Ruby master - Feature #18915

New error class: NotImplementedYetError or scope change for NotImplementedYet

07/14/2022 01:39 PM - Quintasan (Michał Zając)

Status: Open
Priority: Normal
Assignee:
Target version:

Description

Abstract

Introduce NotImplementedYetError exception that should be used in case when the codepath has not been implemented by the developer for some reason (maybe they're designing an abstract class or are designing some sort of interface to reuse later on) OR extend the meaning of NotImplementedYet to cover those usecases so we don't have to introduce another exception

Background

NotImplementedError is supposed to be raised if the underlying operating system or Ruby runtime does not support them (https://ruby-doc.org/core-3.1.2/NotImplementedError.html)

However it appears that many people are misusing this exception by raising this in a superclass from which they later inherit from. I do realize that Ruby promotes duck-typing (the default RuboCop style guide has a cop for this – https://github.com/rubocop/ruby-style-guide#duck-typing). However I have seen this being discussed numerous times:

- https://github.com/rubocop/ruby-style-guide/issues/458
- http://chrisstump.online/2016/03/23/stop-abusing-notimplementederror/
- https://oleg0potapov.medium.com/ruby-notimplementederror-dont-use-it-dff1d7228e5
- https://gitlab.com/gitlab-org/gitlab/-/issues/354314 (which I'm the author of)
- https://github.com/rmosolgo/graphql-ruby/issues/2067 (here the author actually confused it with Python's NotImplementedYet)

Proposal

Create NotImplementedYetError exception

OR

Allow raising NotImplementedError in cases other than OS or Ruby runtime incompatibilities

Evaluation

Add NotImplementedYetError

I think a new exception is a better idea than changing the usage of an existing one just because "everyone is using it". That said it would require people to refactor their code which might prevent wider adoption of the new exception.

Change scope of NotImplementedYet

This would require the least amount of changes possible (only a documentation change) and I believe there would be no compatibility problems whatsoever.