Job-Portal

Akshat Savla
asavla@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Anagha Patil
apatil@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Madiha Mansoori
mmansoo@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Mohit Soni
msoni@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Tilak Satra
tsatra@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

JobPortals is a simple portal which allows recruiters and dco-ordinators to post jobs for students and jobseekers, it additionally allows jobseekers to view these postings. The aim of this portal is to provide jobs spanning not only technical domains but simpler domains like dining and student centers, thereby it can act as a unifying means for students and recruiters.

ACM Reference Format:
Akshat Savla, Anagha Patil, Madiha Mansoori, Mohit Soni, and Tilak Satra. 2022. Job-Portal. 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

The Linux Kernel is an excellent example of the strength of open-source technologies and how teamwork can result in a successful product. This was made feasible by the team members' willingness to be flexible and by the standards that were upheld while the kernel was being developed. These best practices are ready to be used in the creation of subsequent projects and have been documented. These practices are evaluated and updated as needed because they are open source.

2 DISTRIBUTED DEVELOPMENT MODEL

Teams operating in dispersed locations are required to collaborate in a time- and location-neutral manner under the distributed software development model. Even though members of a remote software development team may live in the same ZIP code, they never share a workspace.Modules can be distributed across team members in a distributed paradigm, and pull requests can be reviewed cooperatively. This decreases the amount of work that needs to be concentrated at one time and boosts productivity.

2.1 Workload is Spread across all team members

In order to avoid issues like delivery delays, disgruntled or unmotivated resources, and many others in a distributed setting, spread the task among the members of your distributed agile teams equally. Our project followed a distributed project model where our workload was divided between the multiple team members in advance and the developed models were tested by the rest of the team members for an all rounded vision strategy.

2.2 Significance of number of commits

The number of commits can be used as a measure of the amount of work done by the team to better understand the efficiency of the team. Greater number of commits can signify substantial amounts of work on the project. Our project followed a distributed model and individual members were assigned certain modules which were developed on their branches and eventually committed.

2.3 Issue reports

You can track your work on GitHub, where development takes place, using issues. The issue's timeline displays the cross-reference so you can keep track of related work when you refer to an issue in another issue or pull request. You can connect an issue to a pull request to show that work is in progress. Issues are assigned to different team members and closed in a timely fashion.

2.4 DOI badge

A repository's unique identifier is called a Digital Object Identifier (DOI). A DOI can be used to specifically identify each project in a large team working on several in a distributed fashion. Zenodo has been integrated into our project.

2.5 Docs, doco generated

Doco, also known as software documentation helps users to better understand the application that has been developed and provides insights on its features and usage.Any software project contains several modules, the specification of all of these modules is documented by doco.

2.6 Docs

Docs are useful for informing the new development team about functionalities of the software.It includes details such as the performance of functions, input output details, how to call the functions, and a basic description of the function.

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

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.7 How, Why and Short Video

The how why what section of the readme offers useful information for the user to understand the cause and functions of the project. The readme offers general guidelines on how to run the system, additionally it includes a video on the functioning of the system.The readme also contains visual descriptions to explain the systems

3 TOOLS

The usage of the right kind of tools is useful for maintenance and development of software that meets industry standards.

3.1 Version Control

Version control allows multiple developers to work on the same project simultaneously while tracking changes made by other developers.JobPortal uses git as a version control system to manage concurrent development without affecting the other versions of the software.

3.2 Style checkers, Code formatters and syntax checkers

The tools for style checking, code formatting, and syntax checking assist in recognizing and automatically converting the various coding standards used by the team members. the codeformatter 'prettier' has been used in our project. A object called callForObject() has been used by Jobsportal to check the codes against accepted coding standards and alter them if necessary.

3.3 Testing-Code coverage, Analysis thors and Affiliations

Testing ensures that the provided feature functions correctly and in accordance with plan. Test cases help to deal with different failure scenarios for the application. To find potential issues brought on by fresh contributions or a change in the dependent environment, automated test execution is crucial. Code coverage guarantees that every line of code is tested. Our project mmakes use of maven along with github actions to check for code coverage and test cases.

4 CONSENSUS ORIENTED MODEL

Effective decision-making in a group or team can be challenging at times. This can be accomplished step-by-step using the Consensus Oriented Decision Making Paradigm, often known as the CODM model, resulting in successful group collaboration. Everyone must concur with the functionality created for the system, and the consensus oriented model ensures this. Additionally, it provides everyone participant an equal voice, which produces creative and effective ideas.

4.1 Contributing.md

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

4.2 Discussion of issues, chat channel and failing test cases

Often resolution of failing test cases and issues that arise due to merging, requires proper communication amongst the various team members. Conversations about these test cases that were failing and merge conflict resolution was done through discord in the development team of jobsportal.Regular meetings and active discussions were done across our team to conclude on best solution.

5 ZERO INTERNAL BOUNDARIES

The principle of zero internal boundaries allows for any team member to modify the code of any other team . The majority of the time, developers focus on particular areas of the kernel, although this does not exclude them from modifying any other area as long as the changes are justified. This procedure makes sure that issues are resolved at the source, preventing the need for several workarounds, which are always detrimental for kernel stability. Additionally, it offers the developers a broader perspective of the system as a whole.

5.0.1 Using the same tools. It is essential for the project development, that all team members use the same tools and technologies functioning at the same version to wade through issues that can arise from deprecated functions and technologies. All the team members worked with the technologies and versions listed out in requirements.txt which specifies the packages

6 SHORT RELEASE CYCLES

Since this is a small scale project, it doesn't have multiple release cycles.

7 NO REGRESSION RULE

The majority of the time, developers focus on particular areas of the system, although this does not exclude them from altering any other area as long as the changes are justified. This procedure makes sure that issues are resolved at the source, preventing the need for several workarounds, which are always detrimental for kernel stability. Additionally, it offers the developers a broader perspective of the system as a whole. A key feature of the regression rule that the additional functionalities should not break the functioning of the existing functionalities.