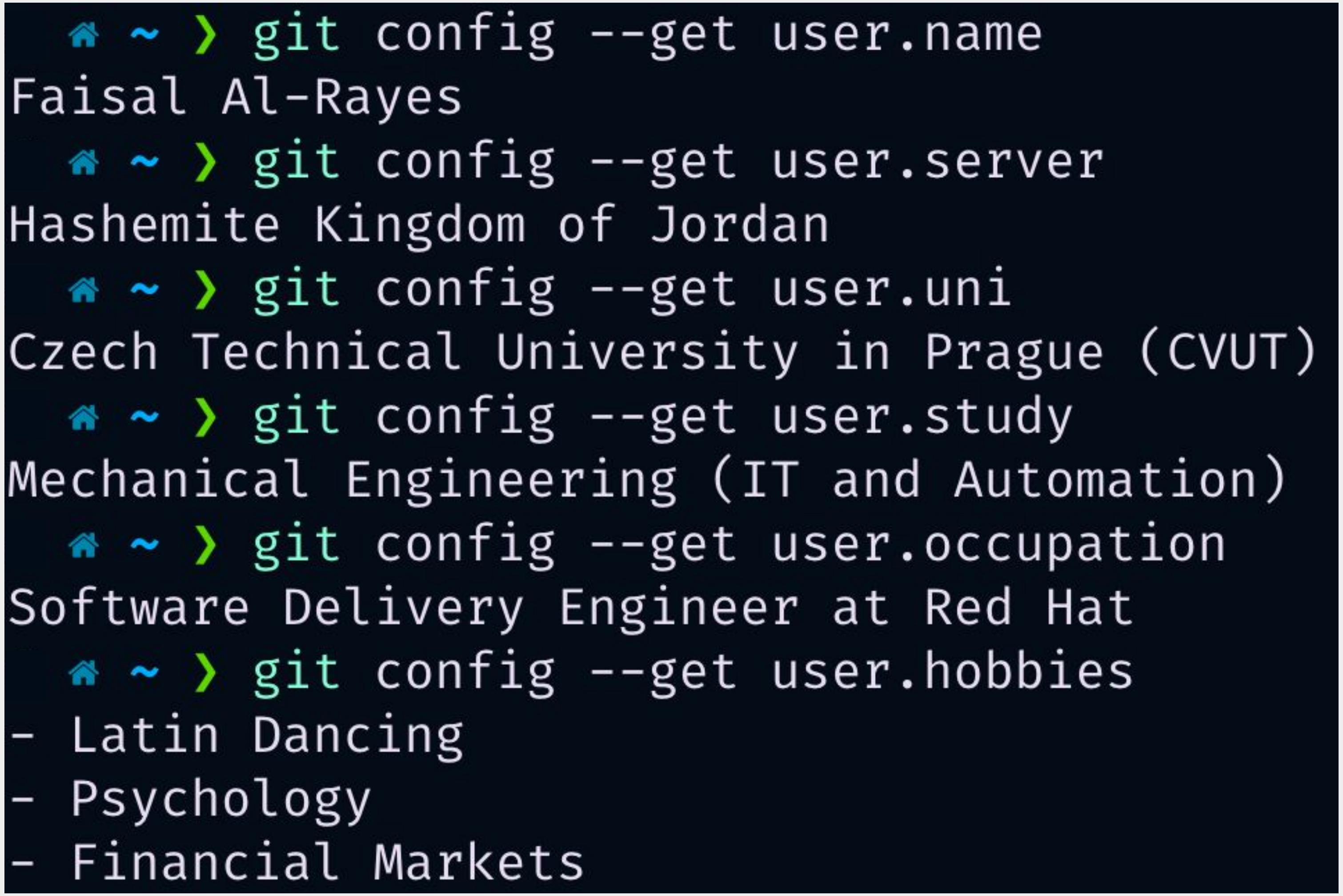
Commit messages as a long term investment

Structuring history for the future.

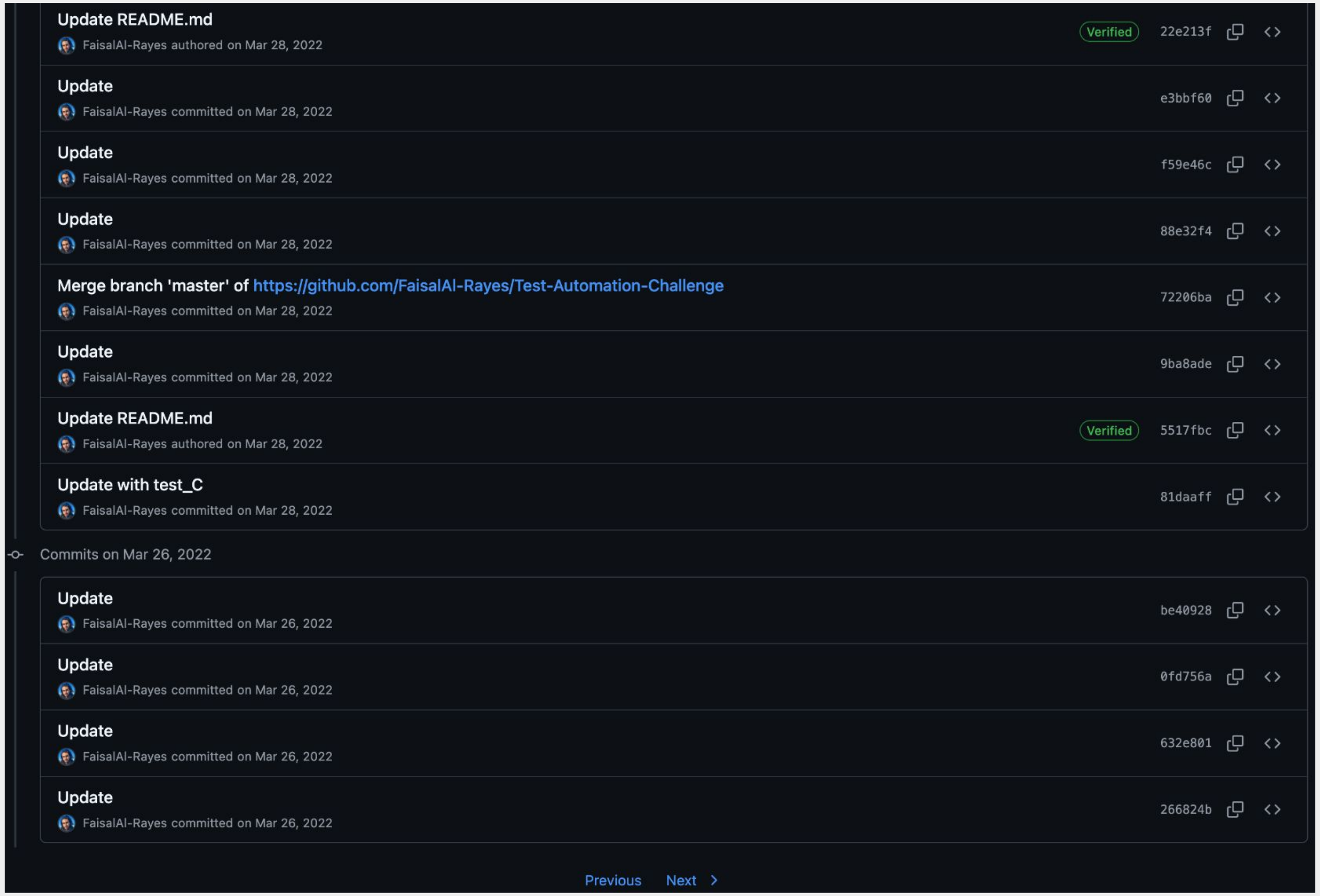
Faisal Al-Rayes
Software Delivery Engineer

whoami



```
  🏠 ~ ❯ git config --get user.name
Faisal Al-Rayes
  🏠 ~ ❯ git config --get user.server
Hashemite Kingdom of Jordan
  🏠 ~ ❯ git config --get user.uni
Czech Technical University in Prague (CVUT)
  🏠 ~ ❯ git config --get user.study
Mechanical Engineering (IT and Automation)
  🏠 ~ ❯ git config --get user.occupation
Software Delivery Engineer at Red Hat
  🏠 ~ ❯ git config --get user.hobbies
- Latin Dancing
- Psychology
- Financial Markets
```

Git log of shame



Update README.md
FaisalAl-Rayes authored on Mar 28, 2022 Verified 22e213f

Update
FaisalAl-Rayes committed on Mar 28, 2022 e3bbf60

Update
FaisalAl-Rayes committed on Mar 28, 2022 f59e46c

Update
FaisalAl-Rayes committed on Mar 28, 2022 88e32f4

Merge branch 'master' of https://github.com/FaisalAl-Rayes/Test-Automation-Challenge
FaisalAl-Rayes committed on Mar 28, 2022 72206ba

Update
FaisalAl-Rayes committed on Mar 28, 2022 9ba8ade

Update README.md
FaisalAl-Rayes authored on Mar 28, 2022 Verified 5517fbc

Update with test_C
FaisalAl-Rayes committed on Mar 28, 2022 81daaff

Commits on Mar 26, 2022

Update
FaisalAl-Rayes committed on Mar 26, 2022 be40928

Update
FaisalAl-Rayes committed on Mar 26, 2022 0fd756a

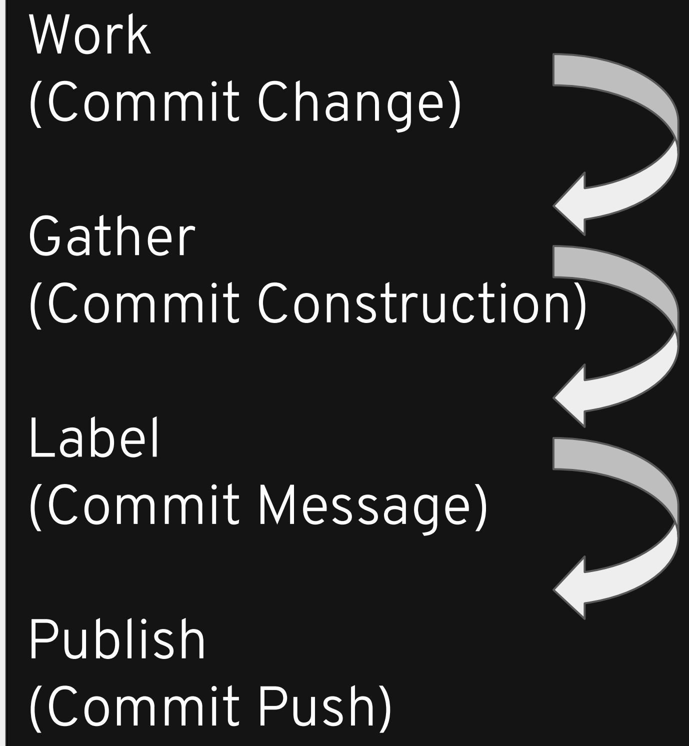
Update
FaisalAl-Rayes committed on Mar 26, 2022 632e801

Update
FaisalAl-Rayes committed on Mar 26, 2022 266824b

What value can I extract from this talk?

- Think at the forefront of your mind of the way you structure your commits.

- Think of commit message structures that would make sense for the projects you work on.



Work
(Commit Change)

Gather
(Commit Construction)

Label
(Commit Message)

Publish
(Commit Push)

Red Hat

Agenda

{

“0”: “Understand commits on a high level”,

“1”: “Commit structure elements”,

“2”: “Commit message example”,

“3”: “tools that could be helpful”

}

Ground up understanding of git commits

Git commits are meant to package a set of changes into a single unit of storage and need a good label to perform their job well.

Commit structure elements

These are some of the most common structure elements when it comes to commit messages.

Commit Title

What does the commit enable/do? (start with uppercase imperative verb)

Reasoning

Why was this commit necessary?

Methodology

How was this change implemented?

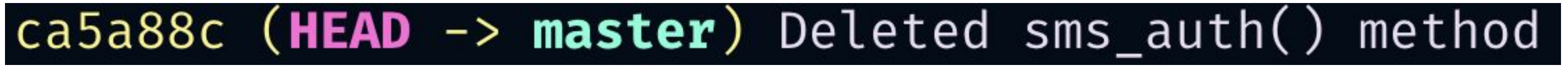
Line lengths

Keep the title length up to 50 characters and the rest up to 72 characters.

Commit title

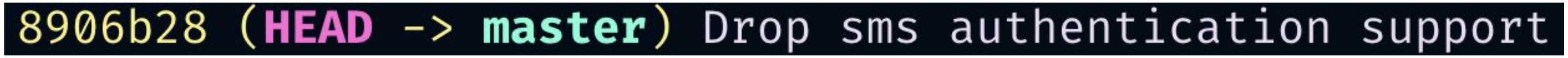
An illustration of what this commit represents.

✗



```
ca5a88c (HEAD -> master) Deleted sms_auth() method
```

✓

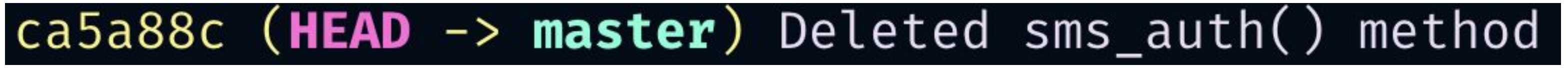


```
8906b28 (HEAD -> master) Drop sms authentication support
```

Commit reasoning

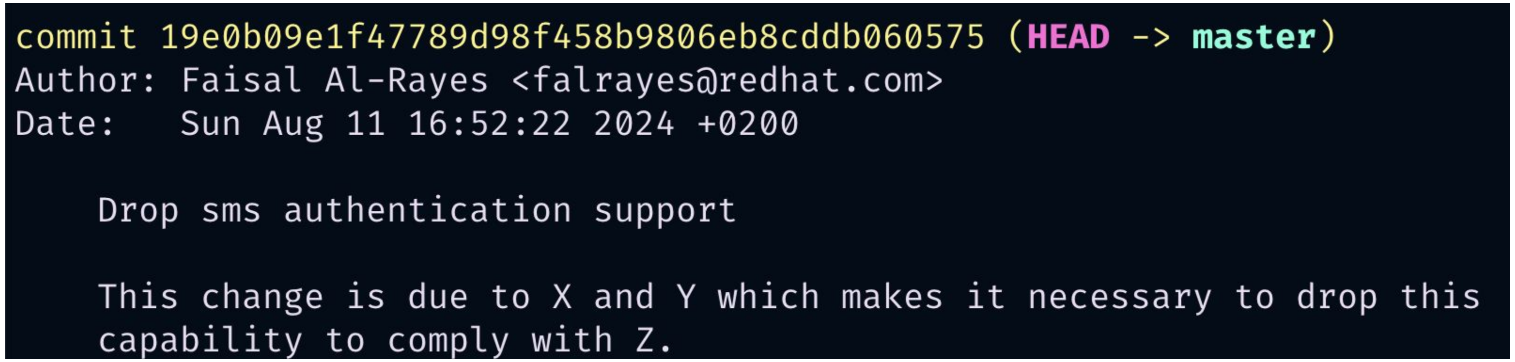
An illustration of why this commit was necessary.

✗



```
ca5a88c (HEAD -> master) Deleted sms_auth() method
```

✓

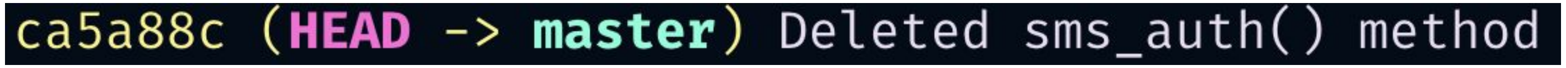


```
commit 19e0b09e1f47789d98f458b9806eb8cddb060575 (HEAD -> master)
Author: Faisal Al-Rayes <falrayes@redhat.com>
Date:   Sun Aug 11 16:52:22 2024 +0200

    Drop sms authentication support

    This change is due to X and Y which makes it necessary to drop this
    capability to comply with Z.
```

Commit methodology

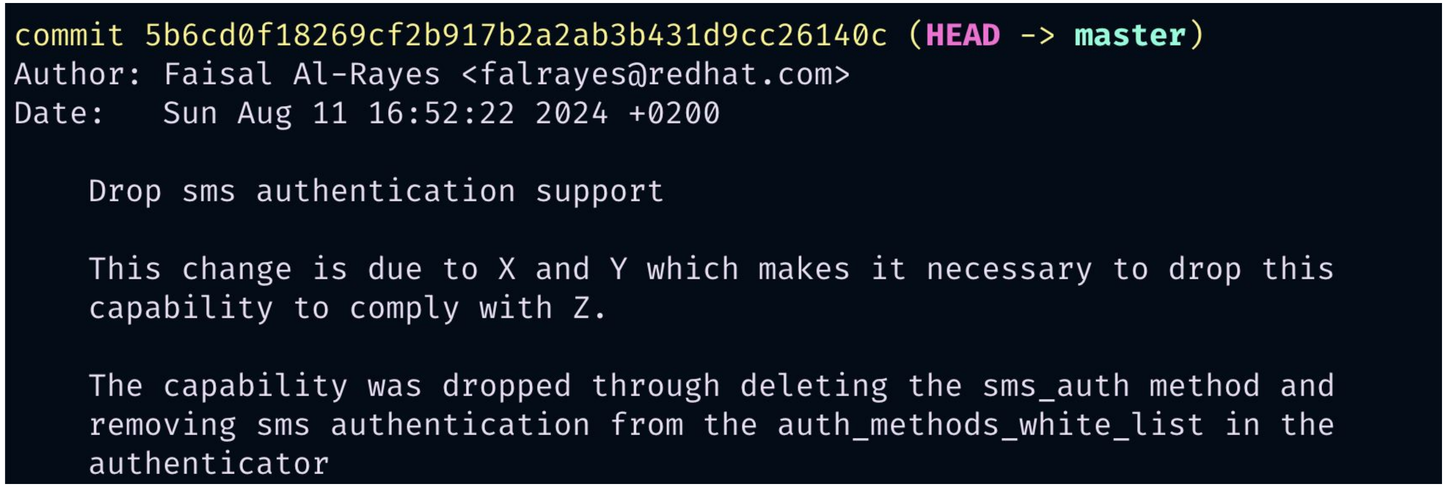
An illustration of commit information on how the change was done.

❌



```
ca5a88c (HEAD -> master) Deleted sms_auth() method
```

✅

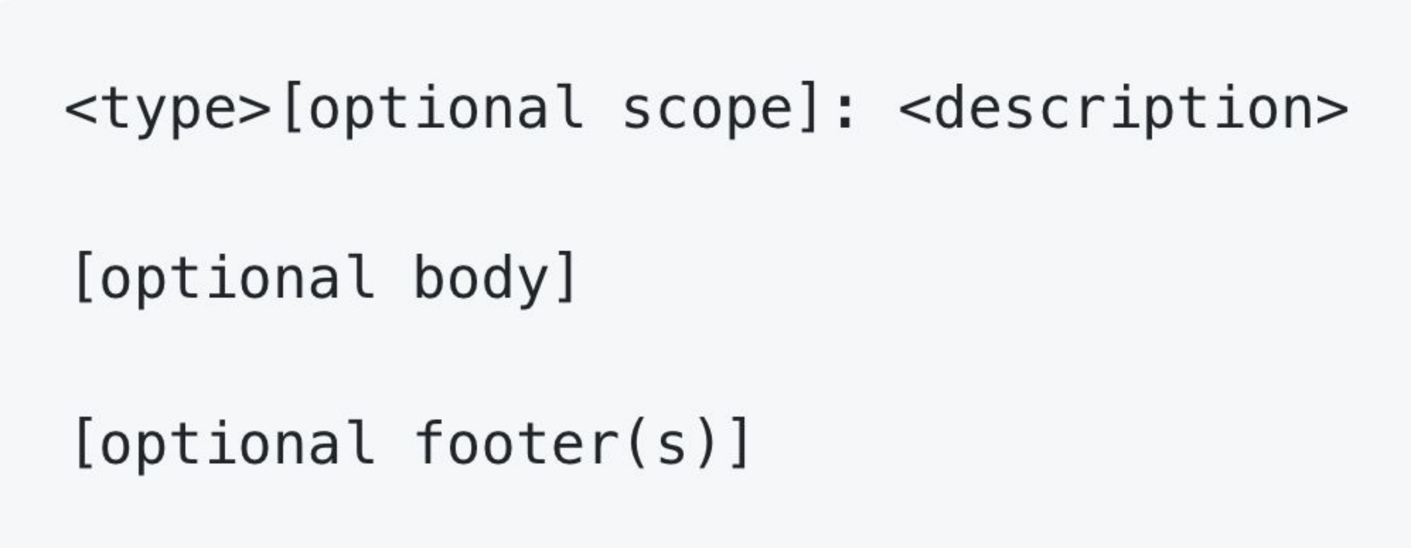


```
commit 5b6cd0f18269cf2b917b2a2ab3b431d9cc26140c (HEAD -> master)
Author: Faisal Al-Rayes <falrayes@redhat.com>
Date:   Sun Aug 11 16:52:22 2024 +0200

    Drop sms authentication support

    This change is due to X and Y which makes it necessary to drop this
    capability to comply with Z.

    The capability was dropped through deleting the sms_auth method and
    removing sms authentication from the auth_methods_white_list in the
    authenticator
```

Resources and Tools

Conventional Commits

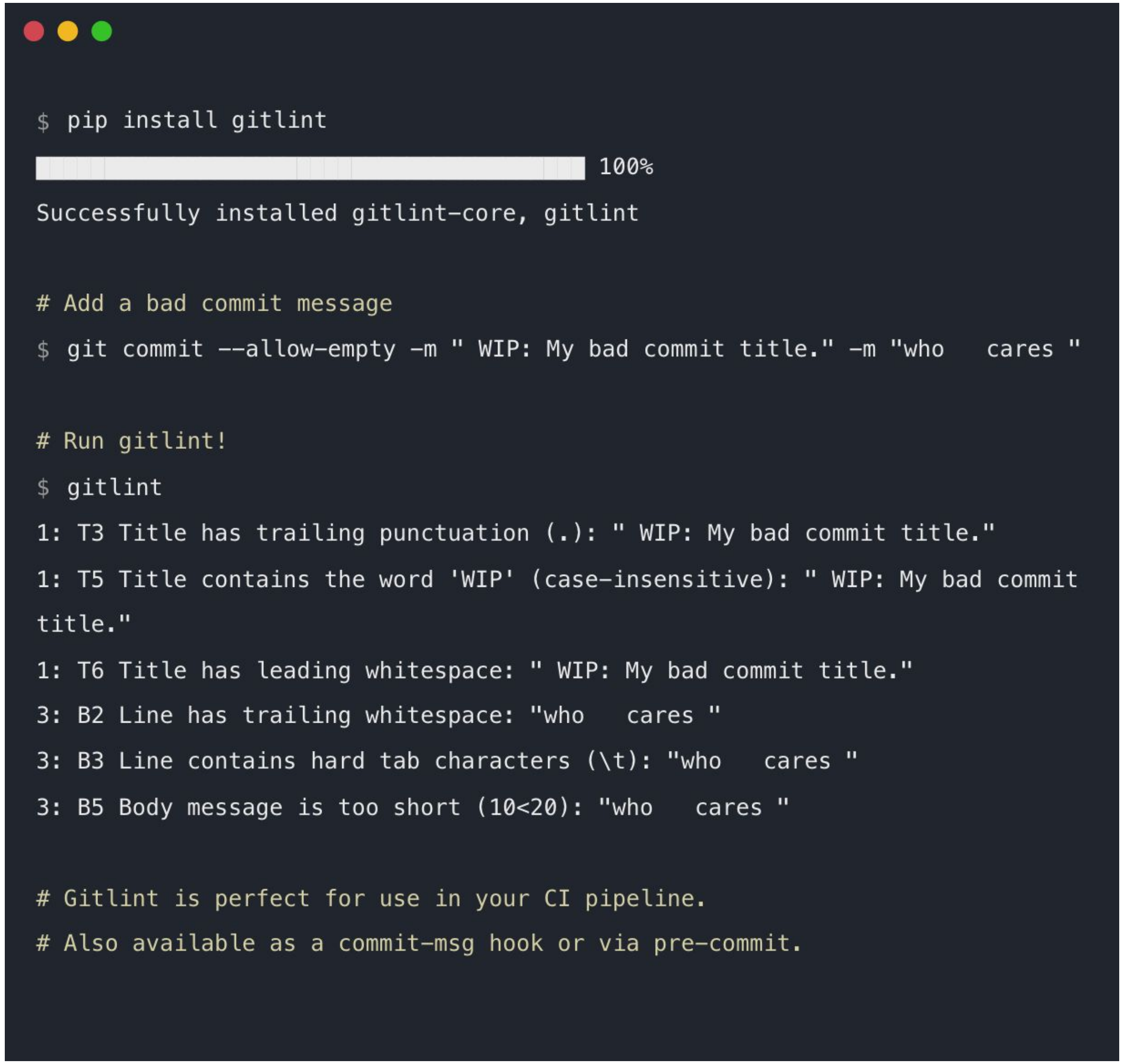
https://www.conventionalcommits.org/



```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

Gitlint

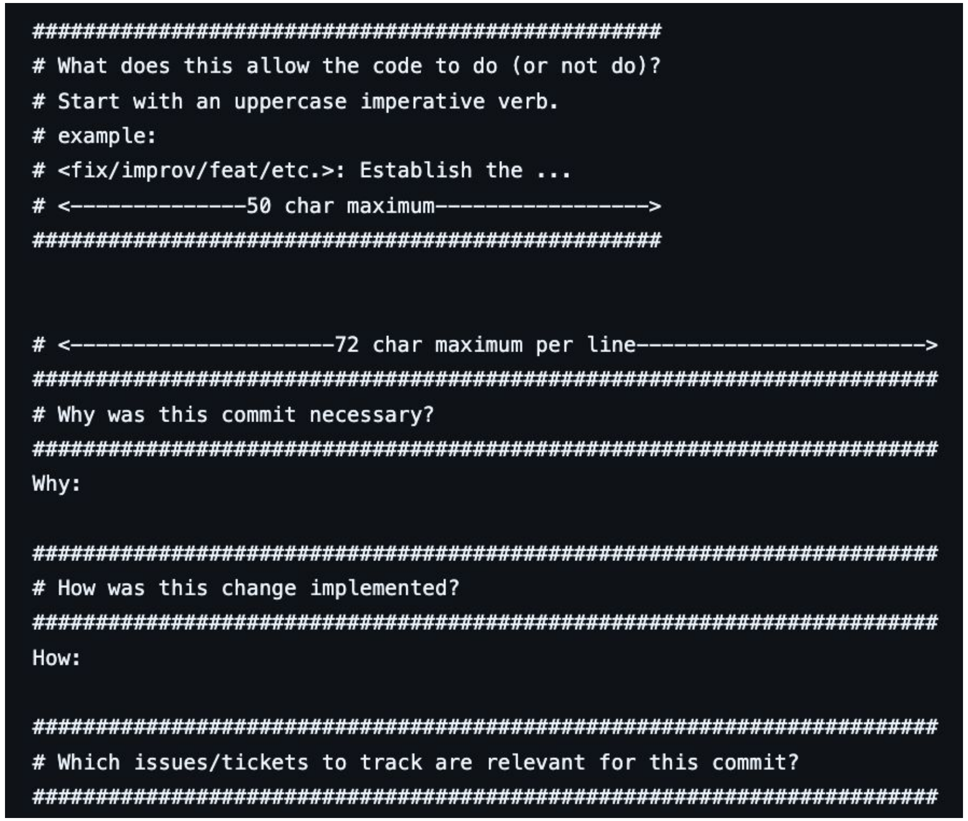
https://jorisroovers.com/gitlint/



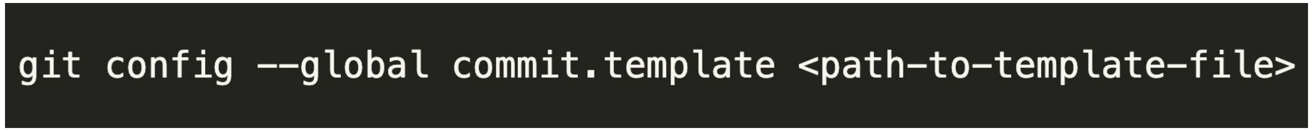
```
$ pip install gitlint
████████████████████████████████████████ 100%
Successfully installed gitlint-core, gitlint

# Add a bad commit message
$ git commit --allow-empty -m " WIP: My bad commit title." -m "who	cares "

# Run gitlint!
$ gitlint
1: T3 Title has trailing punctuation (.): " WIP: My bad commit title."
1: T5 Title contains the word 'WIP' (case-insensitive): " WIP: My bad commit
title."
1: T6 Title has leading whitespace: " WIP: My bad commit title."
3: B2 Line has trailing whitespace: "who	cares "
3: B3 Line contains hard tab characters (\t): "who	cares "
3: B5 Body message is too short (10<20): "who	cares "

# Gitlint is perfect for use in your CI pipeline.
# Also available as a commit-msg hook or via pre-commit.
```

Commit Message Template

https://git-scm.com/docs/git-commit/2.10.5#Documentation/git-commit.txt---templateltfilegt



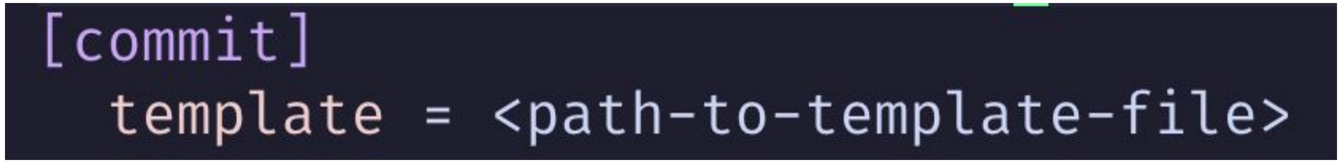
```
##################################################
# What does this allow the code to do (or not do)?
# Start with an uppercase imperative verb.
# example:
# <fix/improv/feat/etc.>: Establish the ...
# <--------------50 char maximum----------------->
##################################################


# <---------------------72 char maximum per line----------------------->
########################################################################
# Why was this commit necessary?
########################################################################
Why:

########################################################################
# How was this change implemented?
########################################################################
How:

########################################################################
# Which issues/tickets to track are relevant for this commit?
########################################################################
```



```
git config --global commit.template <path-to-template-file>
```

Or

add the following to your <cloned-repo>/git/config file



```
[commit]
    template = <path-to-template-file>
```

Thank you for you attention!

linkedin.com/in/faisalalrayyess