

Rubric and Linux Kernel Best Practices CSC 510 Project 1

Janvi Phadatare
jphadta@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Praneya Lal
plal@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Kaksha Mhatre
kpmhatre@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Mitanshu Reshamwala
mresham@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Mitul Patel
mpatel27@ncsu.edu
North Carolina State University
Raleigh, NC, USA

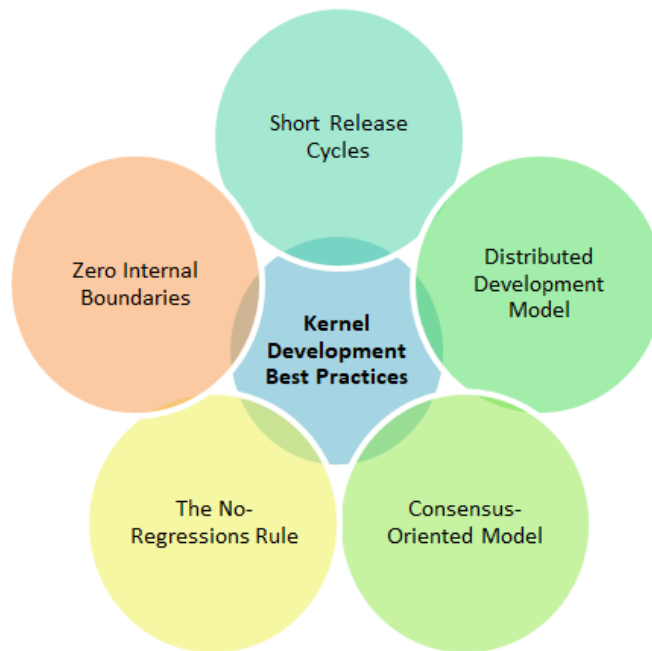


Figure 1: Linux Kernel Best Practices

ABSTRACT

This paper will discuss about how the Linux Kernel Best Practices are linked with the rubric of the CSC 510 Project 1. It will show how the Linux Kernel Best Practices are followed by the rubric and helps in recording the performance of project 1.

CCS CONCEPTS

• Linux Kernel Best Practices;

KEYWORDS

Short Release Cycles, Zero Internal Boundaries, Consensus Oriented Model, Distributed Development Model, No-Regression Rule.

Permission to make digital or hard copies of all or part of this work for personal or Unpublished working draft. Not for distribution.

for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2022, Woodstock, NY

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

2022-10-07 17:04. Page 1 of 1–2.

ACM Reference Format:

Janvi Phadatare, Praneya Lal, Kaksha Mhatre, Mitanshu Reshamwala, and Mitul Patel. 2022. Rubric and Linux Kernel Best Practices CSC 510 Project 1. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

This paper will discuss about most of the Linux Kernel Best practices. For CSC 510 SE Project 1 we are given a rubric with various things, so we need to show the connection between items of the Rubric and Linux Kernel Best Practices. This will help to evaluate the effectiveness of our project with respect to the industry standards. This paper will eventually show how the Rubric items follow the Linux Kernel Best Practices:-

2 LINUX KERNEL BEST PRACTICES

This section will give in-depth description about the Linux Kernel Best Practices. Linux Kernel is one of the most important part of Linux, it has the responsibility to manage the hardware, data integrity and security of the system. Following are some of the best linux kernel practices.

- **Short Release Cycle:** In the earlier time, there were long release cycles, like everything was made up first and large chunks of code was integrated directly into the system, so if there was an issue it took long time to understand and it was not a feasible option. Hence, short release cycles resolved this problem, in this whenever new code was ready it would be made available in a release as this would cause minimal damage.
- **Distributed Development Model:** If massive amount of changes are applied to a project, then even an experienced developer might face an issue, so to avoid this giving maximum number of people the chance to review the code, enhances that the code quality is maintained.
- **The No-Regressions Rule:** The rule is quite important as it guarantees that if the kernel works in a particular setting, all subsequent kernels should work too. We don't need to backtrack previous releases and everything should remain proper in latest releases.
- **Consensus Oriented Models:** This model ensures that all the developers should agree with the changes that are made so that everyone knows what is happening and be aware about the functionality of all the latest versions.
- **Zero Internal Boundaries:** If a developer finds an issue and can justify the changes, then he can change any part of the kernel. This helps in smoother functioning and developers need not wait long time to resolve the issue.

3 CONNECTION BETWEEN THEM

For CSC SE 510 Project 1 we are given a rubric and above in the paper we have described what are the important Linux Kernel Best Practices. So this section will tell about the connection between them.

So firstly, Short release cycles are mentioned in the rubric. It clearly mentioned that for short projects there will not be many short release cycles. But for big projects it will be a great evaluator about how the developers are working around the new releases.

Distributed Development model means many people are there to review the code, so this rubric is defined as all the team members are working and making contributions to the project. It is also checking that everyone is using the same languages or framework so there is proper consistency.

The No Regressions rule means in a particular setting if the kernel works then in subsequent setting it should also work. So this rubric is followed as in any system the developed project can work and contributors in their repository can put instructions or videos to ensure that even new user is understanding the system, this will also ensure and tell about all the features the team members have added.

Consensus Oriented Model means all developers should not disagree on a change. This rubric is evaluated using the issues the developers open or close by discussing. Team members can discuss about possibility of the issues and come towards an consensus result. Every developer will know about the issue and new features as it would have been discussed already.

Zero internal boundaries means that if the developer finds an issue he can resolve it immediately. This rubric is evaluated as there are various people working on different things and they create pull requests for an issue. If they can justify the issue the pull request is approved by other team members and it can be committed into the main branch. Proper documentation is also important part of the rubric, as it provide description of all the steps that need to be followed for proper functioning of the project. This rubric also ensures that everyone is working on different parts of the project and test cases are properly created and developer can see all the failing and passing test cases.

Therefore we have covered the points mentioned in the rubric and described about the Linux Kernel Best Practices in the above section.

4 CONCLUSION

So for CSC 510 SE Project 1, the rubric provided meets all the requirements of the Linux Kernel Best practices and the above sections shows how they are connected. Hence the quality of the project can be clearly assessed by this rubric provided as it follows all the major principles.

Received 24 August 2022; revised 05 October 2022; accepted 7 October 2022