[51]

Mi, Q., Keung, J., & Yu, Y. (2016). Measuring the Stylistic Inconsistency in Software Projects using Hierarchical Agglomerative Clustering. In Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering - PROMISE 2016 (pp. 1–10). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2972958.2972963>

**KEYWORDS AUTOR:**

programming style, stylistic inconsistency, hierarchical agglomerative clustering, empirical software engineering

**TÍTULO**:

Medición de la inconsistencia estilística en proyectos de software mediante agrupación jerárquica aglomerativa

**PARA ANEXAR A DOCUMENTO:**

**\*\* INICIO \*\*\***

**DATASET**:

HERRAMIENTAS:

<https://github.com/google/styleguide>

<http://www.gnu.org/prep/standards/>

<http://prettyprinter.de/>

<http://astyle.sourceforge.net/>

Para extraer las métricas de estilo:

<https://github.com/robertyuyang/StylisticFingerprinting>

Palabras clave en C++:

<https://en.cppreference.com/w/cpp/keyword>

**LENGUAJE DE PROGRAMACIÓN:**

C; C++;

**TAGS**

LEGIBILIDAD; METRICAS; ESTILO DE PROGRAMACION; BUENAS PRACTICAS; CLUSTERIZACION;

Mi et al (2016) desarrollaron un programa en Pyhton con el cual escanearon varios archivos de código fuente en C y C++, generando conjuntos de datos en formato CSV, utilizando técnicas de cluster jerárquico aglomerativo , para extraer características y evaluar la huella estilística. Entre los aspectos evaluados se tuvieron en cuenta varias métricas de legibilidad de programación (PRM por sus siglas en inglés).

**\*\* FIN \*\*\***

**RESUMEN COMPLETO**

Estilo de programación: es un tipo de hábito personal para escribir código fuente y que refleja las preferencias de cada individuo.

En ocasiones el estilo de programación de un desarrollador puede ser opuesto al de otro, dificultando la actividad de mantenimiento. El problema se denomina como Inconsistencia estilística.

Métricas del formato de programación (PFM): Relacionadas al formato como espacios en blanco, alineación e identación.

Métricas de legibilidad de programación (PRM): se incluyen las de preferencia en los nombres, proporción de comentarios y longitud de la línea.

**MÉTRICAS PARA C++**

|  |  |  |
| --- | --- | --- |
| Tipo | Categoría | Métrica |
| PFM | Diseño físico de llaves | Porcentaje de llaves abiertas (f) que están solas en una línea.  Porcentaje de llaves abiertas (f) que son el primer carácter de una línea.  Porcentaje de llaves abiertas (f) que son el último carácter de una línea.  Porcentaje de llaves abiertas (f) que están en el medio de una línea.  Porcentaje de llaves cerradas (g) que están solas en una línea.  Porcentaje de llaves cerradas (g) que son el primer carácter de una línea.  Porcentaje de llaves cerradas (g) que son el último carácter de una línea.  Porcentaje de llaves cerradas (g) que están en el medio de una línea.  Porcentaje de funciones con línea (s) en blanco antes de cerrar llaves (g). |
|  | Estilo de alineación | Promedio de espacios en blanco o pestañas horizontales al lado izquierdo de los operadores de asignación.  Promedio de espacios en blanco o pestañas horizontales al lado derecho de los operadores de asignación.  Promedio de espacios en blanco o tabulaciones horizontales después del símbolo // o /\* |
|  | Estilo de identación | Porcentaje de SLOCa físico sin sangrado.  Porcentaje de SLOC físico con espacios en blanco para sangría.  Porcentaje de SLOC físico con pestañas horizontales para sangría. |
| PRM | Estilo de comentarios | Si el mensaje de Copyright aparece en la parte superior del archivo.  La relación entre el número de líneas en blanco y el número de líneas no en blanco.  PRM03 La relación entre el número de líneas de comentarios y el número de líneas sin comentarios (es decir, líneas de código puro).  PRM04 La relación entre el número de líneas de comentario puro y el número de líneas de comentario.  PRM05 La relación entre el número de líneas de código con comentarios en línea y el número de líneas de comentarios.  PRM06 La relación entre el número de comentarios de una sola línea (// símbolo) y el número de SLOC físicos.  La relación entre el número de comentarios de varias líneas (/ \* y \* / par de símbolos) y el número de SLOC físicos.  La relación entre el número de comentarios TODO y el número de líneas no en blanco |
|  | Preferencia de nombres | Porcentaje de caracteres en mayúscula para el nombre de la función.  PRM10 Porcentaje de caracteres en minúscula para el nombre de la función.  PRM11 Porcentaje de caracteres numéricos para el nombre de la función.  PRM12 Porcentaje de caracteres de subrayado para el nombre de la función.  PRM13 Porcentaje de caracteres en mayúscula para el nombre de la variable.  PRM14 Porcentaje de caracteres en minúscula para el nombre de la variable.  PRM15 Porcentaje de caracteres numéricos para nombre de variable.  PRM16 Porcentaje de caracteres de subrayado para el nombre de la variable.  PRM17 Longitud promedio del nombre de la función.  PRM18 Longitud promedio del nombre de la variable. |
|  | Longitud de línea | Promedio físico de SLOC por función.  PRM20 Longitud de línea promedio de un SLOC físico. |

método utilizado para medir la similaridad de la huella estilística:

Distancia Euclidiana.

Dataset:

archivos de código en C y C++ , de 20 desarrolladores de GitHub.

Propuesta:

Enfoque efectivo y eficiente, capaz de medir cuantitativamente el grado de inconsistencia del estilo de programación dentro de los proyectos de software. Desarrollaron un programa en Python, sobre la base de cpplint, que es una herramienta de código abierto similar a pelusa (lint-like) lanzada por Google. Los conjuntos de datos con formato CSV se generan automáticamente después de escanear todos los archivos fuente involucrados en los proyectos de software.