

  2) TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/ws2015/leeze.ms/spec/models/tour_category_spec.rb
     # Not yet implemented
     # ./spec/models/tour_category_spec.rb:4

Finished in 0.54906 seconds (files took 1.42 seconds to load)
16 examples, 0 failures, 2 pending

[1] guard(main)> ]
```

TASKBOARD

Inbox	In Progress	In Review	Complete
<p>Ein Anwendung soll über ein einfaches Styling verfügen</p>	<p>+</p>	<p>Ein Besucher soll Touren anlegen können, ...</p>	
	<p>Ein Besucher soll Touren nach Kategorie filtern können, ...</p>	<p>Ein Besucher soll Touren nach Kategorie filtern können, ...</p>	

TASKBOARD

Inbox	In Progress	In Review	Complete
<p>Ein Anwendung soll über ein einfaches Styling verfügen</p>	<p>Ein Anwendung soll über ein einfaches Styling verfügen</p>	<p>Ein Besucher soll Touren anlegen können, ...</p>	<p>Ein Besucher soll Touren nach Kategorie filtern können, ...</p>

TASKBOARD - USERS STORY 3

Die Anwendung soll über ein einfaches Styling verfügen

TWITTER BOOTSTRAP

```
26 | gem 'therubyracer'  
27 | gem 'twitter-bootstrap-rails'  
28 | gem 'less-rails'
```

\$ bundle install

TWITTER BOOTSTRAP

- ▶ Nötige Assets in die Anwendung kopieren:
 - ▶ \$ bundle exec rails generate bootstrap:install
- ▶ Application Layout durch Bootstrap ersetzen:
 - ▶ \$ bundle exec rails generate bootstrap:layout application fluid

Tabelle auf der Übersichtsseite anzeigen

```
5 <ul>
6   <% Category.all.each do |category| %>
7     <li><%= link_to category.name, category_tours_path(category) %></li>
8   <% end %>
9 </ul>
10
11 <table class="table table-striped">
12   <tr>
13     <th>Title</th>
14     <th></th>
15     <th></th>
16     <th></th>
17   </tr>
18   <% @tours.each do |tour| %>
19     <tr>
20       <td><%= link_to tour.title %></td>
21       <td><%= link_to 'Show', tour %></td>
22       <td><%= link_to 'Edit', edit_tour_path(tour) %></td>
23       <td><%= link_to 'Destroy', tour, method: :delete, data: { confirm: 'Are you sure?' } %></td>
24     </tr>
25   <% end %>
26 </table>
27 <br>
```

A screenshot of a web browser window titled "LeezeMs". The address bar shows "localhost:3000/1-radtour". The main content area displays a list of tours:

Listing Tours

- Radtour
- Mountainbiketour
- Radwandern

Below the list, there is a table row with the following columns:

Title	Show	Edit	Destroy
Geführte Tour von der FH zu "Burg Erking"	Show	Edit	Destroy

On the right side of the page, there is a sidebar with the following content:

Sidebar

Sidebar

Link1

Link2

Link3