

# Move Fast and (Don't) Break Things

**OR**

How to leverage feature flags for fun and profit

Hi, I'm Gonçalo 🖐️

- Senior UI Engineer @ CrowdStrike
  - We are hiring btw 🙄🙄
- I run, climb, and organise codebar
- [heroesofcomputer.science](#) 🧑🏻💻 🧑🏻💻 🧑🏻💻
- [@gnc1morais](#) pretty much everywhere



*Tomek Nieżurawski*





Thank you for “Feature flags: bad practices”

# I'll be talking about:

- What are feature flags
- How do they look like
- Why you should use them
- Trade-offs they bring
- Lifecycle of a flag

# What I **won't** be talking about:

- Ember's own feature flags
- How to set them up in your system

<a href="#">Home</a> / Features	
Features	
<a href="#">Add Feature</a>	
	search Fully Enabled
	admins 10% of actors and 25% of time
	stats 10% of actors
	logging

# How do they look like?

## Simple show/hide

```
{{#if (has-feature-flag 'amazing-new-button')}}  
  <button class="favorite styled" type="button">  
    Add to favorites  
  </button>  
{{/if}}
```

# How do they look like?

## Old/new (HBS)

```
{{#if (has-feature-flag 'amazing-new-button')}}  
  <button type="button" class="favorite styled">  
    Add to favorites  
  </button>  
{{else}}  
  <button type="button">  
    Fav  
  </button>  
{{/if}}
```

# How do they look like?

## Old/new (JS)

```
export default class AwesomeComponent extends Component {  
  @hasFeatureFlag('amazing-new-button') hasNewButton;  
  
  button() {  
    return this.hasNewButton ? 'new' : 'old';  
  }  
  
  calculate() {  
    if (!this.service.hasFeatureFlag('amazing-new-button')) {  
      return 15;  
    }  
  
    return 42;  
  }  
}
```



# How do they look like?

*Old old, new old, old new, new new...*

```
{{#if (has-feature-flag 'amazing-new-button')}}  
  <button type="button" class="favorite styled">  
    Add to favorites  
  </button>  
{{else}}  
  {{#if (has-feature-flag 'new-button')}}  
    <button type="button">  
      Fav  
    </button>  
  {{else}}  
    {{-- Turles all the way down... --}}  
  {{/if}}  
{{/if}}
```

# Why feature flags?

- Keep things hidden until ready
- Use WIP code with production data
- Work on smaller tasks
- Planned phased rollouts

# What are the trade-offs?

- Additional overhead
- Increased complexity
- Multiple environments? Multiple problems

# Lifecycle of a feature flag

## Ideation

- Start with good naming
- Documentation documentation documentation
- Separate frontend and backend flags
- **Keep a record of the imminent flag removal**

# Lifecycle of a feature flag

## Delivery

- Write code with removal in sight
- Ensure your logical forks are simple
- Introduce flags **on by default** in tests 🙄🙄
- Ship with Elephant Carpaccio in mind

# Lifecycle of a feature flag

## Retirement

- Remember when we decided when to remove the flag?
- Remember the Jira ticket or pull/merge request we opened before?
- Remember when we turned the flag on by default on our test suite?

# Summary

- Feature flags are awesome, they allow you to...
- Deploy work in progress without affecting users
- Ship your work faster by finishing smaller tasks
- **But** there are trade-offs
- Write down a team plan of how to deal with them



Thank you!

[twitter.com/gnclmoraais](https://twitter.com/gnclmoraais)