Technical Specification

Instructions

Use the technical specification guide below to assist your team in completing a technical specification documentation of the work your team completed throughout the project. Please consider any extra criteria that may be necessary to include that are project-specific.

Use Case: The collective documentation that your team collects will be turned in. For on-going projects the documentation will be shared with future years, so make sure to include details that are pertinent to each step. This technical specification documentation should be continuously updated every sprint.

Contribution: Students will be graded on their contribution to the Documentation Guide on a bi-weekly basis. See the student course schedule for due dates.

Reminder: Documentation should ONLY be shared with your team. It should not be shared with anyone outside of your Corporate Partner Team.

Table of Contents

Technical Specification

Title Page

The title page should include:

- Title
- Author(s)
- Team
- Reviewers(s)
- Created on
- Last Updated

Technical Specification

Introduction

Overview of the project and what problem the project is trying to solve, like a summary or an abstract

A. **Glossary/Terminology**
This includes new terms that have come up during the research or design of the project. i.e., Anvil – supercomputing platform utilized by the project team GPU – graphic processing unit.

B. **Context/Background**
   a. Reasons why the problem is worth solving
   b. Origin of the problem
   c. How the problem affects users and company goals
   d. Past efforts made to solve the solution and why they were not effective
   e. How the product relates to team goals, OKRs
   f. How the solution fits into the overall product roadmap and strategy
   g. How the solution fits into the technical strategy

C. **Product and Technical Requirements**
This is all about the MVP – minimum viable product. Here is where the team will outline what is required for the project to be successful.

D. **Non-goals or Out of Scope**
As important as defining project scope is, so is defining what is out of scope. Creating a project that keeps growing in size and requirements is easy, often called scope creep. However, this can also cause the project never to reach completion. It's crucial to outline what will not be accomplished during a project and what will be left for future discovery.

E. **Future Goals**
What could be accomplished if this project succeeds, and where could the research lead?

F. **Assumptions**
Conditions and resources must be present and accessible for the solution to work as described.

G. **Solutions**
   a. Current Solution – without doing this project, this is where the research or system stands currently without any work being done. i.e., if this project is not completed, here is where the company stands presently.

Technical Specification

b. Pros & Cons - What are the pros and cons of the current existing solution?

c. Proposed Project – Explain the project and how it will solve or meet the current needs of the business. How will the project impact the company or industry to move forward?

   This section will also include the details below:

   I. **Data Model / Schema Changes** – What data is used?

   II. **Business Logic** – What business processes will this impact?

   III. **Presentation Layer** – Are there any front-facing interfaces (i.e., webpages) that this project will create

   IV. **Other questions to answer**

   - How will the solution scale?
   - What are the limitations of the solution?
   - How will it recover in the event of a failure?
   - How will it cope with future requirements?

d. Test Plan

   i. Explanation of how the project plans to test the success of the end product

   ii. QA

e. Alternate Solutions / Designs

   i. Summary statement for each alternative solution

   ii. Pros and cons for each alternative

   iii. Reasons why each solution couldn't work

   iv. Ways in which alternatives were inferior to the proposed solution

      1. Migration plans to next best alternative in case the proposed solution falls through

Further Considerations

a. **Impact on other teams** - How will this increase the work of other people?

b. **Third-party services and platforms considerations**
   - Is it worth it compared to building the service in-house?
   - What are some of the security and privacy concerns associated with the services/platforms?
   - How much will it cost?
   - How will it scale?
   - What possible future issues are anticipated?

c. **Cost analysis**
   - What is the cost to run the solution per day?
   - What does it cost to roll it out?

d. **Security considerations**
   - What are the potential threats?
   - How will they be mitigated?
   - How will the solution affect the security of other components, services, and systems?

e. **Privacy considerations**
   - Does the solution follow local laws and legal policies on data privacy?
   - How does the solution protect users' data privacy?
   - What are some of the tradeoffs between personalization and privacy in the solution?

f. **Accessibility considerations**
   - How accessible is the solution?
   - What tools will you use to evaluate its accessibility?

g. **Operational considerations**
   - Does this solution cause adverse aftereffects?
   - How will data be recovered in case of failure?
   - How will the solution recover in case of failure?
   - How will operational costs be kept low while delivering increased user value?

h. **Risks**
   - What risks are being undertaken with this solution?
   - Are there risks that, once taken, can't be walked back?
   - What is the cost-benefit analysis of taking these risks?

i. **Support considerations**
   - How will the support team get across information to users about common issues they may face while interacting with the changes?
   - How will we ensure the users are satisfied with the solution and can interact with it with minimal support?

Technical Specification

- Who is responsible for the maintenance of the solution?

Success Evaluation

a. **Impact**
   - Security Impact
   - Performance impact
   - Cost impact
   - Impact on other components and services

b. Metrics
   - List of metrics to capture
   - Tools to capture and measure metrics

Work

a. **Work estimates and timelines**
   - List of specific, measurable, and time-bound tasks

  - Resources needed to finish each task
  - Time estimates for how long each task needs to be completed

b. **Prioritization** - Categorization of tasks by urgency and impact

c. **Milestones** - Dated checkpoints when significant chunks of work will have been completed

d. **Future work** - List of tasks that will be completed in the future

Technical Specification

Deliberation

A. **Discussion** - Elements of the solution that team members disagree on need to be debated further to reach a consensus
B. **Open Questions** - Questions about things you need help knowing the answers to or are unsure that you pose to the team and stakeholders for their input. These may include aspects of the problem you don't understand how to resolve yet.

End Matter

a. **Related Work** - Any work external to the proposed solution that is similar and worked on by different teams. It's important to know this to enable knowledge sharing between such teams when faced with related problems.
b. **References** - Links to documents and resources you used when creating your design and wishing to credit.
c. **Acknowledgments** - Credit people who have contributed to the design that you wish to recognize.

Return to Page 1