Google Season of Docs Case Study Template

README: The purpose of this case study is for others to learn from your experience. You should provide an honest assessment of how things went. We are interested in hearing about mistakes and challenges as well as successes. Do your best to answer ALL of the following questions to the best of your ability.

Cloud Native Buildpacks: 2024 Google Season of Docs Case Study

Organization or Project Name: Cloud Native Buildpacks

Season of Docs link: https://github.com/buildpacks/docs/issues/677

Organization Description:

Buildpacks streamline the process of building, deploying, and managing applications by automating the creation of container images from source code. Their key objectives include detecting application dependencies, configuring runtime environments, and ensuring consistent builds across different platforms. By abstracting away complex infrastructure details, buildpacks enable faster deployments, enhance developer productivity, and ensure better security through standardized and reproducible workflows. Buildpacks have particular advantages in modern cloud-native environments, reducing manual effort and simplifying application scaling.

Authors: Aidan Delaney and Natalie Arellano

Summary

Fill out the following table and provide brief answers to the questions. You can provide more details in later sections. ***We recommend you write out the detailed sections first then come back to write the summary.***

# Tech Writers	TW Project Hours	Budget	% Project Completed
1	400	$8000	100%

What problem were you trying to solve? And how did you try to solve it?

We needed to translate our technical specification into language, examples and explanations suitable for three audiences: application developers, buildpack authors and platform operators.

What were the outcomes of your project?

We now have better cross-linked documentation for application developers and buildpack authors. In addition, we have much better tutorial content for platform operators.

What went well? And what would you do differently?

Our technical writer was excellent. They asked questions about our core specification which led the mentorship team to write rough notes. From the rough notes the technical writer produced copy that our end-users are comfortable reading.

What advice would you give to other projects trying to solve a similar problem with documentation?

We found our approach of identifying “stubs” that needed improvement was a good approach to onboard a technical writer. We started with stubs we considered straightforward and progressed to more complex stubs. This built confidence in our process and allowed the technical writer to develop their own direction for later project contributions.

Project Description

Describe the project by filling out the following sections.

Project Proposal

https://github.com/buildpacks/docs/issues/677

Proposal Creation Process

The project proposal evolved from an earlier user survey which highlighted issues with our documentation. The leadership team collaborated on developing the project scope and the detail was delegated to the learning team.

Budget

Answer the questions in the table below.

How much money did you ask for?	$8250.00
How did you come up with this estimate?	We estimated the number of hours necessary to identify the project tasks. That number was multiplied by a minimum hourly wage valid in the US and all European countries.
How many hours of work did you budget for the project?	~400
How many hours of work were actually needed for the project?	The technical writer worked their expected hours. Two mentors contributed several hours per-week for interviews, synchronous meetings, asynchronous queries and for reviews.
What other expenses did you include in your budget?	Small budget for t-shirts to publicise documentation upgrades at conferences.
Did you run into any budget surprises during the project (e.g. misestimates)? If so, please explain.	We ran into some small surprises to do with the cost of international money transfers ~$80.

Tech Writer Recruitment

A Google Form was advertised to prospective technical writers. A list of criteria for strong submissions was established. Submissions were gathered and compared with the criteria for strong submissions. This process resulted in a ranked list of submissions. The top submissions of our ranked list were invited for an interview.

The interview process consisted of a common set of questions, with time set aside for candidates to distinguish themselves. After the interview process, the Cloud Native Buildpacks learning team reviewed the candidate performance and our top three preferences were noted. We were delighted that our most preferred candidate accepted the position.

We quickly established a monthly set of goals for our technical writer and a weekly synchronous catch up. Our technical writer contributed up until the end of the GSoD project and her contributions have influenced the future direction of our documentation.

Other Participants

In addition to our technical writer, and the Cloud Native Buildpacks Learning Team, we are greatly indebted to Nancy Lancaster for organizing the interview process. We have also had significant support from Nate Waddington the head of Mentorship & Documentation at the CNCF.

Timeline

Our initial timeline is contained in the following table:

What	Timeframe	Duration
Applicants submit details to form	April 15–April 22	1 week
Team reviews applications	April 22–April 24	3 days
Team narrows down the top 5 candidates	April 25	1 day
Team sends notices out and asks for applicant to book a time for an interview	April 26	1 day
Team conducts interviews	April 30–May 2	3 days
Team meets and chooses top candidate to hire	May 3	1 day
Candidate is hired	May 6	1 day
Tech writer performs first step in proposal “Project familiarization (go through existing documentation to understand personas)”	May 7–May 28	3 weeks
Tech writer performs next step in timeline of proposal “Review buildpack spec through the lens of a buildpack author to identify and fill in missing content on docs website”	May 29–June 30	4.5 weeks
Tech writer performs next step, “Review platform spec through the lens of a platform operator to identify and fill in missing content on docs website”	July 1–August 31	8.5 weeks

Tech writer performs next step, “With learnings from buildpack and platform specs, enumerate buildpacks features and the subset of concepts that are relevant to app developers; identify and fill in missing content on docs website	September 1–October 31	8.5 weeks
Finalize project, gather community feedback	November 1–November 22	3 weeks
Organization administrators submit their case study and final project evaluation	November 22–December 10	2 weeks

The September 1 to October 31 period developed a focus on documenting `lifecycle` for platform operators. This was a complex challenge given the lack of existing documentation in this area and the level of technicality contained in the specification. The discussion thread around this contribution runs to 134 comments

Deliverables

Planned Deliverables

Relevant links might include published docs, pull requests, or other artifacts.

Deliverable	% Complete	Relevant Links	Notes
Buildpack Authors Stubs	100%	• https://github.com/buildpacks/docs/pull/734 • https://github.com/buildpacks/docs/pull/737 • https://github.com/buildpacks/docs/pull/739 • https://github.com/buildpacks/docs/pull/744 • https://github.com/buildpacks/docs/pull/752	
Buildpack Users Stubs	100%	• https://github.com/buildpacks/docs/pull/769 • https://github.com/buildpacks/docs/pull/777	
Buildpack Platform Stubs	100%	• https://github.com/buildpacks/docs/pull/770	

Unplanned Deliverables

Deliverable	% Complete	Relevant Links	Notes
Platform Tutorial	100%	• https://github.com/buildpacks/docs/pull/769	Our technical writer identified a lack of documentation about the `lifecycle` component.

Metrics

Our technical writer is now the 8 largest contributor to our documentation codebase, and the second largest contributor over the last 6 months. In total, they are directly responsible for 20 commits and have had influence on most of the other documentation commits over the past 6 months.

We have seen a small uptick in different contributors since our technical writer updated our tutorials. In total, we have had 6 additional contributors since our technical writers' contributions which compares with 2 additional contributions in the 6 months prior to her work.

Analysis

Provide a short narrative about how the project went. Was the project successful? Why or why not? Or when will you be able to judge success? Did you face any unexpected hurdles or setbacks? Did the project result in any new or updated processes or procedures in your organization?

This GSoD project successfully delivered on our plans to expand the identified "stubs" in our documentation. We successfully improved our documentation for developers who are interested in a key, but previously poorly documented, `lifecycle` component of the buildpacks process. The Learning Team also benefited from the insight of a technical writer into the parts of our Buildpacks specification that are particularly difficult for end-users to untangle.

The Cloud Native Buildpacks Learning Team has had to make improvements to their process to provide a "rough translation" of the buildpacks specification as source notes for technical writers. These "rough translations" are not exposed to end-users as they mainly scaffold the communication between technical writers and the learning team.

The Learning Team created "stubs" in the documentation for our technical writer to improve. We believe that these small, well-structured tasks made it easier for the technical writer to onboard. By the end of the project our technical writer was driving the direction of her contributions. This resulted in a tutorial on "Orchestrate a build with the CNB lifecycle" which filled a large gap in our documentation.

The project was successful and overcame hurdles that are commonly reported in open-source projects. Mentors contributed to the efforts in their own-time and synchronous meetings were scheduled at times appropriate for a project spanning multiple time-zones. In addition, there were mid-project changes to staffing within some company departments that regularly contribute to Cloud Native Buildpacks. This resulted in the technical writer having to adopt a flexible approach to the contacts with whom they were communicating. We also recognize and thank contributors who had to pick up extra organizational roles due to personnel changes.

Lessons Learned

Use the tables below to highlight lessons learned -- for things that went well and things that could be improved. Use the listed topic categories as appropriate or add your own. Feel free to have multiple rows for a given topic as needed. These lessons learned will feed into our aggregated list of best practices for open source documentation projects.

What went well?

For lessons learned, add your own or indicate a "plus one" for any of the existing GSoD Best Practices.

Topic	What we did	Lesson Learned
Budget	We adopted best practice and ensured our hourly rate was in-line with employment law globally.	Expect small costs for international money transfer. We also did not need our budget for t-shirts as not everyone could travel to the same conference.
Communication	Regular updates to our leadership team and within our bi-monthly open-meetings.	
Mentorship	We assigned two mentors to the project, who were aligned on the project goals.	A larger mentor group consisting of core and peripheral mentors could have reduced some of the work on core mentors.
Metrics	Our key metric was the number of "stubs" that were completed.	Getting early coverage of "stubs" developed confidence in our approach and empowered the technical writer to drive their own goals.
Onboarding	We developed a list of "quick wins" to get the technical writer familiar with our project.	
Participants	We had regular scheduled synchronous catch ups.	Be flexible on times. There is no time-slot that suits a global open-source team.
Project Deliverables	Defined the deliverables as part of the project scope	
Project Management	Bi-monthly updates within our open Buildpacks community meetings	
Recruiting & Hiring	Adopted best-practice from HR departments with whom we were familiar	Establish a list of criteria by which your team evaluates applications. We believe this strongly contributed to our evaluation process being fair.

What could be improved?

For lessons learned, add your own or indicate a "plus one" for any of the existing GSoD Best Practices.

Topic	What we did	What we would do differently	Lesson Learned
Budget	We budgeted at a rate that would encourage global applications	We adopted best-practice for our budget and would not do this differently.	Stick closely to documented best-practice.
Communication	Bi-monthly updates in our open community meetings and regular updates on our asynchronous leadership channel	Involve more of the leadership team into communicating more widely	
Mentorship	Established a core mentorship team	Spread the mentorship responsibilities more widely	All mentors have a different combination of technical leadership, project management and communication skills. A wider set of mentors would ensure strength in all areas.
Metrics	Our user-survey identified gaps in our documentation. We turned these gaps into small tasks. Our key metric was then to complete all small tasks.		
Onboarding	The small tasks that formed our metrics were useful for onboarding.		Start with small tasks that exemplify the project direction.
Project Management	Reported regularly to the leadership team		
Recruiting & Hiring	Learned from colleagues in academia and HR about how to run a fair process.		Established a list of criteria to evaluate applications. Socialised the criteria with the leadership team so that everyone knew what to expect.

Appendix

If you have other materials you'd like to link to (for example, if you created a contract for working with your technical writer that you'd like to share, or templates for your documentation project, or other open documentation resources, you can list and link them here). The Appendix is also a good place to list links to any documentation tools or resources you used, or a place to add thanks or acknowledgments that might not fit into the sections above.