

Comparativa de Paradigmas de programación utilizando metricas de calidad

Yaneth Aquino, Bianca Chura ,Adnner Esperilla y Johanna Torres

June 13, 2020

Abstract

La calidad es un concepto relativo y multidimensional, referido a las expectativas y cualidades solicitados por el cliente, a su vez, está ligada a restricciones y compromisos (presupuesto y tiempo de desarrollo, entre otros). Sin embargo, existe algo que nadie puede negar, cuando algo es de calidad suele pasar desapercibido, pero, por el contrario, la mala calidad es algo que destaca negativamente.

I. INTRODUCCION

Para producir software de calidad y cumplir con la expectativas de nuestros usuarios es que existen diferentes métricas de calidad de software. Métricas de calidad de software es un conjunto de medidas utilizadas para estimar la calidad de un proyecto a desarrollar, entre otros conceptos, y que permiten comparar o planificar estas aplicaciones. son diseñados, especificados e implementados con el propósito de estar al día con la evolución de la programación paradigmas (por ejemplo, imperativos, orientados a objetos, orientados a aspectos, reflexivos y funcionales, por nombrar algunos).

II. DESARROLLO

i. Ruby

Ruby es un lenguaje de programación dinámico, **reflexivo y orientado a objetos** de propósito general que combina sintaxis inspirado por Perl con características similares a Smalltalk. Ruby se originó en Japón a mediados de la década de 1990 y fue desarrollado y diseñado por Yukihiro "Matz" Matsumoto. Ruby es un lenguaje de programación puro orientado a objetos con una sintaxis súper limpia

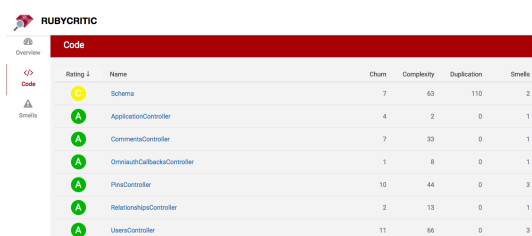
que hace que la programación sea elegante y divertida. Ruby combina con éxito Smalltalk's elegancia conceptual, la facilidad de uso y aprendizaje de Python, y el pragmatismo de Perl. Ruby ha comenzado a convertirse en popular en todo el mundo. Ruby soporta múltiples paradigmas de programación, incluyendo **funcionales, orientados a objetos, imperativos, y reflexivo**. También cuenta con un sistema de tipo dinámico y gestión automática de la memoria



ii. Métricas principales Ruby

- rotación
- la complejidad costo
- la calificación .

- Ejemplo: para proporcionar un informe de calidad de su código Ruby
- Puntuación :** un número genérico que representa la calidad general del código analizado
- Churn :** número de veces que se cambió un archivo
- Complejidad :** cantidad de dolor en su código
- Calificación :** la calificación asignada a un archivo



| Rating | Name | Churn | Complexity | Duplication | Smells |
|--------|-----------------------------|-------|------------|-------------|--------|
| C | Schema | 7 | 63 | 110 | 2 |
| A | ApplicationController | 4 | 2 | 0 | 1 |
| A | CommentsController | 7 | 33 | 0 | 1 |
| A | OmniauthCallbacksController | 1 | 8 | 0 | 1 |
| A | PinsController | 10 | 44 | 0 | 3 |
| A | RelationshipsController | 2 | 13 | 0 | 1 |
| A | UserController | 11 | 66 | 0 | 3 |

<https://subvisual.com/blog/posts/19-solid-principles-in-ruby/> -

<https://github.com/whitesmith/rubycritic> -

HASP. 2008. Tim Chevalier, Darin Morrison, Rebekah Leslie, Tom Harke, John Matthews, Brian Huffman, Andrew McCreight, Mark Jones, Andrew Tolmach, Peter White, James Hook. The High-Assurance Systems Programming. <http://hasp.cs.pdx.edu/overview.html> (accessed July 13, 2010) - Ruby Programming Language. About Ruby. Available at <http://www.ruby-lang.org/en/about/>. -

R. Garcia, J. Jarvi, A. Lumsdaine, J. Siek, and J. Willcock. A Comparative Study of Language Support for Generic Programming. Open Systems Lab, Indiana University Bloomington, IA. November 2003

III. CONCLUSIONES

Es importante tomar decisiones tecnológicas en el momento adecuado y por las razones correctas. Buen negocio las decisiones proporcionan a las buenas personas las herramientas de apoyo adecuadas para que puedan producir buenos productos. Cuando se trata de desarrollo de software, lidiar con problemas de lenguaje difíciles de frente es una requisito para el gerente visionario de hoy. Cuando se combina con otra ingeniería de software consideraciones, una buena decisión de lenguaje puede apoyar el desarrollo de software rentable.

REFERENCES

- Al-Qahtani, S. S., Pietrzynski, P., Guzman, L. F., Arif, R., Tevoedjre, A. (2010). Comparing selected criteria of programming languages java, php, c++, perl, haskell, aspectj, ruby, cobol, bash scripts and scheme revision 1.0-a team cplgroup comp6411-s10 term report. arXiv preprint arXiv:1008.3434. -