SWEN 438 — Everything as Code

James Quilty

School of Engineering and Computer Science
Victoria University of Wellington

Everything as Code

It's become a core tenet of DevOps:

> *"wherever possible, treat it as code."*

The secret: it's not about the *automation* it's about the culture.

[Image] Figure 1 of Humble *et al.*, *Why Enterprises Must Adopt DevOps to Enable Continuous Delivery*, Cutter IT Journal **24**, 6 (2011).

Everything as Code

It's become a core tenet of DevOps:

> *"wherever possible, treat it as code."*

The secret: it's not about the *automation* it's about the culture.

[Image: Figure 3 of Humble *et al.*, *Why Enterprises Must Adopt DevOps to Enable Continuous Delivery*, Cutter IT Journal **24**, 6 (2011).]

Everything as Code

"DevOps also extends and builds upon the practices of infrastructure as code, . . .

. . . ***the work of Operations is automated and treated like application code, so that modern development practices can be applied to the entire development stream.***

This further enabled fast deployment flow, including continuous integration, continuous delivery, and continuous deployment. . . ."

Humble *et al.*, *The DevOps Handbook* (2016)

Infrastructure as Code

[Image: DevOps Technology: Everything as Code, VMware Cloud https://cloud.vmware.com/community/2021/02/08/devops-everything-as-code/]

The Place of Version Control

[Image: Figure 3 of Humble *et al.*, *Why Enterprises Must Adopt DevOps to Enable Continuous Delivery*, Cutter IT Journal **24**, 6 (2011).]

The Place of Version Control

- Source code
- Infrastructure configuration
- Automated tests
- Database scripts
- Tool chain

The Practicalities

On the Dev side of DevOps:

- Version control
- Coding style standards
- Document standards

Often called “linting”

Version Control with git

git is not a great tool

> *"Git allows a wide variety of branching strategies and workflows. Because of this, many organisations end up with workflows that are too complicated, not clearly defined, or not integrated with issue tracking systems."*

GitLab Flow
`https://docs.gitlab.com/ee/topics/gitlab_flow.html`

Take care of your repository and it will take care of you.

Branching Strategies

You will always have one.

The tension:

> ***Optimise for individual productivity***
>
> *vs.*
>
> ***Optimise for team productivity***

Humble *et al.*, *The DevOps Handbook* (2016)

Branching Strategies

The problems with git

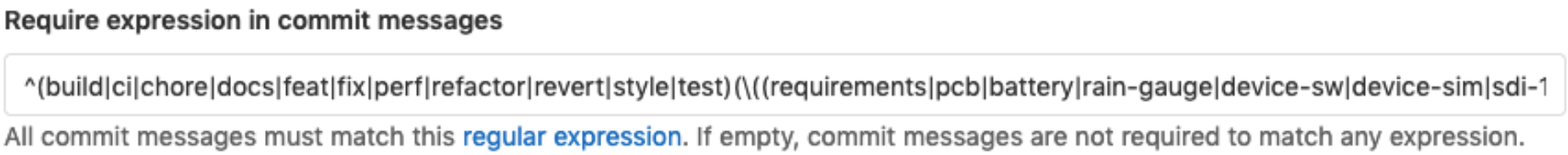
[Image: git flow from GitLab Flow page]

GitLab Flow
`https://docs.gitlab.com/ee/topics/gitlab_flow.html`

Commit Message Standards

Example: `https://github.com/angular/angular/blob/master/CONTRIBUTING.md`

Implementation in GitLab via push rules
(Settings→Repository→ Push rules)



Linting and Pre-Commit

Linting is a form of static analysis.

Automatic enforcement for the benefit of consistency.

Integrate with version control via pre-commit!
`https://pre-commit.com/`

[Image: pre-commit logo from `https://pre-commit.com/`]