

GraphQL

A query language for your API.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
project(name: "GraphQL") {  
  tagline  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Agenda

- About Me (Tiago de Castro Freire)
- About Pipefy Company
- GraphQL
 - What is GraphQL?
 - REST vs GraphQL
 - Queries & Mutations
 - Fields
 - Arguments



Tiago de C. Freire

- Developer at [Pipefy](#)
- Past: Codeminer42 / Locaweb
- Ruby / Elixir / Javascript



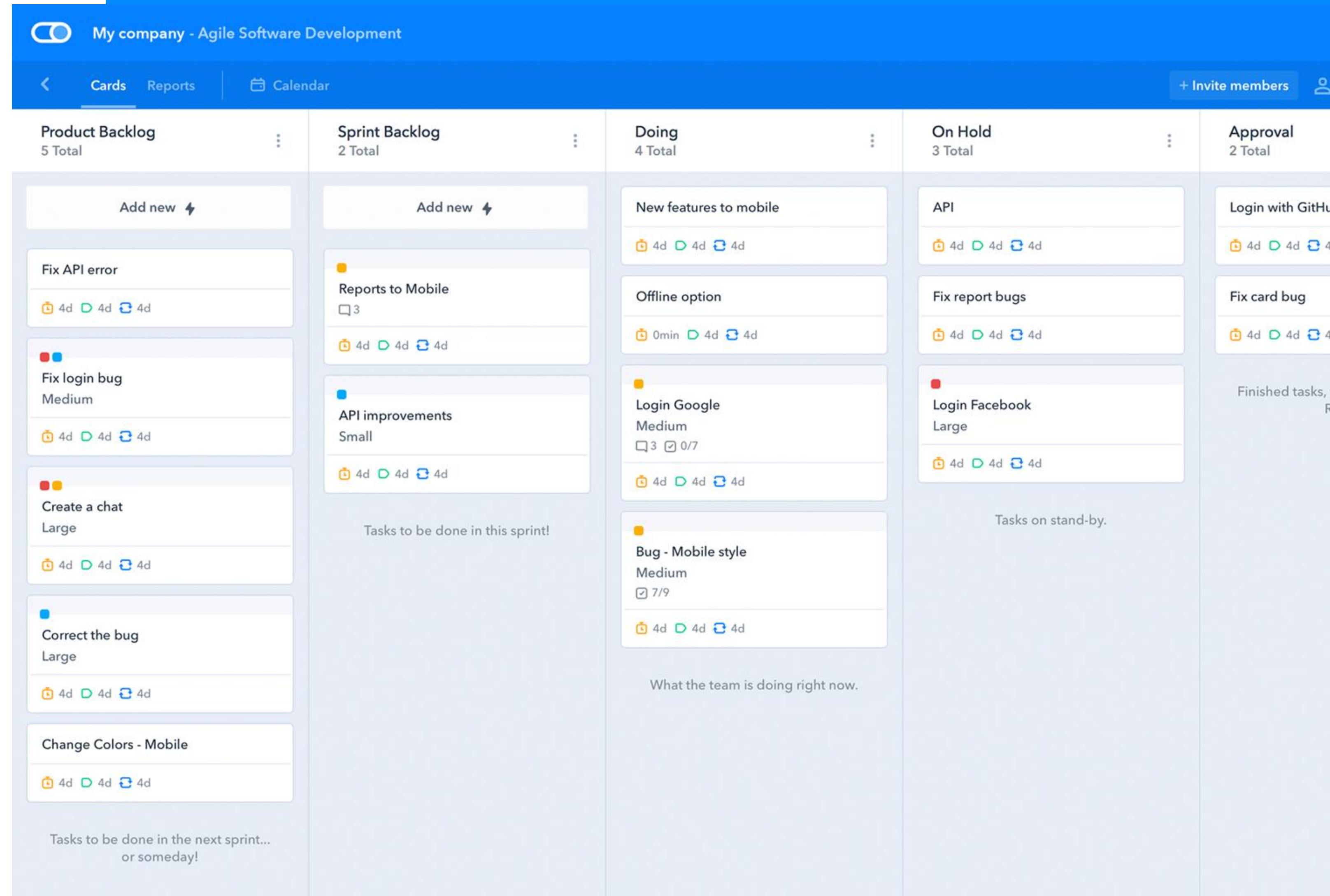
```
VIM - VI Improved
      version 7.4.52
      by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

      Become a registered Vim user!
type  :help register<Enter>  for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version7<Enter>  for version info
```

What is Pipefy?

Pipefy is a cloud-based business process management platform that empowers managers to build and execute any type of workflow. Through a simple Kanban-style interface companies can achieve process excellence without the need for IT, technical skills or professional services.



Pipefy Technologies

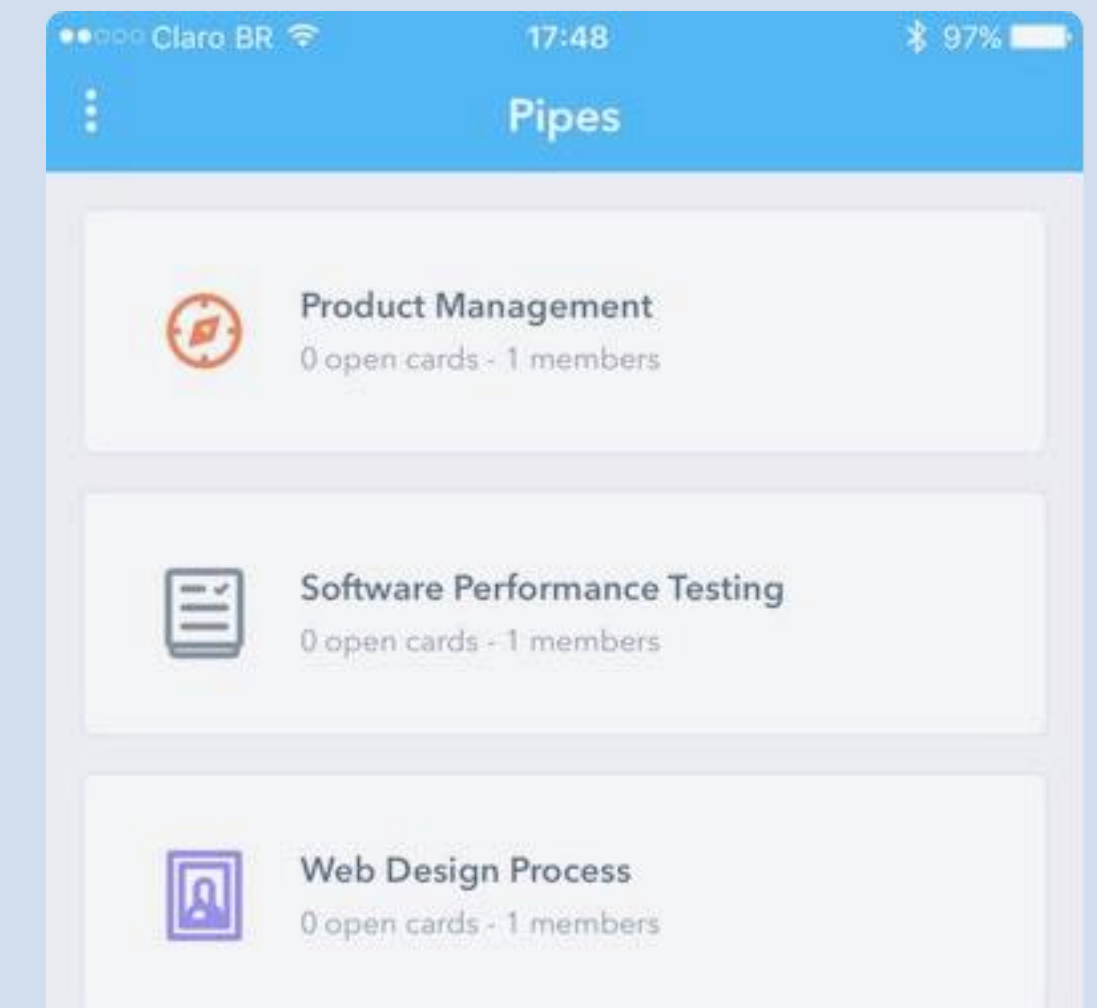
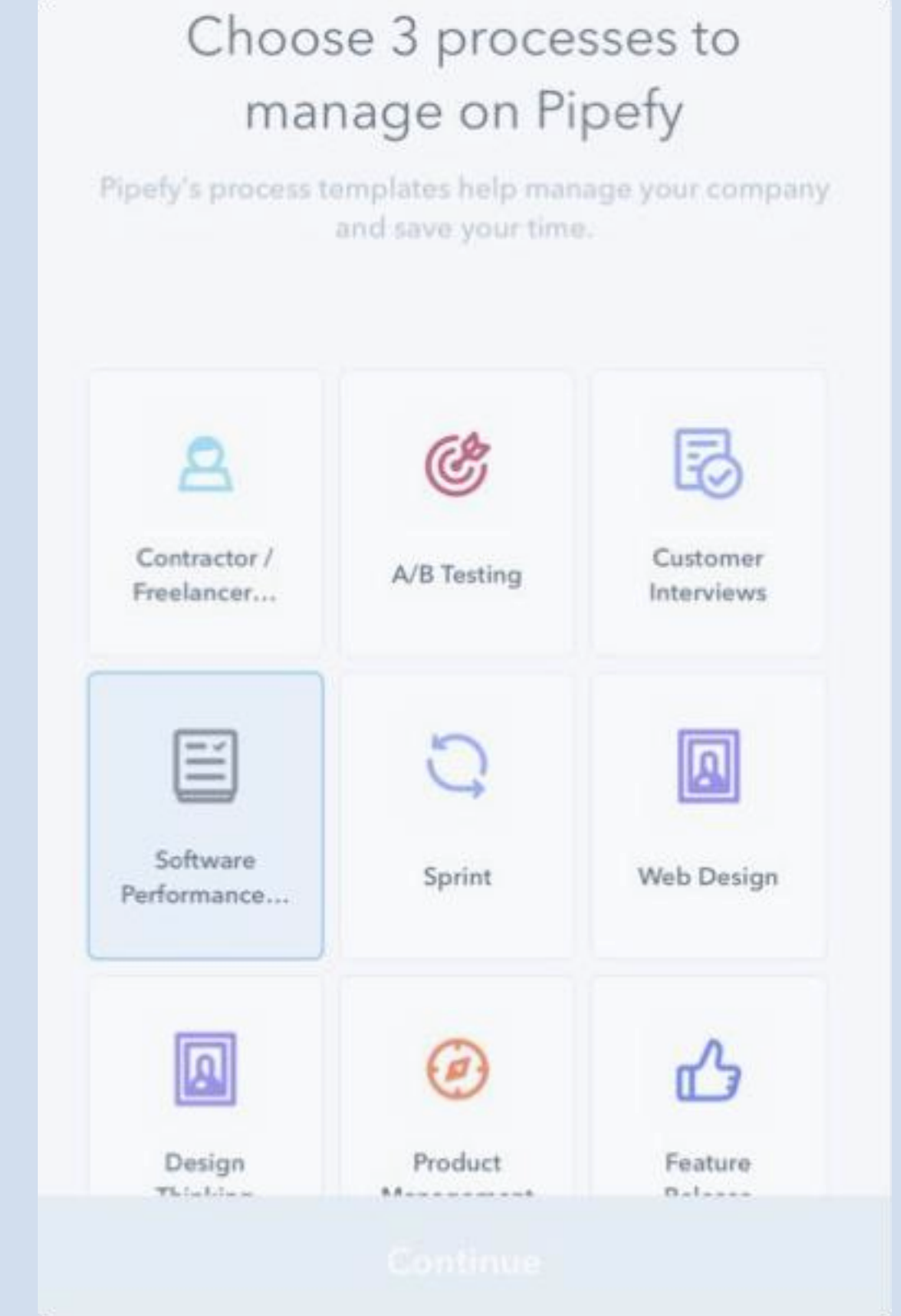
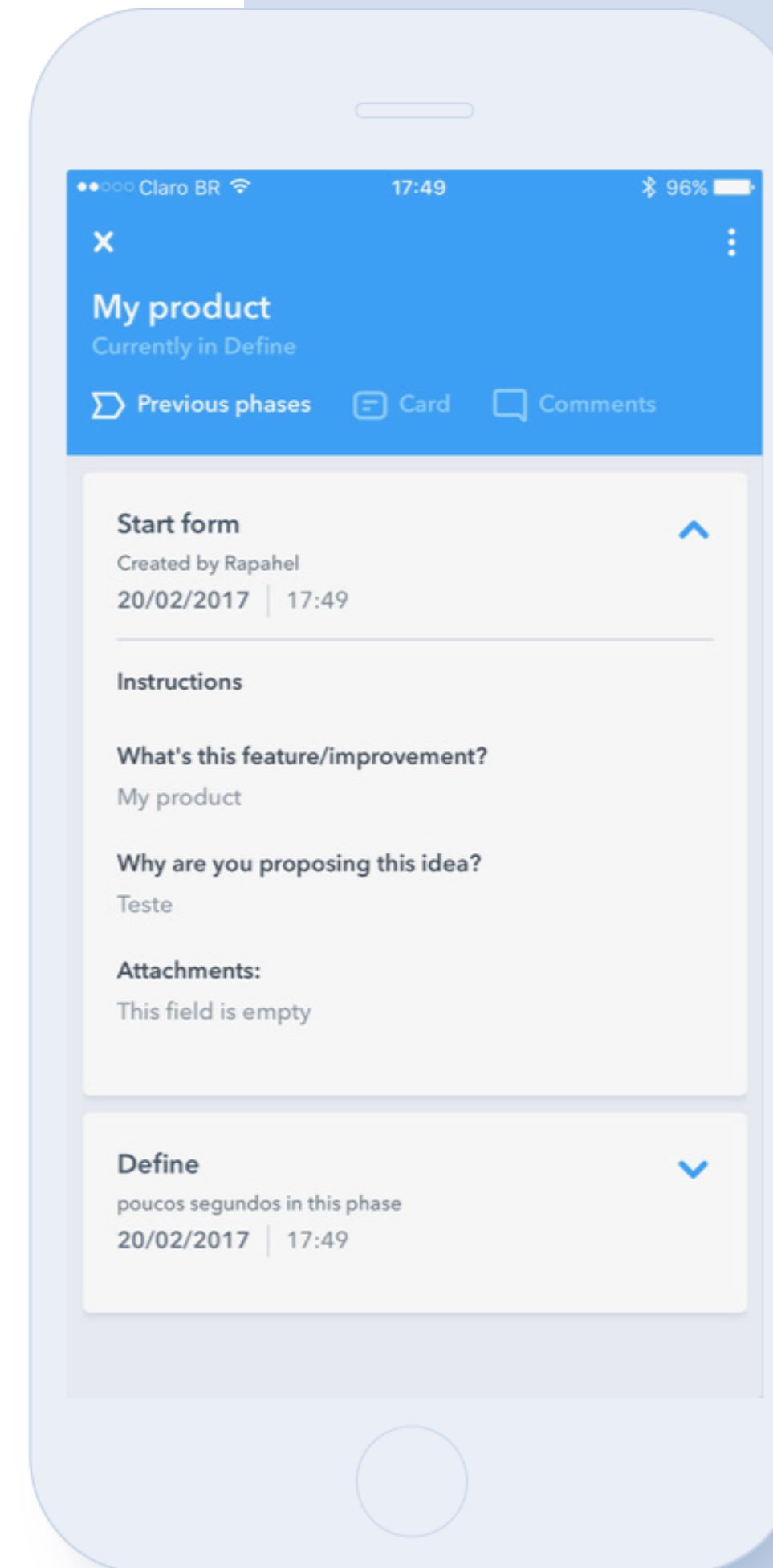
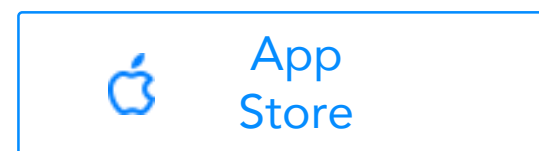
- GraphQL API
- React / Redux / Apollo
- Ruby / Rails
- Elixir / Phoenix



A screenshot of the Pipefy agile software development dashboard. The interface is titled "My company - Agile Software Development" and includes navigation tabs for "Cards", "Reports", and "Calendar". The main area is divided into five columns representing different stages of the workflow: "Product Backlog" (5 Total), "Sprint Backlog" (2 Total), "Doing" (4 Total), "On Hold" (3 Total), and "Approval" (2 Total). Each column contains a list of tasks with details such as priority, status, and due dates. The "Doing" column is currently active, showing tasks like "New features to mobile", "Offline option", "Login Google", and "Bug - Mobile style". The "On Hold" column shows tasks like "API", "Fix report bugs", and "Login Facebook". The "Approval" column shows tasks like "Login with GitHub" and "Fix card bug". The interface also includes an "Add new" button in each column and a "Finished tasks" section at the bottom right.

Pipefy Mobile App

- GraphQL API
- React Native
- Expo
- Apollo Client



GraphQL

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


REST vs GraphQL

- HTTP request methods
- Endpoint
- Body



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Queries

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Mutations

In REST, any request might end up causing some side-effects on the server, but by convention it's suggested that one doesn't use `GET` requests to modify data. GraphQL is similar - technically any query could be implemented to cause a data write. However, it's useful to establish a convention that any operations that cause writes should be sent explicitly via a mutation.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Queries & Mutations

- Fields
- Types
- Arguments



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Fields

GraphQL is about asking for specific fields on objects.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
project(name: "GraphQL") {  
  tagline  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Types

It describes what data can be queried.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
project(name: "GraphQL") {  
  tagline  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Types

- Default: Int, Float, String, Boolean, ID
- Scalar Types
- Enumeration types
- Interfaces
- Union types
- Input types



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Arguments

When you add the ability to pass arguments to fields, things get much more interesting.



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
project(name: "GraphQL") {  
  tagline  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

Pipefy Platform

Extend the potential of Pipefy's process platform, build powerful products with our well documented APIs and Apps.

<https://developers.pipefy.com>

 **pipefy** developers



Thank you ;-)

Twitter: @akaKuruma

Email: tiago.freire@pipefy.com

