



Riffing on Rails: Sketch your way to better designed code



**KASPER
TIMM HANSEN**

Ruby Consultant



TUE MAY 7, 2024



4:00 PM

SPEAKER: Kasper Timm Hansen

Speaker Bio: Kasper is an independent Rails consultant with a strong focus on clear naming and Domain Modeling. He used to be on the Rails team where he reviewed & merged over 1000 contributor PRs. Kasper is still involved in Open Source with his own gems that take Rails further. He's contracted for Bullet Train too, so he's more than happy to talk about that work.

He's also developing a new software development technique called Riffing, that he's amped to see if it could fit and level up you and your team's productivity.

ROOM: 140 E-G

Riffing on Rails

Sketch your way to better
designed code

Hi, it's Kasper 🖐️

**Today let's talk about code
design**

Figuring out

the names, responsibilities,
and relationships

**How do we
get better
designs?**

I don't actually know.

Sorry.

**Better design requires a
process**

**How this
came about**

Overwhelmed
→ Anxious

**"I just want to
ship"**

My way out
with riffing

1. what riffing is
2. benefits
3. obstacles
4. demo

Our meal plan

Non compete

Just one more tool in your toolbox

how it works

- use a single Ruby file
- lean on conventions as much as possible
- use near runnable code

we're aiming to

- learn about the problem
- surface unknowns

Sessions are
usually ~40-60min

we do a

Lil' Demo

as a treat

```
# Users can favorite other objects in our system

class User
  has_many :favorites

  def favorite(record) = favorites.create!(record:)
end

class User::Favorite
  belongs_to :user
  belongs_to :record, polymorphic: true
end

class Users::FavoritesController
  def create
    Current.user.favorite SignedGlobalID.find(params[:id],
only: [Post, Comment], for: :user_favorites)
  end
end
```

Riffing

Benefits

Fewer sunk costs
Low Friction

**Compare
Alternatives**

**more XP per
feature**

**Embody the
problem**

**Get things out
of my head**

**Negative
Emotions**
are now really useful

Riffing

Obstacles

**You need to
know some
APIs**

A blank file

Uhh, no tests

Getting stuck

**When in
doubt,
experiment
more**

Let's riff

Wrap up

Receipts

~6 hours from live sessions

youtube.com/@kaspth-final1-psd

How you can start to riff

Pick out UI you like

ex. Overcast settings, GitHub sponsors

Start with a coworker

Team leads

- write up a prompt
- ask team to riff for 30 min
- compare

Cheap to run

so it's never a loss

One more thing: make it executable

```
require "bundler/inline"; gemfile do
  gem "activerecord", require: "active_record"
  gem "sqlite3"
end

ActiveRecord::Base.establish_connection adapter: "sqlite3", database: ":memory:"
ActiveRecord::Base.logger = Logger.new STDOUT

ActiveRecord::Schema.define do
  create_table :users

  create_table :posts do |t|
    t.references :user, null: false, index: true
    t.string :title, null: false
  end
end

class User < ActiveRecord::Base
  has_many :posts
end

class Post < ActiveRecord::Base
  belongs_to :user
end

user = User.create!
post = Post.create! user:, title: "First"
binding.irb
```

Hack Day

10am-noon

Oaken

2pm-4pm

We riff on your thing

Taking Rails further

active_record-
associated_object

active_job-
performs

Bullet Train

nice_partials

showcase-rails

...

active_record-
ingress

action_controller-
stashed_redirects

Experimental

brb-templates

hercule-poro

Work with me



hey@kaspth.com

github.com/kaspth

twitter.com/kaspth

ruby.social/@kaspth