

Title and People

The title of your design doc, the author(s) (should be the same as the list of people planning to work on this project), the reviewer(s) of the doc, and the date this document was last updated. Specify the status of the doc (draft, pending review, reviewed, etc.). For example:

- Title: RuFaS Design Document Template
- Author(s): Pooya Hekmati
- Reviewer(s): [Names of Reviewers]
- Date: 10/02/2023
- Status: Draft

Overview

A high level summary that every developer and SME at RuFaS should understand and use to decide if it's useful for them to read the rest of the doc. It should be 3 paragraphs max. For example:

This design document outlines the development of a science-based product at RuFaS. It provides a high-level understanding of the project for developers, subject matter experts (SMEs), and stakeholders to determine its relevance and importance. The document aims to present a concise summary.

Context

A description of the problem at hand, why this project is necessary, what people need to know to assess this project, and how it fits into the technical strategy, product strategy, or the team's quarterly goals. The project is centered around the development of a science-based product to address a specific problem. It is vital to understand the problem's significance and how this project aligns with our technical strategy, product strategy, and quarterly goals. The context section will clarify why this project is essential and what knowledge is required to assess its scope and impact.

Requirements

A list of things that can be used to define project's success. Preferably, break the list into functional and non-functional requirements; as well as must to have and nice to have.

Milestones

Project milestones provide measurable checkpoints, helping project managers and stakeholders track progress. They encompass the design, implementation, validation, testing, and rollout phases of the project. Including calendar dates accommodates potential delays and ensures clarity about the project's timeline. As a rough rule of thumb, each milestone corresponds to a PR. Use calendar dates so you take into account unrelated delays, vacations, meetings, and so on. It should look something like this:

- Start Date: [Start Date]
- Milestone 1 — Design Phase Kick-off: [Date]
- Milestone 2 — Scientific Solution Iteration 1: [Date]
- Milestone 3 — Technical Solution Iteration 1: [Date]
- Milestone 4 — Scientific Solution Iteration 2: [Date]
- Milestone 5 — Technical Solution Iteration 2: [Date]
- Milestone 6 — Iterative Refinement: [Date]
- Milestone 7 — Implementation and Validation: [Date]
- Milestone 8 — Testing and Bug Fixes: [Date]
- Milestone 9 — Product Rollout: [Date]
- End Date: [End Date]

Example:

Start Date: June 7, 2018

Milestone 1 - New system MVP running in dark-mode: June 28, 2018

Milestone 2 - Retire old system: July 4th, 2018

End Date: Add feature X, Y, Z to new system: July 14th, 2018

Add an [Update] subsection here if the ETA of some of these milestone changes, so the stakeholders can easily see the most up-to-date estimates.

Existing Solution

In addition to describing the current implementation, you should also walk through a high level example flow to illustrate how users interact with this system and/or how data flow through it.

A **user story** is a great way to frame this. Keep in mind that your system might have different types of users with different use cases.

Proposed Solution

The proposed solution will be developed iteratively, with the scientific and technical solutions being refined together. This iterative process ensures that both aspects of the solution evolve in tandem. Scientific and Technical Solution Iterations:

- Initial Overview: [Provide a high-level overview of the combined scientific and technical solution.]
- Iteration 1: [Add details to both the scientific and technical solutions based on initial feedback.]
- Iteration 2: [Refine the scientific and technical solutions further.]
- Iterative Refinement: [Continue to improve both solutions collaboratively.]

Provide a big picture first, then fill in *lots* of details. Aim for a world where you can write this, then take a vacation on some deserted island, and another engineer on the team can just read it and implement the solution as you described.

Scientific Solution

1. High-Level Overview: [Provide a high-level overview of the scientific solution.]
2. Review by SWE: [Ask SWEs to review the scientific solution and propose a technical solution.]
3. Detailed Plan: [Add more details to the scientific solution based on SWE feedback.]
4. Review by SWE: [Ask SWEs to provide corresponding technical details for the scientific solution.]

Technical Solution

1. Technical Overview: [Provide a high-level overview of the technical solution.]
2. Review by SMEs: [Ask SMEs to review the technical solution and suggest improvements.]
3. Detailed Implementation: [Elaborate on the technical solution based on SME feedback.]

4. Review by SMEs: [Request SMEs to provide further insights and refinements to the technical solution.]

Alternative Solutions

What else did you consider when coming up with the solution above? What are the pros and cons of the alternatives? Have you considered buying a 3rd-party solution — or using an open source one — that solves this problem as opposed to building your own?

Testability, Monitoring and Alerting

This section is important, because people often treat this as an afterthought or skip it all together, and it almost always comes back to bite them later when things break and they have no idea how or why. Here, try to address this question: in future, how can one know if what you implemented is still working? How can they make sure their changes have not broken yours? Remember, this person can very well be you, so you are doing your future self a favor here.

Cross-Team Impact

Answer questions similar but not limited to:

- Does this impact other parts of the codebase (modules)?
- Does it cause any latency regression to the system?
- Does it expose any security vulnerabilities?
- What are some negative consequences and side effects?

Open Questions

Any open issues that you aren't sure about, contentious decisions that you'd like readers to weigh in on, suggested future work, and so on. A tongue-in-cheek name for this section is the "known unknowns".

Detailed Scoping and Timeline

This section is mostly going to be read only by the those working on this project, the tech lead, and project director. Hence this section is at the end of the doc.

Essentially, this is the breakdown of how and when you plan on executing each part of the project. There's a lot that goes into scoping accurately, so you can read [this post](#) to learn more about scoping.

This section of the design doc can be treated as an ongoing project task tracker, so it should be updated whenever scoping estimate changes.