Part1:

Good afternoon, everyone. My name is Zhou Zhiqiang and I’m very delighted to have the opportunity to present my internship work here. My presentation is in four parts. Firstly, I’ll introduce the background and key metrics in the dashboard. Secondly, a demonstration is given. Thirdly, I’ll talk about the workflow behind the dashboard. And finally, Q & A. Please feel free to interrupt me whenever you have any question.

Let’s start with a story. This year the goal of our insight team is to attract more people to make contributions on Docs website and Alex is in charge of this task.

Firstly, take a look at this funnel. The absolute number makes not much sense here and we care more about the percentage. We can see that in July, the percentage of people who click the contribution button is 0.23% and the percentage of people who make a pull request is 1.70%, which are relatively low.

So for Alex, how to increase the percentage?

Actually, Alex has organized us to bring up some hypotheses, for example, we may give the people who make contributions some awards. It sounds reasonable but how to verify its feasibility. Currently, we lack an integrated UI hub to present the contribution situations. Thus a dashboard is made to help Alex judge whether a hypothesis works or not.

I’d like to introduce the key metrics in dashboard briefly: the publish here where people can modify one or more topics. For people outside Microsoft, the publish is mainly accomplished by a pull request, so the pull request should be considered and a pull request will in turn cause a contribution in GitHub concept, the contribution and contributors are included.

Part2:

So Let’s turn to the dashboard. This is the overall view of the dashboard, which can be divided into four parts: the publish, pull Request, Contribution & Contributors and here on top. All the items are categorized by repository, tenant and site. And a time selector is used here to configure the time range.

For the three key metrics, the per day trend or per month trend and the total number are shown. Suppose that Alex has implemented an A-B testing, if the hypothesis works, he will see an obvious increase in these charts. As to publish, if Alex wants more detailed information, he can look at these four charts which are further categorizations for publish.

For example, he can see the statistics of how many lines or words inserted or deleted in a topic. (If the proportion of small changes is high, we may guess people are usually willing to make some small adjustments such as word spelling)

As I said just now, for people outside Microsoft, the publish is mainly accomplished by a pull request. Alex then wants to know the behaviors for these parts of people. Then he will turn to these three charts where the pull request is categorized by its status, the number of commits in a single pull request and the duration of the pull request from created to merged.

For time reason, I will not introduce these charts in details. You can know more about them by fetching the document whose link is below.

What’s more, for Alex, by selecting the repo list, he can compare the performance of repos internally. If some certain repo performs much better than others, he can conversely look into this repo to find the reason and make further hypothesis. And look at this time selector, for example, the Docs is open to public in May, so Alex can drag it to see the data after May.

Part3:

To conclude, the dashboard offers a unified place to present the overall contribution situation and can help judge whether a hypothesis is feasible and efficient, make further hypothesis. Then I’d like to move on to the workflow behind the dashboard.

Firstly, we need to acquire the JSON data from GitHub and VSO by calling REST API. Secondly, the JSON data will be parsed. Thirdly, loading the information to database and finally using PowerBI to visualize the data.

So what are the challenges in this workflow? Firstly, I will gather the data that is missing. Do you still remember the changed lines and words in dashboard? Actually, GitHub only offers us the data of changed lines but as the topics are usually documents, the word change will be more reasonable and accurate. Thus, I design an algorithm to get this part of data.

Secondly, I will complement the data that is not complete previously. And thirdly, I will format the data so that it can be well displayed, which takes quite a lot of efforts.

Well, that’s all for my presentation. Thank you for listening and sincere thanks for Alex and Nanxuan ‘s helps in my internship. Any question?