DP Evaluation — Process for Teams
6.033 Spring 2020

This document builds on the evaluation overview; be sure to read that first. You can use this document with the rest of your DP team to develop your evaluation. As you do this, if there are places where you feel stuck or have a question, you should reach out to your TA.

Before you Start

- **Identify the limitations imposed by the DP spec.** For instance, network capacities, storage capacities, etc.

- **Identify the requirements of the project.** For instance, how quickly do certain types of data have to be delivered, how reliably do certain types of data have to be delivered, etc.

  Your evaluation should also address the use-cases presented in the design project write-up. As you write your final report, you may pull those out into their own subsection, or mention them at different points within your evaluation. Whatever structure works best for your report is fine.

- **Identify your design choices, especially places where you made trade-offs.** Hopefully you did this as part of your DPPR.

Meeting the Requirements

Once you've identified the requirements of the projects, identify the evaluation components needed to convince us that you've met the requirements.

- In many cases, you will find that you can do a direct calculation. For instance, calculating how long it takes a software update to get to every device, and stating that that it means the requirement. Remember, you control many aspects of this design: how bundles are stored (where, for how long, etc.), whether additional network communication is used (and if so, how much), etc.

  If you find that you cannot make a direct calculation, consider the following:

  - Can you make a reasonable estimate of a quantity that would help you make a direct calculation? You can use your own experiences, other course content, etc., to estimate things such as likely failure rates, speeds of different components, etc.
  - Can you make a calculation that describes the worst case?

- Pay special attention to values that change depending on the circumstances. You may need to make a calculation for multiple scenarios, or under the worst case, to illustrate that you've met the requirements.

Justifying your Design

In many cases where you made a design decision, you can think about it in comparison to some alternative design:

- What was the alternative design? Sometimes the alternative design is just your design, but without one component. For example, if you have developed a very fancy data structure for storing bundles, an alternative design would be your system with a more basic data structure.

- What is better about your design compared to the alternative? Often, this is something you can calculate directly: for instance, your design has an overhead of X KB/sec, and the alternative has Y KB/sec, where Y > X. In some cases, though, we make design decisions in favor of difficult-to-calculate principles; for instance, simplicity.

- What is *worse* about your design compared to the alternative? For example, if you have less network overhead, do you also have less reliability?

- Explain why the good outweighed the bad in this decision.