

Ingeniería de software

Nombre del Estudiante: Wladimir Aldo Carlosviza Amanqui

Affiliation

VII semestre Código del Alumno: 213842

Nombre del docente: Ingeniería de software

Año: 2024

Calidad de Documentación

Esta métrica evalúa la calidad de la documentación que acompaña al software, como manuales de usuario, guías de instalación, documentación técnica. Una buena documentación es esencial para que los usuarios puedan comprender y utilizar correctamente el software. También es importante para el mantenimiento y actualización del software por parte de los desarrolladores. La calidad de la documentación se puede medir evaluando aspectos como la claridad, la completitud, la precisión, la organización y la facilidad de comprensión de los documentos. Una documentación de alta calidad facilita el uso y el mantenimiento del software, lo que puede mejorar la satisfacción del usuario y reducir los costos de soporte.

Métricas

Completitud

- **Cobertura de Funcionalidades:** Porcentaje de las funcionalidades del software que están documentadas.
- **Cobertura de Casos de Uso:** Porcentaje de casos de uso o escenarios de aplicación cubiertos por la documentación.

Claridad y Comprensibilidad

- **Lenguaje Claro:** Evaluaciones cualitativas o cuantitativas del lenguaje utilizado en la documentación.
- **Estructura y Organización:** Uso de encabezados, subencabezados, listas y otras herramientas de formato para mejorar la legibilidad y organización.

Precisión

- **Exactitud Técnica:** Grado en el que la documentación describe correctamente las funcionalidades y comportamientos del software.
- **Actualización:** Frecuencia y velocidad con la que se actualiza la documentación para reflejar cambios en el software.

Utilidad

- **Feedback de Usuarios:** Opiniones y comentarios de los usuarios sobre la utilidad de la documentación.
- **Ejemplos y Casos Prácticos:** Cantidad y calidad de ejemplos prácticos que ayudan a los usuarios a entender cómo utilizar el software.

Accesibilidad

- **Disponibilidad:** Facilidad con la que los usuarios pueden acceder a la documentación (por ejemplo, accesible en línea, descargable, etc.).
- **Índice de Búsqueda:** Presencia y calidad de un índice de búsqueda o un sistema de navegación que permita a los usuarios encontrar rápidamente la información que necesitan.

Consistencia

- **Estilo Consistente:** Uniformidad en el estilo de escritura y terminología a lo largo de toda la documentación.
- **Formato Consistente:** Uniformidad en el uso de formatos, estructuras de sección y esquemas de presentación.

Interactividad y Ejemplos

- **Ejemplos de Código:** Inclusión de ejemplos de código funcionales y claros.
- **Tutoriales y Guías Paso a Paso:** Disponibilidad de tutoriales interactivos o guías que guíen al usuario a través de tareas específicas.

Eficiencia

- **Tiempo de Resolución de Dudas:** Tiempo promedio que los usuarios tardan en encontrar respuestas a sus preguntas en la documentación.

Evaluating Software Documentation Quality

Aborda la importancia de la documentación en el uso y aprendizaje de bibliotecas de software. Se aborda que la documentación de mala calidad puede ser ineficaz o incluso perjudicial si contiene información incorrecta o desactualizada. Se centra en desarrollar una métrica para evaluar la calidad de la documentación, con el objetivo de facilitar a los desarrolladores la tarea de seleccionar y aprender a usar bibliotecas de software Tang y Nadi (2023).

1. Importancia de la Documentación

- La documentación es crucial para que los desarrolladores puedan comparar, seleccionar y aprender a usar bibliotecas de software.
- Documentación de mala calidad puede causar confusión y errores, lo que subraya la necesidad de evaluar su calidad antes de invertir tiempo en aprender una biblioteca.

2. Aspectos de Calidad de la Documentación

- Tang identifica seis aspectos clave de la calidad de la documentación: Ejemplos de Código, Completeness (Compleitud), Facilidad de Uso, Legibilidad y Actualización.
- Estos aspectos se derivan de una revisión de la literatura existente y se validan a través de entrevistas con profesionales de la industria.

3. Desarrollo de Métricas

- Utilizando el paradigma Goal-Question-Metric (GQM), Tang desarrolla métricas específicas para evaluar los aspectos seleccionados de la documentación.
- Estas métricas incluyen indicadores automáticos y semi-automáticos, como el índice de legibilidad de Flesch y la proporción de clases y paquetes documentados respecto al código fuente.

4. Validación y Herramienta de Evaluación

- Las métricas desarrolladas se validan mediante una encuesta con 25 desarrolladores, quienes encuentran útiles las métricas relacionadas con la legibilidad del texto y los ejemplos de código.
- Se presenta una herramienta de resumen de calidad de la documentación que permite a los desarrolladores evaluar rápidamente la calidad de la documentación de una biblioteca.

5. Resultados y Aplicaciones Futuras

- Los participantes en el estudio encuentran la herramienta y las métricas útiles, con una puntuación media de cuatro en una escala Likert de cinco puntos.


| Aspecto | Definición | Referencias |
|------------------------------------|--|--|
| Accesibilidad | ¿Qué tan difícil es encontrar la documentación? | [20], [61], [72] |
| Apelar | ¿Qué tan interesante es leer? | [sesenta y cinco] |
| Comentarios de idoneidad | Densidad de comentarios del código fuente. | [3] |
| Autoridad | ¿Cuánta autoridad tiene la documentación? | [19], [20], [72] |
| Ejemplos de código | La existencia de ejemplos de código en la documentación. | [20], [21], [52], [53], [67], [70] |
| Cohesión | ¿Qué tan bien encaja la documentación? | [sesenta y cinco] |
| Lo completo | ¿Está completa la información de la documentación? ¿Está todo el código fuente documentado? ¿Están documentadas todas las tareas/características? | [2], [3], [8], [21], [53], [61], [67], [72]–[74] |
| Consistencia | ¿La documentación es coherente consigo misma? Misma terminología, mismo formato, etc. | [8], [21], [29], [65], [67], [72] |
| Consistencia (según el estándar) | ¿La documentación se ajusta a un estándar de documentación definido por una autoridad externa? | [3], [61] |
| Exactitud | ¿Es exacta la información de la documentación? | [2], [21], [67], [72] |
| Punto de vista de la documentación | La documentación debe estar escrita desde el punto de vista del lector. | [4] |
| Facilidad de uso | ¿Qué tan fácil es utilizar la documentación? por ejemplo, navegación, enlaces internos y externos | [3], [20], [52], [67], [70] |


| | | |
|----------------------------|--|--|
| Eficacia | ¿La documentación hace un uso eficaz del vocabulario técnico? | [sesenta y cinco] |
| Aptitud del propósito | ¿La documentación se ajusta al propósito previsto? | [4] |
| Formato | ¿Qué formato de archivo es la documentación? por ejemplo, HTML, PDF, etc. | [20] |
| Soporte gráfico | ¿La documentación utiliza imágenes? | [3], [20], [21] |
| Longitud | La longitud de las frases en la documentación. | [3], [20] |
| Mantenibilidad | ¿Qué tan fácil es actualizar la documentación? | [2] |
| Organización | ¿La información de la documentación está organizada de manera eficiente? por ejemplo, secciones/subsecciones | [3], [4], [20], [21], [53], [61], [65], [67], [72] |
| Precisión | ¿Qué tan precisa es la documentación? | [21] |
| Calidad | ¿Qué tan bien escrita está la documentación? | [sesenta y cinco] |
| Legibilidad | ¿Qué tan fácil es leer la documentación? | [2], [3], [20], [21], [29], [65], [67], [72] |
| Justificación del registro | ¿La documentación incluye decisiones de diseño? | [4] |
| Relevancia del contenido | ¿Es relevante la información de la documentación? | [4], [20], [21], [53], [65], [67] |
| Ortografía y gramática | Ortografía y gramática de la documentación. | [20], [72] |
| Admite muchos escenarios | ¿La documentación admite muchos escenarios? | [53] |
| Trazabilidad | ¿Hasta qué punto se pueden rastrear los cambios en la documentación? | [72] |
| Tipo | ¿Cuál es el tipo de documentación? por ejemplo, requisitos, especificaciones, pruebas, etc. | [20] |
| Comprensibilidad | ¿Qué tan comprensible es la información en la documentación? | [4], [65], [67] |
| A hoy | ¿Qué tan actualizada está la documentación relativa al código fuente? | [2]–[4], [8], [19]–[21], [29], [61], [67], [72] |


Figura 1
Resumen de aspectos de calidad de la documentación

←

[Orjson](#)
 Documentation
 Quality Evaluation







Instructions
Hovering over any of the presented metrics/information provides details on what it represents. You can provide feedback by clicking on the bottom "Answer Survey Questions" button. Please note that this feedback is optional, and you are under no obligation to provide any. Also note that when processing a new library, calculating the different metrics will take time; a page refresh will automatically occur when each process is complete.

Answer Survey Questions

Documented Library Tasks
Each link below goes to the section of documentation where the task is found.

- ✓ [Serialize float by default](#)
- ✓ [Serialize dict keys](#)
- ✓ [Serialize dictionaries to custom format](#)
- ✓ [Customize output](#)
- ✓ [Insert to dict](#)
- ✓ [Serialize to same str](#)
- ✓ [Serialize None](#)
- ✓ [Serialize bool for comparison](#)
- ✓ [Serialize bool by default](#)
- ✓ [Serialize float for comparison](#)
- ✓ [Serialize None for comparison](#)
- ✓ [Serialize subclass arbitrary types specify default as callable](#)
- ✓ [Serialize None](#)
- ✓ [Serialize datetime.datetime objects without %p](#)
- ✓ [Serialize None by default](#)
- ✓ [Serialize str for comparison](#)
- ✓ [Serialize int for comparison](#)
- ✓ [Serialize int by default](#)
- ✓ [Serialize twitter/json future](#)
- ✓ [Serialize str by default](#)

Library Description
Orjson is a fast, correct JSON library for Python. It benchmarks as the fastest Python library for JSON and is more correct than the standard json library or other third-party libraries. It serializes datadict, datetime, numpy, and UUID instances natively.

Documentation Quality Metrics

| | |
|--------------------------------------|-------|
| Methods with code examples | ★★★★★ |
| Classes with code examples | ★★★★★ |
| Readability of text | ★★★★☆ |
| Readability of code examples | ★★★★☆ |
| Documentation/Source code similarity | ★★★★★ |
| Navigation rating | ★★★★★ |

Star rating at top

Please submit the form by scrolling down and clicking the "Submit" button. Note that this survey form is scrollable.

1. How useful is having a general rating of orjson documentation's quality?

☐ Not useful
☐ Somewhat not useful
☐ Neither useful nor not useful
☐ Somewhat useful
☐ Very useful

Documented Library Tasks

2. How useful is having a list of documented library tasks found in orjson's documentation?

☐ Not useful
☐ Somewhat not useful
☐ Neither useful nor not useful
☐ Somewhat useful
☐ Very useful

Methods with Code Examples

3. How useful is having a rating of methods with a code example in orjson's documentation?

Figura 2
Interfaz de la herramienta web

Referencias

- Tang, H., y Nadi, S. (2023). Evaluating software documentation quality. *Proceedings - 2023 IEEE/ACM 20th International Conference on Mining Software Repositories, MSR 2023*, 67-78. doi: 10.1109/MSR59073.2023.00023