

# Not the Rails Way

Игорь Александров, JetRockets Co-Founder

# Игорь Александров

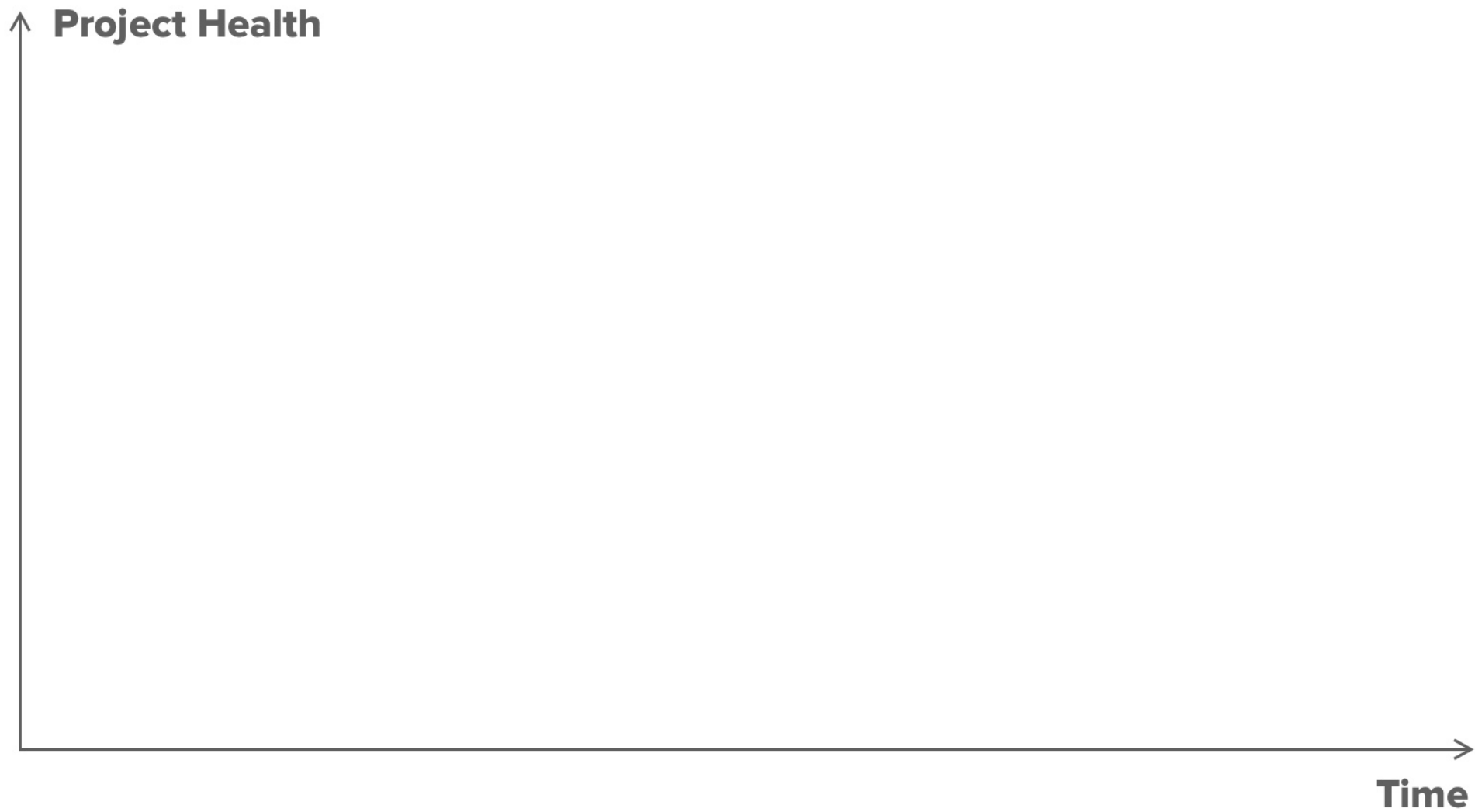
---

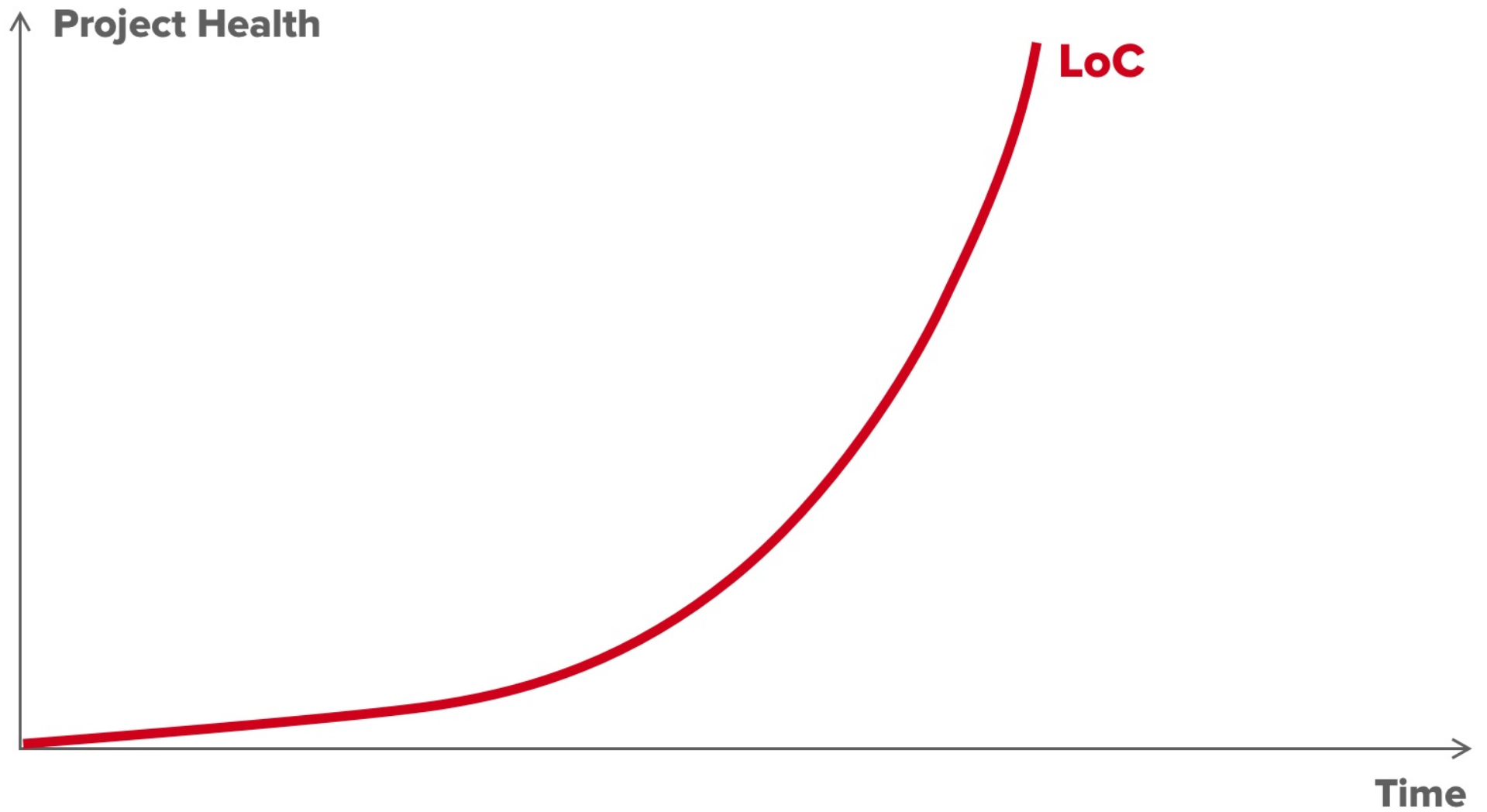


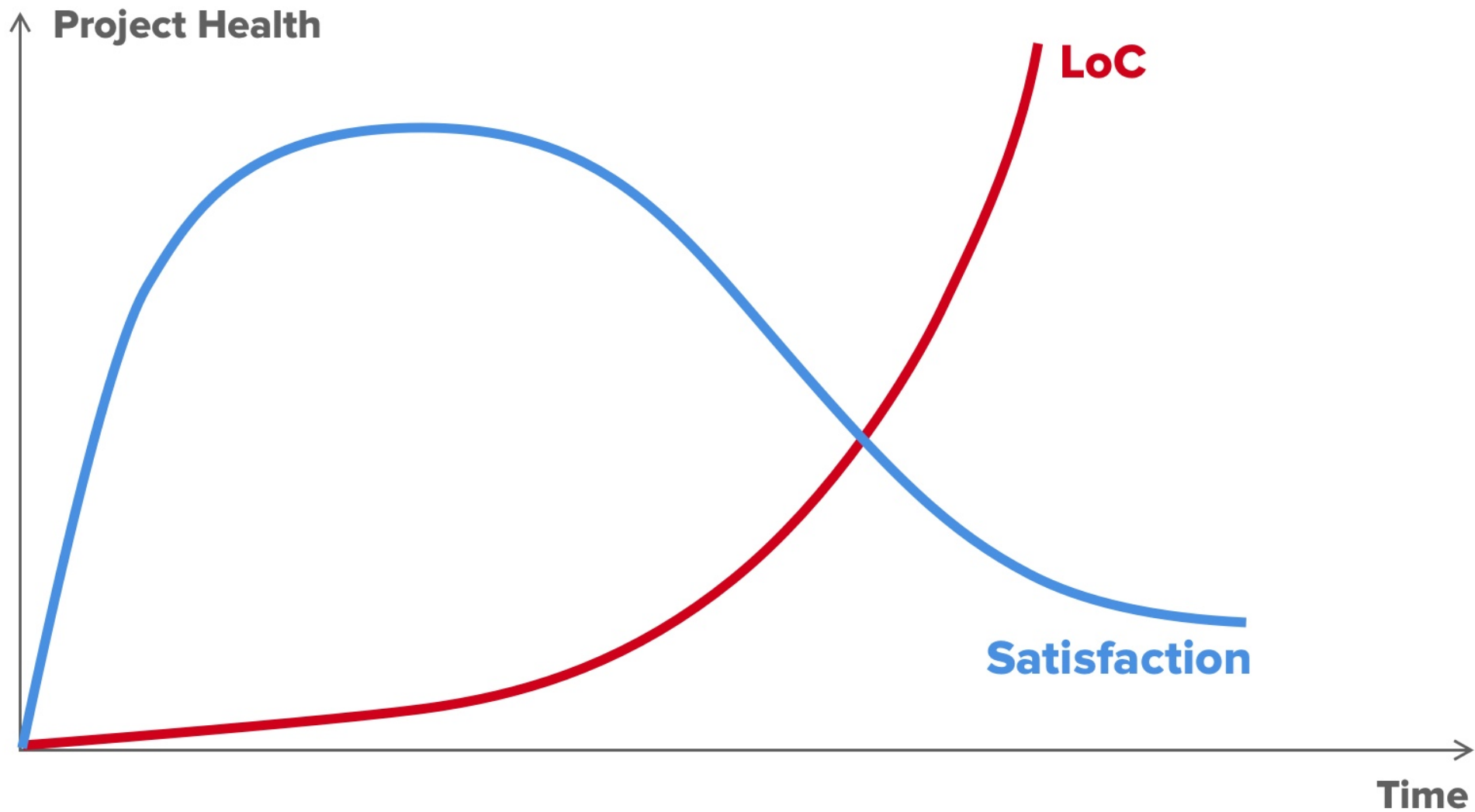
[jetrockets.pro](http://jetrockets.pro)

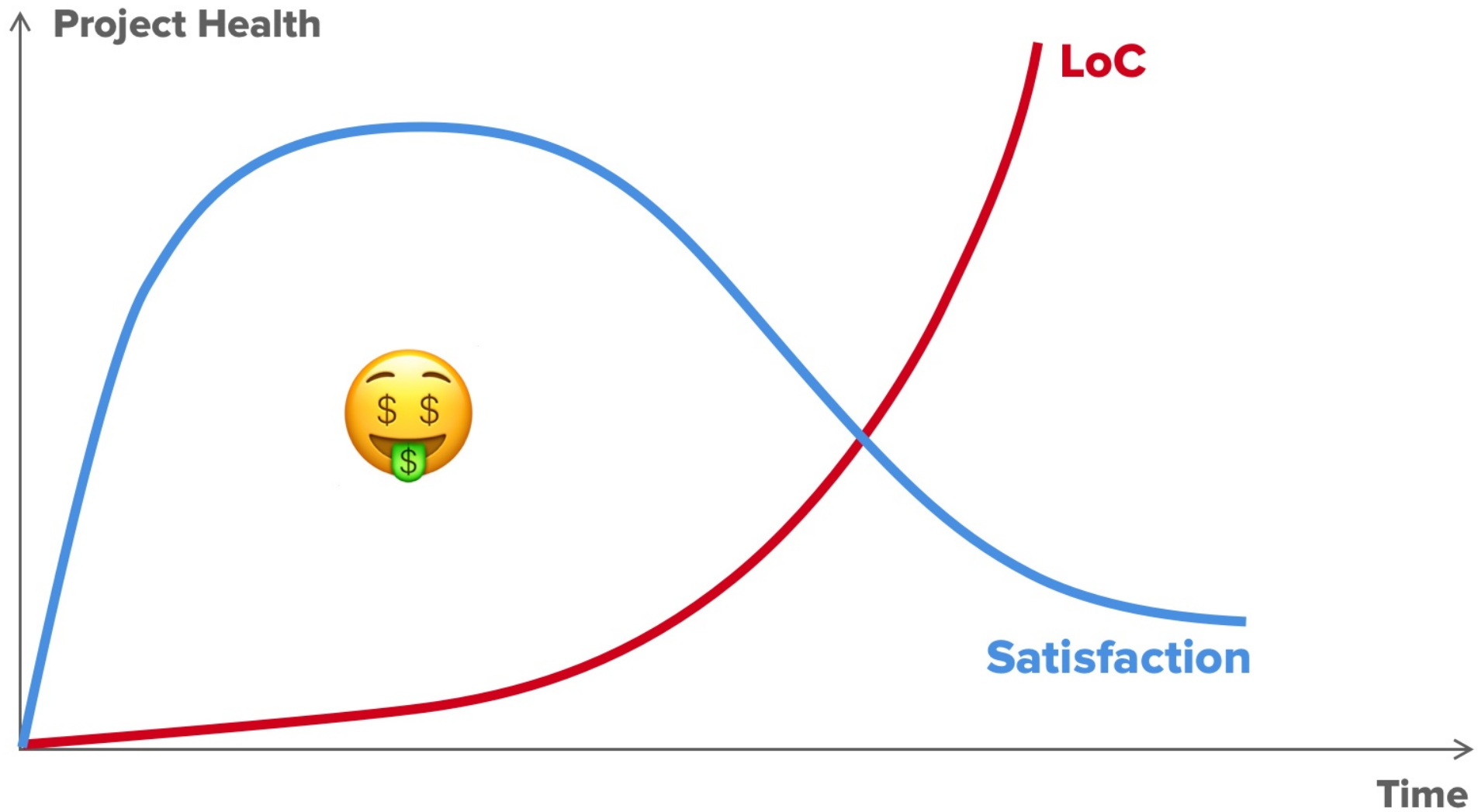
[github.com/jetrockets](https://github.com/jetrockets)

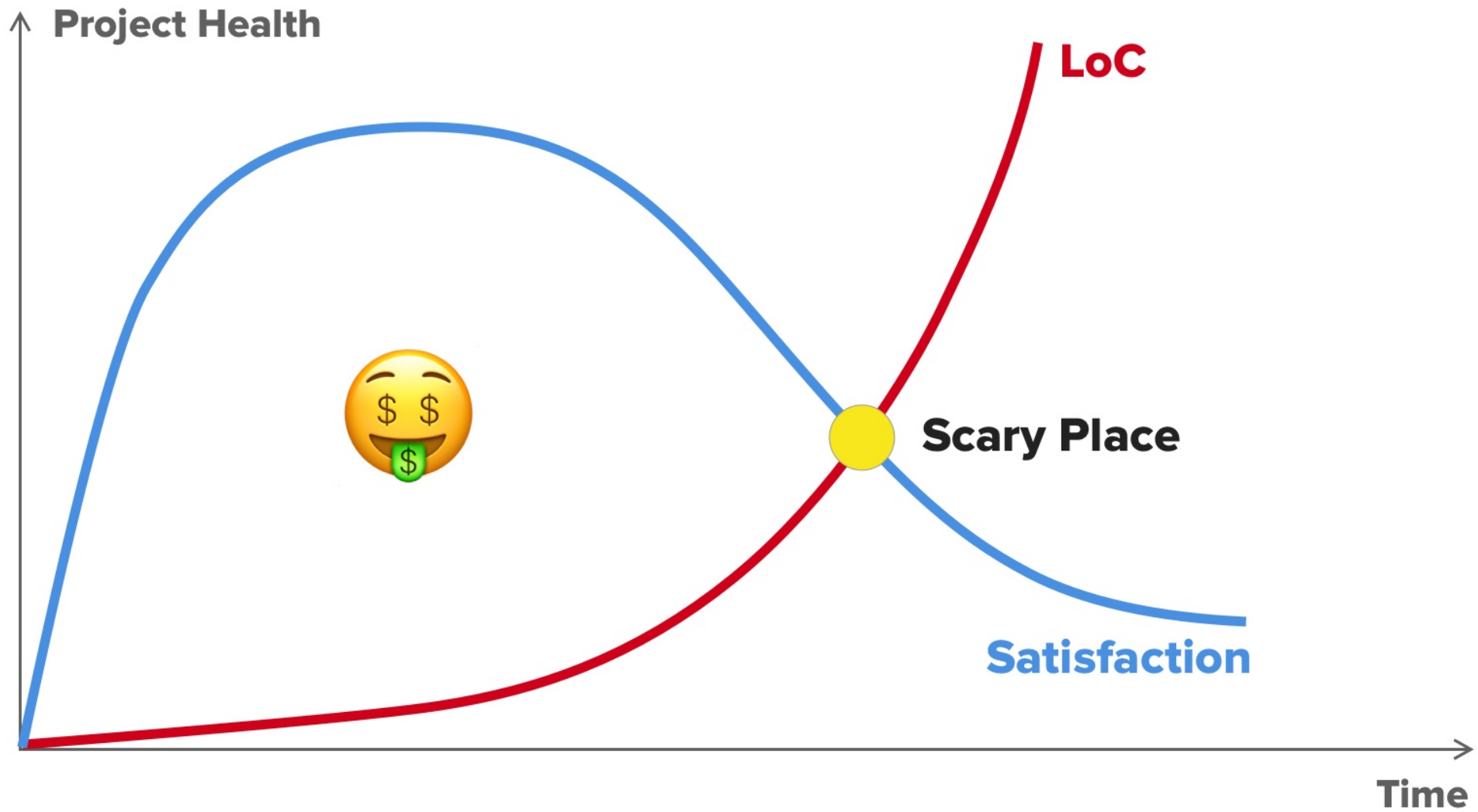
[github.com/igor-alexandrov](https://github.com/igor-alexandrov)

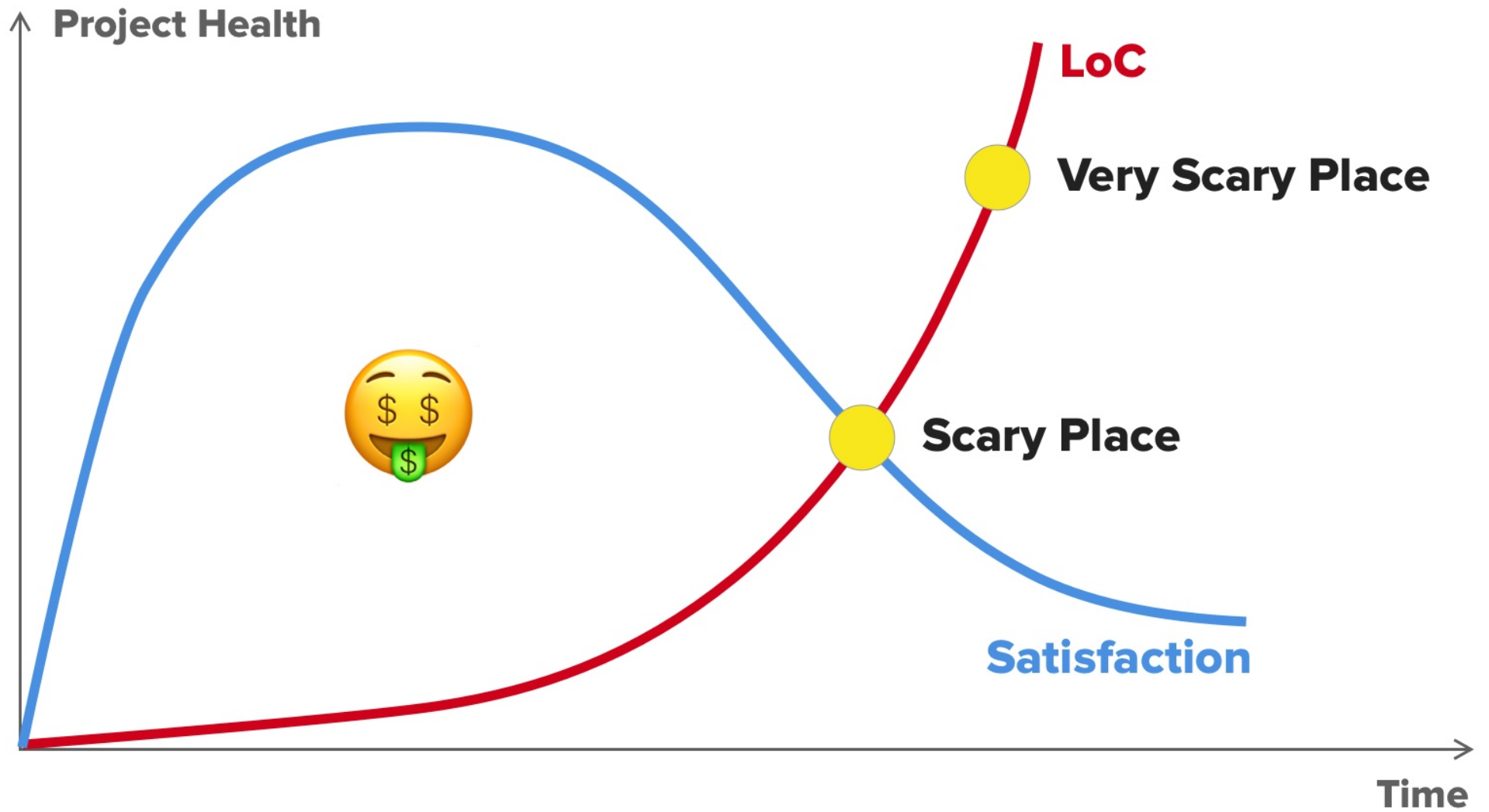














**Знакомо?**

**Разве проблема  
в Ruby?**

# **Проблема N°1 — архитектура**

```
class DocumentsController < ApplicationController
  def create
    @document = Document.new(params[:document])

    # 90% of developers don't care about 'magic'
    if @document.save
      # ...
    else
      # ...
    end
  end
end
```

```
# app/models/documents.rb
```

```
619 # #####
```

```
620 # Callbacks
```

```
621
```

```
622 before_validation :on => :create do
```

```
623   self.scorer = applicant
```

```
624   self.closing_status ||= 'likely'
```

```
    # ...
```

```
728
```

```
729 # #####
```

```
730 # Class Methods
```

# **Приложение — цепочка сервисов**

**Запрос => [\*] => [\*] => Ответ**

**Приложение —  
цепочка функций**



```
class CreateDocument
```

```
  # ...
```

```
end
```

```
class CreateDocument
  def call
    # ...
  end
end
```

```
class CreateDocument
  def call(document, params)
    # ...
  end
end
```

```
class CreateDocument
  def call(document, params)
    document.attributes = params
    document.save!
  end
end
```

```
class CreateDocument
  def call(document, params)
    document.attributes = params
    document.save!
  end
end
```

```
class CreateDocument
  def call(document, params)
    document.attributes = params

    # before_create
    rename_document(document) if document.valid?

    document.save!
  end
end
```

```
class CreateDocument
  def call(document, params)
    document.attributes = params
    rename_document(document) if document.valid?

    document.save!

    # after_commit on: :create
    make_async_call_with_document(document)
  end
end
```

# Pros

---

- Единый интерфейс сервисов;
- частично избавились от колбеков;
- тестирование функциональных объектов – проще;
- композиция сервисов.



# Cons

---

- Валидации остались в моделях:

```
document.save(validate: false)
```

```
document.from_crm = true
```

```
document.save
```

- композиция сервисов получалась не всегда, получалось дублирование;
- тестирование не стало проще из-за невозможности управлять зависимостями;
- не получалось сделать композицию объектов (Domain).

**Silver Bullet?**

# **Form Object!**

<http://trailblazer.to/gems/reform>

```
class CreateDocument
  def call(document, params)
    form = CreateDocumentForm.new(document)

    if form.validate(params)
      document = form.sync
      rename_document(document)
    end

    document.save!
  end
end
```

```
class CreateDocument
  attr_reader :form_class
  attr_reader :rename_document

  def initialize(form_class:, rename_document:)
    @form_class = form_class
    @rename_document = rename_document
  end

  def call(document, params)
    # ...
  end
end
```

```
class CreateDocument
  def call(document, params)
    form = form_class.new(document)
    if form.validate(params)
      document = form.sync
      rename_document(document)

      document.save!
    end

    document
  end
end
```

# **dry-validation**

<https://dry-rb.org/gems/dry-validation>

```
# app/services/document/create_document.rb
```

```
class Document::CreateDocument
```

```
  def call(document, params)
```

```
    result = schema(document).(params)
```

```
    if result.success?
```

```
      # ...
```

```
    end
```

```
  end
```

```
end
```

```
class CreateDocumentSchema < Dry::Validation::Schema
```

```
end
```

*# Initialize once*

```
command = CreateDocument.new(  
  rename_document: Document::RenameDocument.new  
)
```

*# Use several times*

```
command.call(passport, params[:passport])  
command.call(visa, params[:visa])
```



**Profit?**

**Не совсем...**

```
class Document::CreateDocument
  attr_reader :rename_document
  attr_reader :upload_document_to_perfect_audit
  attr_reader :update_expiration_dates
  attr_reader :convert_document_to_pdf
  attr_reader :create_affiliations
  attr_reader :update_cpl_records
  attr_reader :update_application_status

  # hundreds of other stuff here
  # ...
end
```

**Silver Bullet?**

**dry-container**

<https://dry-rb.org/gems/dry-container>

# **dry-container**

- IoC Container
- Stubs!
- Thread safe

```
# app/containers/global_container.rb
module GlobalContainer
  extend Dry::Container::Mixin

  namespace('services') do
    register('create_document') do
      Document::CreateDocument.new(
        rename_document: self['services.rename_document']
      )
    end

    register('rename_document') do
      Document::RenameDocument.new
    end
  end
end
```

*# создание документа от заёмщика*

```
command = GlobalContainer['services.create_document']
```

```
command.rename_document    # Document::RenameDocument
```

```
command.class              # Document::CreateDocument
```

*# создания документа из CRM*

```
command = GlobalContainer['crm.services.create_document']
```

```
command.rename_document    # Crm::Document::RenameDocument
```

```
command.class              # Document::CreateDocument
```

**Profit?**



```
# app/lib/containers/global_container.rb
1 module GlobalContainer
2   extend Dry::Container::Mixin
      namespace( 'services' ) do
        # ...
2044   end
2045 end
2046 end
```

**2046 LOC???**

**ИОС головного мозга**

```
# app/lib/containers/api_v2_container.rb
```

```
1 module ApiV2Container
```

```
2   extend Dry::Container::Mixin
```

```
   namespace( 'auth' ) do
```

```
     # ...
```

```
124   end
```

```
125 end
```

```
126 end
```

```
# app/services/document/create_document.rb
class Document::CreateDocument
  def call(document, params)
    result = schema(document).(params)
    if result.success?
      # ...
    else
      return false
    end
  end
end
```

```
# app/services/document/create_document.rb
class Document::CreateDocument
  def call(document, params)
    result = schema(document).(params)
    if result.success?
      # ...
    else
      return false
    end
  end
end
end
```

**Silver Bullet?**

**dry-monads**

<https://dry-rb.org/gems/dry-monads>

# **dry-monads**

- ~~Монада — это то, что все знают, но не могут объяснить;~~
- монада – контейнер, в котором есть результат с объектом;
- chaining;
- синтаксический сахар.

```
class CreateDocument
  def call(document, params)
    result = schema(document).(params)
    if result.success?
      # ...
      Success(document)
    else
      Failure(result.errors)
    end
  end
end
```

```
result = command.call(document, params[:document])

case result
when Dry::Monads::Success
  # ...
when Dry::Monads.Failure(LendingOne::ValidationError)
  # ...
when Dry::Monads::Failure
  # ...
end
```



# Pros

---

- Тестирование функциональных объектов;
- простое расширение логики;
- композиция – основа моделирования процессов;
- масштабирование и слабая связанность компонентов приложения.

# Cons

---

- Имитация функций объектами;
- концептуально множатся сущности (которые не являются сущностями по факту);
- IoC с доступом по строкам;
- можно написать монадический метод и вернуть не монадическое значение;

Проблема N°2.

**Nobody wants  
to ~~think~~ hack**

**ActiveSupport**

**number\_to\_currency**

# number\_to\_currency

```
report = MemoryProfiler.report do
  # ActiveSupport::NumberHelper::NumberToCurrencyConverter
  number_to_currency(BigDecimal(1543679.753))
end
```

```
report.total_allocated_memsize
# 10168
```

**10 килобайт?**

# FasterSupport

[https://github.com/jetrockets/faster\\_support](https://github.com/jetrockets/faster_support)

# FasterSupport

---

```
report = MemoryProfiler.report do
  d = BigDecimal(1543679.753)
  FasterSupport::Numbers.number_to_currency_n_u(d)
end
```

```
report.total_allocated_memsize
# 80
```



# Performance

---

Checking for BigDecimal

Warming up -----

ActiveSupport 874.000 i/100ms

FasterSupport 30.613k i/100ms

Calculating -----

ActiveSupport 9.263k ( $\pm$  5.2%) i/s - 46.322k in 5.015584s

FasterSupport 386.647k ( $\pm$  2.9%) i/s - 1.959M in 5.071773s

Comparison:

FasterSupport: 386647.3 i/s

ActiveSupport: 9262.8 i/s - 41.74x slower

# Counter Cache

---

```
d = Document.first  
d.comments_count # => 0  
Document.increment_counter(:comments_count, 1)  
  
d.comments_count # => 0
```

# Counter Cache

---

```
d = Document.first  
d.comments_count # => 0  
d.increment(:comments_count)  
d.reload
```

```
d.comments_count # => 0
```

PostgreSQL

# **RETURNING** **statement**

# RETURNING statement

---

```
UPDATE "documents"
```

```
SET
```

```
    "comments_count" = COALESCE("comments_count", 0) + 1
```

```
WHERE
```

```
    "documents"."id" = 343195
```

```
RETURNING
```

```
    "id", "comments_count"
```

# RETURNING statement

---

```
d = Document.first
```

```
d.comments_count # => 0
```

```
d.increment_counter_with_value(:comments_count)
```

```
d.comments_count # => 1
```

# PostgreSQL RETURNING statement

[activerecord-update\\_counters\\_with\\_values](#)

# Links

---

[jetrockets.pro](https://jetrockets.pro)

[github.com/jetrockets](https://github.com/jetrockets)

[github.com/igor-alexandrov](https://github.com/igor-alexandrov)



*JetRockets*