Season of Docs Case Study: MicroPython

Organization or Project: MicroPython

Organization Description: MicroPython is a version of the Python programming language written specifically to run on microcontrollers, embedded systems, and other computing devices that have minimal resources.

Authors: Damien George, Matt Trentini

Problem Statement

MicroPython has become a popular platform in the embedded space and the language and ecosystem have grown rapidly. Unfortunately the documentation hasn't kept up with the rapid pace of feature development. Improving the documentation will help many people - particularly newcomers! - to use the platform and allow it to continue growing.

Proposal Abstract

The original GSoD proposal, Update MicroPython port documentation, discussed the goal of adding documentation for MicroPython *ports*, or families of microcontrollers, that had little or no documentation. The purpose was to make it "...easier for new people to start with the project, for experienced developers to use it as a reference, and for a reduction in the support load".

Project Description

Creating the proposal

The process began by reviewing the previous GSoD proposals to consider what did, and did *not*, work well in the past. Then a core group of MicroPython developers came together and discussed some of the areas that could be best served with improved documentation. The proposal was born out of these activities.

Budget

It proved difficult to estimate the budget. Although experienced in estimation, the authors knew that the documentation improvements were somewhat open to negotiation, the effort would be *highly* dependent on the experience of the technical writer and the tasks had not been broken into sufficiently small enough components to produce an accurate estimate.

The budget selected did not appear to be wildly inappropriate for the scope of work and the Technical Writer was satisfied with the payment so the estimate could be considered reasonable.

To refine budget estimations further in the future, we note that the hardware costs were overestimated and the amount of time required to be invested by the mentors was significantly underestimated.

Participants

The team working on this project was:

- Damien George (MicroPython creator and core developer)
- Matt Trentini (MicroPython community member)
- Jim Mussared (MicroPython core developer)
- Niti Kaur (Technical Writer)

Many Technical Writers applied, via the GSoD system, to work on the MicroPython project. The proposals were thoroughly read and carefully considered and, through consensus, Niti was selected. Her proposal was well written, showed interest and prior knowledge of the MicroPython project, and covered most of the tasks we had suggested.

Selecting Niti wasn't trivial; there were other very capable candidates. Each MicroPython member rated each of the Technical Writers by the quality of their proposal. Lower ranked Writers were quickly eliminated while a 'top 5' was slowly established. A spirited discussion followed to present the case for the more well-liked candidates; this continued until one candidate - Niti - was selected.

Timeline

Below is a timeline of features that were added to the project during the course of this Season of Docs.

Stage	*Completed By*
General rp2 documentation - including quickref - added	20 July
rp2 PIO assembly instructions and tutorial added	13 October
Random module documented	13 October
Mpremote documented	14 October
Manifest system documented	14 October
Stm module documented	14 October
Esp32 register tutorial added	14 October
Sys.settrace documented	19 November

Python language support	*9 December*
Improve download manager board images	*9 December*

Results

While the project started by following the original proposal, priorities changed as it became clear that other documentation activities were more pressing. The team met online with Niti weekly to provide feedback on work completed in the past week, support with issues, mentoring for improving documentation writing, and guidance on what should be worked on next.

The original proposal focussed on providing per-port documentation for a number of microcontroller families. The most pressing of these was the rp2 and that was an early focus of the project; this was a large task, split in two, completed in July and October. Other ports were deemed less important and a collection of documentation tasks were then selected to undertake (see the Timeline, above).

Metrics

It was expected that the MicroPython forum could be a source of metrics; an analysis could be performed to determine if forum questions could be answered with links to new documentation entries. It's too early to say but there are examples of links to the new documentation.

It was also expected that the addition of the documentation updates may trigger further improvements to the documentation. That has been observed, particularly with the addition of the rp2 documentation which was previously lacking.

Finally, it was thought that more issues may be raised against the new additions to the documentation. Currently that hasn't happened; it's not yet clear if that's because the documentation contains few defects or if more issues will arise over time.

Analysis

MicroPython now has some additional, valuable documentation due to the efforts arising from the Season of Docs project, which is great! Feedback has already been received from the community that these have been welcome additions.

However, the improvements did come at a cost.

Some of the most useful additions were deeply technical and complex in nature and really beyond Niti's knowledge, through no fault of her own! We found that the core team was required to invest a *significant* effort to provide Niti with the appropriate level of information and background so that she was able to appropriately write about the topics.

Given that there are few core developers on the project it’s difficult to say whether that investment in time would have been better spent on implementation.

Communication was, at times, difficult. Internet access was patchy for Niti which made voice communications challenging.

Niti was also new to using git and sphinx - tools needed to create the documentation. The core team was so used to them that they had forgotten that there is a large ramp-up to learn them.

Summary

Although there were differences from the original proposal, and some challenges, the Google Season of Docs experience was very positive. MicroPython now has much-needed documentation on a new, exciting microcontroller family and has filled gaps in many areas of the online documentation. New users will benefit from the tutorials and introductory documentation and experienced developers will appreciate the additions to the reference material.

Niti was easy to work with and keen to learn, we thank her for her contributions! We hope that she’ll continue building on expanding her MicroPython knowledge and being a part of the community.

Advice for other projects:

- Invest in training your Technical Writer early in the schedule. Make it as easy as possible for them to learn the necessary tools for your project. Allow them the time to practice and become competent with the tools.
- Establish clear processes around setting short-term tasks and goals. We had some issues with voice communication that was later alleviated by using a Slack channel to capture the pertinent information; it would have been more efficient to make use of this earlier. Even consider a separate issue list or Kanban board solely for the Season of Docs tasks.
- Perhaps scale back your proposals or select appropriate goals. Many of the Technical Writers involved in the Season of Docs project are inexperienced and trying to learn their craft and, often, a new domain. Tasks we chose were selected because they were of most value to developers in the community but they were deeply technical in nature - which meant they were *especially* difficult to write about. Perhaps selecting different tasks - tutorials or examples in our case - may mean your Technical Writer will be able to generate better outcomes.
- Encourage members of your community to apply for a Technical Writer position. A large part of the effort of writing is learning the domain; if you can select someone who knows the domain they’ll be much more effective.