BurnOut

Shantanu Pandey
spandey4@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Leanne Serrao
lserrao@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Sahil Rajesh Nande
snande@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Varun Garg
vgarg4@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Urmi Kashyap Pathak
upathak@ncsu.edu
North Carolina State University
Raleigh, NC, USA

ABSTRACT

BurnOut is an easy to use application that keeps track of a user's daily calories gained and burnt. It can help the user to set goals such as weight loss/gain. User can edit his profile to enter weight, height and goal. This document provides a major perspective for the users to understand and take up the project as an Open source software and add on multiple features before releasing to the market. Also, the document aids the developers in understanding the code and acts as a reference point for continuing the project.

ACM Reference Format:
Shantanu Pandey, Leanne Serrao, Sahil Rajesh Nande, Varun Garg, and Urmi Kashyap Pathak. 2021. BurnOut. In *Conference '17: In Proceedings of ACM Conference (Conference '17)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.5281/zenodo.5534872

1 KERNEL DEVELOPMENT BEST PRACTICES

The Linux kernel is a prime example of the power of open source and the merits of constructive collaboration that can lead to a successful product. It can be made possible via coordination and flexibility between team members and a set of standards followed during the development of the kernel.

1.1 Short Release Cycles

Short Release Cycles were introduced to eliminate glitches and problems occurring in the long release cycles.Long release cycles meant that vast chunks of code had to be integrated at once, which proved to be rather inefficient. It also translated into a lot of pressure for the developers to integrate features in the upcoming release even if they were not completely stable or ready.

1.2 Distributed Development Model

All the team members working on a single module could be difficult to implement and would in turn reduce the efficiency of the project. Alternatively, a distributed model can be implemented where different modules are distributed among members. This reduces congestion of work and increases efficiency.

1.2.1 ***Workload is spread over the team****.*
Different features and modules of the project are divided among team members so that completion of the tasks can be done in an efficient manner. BurnOut development team distributed the tasks according to their strengths which in turn helped the overall development process.

1.2.2 ***Number of commits and Number of commits: by different people****.*
The number of commits specify the amount of work done collectively as the project is divided into different modules. The number of commits by different people specify the amount of work done by each team member. Each team member was assigned a distinct module which was committed successfully to Github.

1.2.3 ***Issue reports: there are many and Issues are being closed****.*
Issues are assigned to different team members. The frontend and backend GitHub repos of BurnOut have different issues which are assigned to different team members to close in a timely manner.

1.2.4 ***DOI Badge exists****.*
DOI(Digital Object Identifier) is a unique identifier for a repository. It can be used to uniquely identify the project. Zenodo has been used to generate DOI for BurnOut project.

1.2.5 ***Docs doco generated, format not ugly****.*
Any software project consists of different modules developed by different team members. The doco will specify the usage of different modules. BurnOut has used docopt for generating the documentation.

1.2.6 ***Docs:what:point of each class/function (in isolation)****.*
The description of each function lets the new developer team members know of the type of methods, input parameters and returned JSON output from each API calls.The returned JSON output is made visible to the user using React components. BurnOut has incorporated the following format for the description: Parameter Method Description Output

1.2.7 ***How, Why and Short Video****.*
The user should be interested to use the product which can be achieved by adding pictures, short videos or GIFs . BurnOut has a README.md created in the GitHub repository which contains an

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed 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 '17, July, 2017, New York, NY, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.5281/zenodo.5534872

attractive logo for the application, punchline, GIFs that show the workings of the project and a short video.

1.2.8 *Version Control*.

BurnOut uses GitHub to track changes and version of the application by allowing team members to push their changes.

1.2.9 *Testing*.

Testing validates the working of the functionality without any errors. BurnOut team has incorporated two test cases to create session for login/signup and also created another test case to return the calorie with the associated food item. BurnOut uses Postman for automated debugging/testing.

1.3 Consensus Oriented Model

A Consensus Oriented Model states that a proposed change cannot be integrated into the code base as long as a any developer in the project is opposed to it. It also gives equal opportunity to each member for implementing their ideas to the application.

1.4 Contributing.md

In order to maintain the application, the team members must follow the guidelines specified in Contributing.md. BurnOut has a Contributing.md in the GitHub repository which must be followed by all the team members at all times.

1.5 The No-Regression Rule

The No-regression rule conveys that there should not be changes made in the system at the cost of quality. Therefore avoiding frequent changes.

1.6 Zero Internal Boundaries

Zero Internal Boundaries imply that all the members working in a project are able to make changes in any part of the project as long as the changes are justifiable. The BurnOut team has distributed an equal workload for the development of frontend, backend and testing. Requirements.txt have been added by the team members which specifies the packages used in the application.