

# WolfJobs

Atharva Patil  
apatil25@ncsu.edu  
North Carolina State University  
Raleigh, NC, USA

Atharva Joshi  
amjoshi3@ncsu.edu  
North Carolina State University  
Raleigh, NC, USA

Banpreet Chhabra  
bchhabr@ncsu.edu  
North Carolina State University  
Raleigh, NC, USA

Vaishnavi Patil  
vpatil6@ncsu.edu  
North Carolina State University  
Raleigh, NC, USA

Shivam Pednekar  
spednek2@ncsu.edu  
North Carolina State University  
Raleigh, NC, USA

## ABSTRACT

WolfJobs is a job portal which provides Recruiters a mean to post temporary job openings and Candidates to apply for the jobs without being concerned about permanent associations. Unlike the Job Portals that display only Permanent job positions, WolfJobs focuses on providing Temporary positions to anyone who wants to work for a limited period of time. Be it a short-time Technical requirement or hiring a short-term Cashier for a dining place, WolfJobs covers it all! Every user associated with WolfJobs has a choice to Post a job or Apply for a job. WolfJobs provides a universal account for a specific user so that He/She can manage the applications as well as job postings efficiently.

### ACM Reference Format:

Atharva Patil, Atharva Joshi, Banpreet Chhabra, Vaishnavi Patil, and Shivam Pednekar. 2021. WolfJobs. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 KERNEL DEVELOPMENT BEST PRACTICES

Linux Kernel truly shows the power of open source technologies and the way collaboration between team members can lead to a successful product. This was possible because of the flexibility the team members maintained as well as a set of standards that were followed during the development of the kernel. These best practices have been documented and available to be incorporated in the development of upcoming projects. Being open source, these practices are reviewed and upgraded as per the needs.

## 2 DISTRIBUTED DEVELOPMENT MODEL

All the team members working on a single module or a single member reviewing all the review requests could be cumbersome and would reduce the productivity and efficiency of the project. Instead a distributed model can be implemented wherein modules are distributed among team members and also pull requests are

reviewed in a collaborative manner. This reduces concentration of work at one point and in turn increases the efficiency.

The various actions implemented as part of the Distributed Development Model are as follows:

### 2.1 Workload is spread over the whole team

The workload consists of developing different features and presenting them. Collaborating with the team members helps in completing the tasks in an efficient manner. WolfJobs development team followed a similar approach wherein the modules were divided between team members. Each team member was responsible for the design, development and testing of the modules. The team tested the modules developed by other team members to have a four-eye check.

### 2.2 Number of commits and Number of commits: by different people

The number of commits signify the amount of work done as the modules are divided into different functionalities and assigned to different team members. The Number of commits: by different people denotes the amount of work done by a particular team member. As the WolfJobs project was divided into a number of modules, each team member was assigned a module and it was successfully committed to Github.

### 2.3 Issue reports: there are many and Issues are being closed

Issue reports are a way to keep a track of different errors/bugs and the development modules yet to be developed. Issues are meant to be assigned to different team members. WolfJobs has created a kanban board on GitHub and the issues were raised, assigned and closed in a timely fashion.

### 2.4 DOI Badge exists

A Digital Object Identifier(DOI) is a unique identifier for a repository. In a large team working on multiple projects in a Distributed manner, a DOI can be used to uniquely identify the projects. As WolfJobs is a singular project, Zenodo has been successfully integrated to provide a DOI for the project.

### 2.5 Docs doco generated, format not ugly

A doco or Software documentation helps a user to know about the application. A project consists of different modules developed

Permission to make digital or hard copies of all or part of this work for personal or academic use, or to republish, is granted by ACM, provided that the copyright notice, this permission notice, and the full citation are included in the copy. This work is distributed under the Creative Commons Attribution 4.0 International License.

**Unpublished working draft. Not for distribution.**

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](mailto:permissions@acm.org).

Conference'17, July 2017, Washington, DC, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

2021-09-30 20:53. Page 1 of 1-2.

by different people. The doco will specify the usage of different modules. WolfJobs has used docopt to generate the documentation.

## 2.6 Docs: what: point descriptions of each class/function (in isolation)

The point description of each class/function lets the new developer team members know what the function does. This consists of the I/O, description and the arguments to be passed into the function. WolfJobs has incorporated the following format for the description:

Description

Input

Output

## 2.7 How, Why and Short Video

The answers to the How's and Why's attract the end user to use the product. This can be in the form of Pictures, Short Video or flow diagrams. WolfJobs has a Readme.md created in its GitHub repository which contains an attractive punchline with the workings of the application in the form of screenshots and a short video

## 3 TOOLS MATTER

With the right tools, the application functions as per the standards and the maintenance and development becomes easier.

### 3.1 Version Control

Version Control helps in managing and tracking the changes to the application being built. WolfJobs makes use of GitHub to track the changes to the code base by allowing different members to work concurrently and push their changes without affecting the other functionalities being developed.

### 3.2 Style checkers, Code formatters and syntax checkers

The style checkers, code formatters and syntax checkers tools help in identifying and automatically converting the different code practises followed by the team members into a uniform coding standard. WolfJobs has made use of autopep8 - a python package that checks the codes against standard coding practises and transforms the code if it does not meet the requirements.

### 3.3 Testing - Code coverage, Analysis Tools, Test Cases and Routinely Execution

Testing makes sure that the functionality developed is working as expected without any kind of errors. Test Cases help in covering different scenarios in which the application might fail. Automated test execution is essential to identify the potential errors due to new commits or a change in the dependent environment. Code coverage ensures that all the lines of codes are being tested.

## 4 CONSENSUS ORIENTED MODEL

The Consensus Oriented Model makes sure that everyone agrees to the functionality developed for the system. It also gives an equal stand to each member which results in innovative/efficient ideas

## 4.1 Contributing.md

Open Source allows anyone to contribute to a software. In order to maintain the application without breaking it, the potential developers must follow the guidelines specified in Contributing.md. WolfJobs has its own Contributing.md in its GitHub repository which must be followed before adding additional functionalities in it.

## 4.2 Discussion of the Issues, Chat Channel, Failing Test cases

Discussion consists of solving different issues, conversing on a platform and breaking changes. Conversations about the issues and test cases helps in coming up efficient resolutions. WolfJobs has a chat channel on Whatsapp and daily video discussions are held on Zoom wherein the issues are agreed upon and the best solution is implemented.

## 5 ZERO INTERNAL BOUNDARIES

Zero Internal Boundaries states that the application can be modified by any team member provided the change is justified.

### 5.1 Using the same tools

It is essential that all the team members use the same tools to develop functionalities in order to comply with the coding standards without breaking the application. All the team member of WolfJobs have agreed upon a requirements.txt which specifies the packages and its corresponding versions used in the application. The Team has also distributed an equal workload on the basis of design, development, documentation and testing.

## 6 SHORT RELEASE CYCLES

Short Release Cycles ensures that the end users get the functionalities developed in an incremental manner. This enables the project team to incorporate any feedbacks/bug fixes in the upcoming cycles.

## 7 NO REGRESSION RULE

The No Regression Rule conveys that the previously released functionalities are not changed frequently. The new functionalities should not break the existing one.