Smart-todo

Chiu, Ching-Lun
cchiu4@ncsu.edu
North Carolina State University
Raleigh, North Carolina, United States

Yu, Hsueh-Yang
hyu25@ncsu.edu
North Carolina State University
Raleigh, North Carolina, United States

Lin, Po-Hsun
plin8@ncsu.edu
North Carolina State University
Raleigh, North Carolina, United States

Ku, Li-Ling
lku@ncsu.edu
North Carolina State University
Raleigh, North Carolina, United States

Chiang, Chen-Hsuan
cchiang5@ncsu.edu
North Carolina State University
Raleigh, North Carolina, United States

ABSTRACT

Smart-todo is a upgrade to the original Smart-todo. Smart-todo is a job portal which provides Recruiters a mean to post temporary job openings and Candidates to apply for the jobs. But, the original Smart-todo lacks features, it's unusable since only a few things works, candidates cannot save their favorite jobs, they cannot update their own profile...etc. In Smart-todo we've added an advanced search mechanism where applicants can filter jobs by their locations, job type or job industry; we also added auto fill so that applicant's info will auto fill when they are filling out an application. Email update is also added so that candidates can keep track on their applied jobs.

ACM Reference Format:
Chiu, Ching-Lun, Yu, Hsueh-Yang, Lin, Po-Hsun, Ku, Li-Ling, and Chiang, Chen-Hsuan. 2018. Smart-todo. 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 KERNEL DEVELOPMENT BEST PRACTICES

Linux Kernel is the prime example of open source technologies and the collaboration between team members. This is possible through a series of practices and keeping apace of the latest trends. These practices have been documented and is available to be incorporated in the development of upcoming projects.

2 DISTRIBUTED DEVELOPMENT MODEL

A single member reviewing all the requests or all the team members working on the same branch could be cumbersome and would waste unnecessary time. A distributed model is the best way of developing, modules can be distributed among team members, pull request can be reviewed by any team members. This method prevents all team members working on a single module and improves efficiency. The following are some actions that one can implement as part of the Distributed Development Model:

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2.1 Workload is spread over the whole team

The workload consists of implementing different features and presenting them. Each team member will be working on their familiar area or they can collaborate with each other. The development of Smart-todo follows such practice where features were divided between team members and each team member can review others features.

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

The number of commits shows the amount of work done by each member on their module. For Smart-todo each feature is assigned to a team member and each team member commits their code to Github

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

Issue reports is a indication that codes are regularly tested and inspected, it is also a good way to keep track of different errors and bugs for each developing feature. For Smart-todo several issues are reported and discussed, and then closed.

2.4 DOI Badge exists

Digital Object Identifier(DOI)is a persistent identifier for a repository. In a large team working on multiple projects at the same time, a DOI can be used to identify projects. Since Smart-todo is a single project, Zenodo was integrated as the DOI for the project.

2.5 Docs doco generated, format not ugly

A document for software project helps user to get familiar with the application. Since a project consists of different features user will get used to the application faster with the help of a doco. Smart-todo 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. It consists of the I/O, description and the arguments to be passed into the function. Smart-todo2.0 has incorporated the following formant for the description:
Description
Input

Output

2.7 How, Why and Short Video

The answer to How, Why can intrigue end user to use the project. It can be in any form, videos, pictures or flow diagrams are some typical choice. For Smart-todo we created a short video explaining the improvement from the original Smart-todo.

3 TOOLS MATTER

With all team members using the same and correct tool, it can ensure that all team members work on the same page and it also makes maintenance and development much more easier.

3.1 Version Control

Version control tools help tracking and managing changes in the project. Smart-todo uses Github as the version control tool. It allows us to work in parallel without collision and push changes without affecting others code.

3.2 Style checkers, Code formatters and syntax checkers

The style checkers, code formatters and syntac checkers tools help in identifying and automatically converting different coding types to a uniform standard. Smart-todo uses autopep8-python package that checks the codes against standard coding practice.

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

Testing makes sure that the functions we developed works as intended and didn't have any errors. Test cases simulate different scenarios that the project will be exposed to. Automated test execution makes sure that new commits and code change will not cause any problem. Code coverage can ensure that every line of code is executed and being tested.

4 CONSENSUS ORIENTED MODEL

Consensus Oriented Model makes sure that everyone is on the same page for the features developed for the project. It also gives each team member chances to provide innovative ideas.

4.1 Contributing.md

Open source allows anyone to contribute to a software. Since Smart-todo is a contribute to a the original Smart-todo, we followed the Contributing.md from the original Smart-todo to add new features.

4.2 Discussion of the Issues, Chat Channel, Failing test cases

Frequent discussion of the issues makes sure that issues are being dealt with. Smart-todo has a chat channel in Line, where daily discussion happens and works on the project are being assigned.

5 ZERO INTERNAL BOUNDARIES

Zero internal boundaries states that the project can be modified by each and every team member provided the changes are justified.

5.1 Using the same tools

It is essential that all team members use the same tools in developing the project, since coding standard need to be applied. For Smart-todo all team members uses tools in requirement.txt which also specifies the version for them.

6 SHORT RELEASE CYCLES

Short release cycle make sure new code is immediately integrated into a stable release, it also enables the team to incorporate any feedbacks in the upcoming cycles.

7 NO REGRESSION RULE

No regression rule makes sure that the previously released functions are not changed frequently and new functions should not break the existing functions.