# Linux Kernel Best Practices and its Implementation in the Project

**Authors**

Abhishek Gupta
guptabhishek785@gmail.com

Vishal Veera Reddy
veerareddyvishal144@gmail.com

Neeloy Gomes
neeloy.gomes@gmail.com

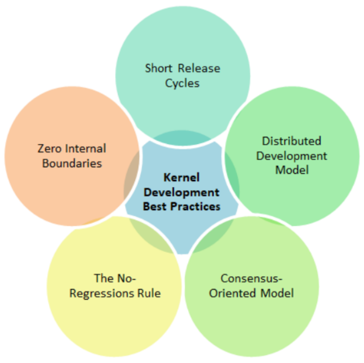
Tyler Craine
tylerkecraine@gmail.com

Abhimanyu Belam
bellamabhimanyu@gmail.com

## Introduction

Linux Kernel is well known for its domination in the open-source software world. It has dominated the open-source industry for nearly 26 years and is still going strong. The practices followed in the development of the Linux Kernel is now widely adapted in every major software development project.

This article talks about the Linux Kernel best practices and draws connection with the Project rubric. The article briefly mentions the Linux Kernel best practices and what it means. The article also stresses the importance of the rubric and how the rubric ensures that these very best practices are followed by the students when working on their projects



- **Short Release Cycles**

  One of the major reasons for the success of the Linux Kernel was that it quickly adapted to the short-release cycle format. A short release cycle ensured that new but stable updates were quickly released for the kernel which ensured that work wasn't getting piled up without affecting the development of the product

  The project rubric ensures that the software created by the students was released in short cycles, ensured that there were a high number of commits which is a good indicator of continuous patches implemented to fix issues as soon as it was created

- **Distributed Development Model**

  The Distributed Development Model stresses that the responsibility and workload need to be efficiently distributed among team members and improve participation for better scalability

  The project rubric gives a lot of importance to this practice. As per the rubric it is very important that the workload is evenly spread over the whole team members and there are high number of commits by different people. It ensures that a suitable version control system is used by the team members to ensure that there is good and equal collaboration among members.

  The project rubric also stresses on the importance of documentation and that the entire team uses the same set of tools to ensure compatibility, avoid hinderances. These practices help to improve transparency and increase participation thereby enhancing the scalability

- Consensus-Oriented Model
  It states that for a feature to be merged/ available all the groups of developers should approve of it.
  This rule is enforced by the fact that the rubric wants the team to have a channel for communication so that they can arrive at a consensus.

- The No-Regressions Rule
  It says that if a feature works in n-1 release then it works in any release after that.

  This rule is enforced by the fact as automated testing and test cases are mentioned in the rubric.

- Zero Internal Boundaries
  The Zero Internal Boundaries stresses that the responsibility can be taken up any developer and there is no specific restrictions to each area.

The project rubric gives a lot of importance to this practice. As per the rubric it is very important that the workload is evenly spread over the whole team members and there are high number of commits by different people.