

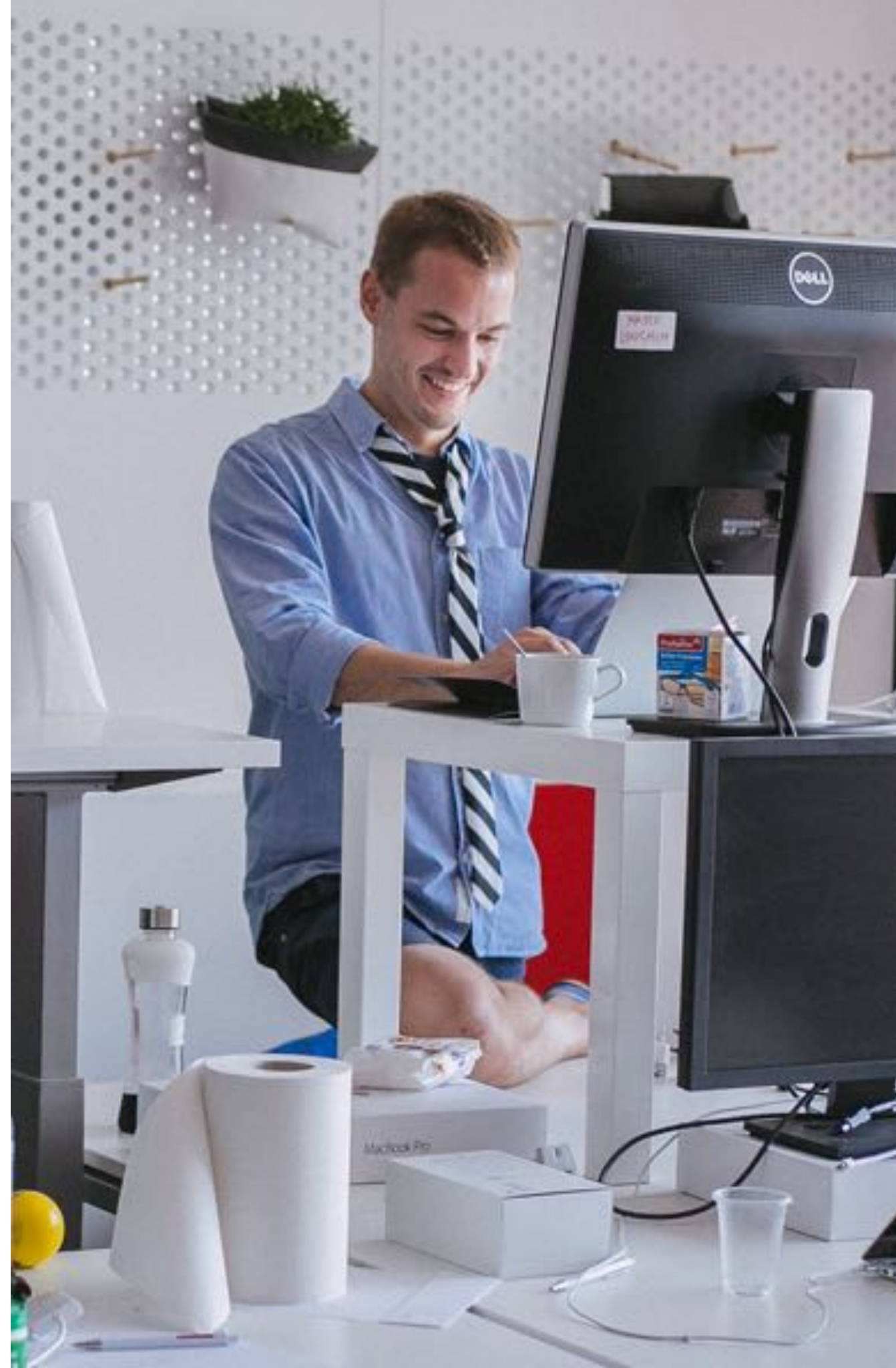
Building a giant

with ~~ember.js~~ legos



Ivan Lučin

Lead Frontend Engineer
@
Productive



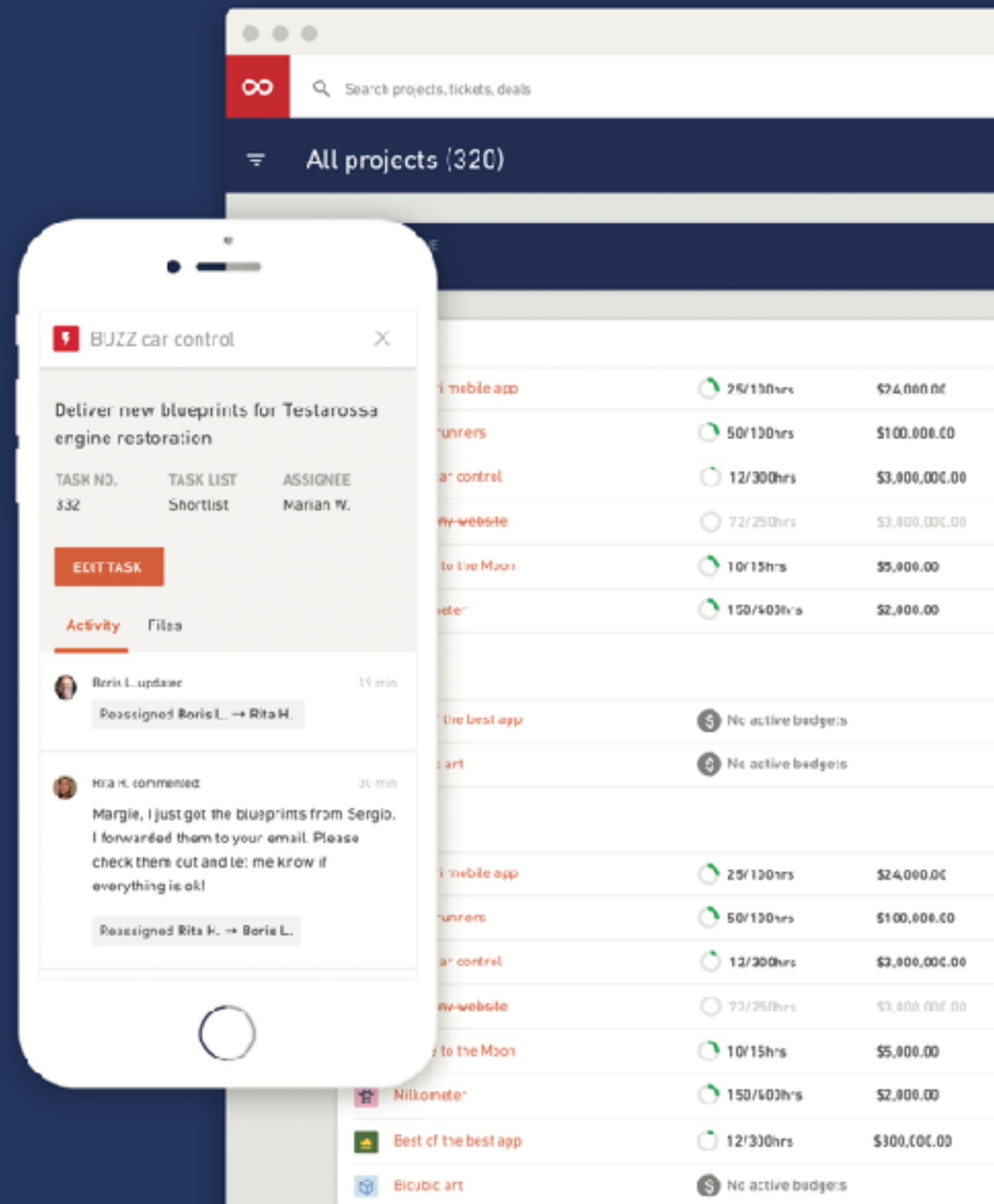
So...
Who's the giant?



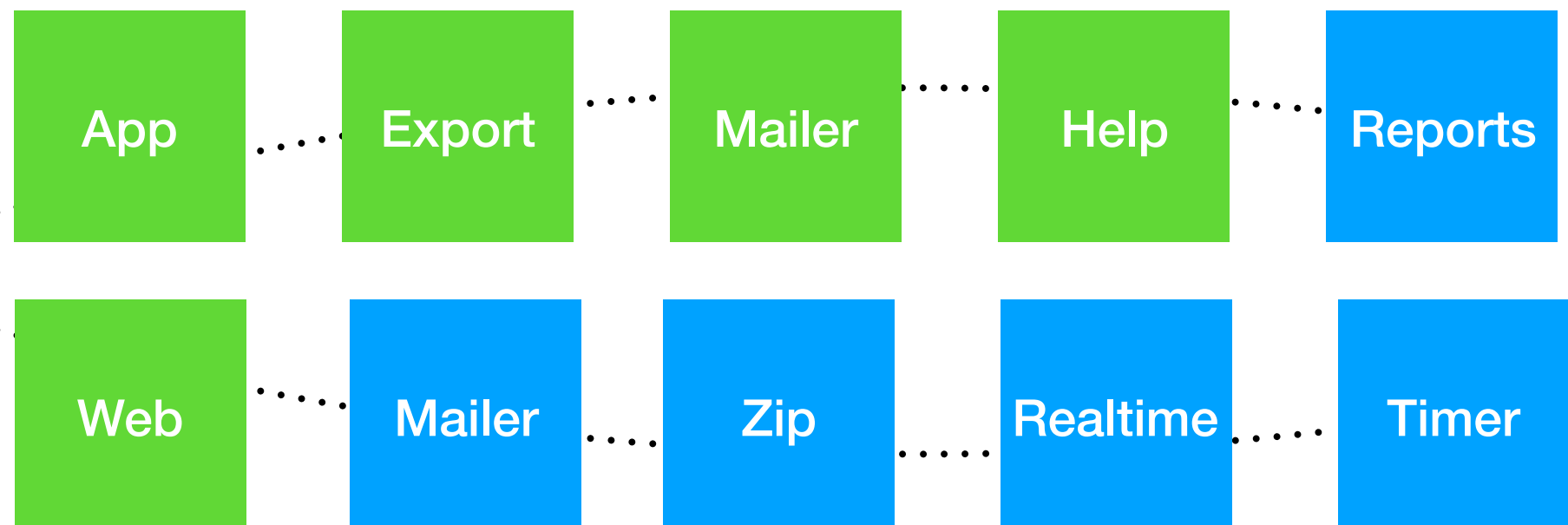
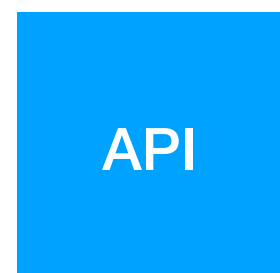
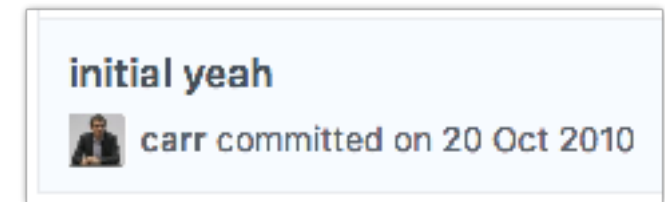
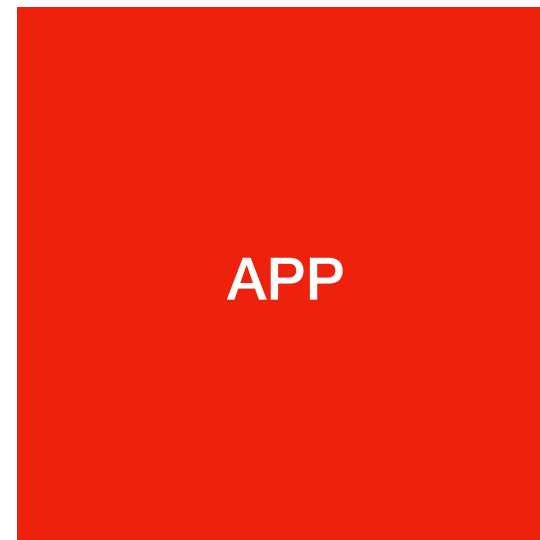
producti^{ve}

The only tool you need to
run a **profitable** agency.

www.productive.io



THEN



NOW



	API	App	Web	Export	...
Age	8 years	3 years	3 years	2 years	
Commits	10,000	13,000	500	600	
LOC	80,000	120,000	10,000	5,000	
Tech	Rails	Ember.js	Handlebars Webpack	Rails	

App		
Age	3 years	
Commits	13,000	
LOC	120,000	
Tech	Ember.js	

Main objective?

Moving forward fast ...

**... without breaking
things.**





ember

- Full-fledged JS framework (like Angular)
- Mature community
- Stability without stagnation
- Glimmer | Ember CLI | Ember Data | Fastboot | Engines



How to communicate with your API?



{json:api}

A SPECIFICATION FOR BUILDING APIS IN JSON

- Supported in Ember & Rails
- Extensible
- Don't waste time thinking about the data serialization

```
{
  "data": [{
    "type": "articles",
    "id": "1",
    "attributes": {
      "title": "JSON API is cool!"
    },
    "relationships": {
      "author": {
        "data": {
          "id": "42", "type": "people"
        }
      }
    }
  }],
  "included": [
    {
      "type": "people",
      "id": "42",
      "attributes": {
        "name": "John",
        "age": 80
      }
    }
  ]
}
```



**How to deal with
complexity and size?**



DOs

- Use simple patterns
- Presentational vs. Controller components
- DDAU - Data down actions up
- BEM - Block Element Modifier
- Prefer KISS over DRY (& others)

Adapter

Model

View

Form

Serializer

Controller

Helper

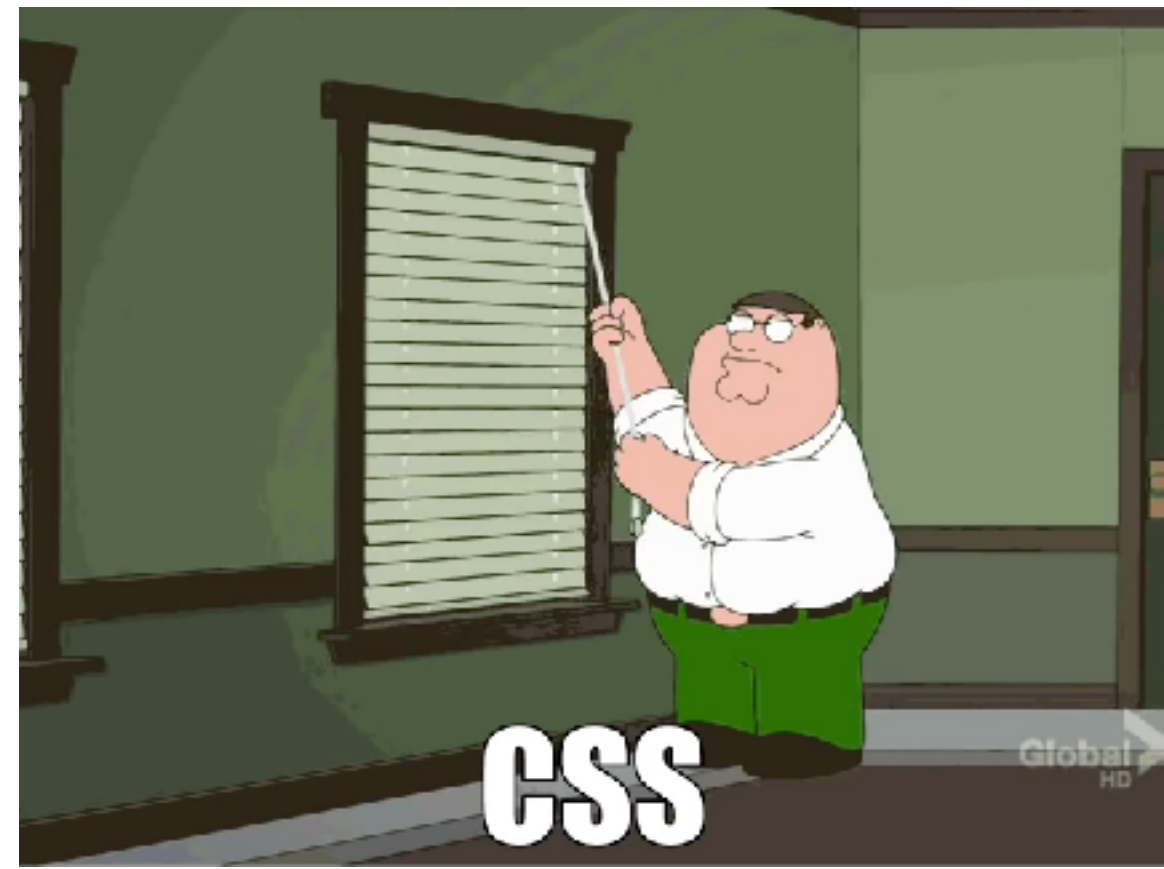
Route

Service



DON'Ts

- Excessive usage of external libraries
 - because they suck more often than not
- Premature / BAD abstractions
- Meta programming
- Over-engineering



Refactor
~30%
of total time



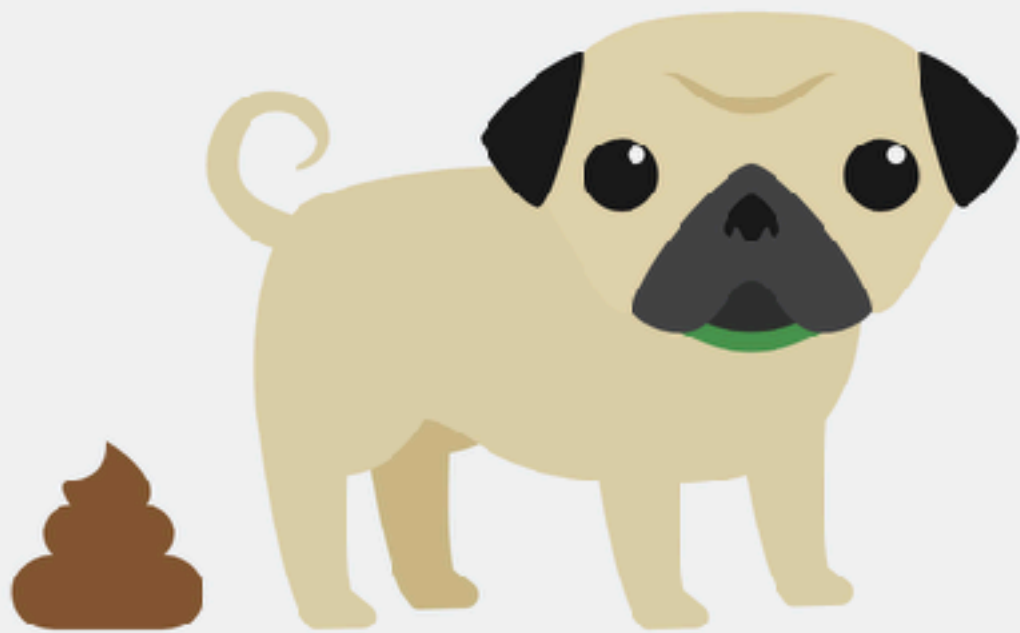
Refactoring rules

1. Write new code separately
2. Deploy both new & old code
3. Split across users / occurrences
4. Make sure nothing is broken
5. Delete old code



**How (not) to break the
app in production?**





Error

Crap.

Our team will take care of this problem, and you should go back home (or check out your console for details).

[GO BACK HOME](#)



Testing

- Ain't nobody got time!
 - Focus on testing integrations
- Always (at least) render the component

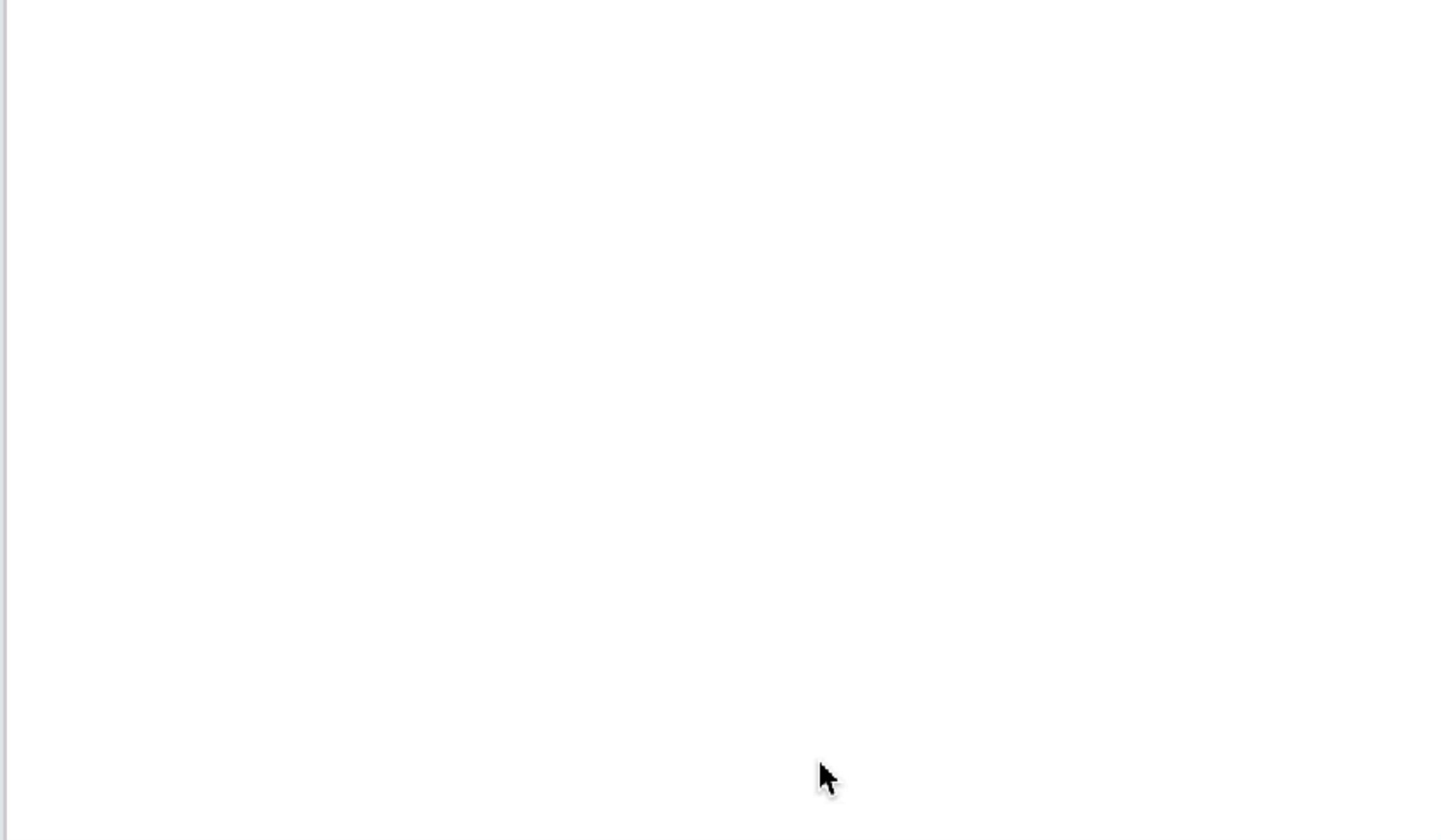
"If you test it, you can say it's a feature and not a bug!"



Acceptance tests






```
test('it allows user to login', async function(assert) {  
  loginServerSetup(server);  
  
  await visit('/login');  
  fillIn(ts('login-form-email-field form-input-field-input'), 'valid@email.com');  
  fillIn(ts('login-form-password-field form-input-field-input'), 'valid-password');  
  await click(ts('login-form form-actions-primary-button'));  
  
  assert.equal(currentPath(), 'organization.dashboard.widgets.index');  
});
```





Monitor for errors

- <https://bugsnag.com>
or similar

Go to project...		
	Productive Blog JS	0 OPEN
	Productive Electron	6 OPEN
	Productive Ember	84 OPEN
	Productive Mailer	0 OPEN
	Productive realtime	0 OPEN

Continuous Integration

- Setup CI server & automatic deployment
- <https://semaphoreci.com>



**Make users test new
stuff without knowing**



CI Server	Autodeploy branch	API & DB	Domain
STAGING	/	staging	<u>app-staging.productive.io</u>
EDGE	develop	production	<u>edge.productive.io</u>
LATEST	develop	production	<u>app.productive.io</u> + sugar on top
STABLE	master	production	<u>app.productive.io</u>

Feature flags

- Deploy untested stuff to production and hide behind a feature flag
- Avoid “it works on my machine” incidents



Metrics

- Collect metrics about your feature usage
- Instead of adding, remove rarely used stuff



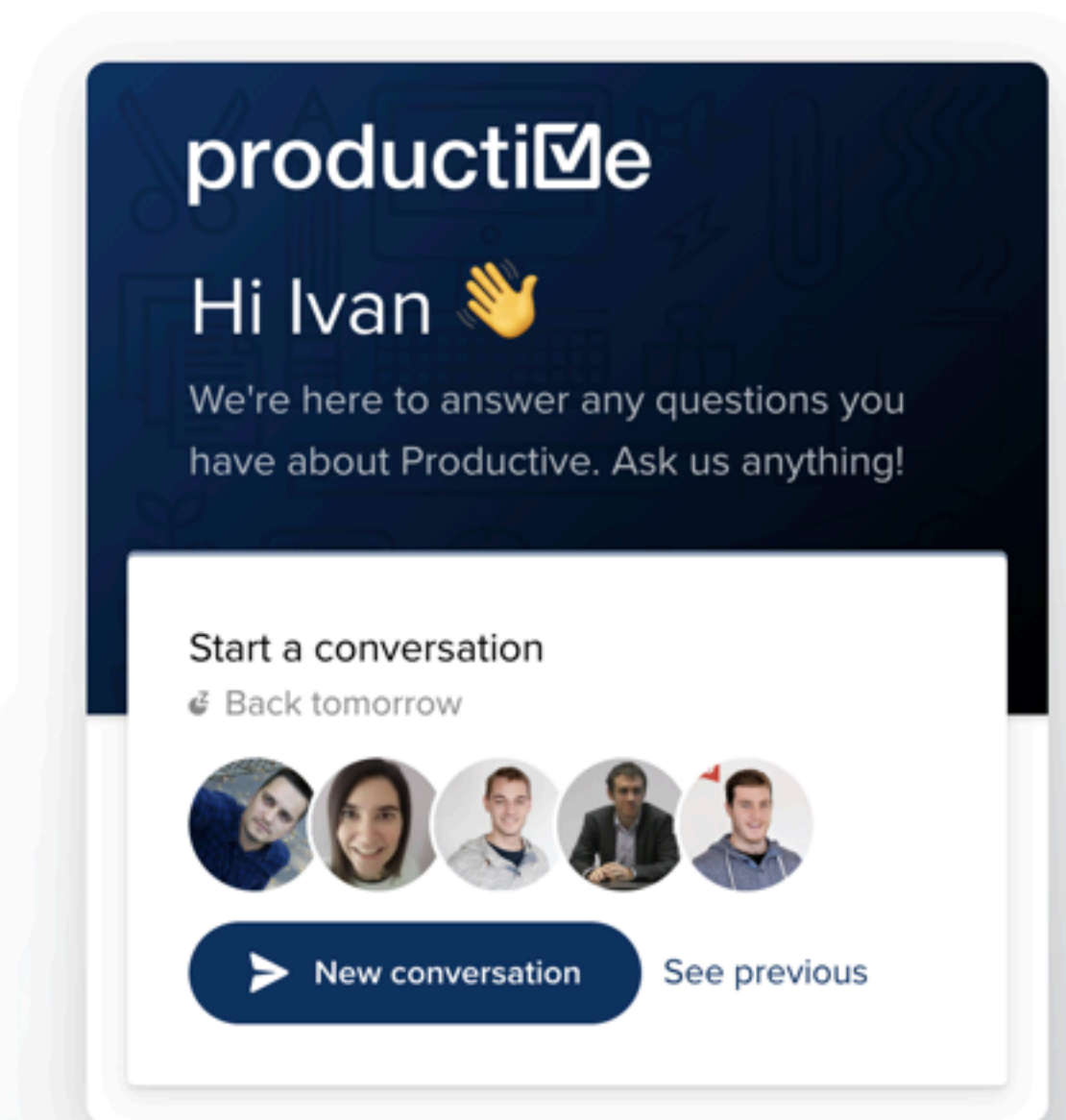
Review & be reviewed

- Code review is a valuable process
- Everybody should be reviewed by everybody
- Always let somebody else test your code



Listen to your clients

- <https://intercom.com> or similar
- They make you reevaluate your work
- Give customer support



Thanks for listening!

Ivan Lučin

Frontend Lead @ Productive

