

Towards Streamlining Auditing for Compliance with Requirements in Opensource Software at NASA

Carlos Paradis
KBR, Inc.

Ivan Dominguez
KBR, Inc.

Misty Davies
NASA Ames Research Center

Introduction

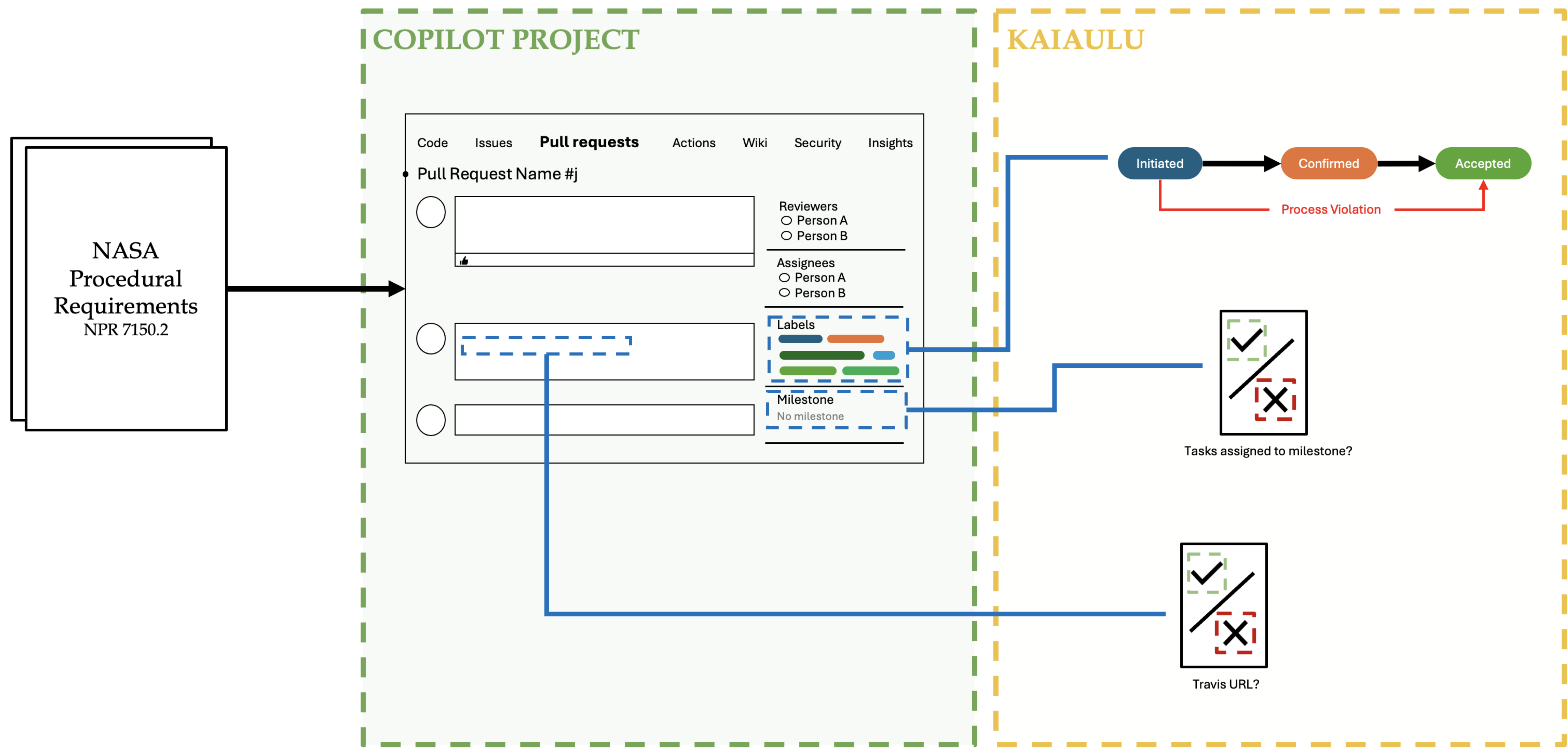
Software that operates in critical environments must be developed and maintained following strict software engineering and development processes. The instantiation of such processes may vary per project; however, once decided upon, projects must undergo audits to evaluate compliance with such requirements.

Aim

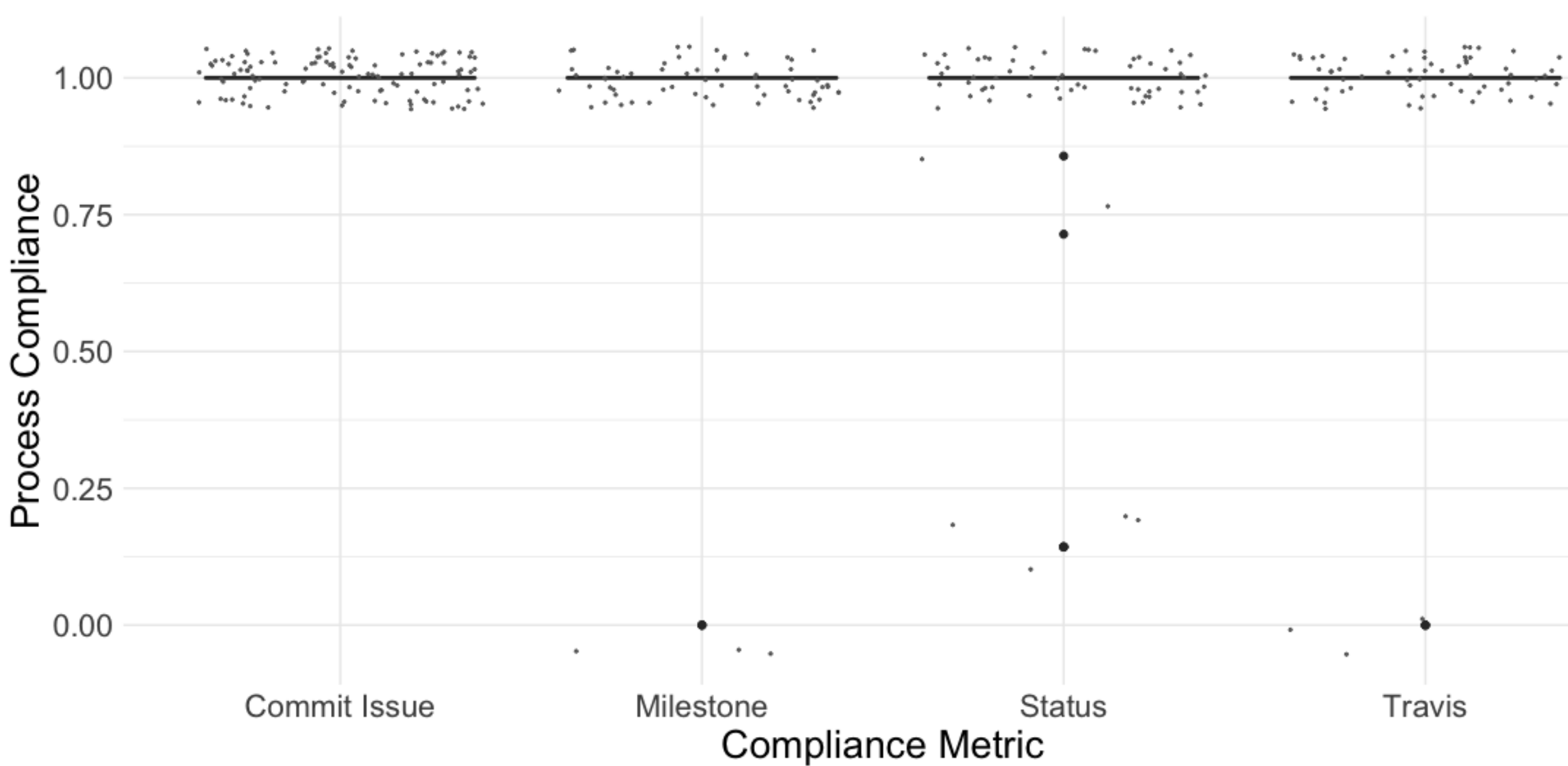
We propose that audit effort can be reduced when requirements are realized by leveraging commonly used open-source infrastructure for version control, issue tracking and continuous integration, and the generated records are analyzed using a repository mining software tool to quantify process compliance.

Method

We perform a case study in the NASA-funded Copilot project, utilizing Kaiaulu, a repository mining software tool. We define four software compliance metrics based on the Copilot's requirements, and analyze their impact on source code quality. Tool: github.com/sailuh/kaiaulu Repo: github.com/Copilot-Language/copilot



Results



Results Highlight: The **Status** issue compliance metric measures if status tags with different labels follow the expected order towards the issue completion. The lowest compliance issues, Issues #332 and #278 reflect errors on the part of the change manager in enforcing compliance with the process.

References

1. C. Paradis and R. Kazman, "Building the msr tool kaiaulu: Design principles and experiences," in Software Architecture (P. Scandurra, M. Galster, R. Mirandola, and D. Weyns, eds.), (Cham), pp. 107–129, Springer International Publishing, 2022
2. C. Jensen, "Data mining for software process discovery in open source software development communities," pp. 96–100, 01 2004
3. W. Poncin, A. Serebrenik, and M. Brand, "Process mining software repositories," pp. 5–14, 03 2011.
4. M. Gupta and A. Sureka, "Nirikshan: Mining bug report history for discovering process maps, inefficiencies and inconsistencies," in Proceedings of the 7th India Software Engineering Conference, ISEC '14, (New York, NY, USA), Association for Computing Machinery, 2014.
5. S. Bala and J. Mendling, Monitoring the Software Development Process with Process Mining, pp. 432–442. 07 2018.