

Una comparación colaborativa del rendimiento en proyectos de software libre

José-Manuel Sánchez¹ , Miguel Angel Olivero¹ , FJ Dominguez-Mayo¹ ,
and David Benavides¹ 

Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de
Ingeniería Informática, Seville, Spain
{jsanchez7, molivero, fjdominguez, benavides}@us.es

Abstract. En los últimos años, los proyectos de software de código abierto (OSS) se han vuelto cada vez más importantes para muchas organizaciones. A medida que estos proyectos crecen en tamaño y complejidad, aumenta la necesidad de un desarrollo de software de alta calidad. Al mismo tiempo se ha comprobado que el uso de prácticas de DevOps mejoran la calidad y el rendimiento organizacional. Sin embargo, es difícil medir el impacto real que supone aplicar estas prácticas porque los métodos de evaluación existentes, como los informes de DORA, se centran principalmente en la implementación continua y la entrega en producción. Esto se diferencia de las prioridades de los proyectos OSS, que enfatizan la liberación continua de código y su impacto en los usuarios en lugar de en sus implementaciones o entregas en producción. Para abordar esta situación se emplea un sistema colaborativo de evaluación, Performance-Tracker (PT), diseñado para evaluar y comparar el rendimiento de proyectos OSS usando varios factores. PT extrae información pública de proyectos OSS y ha permitido generar una base de conocimientos compartida que supone la primera base de conocimiento del marco de referencia. Esto se ha logrado evaluando el rendimiento de 50 proyectos OSS en su primera versión. Este enfoque permite que puedan evaluarse más proyectos y que comparen su rendimiento en base a los puntos de referencia de la base de conocimiento colaborativa. Usando PT y con las métricas propuestas, los proyectos OSS pueden analizar y comparar su rendimiento con respecto al resto de proyectos participantes. Esto, a su vez, permite que sus resultados alimenten la base de conocimiento y enriquezcan el marco de referencia. PT, junto a su base de conocimiento inicial, es una propuesta integral para la evaluación de proyectos OSS, abordando los desafíos propios de este tipo de proyectos. Además, el entorno de aprendizaje colaborativo persigue fomentar un proceso de desarrollo más eficiente en los proyectos OSS. Al permitir una comparación de métricas, los equipos de desarrollo pueden identificar claramente cómo mejorar su rendimiento.

Keywords: Desarrollo software · DevOps · Evaluación comparativa · Software delivery · Software libre



1 Introducción y contexto

Cada vez más son las organizaciones que comienzan a usar proyectos de software de código abierto (OSS) [3]. Esto motiva la creciente popularidad de proyectos de esta naturaleza que hace que los proyectos cuenten con más participación. De este modo, los proyectos OSS tienen cada vez una embergadura mayor y necesitan de estrategias y tácticas que ayuden al mantenimiento de los proyectos [2]. Prácticas como DevOps han ayudado a mejorar el rendimiento de la organización y la calidad de los resultados de los proyectos, aunque no permiten contrastar el impacto que implica su uso. Métricas como DORA [1] permiten extraer información para comparar el rendimiento que obtienen las organizaciones tras el uso de las prácticas. Sin embargo, DORA no permite comprender el impacto de las estrategias aplicadas a proyectos OSS. Mientras que DORA se centra en la producción y las operaciones de servicio, los proyectos OSS persiguen la liberación de código continua y evaluar el impacto en terceras partes. Sánchez-Ruiz et al. [4] exploró las limitaciones que presentan estas métricas al aplicarse a proyectos OSS y propone una adaptación de las métricas DORA para que éstas puedan ajustarse al contexto de los proyectos OSS. Estas métricas, que pretenden concretar conceptos en contextos de desarrollos y operaciones, permiten comprender la frecuencia de liberación de código, el tiempo que tarda un bug en resolverse y un cambio en estar disponible, y la tasa de errores que se reportan. Al considerar proyectos de código abierto, esta información puede encontrarse en las plataformas como GitHub. Para automatizar el proceso de captura de las métricas que permitan comprender el impacto de aplicar prácticas como DevOps, Sánchez-Ruiz et al. diseñaron el sistema "Performance-Tracker" [5] que enriquece la información disponible en GitHub. De esta forma, al visitar un proyecto de GitHub es posible observar los valores de las cuatro métricas y permite comparar el rendimiento de los proyectos. En este artículo se usan las métricas y herramientas de Sánchez-Ruiz et al. [4,5] para explorar los valores medidos en varios proyectos OSS y estudiar las limitaciones actuales de las métricas y la herramienta propuesta.

2 Uso del benchmarking

Para crear la base de conocimientos inicial que permita conocer el grado de rendimiento de los proyectos se han tomado 50 proyectos OSS: **1.** Angular, **2.** Kubernetes, **3.** TensorFlow, **4.** Visual Studio Code, **5.** Vue.js, **6.** Dubbo, **7.** Spring Framework, **8.** Matplotlib, **9.** PyTorch, **10.** Terraform, **11.** Node.js, **12.** OBS Studio, **13.** Electron, **14.** Docker Compose, **15.** Charts.js, **16.** Elasticsearch, **17.** Nextcloud, **18.** SQLAlchemy, **19.** TypeScript, **20.** Babel, **21.** React Native, **22.** Godot, **23.** Next.js, **24.** Swift, **25.** Material UI, **26.** Jax, **27.** Windows Terminal, **28.** DeepSpeed, **29.** Jest, **30.** NocoDB, **31.** Strapi, **32.** Docusaurus, **33.** Directus, **34.** Keycloak, **35.** Grafana, **36.** Sentry, **37.** Prometheus, **38.** Chatwoot, **39.** Renovate, **40.** Playwright, **41.** .NET MAUI, **42.** Remix, **43.** Netbox, **44.** CloudQuery, **45.** Focalboard, **46.** Matomo, **47.** Bazel, **48.** WooCommerce, **49.** Superset, **50.** Airflow.

La extracción de datos, realizada en septiembre de 2023, de cada uno de estos proyectos incluye el nombre, la dirección donde se aloja el proyecto, la cantidad de contribuidores, el número de incidencias abiertas y cerradas, la cantidad de solicitudes de incorporación de código (pull requests) abiertas y completadas, la cantidad de liberaciones de código y la fecha de inicio del proyecto. Aunque todos los proyectos estudiados están alojados en la plataforma GitHub, cada uno de ellos se autogestiona de manera diferente. Al obtener los datos de los proyectos se observó que el etiquetado y clasificación de las incidencias era heterogéneo, dificultando la captura automatizada de estos datos. Por ejemplo, los valores necesarios para las métricas "Time To Repair Code" y "Bug Issues Rate" [4] se obtienen a partir de la cantidad de bugs de cada proyecto. En el caso de identificación de los bugs se han encontrado etiquetas variadas como las siguientes: 'bug', 'type: bug/fix', 'Needs Reproduction', 'Release critical', entre otras muchas. A pesar de que existen diferencias en el etiquetado de las incidencias, ha sido posible recoger datos de los proyectos. Sin embargo, se han encontrado 12 proyectos en los que ha sido imposible incluir sus datos para la versión inicial del banco de información: **1.** odoo, **2.** linux, **3.** mockito, **4.** guava, **5.** pygments, **6.** express, **7.** bootstrap, **8.** axios, **9.** geany, **10.** webiny, **11.** appwrite, **12.** medusa.

Proyectos como 'odoo' o 'Linux' no usan la característica GitHub Release, haciendo que no sea posible calcular las métricas que dependen de las liberaciones de código. 'Bootstrap' o 'Axios' no usan sistema de etiquetado para identificar las incidencias que son bug. El resto de los repositorios mencionados no tienen datos públicos disponibles que puedan ser usados para alimentar las métricas.

3 Resultados del estudio

Una vez recogidos los datos, se han estudiado y se han ordenado los proyectos en terciles (tabla 1). De esta forma se han creado tres niveles (Bajo [Low], Medio [Medium], Alto [High]) para cada una de las métricas. En base a estos niveles, un proyecto puede compararse con otros de su entorno para identificar en qué categoría se encuentra para cada una de estas métricas, e identificar áreas de mejora fácilmente. Esta primera versión de los resultados podrá enriquecerse con una mayor cantidad de proyectos, haciendo que los resultados y la separación de los niveles sean más precisos. Así mismo, los proyectos que quieran comparar sus resultados con los de la base inicial necesitarán enviar sus datos para ser cotejados, lo que a su vez ayudarán a ajustar los valores de cada métrica. El aspecto colaborativo de la plataforma se centra en compartir información sobre el rendimiento de los proyectos entre todos los participantes, de forma que los valores de las métricas representen la realidad de los proyectos OSS y los participantes puedan comparar su rendimiento con el del resto.

4 Conclusiones y trabajos futuros

Aplicar las métricas DORA adaptadas a proyectos OSS con los ajustes propuestos por Sánchez-Ruiz [4] junto a la herramienta Performance-Tracker [5], nos

Table 1. 2022 OSS projects performance

Performance Level	Release Frequency	Frequency	Lead Time For Released Changes	Time To Repair Code	Bug Issues Rate
Low	More than 22.5 days	More than 21 days	More than 21 days	More than 151 days	More than 54%
Medium	Between 8.5 days and 22.5 days	Between 7.5 days and 21 days	Between 7.5 days and 21 days	Between 41 days and 151 days	Between 10% and 54%
High	Less than 8.5 days	Less than 7.5 days	Less than 7.5 days	Less than 41 days	Less than 10%

ha permitido recoger información que evalúa el rendimiento de cada proyecto, y ayuda a identificar áreas de mejora. Al explorar los diferentes proyectos se han encontrado además restricciones que han impedido aplicar las métricas en la totalidad de ellos. Uno de los mayores desafíos actuales para la recogida de muestras es la normalización en la identificación de bug. Al mismo tiempo, se necesita validar la cobertura de las métricas propuestas de forma que ésta pueda reflejar con mayor precisión el rendimiento de los proyectos OSS. Como trabajo futuro se propone descubrir alternativas en la identificación de bugs que ofrezcan mayor confiabilidad a los datos ofrecidos. Por otro lado, para comprender el impacto que tiene esa propuesta en el ámbito académico e industrial, se estima conveniente medir el valor que aporta la propuesta actual a las partes interesadas para identificar áreas de mejora. Además, para enriquecer la información ofrecida por la herramienta PT, se propone estudiar posibles correlaciones entre las métricas propuestas y las métricas DORA.

Acknowledgments. Trabajo parcialmente financiado por FEDER/Ministerio de ciencia, innovación y universidades/Junta de Andalucía/Agencia Estatal de Investigación/CDTI con los proyectos: *Data-pl*(PID2022-138486OB-I00) y MIDAS (IDI-20230256) y la red TASOVA PLUS (RED2022-134337-T).

References

References

1. Cloud, D.G.: 2022 accelerate. state of devops report, <https://cloud.google.com/devops/state-of-devops>
2. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P.: A survey of devops concepts and challenges. *ACM Computing Surveys (CSUR)* **52**(6), 1–35 (2019)
3. OpenLogic by Perforce, O.S.I.: 2023 state of open source report. open source usage, market trends, analysis, <https://www.openlogic.com/resources/2023-state-open-source-report>
4. Ruiz, J.M.S., Mayo, F.J.D., Oriol, X., Crespo, J.F., Benavides, D., Teniente, E.: A benchmarking proposal for devops practices on open source software projects. arXiv preprint arXiv:2304.14790 (2023)
5. Sánchez Ruiz, J.M., Olivero González, M.Á., Domínguez Mayo, F.J., Oriol, X., Benavides Cuevas, D.F.: Benchmarking del rendimiento de proyectos software de código abierto mediante una herramienta colaborativa. In: XXVII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2023) (2023), <https://hdl.handle.net/11705/JISBD/2023/7888>