daftcode

RUBY ON RAILS Level UP

4. Standardy tworzenia oprogramowania

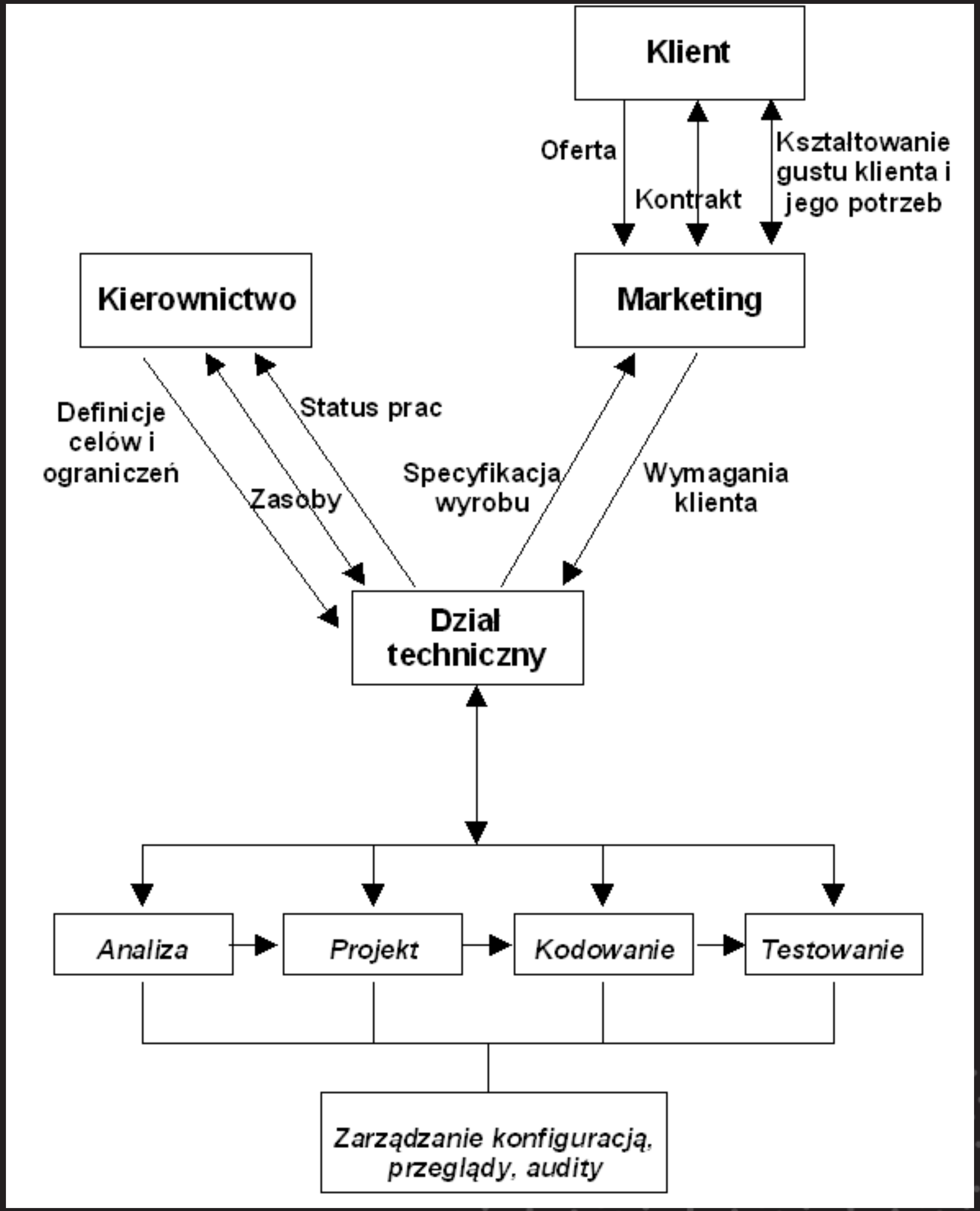
Praca domowa

RUBY ON RAILS
Level UP

Jakie mamy standardy?

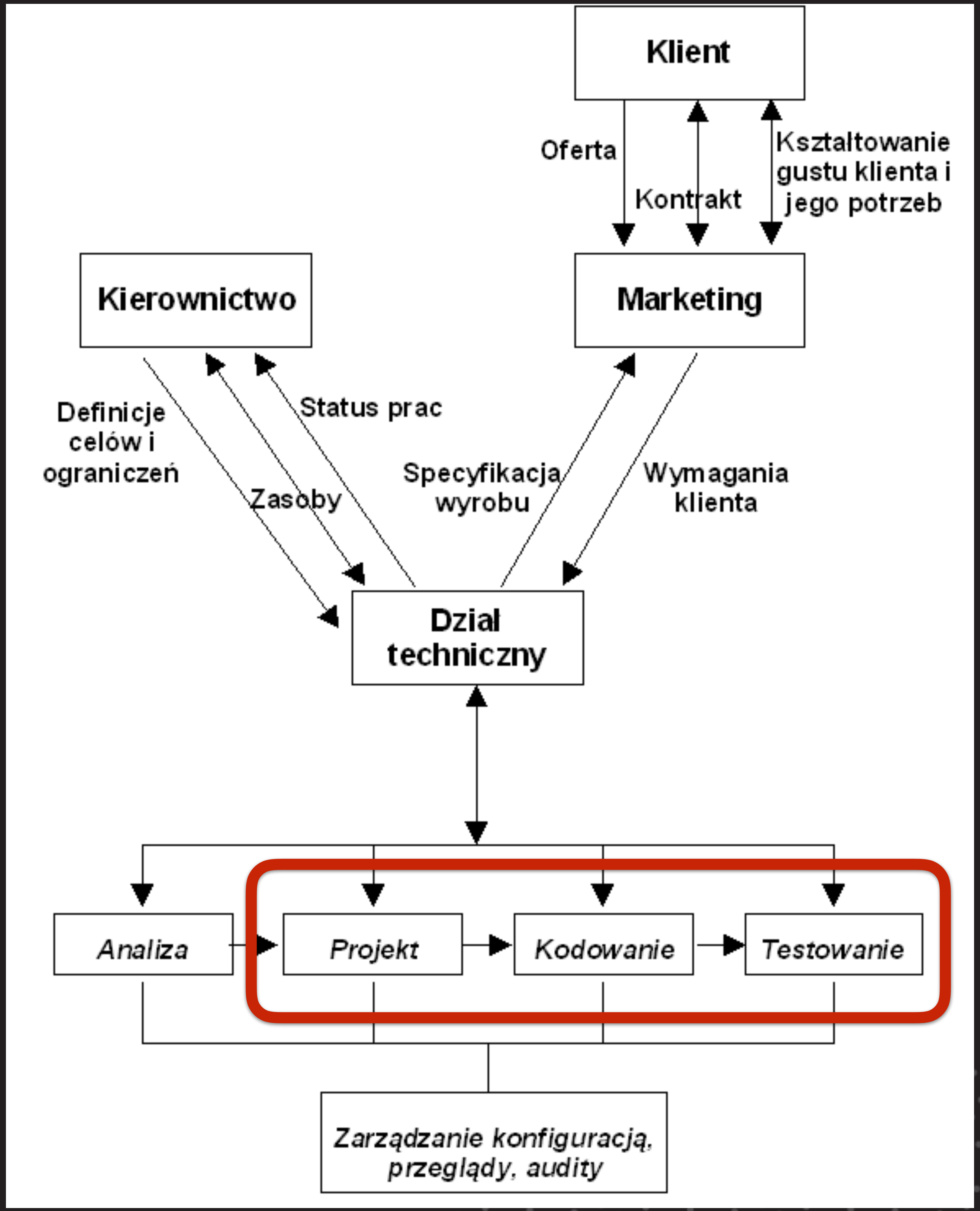
Norma ISO/IEC 90003

Software engineering -- Guidelines for the application of ISO 9001:2008 to computer software



Norma ISO/IEC 90003

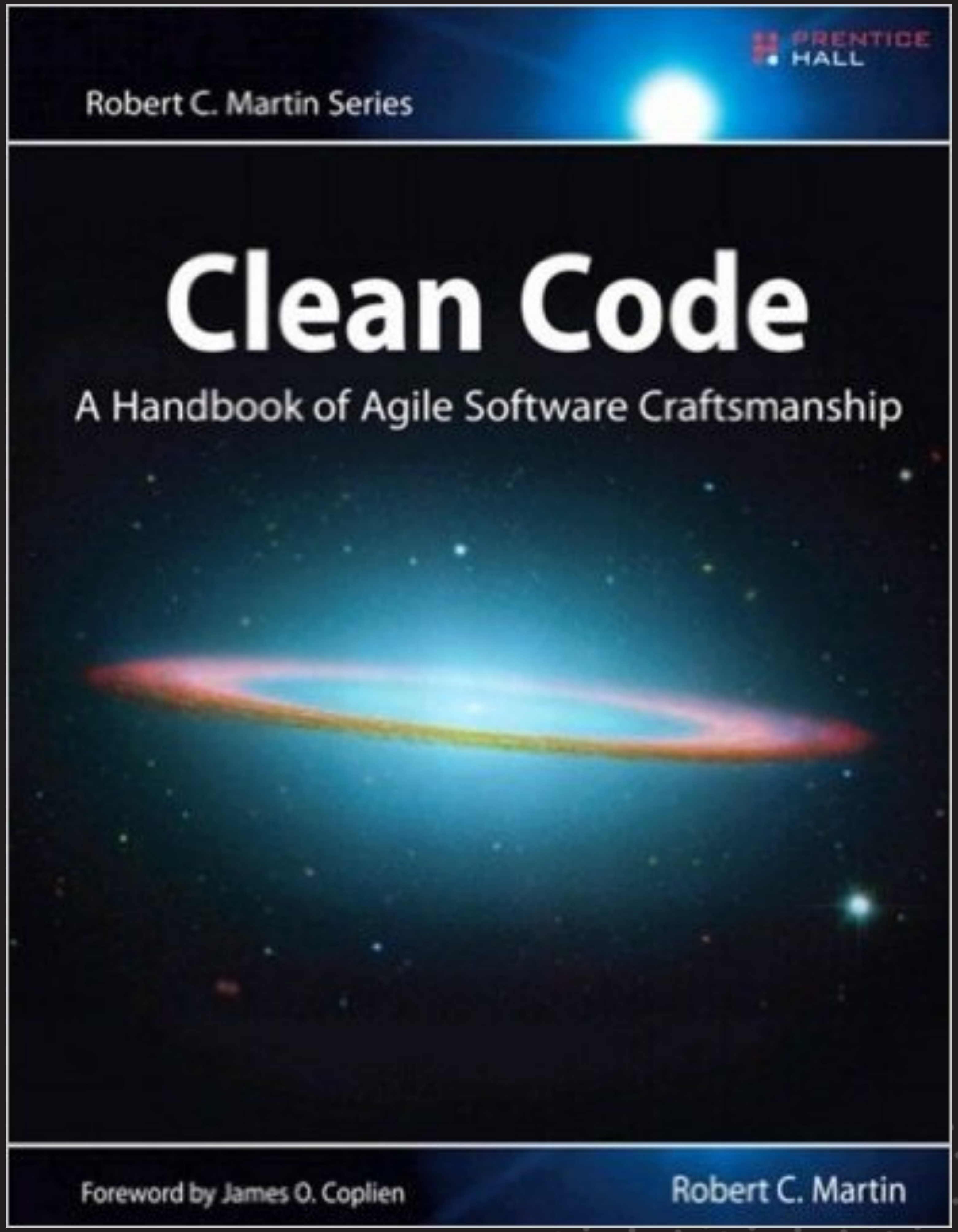
Software engineering -- Guidelines for the application of ISO 9001:2008 to computer software



RUBY ON RAILS
Level UP

"Clean Code"

Robert C. Martin


Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin

RUBY ON RAILS
Level UP

Organizacja projektu

RUBY ON RAILS
Level UP

Planowanie zadań

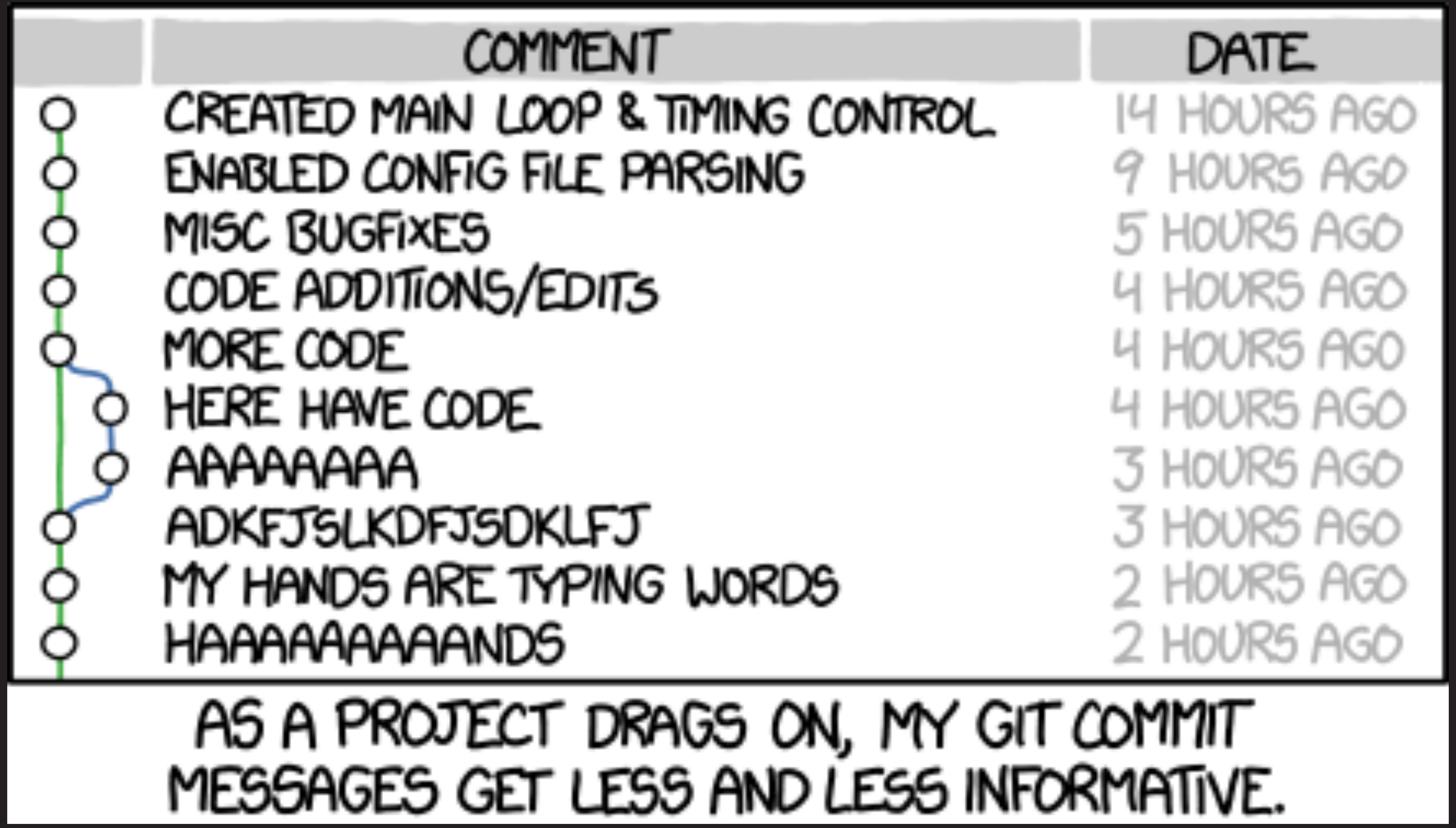
- Bieżące zadania
- Planowanie krótkoterminowe
- Dokładne opisywanie wymagań

Repozytorium kodu

- Git
- SVN
- Mercurial

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Repozytorium kodu



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

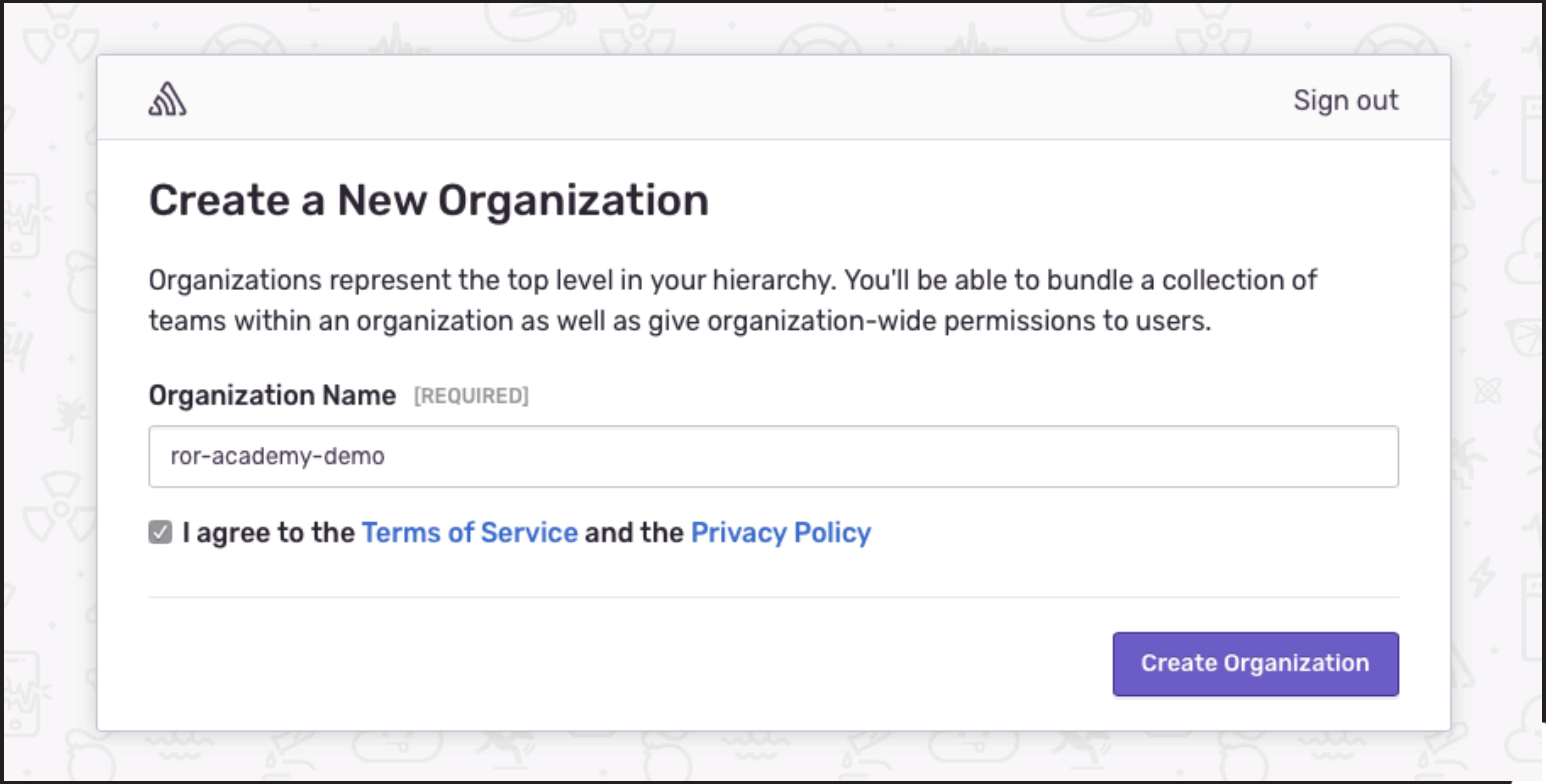
Raportowanie błędów

```
class UsersController < ApplicationController
  def create
    User.crate!(user_params)
  end

  private

  def user_params
    params.permit(:username, :password)
  end
end
```

RUBY ON RAILS
Level UP



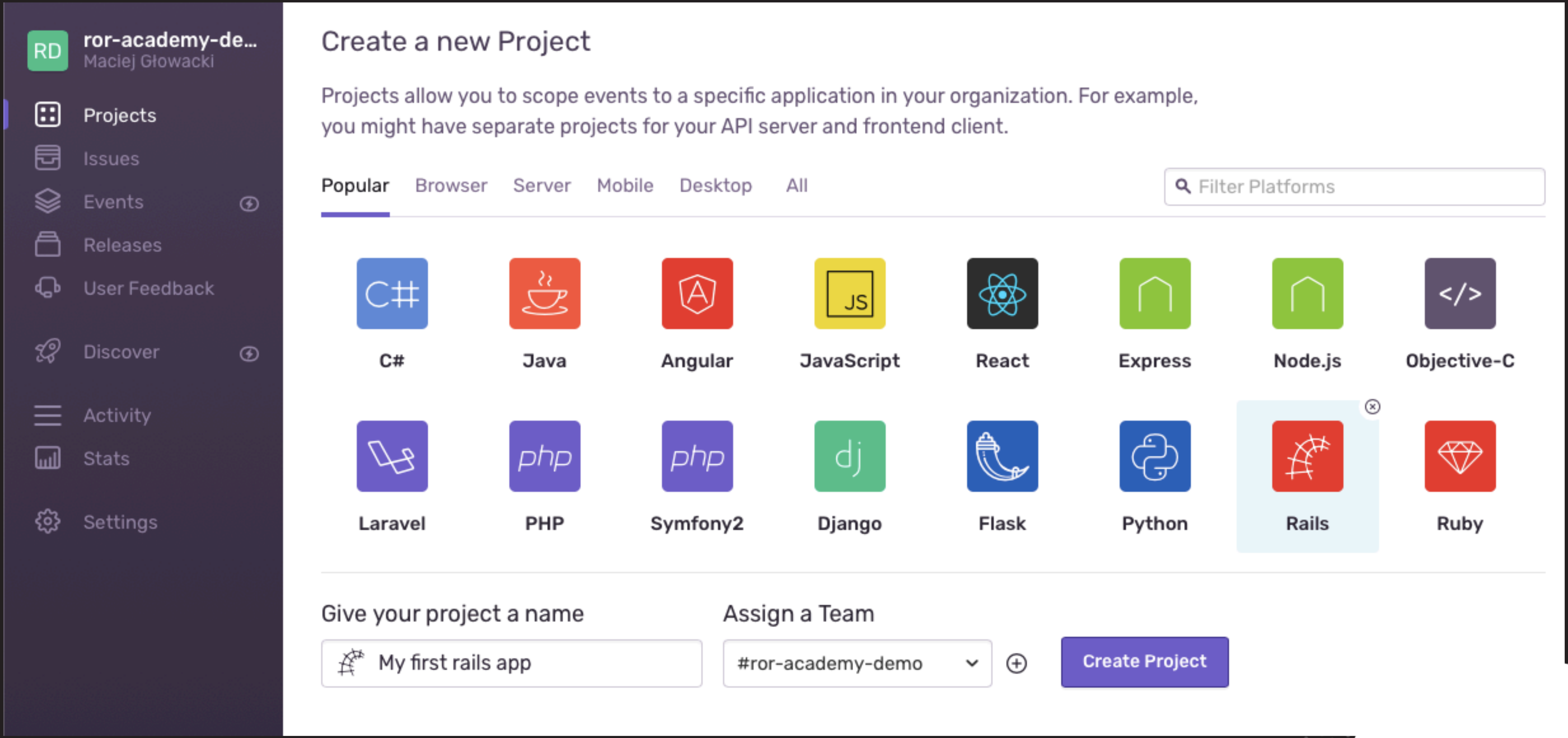
Create a New Organization

Organizations represent the top level in your hierarchy. You'll be able to bundle a collection of teams within an organization as well as give organization-wide permissions to users.

Organization Name [REQUIRED]

ror-academy-demo

✓ I agree to the Terms of Service and the Privacy Policy



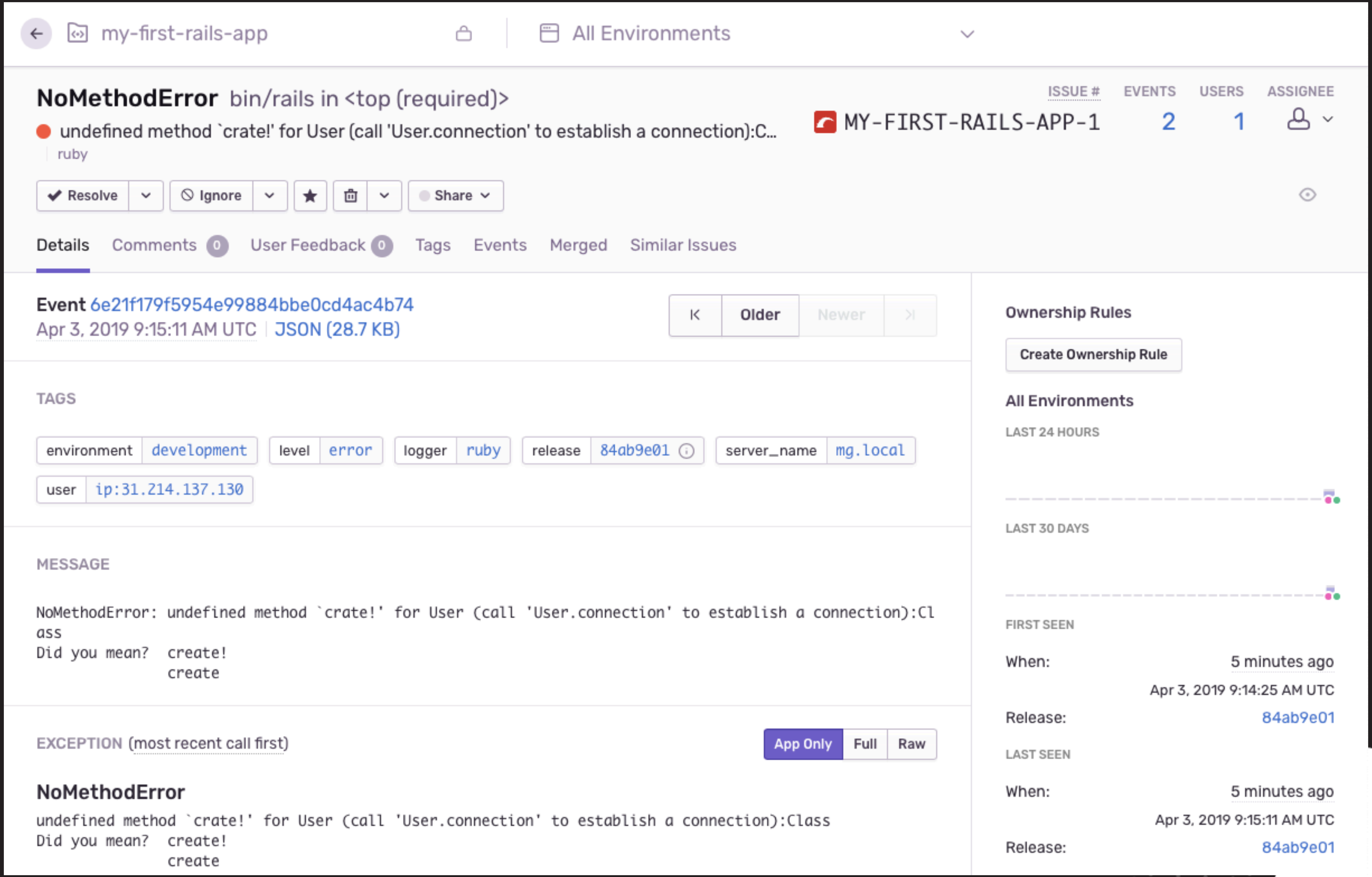
Create a new Project

Projects allow you to scope events to a specific application in your organization. For example, you might have separate projects for your API server and frontend client.

Give your project a name: My first rails app

Assign a Team: #ror-academy-demo

RUBY ON RAILS
Level UP



NoMethodError bin/rails in <top (required)>

undefined method `crate!' for User (call 'User.connection' to establish a connection):C...

ruby

ISSUE #	EVENTS	USERS	ASSIGNEE
MY-FIRST-RAILS-APP-1	2	1	

Event 6e21f179f5954e99884bbe0cd4ac4b74
Apr 3, 2019 9:15:11 AM UTC | JSON (28.7 KB)

TAGS

environment	development
level	error
logger	ruby
release	84ab9e01
server_name	mg.local
user	ip:31.214.137.130

MESSAGE

```
NoMethodError: undefined method `crate!' for User (call 'User.connection' to establish a connection):Cl
ass
Did you mean?  create!
               create
```

EXCEPTION (most recent call first)

NoMethodError

```
undefined method `crate!' for User (call 'User.connection' to establish a connection):Class
Did you mean?  create!
               create
```

Ownership Rules

All Environments

LAST 24 HOURS

LAST 30 DAYS

FIRST SEEN

When:	5 minutes ago
	Apr 3, 2019 9:14:25 AM UTC
Release:	84ab9e01

LAST SEEN

When:	5 minutes ago
	Apr 3, 2019 9:15:11 AM UTC
Release:	84ab9e01

RUBY ON RAILS
Level UP

Styl pisania kodu

- Wcięcia
- Nazwy klas i metod
- Sposób przełamywania linii
- Wyrównanie nawiasów
- Styl pisania hashy, tablic etc.
- 438 reguł w bazowym Rubocopie!

Styl pisania kodu

```
gem 'rubocop', require: false

$ rubocop
Inspecting 1026 files
.(...).C.............

Offenses:

app/models/user.rb:10:1: C: Metrics/ClassLength: Class has too many lines. [130/100]
class User < ApplicationRecord ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
app/models/user.rb:64:28: C: Layout/SpaceInsideHashLiteralBraces: Space inside { missing.
  validates :name, length: {maximum: FieldsLimits::MEDIUM }
                           ^

1026 files inspected, 2 offenses detected
```

RUBY ON RAILS
Level UP

Wzorce projektowe

RUBY ON RAILS
Level UP

Service Object

```ruby
class WarriorCreator
  def initialize(warrior_params)
    @params = warrior_params
  end

  def call
    create_warrior
    send_congratulations
    initialize_training
  end

  private

  attr_reader :params

  def create_warrior
    @warrior = Warrior.create!(name: params[:name], armor: params[:armor])
  end

  # ...
end
```

RUBY ON RAILS
Level UP

Query Object

```
class WarriorsController < ApplicationController
  def index
    warriors = if params[:alive]
                 Warrior.alive.where(clan_id: params[:clan_id])
               else
                 Warrior.where(clan_id: params[:clan_id])
               end
    render json: warriors
  end
end
```

Query Object

```ruby
class WarriorsQuery
  def self.belonging_to_clan(clan_id:, relation: Warrior)
    relation.where(clan_id: clan_id)
  end

  def self.having_birthday(relation: Warrior)
    relation.where(birthday: Date.today)
  end
end
```

RUBY ON RAILS
Level UP

Query Object

```ruby
class WarriorsController < ApplicationController
  def index
    warriors = params[:alive] ? Warrior.alive : Warrior
    warriors = WarriorsQuery.belonging_to_clan(clan_id: params[:clan_id],
                                               relation: warriors)
    warriors = WarriorsQuery.having_birthday(relation: warriors) if params[:birthday]
    render json: warriors
  end
end
```

Null Object

RUBY ON RAILS
Level UP

SOLID

RUBY ON RAILS
Level UP

Single Responsibility Principle

```ruby
class WarriorRecruiter
  def run
    warrior = find_warrior
    Invitation.create!(warrior: warrior)
    message = "Dear #{warrior.name}. Do you want to join my army?"
    send_message(message)
  end

  private

  def find_warrior
    warriors = LinkedinApi.search(profession: 'warrior')
    warriors.select(&:unemployed?).first
  end

  def send_message(message)
    PostalPidgeon.deliver(message)
  end
end
```

RUBY ON RAILS
Level UP

Single Responsibility Principle

```
class WarriorRecruiter
  def run
    warrior = find_warrior
    Invitation.create!(warrior: warrior) 2️⃣
    message = "Dear #{warrior.name}. Do you want to join my army?" 3️⃣
    send_message(message)
  end

  private

  def find_warrior
    warriors = LinkedinApi.search(profession: 'warrior')
1️⃣  warriors.select(&:unemployed?).first
  end

  def send_message(message)
    PostalPidgeon.deliver(message)
  end     4️⃣
end
```

Single Responsibility Principle

```ruby
class WarriorRecruiter
  def run
    warrior = WarriorFinder.new.call
    WarriorInviter.new(warrior: warrior).call
  end
end

class WarriorFinder
  def run
    warriors = LinkedinApi.search(profession: 'warrior')
    warriors.select(&:unemployed).first
  end
end
```

Single Responsibility Principle

```
class WarriorRecruiter
  def run
    warrior = find_warrior
    Invitation.create!(warrior: warrior) 2
    message = "Dear #{warrior.name}. Do you want to join my army?" 3
    send_message(message)
  end

  private

  def find_warrior
    warriors = LinkedinApi.search(profession: 'warrior')
1   warriors.select(&:unemployed?).first
  end

  def send_message(message)
    PostalPidgeon.deliver(message)
  end      4
end
```

Single Responsibility Principle

```ruby
class WarriorRecruiter
  def run
    warrior = WarriorFinder.new.call
    WarriorInviter.new(warrior: warrior).call
  end
end

class WarriorInviter
  def initialize(warrior:)
    @warrior = warrior
  end

  def run
    Invitation.create!(warrior: @warrior)
    WarriorNotifier.new(warrior: @warrior)
  end
end
```

RUBY ON RAILS
Level UP

Single Responsibility Principle

```ruby
class WarriorNotifier
  def initialize(warrior:)
    @warrior = warrior
  end

  def run
    PostalPidgeon.deliver(message)
  end

  private

  def message
    "Dear #{@warrior.name}. Do you want to join my army?"
  end
end
```

Open/closed principle

```ruby
class WarriorNotifier
  def initialize(warrior:)
    @warrior = warrior
  end

  def run
    PostalPidgeon.deliver(message)
  end

  private

  def message
    "Dear #{@warrior.name}. Do you want to join my army?"
  end
end
```

RUBY ON RAILS
Level UP

Open/closed principle

```ruby
class WarriorNotifier
  def initialize(warrior:, delivery_service: PostalPidgeon)
    @warrior = warrior
    @delivery_service = delivery_service
  end

  def run
    @delivery_service.deliver(message)
  end

  private

  def message
    "Dear #{@warrior.name}. Do you want to join my army?"
  end
end
```

Open/closed principle

```ruby
class WarriorInviter
  def initialize(warrior:)
    @warrior = warrior
  end

  def run
    Invitation.create!(warrior: @warrior)
    WarriorNotifier.new(warrior: @warrior)
    WarriorNotifier.new(warrior: @warrior, delivery_service: SmokeSigns)
  end
end
```

Liskov substitution principle

```
class Warrior < ApplicationRecord
  # attributes: name
end

class Hussar < Warrior
  def name
    "Pan #{super}"
  end
end

class Samurai < Warrior
  def name
    { jp: super, en: Romajify::Converter.hepburn(super) }
  end
end
```

Liskov substitution principle

```ruby
class WarriorNotifier
  def initialize(warrior:, delivery_service: PostalPidgeon)
    @warrior = warrior
    @delivery_service = delivery_service
  end

  def run
    @delivery_service.deliver(message)
  end

  private

  def message
    "Dear #{@warrior.name}. Do you want to join my army?"
  end
end
```

Interface segregation principle

```ruby
class BattleCalculator
  def results(our_army:, enemy_army:, attack:)
    BattlePerformer.new(enemy_army: enemy_army).call if attack
    our_army.strength > enemy_army.strength ? 'win' : 'defeat'
  end
end
```

RUBY ON RAILS
Level UP

Interface segregation principle

```ruby
class BattleCalculator
  def results(our_army:, enemy_army:)
    our_army.strength > enemy_army.strength ? 'win' : 'defeat'
  end

  def perform(our_army:, enemy_army:)
    BattlePerformer.new(enemy_army: enemy_army).call
    results(our_army: our_army, enemy_army: enemy_army)
  end
end
```

Dependency inversion principle

```ruby
class WarriorNotifier
  def initialize(warrior:, delivery_service: PostalPidgeon)
    @warrior = warrior
    @delivery_service = delivery_service
  end

  def run
    @delivery_service.deliver(message)
  end

  private

  def message
    "Dear #{@warrior.name}. Do you want to join my army?"
  end
end
```

Thanks!