

CS 621

Project Plan

Due: **Tuesday, April 20, 2023, 9:00 AM EDT** via [Gradescope](#). Late assignments will not be accepted without **prior** permission.

Overview

This is a group assignment. Each project group is to submit one project plan. The group will also present for **5 minutes** in class on **Tuesday, April 25, 2023**.

Every member of the group is expected to put in equal effort. The entire group will share the same grade; however, if members identify others who have not contributed (or have contributed above and beyond), the grades can be varied accordingly.

The goal of a project plan is to lay out exactly what you will do for your class project. It is not a proposal. It is not a general idea of a problem you will work on. **It is a write up of exactly the problem you will solve, how you will solve it, how you will evaluate your solutions, and what you expect to find.** The perfect project plan is essentially your final report, except instead of “we did such and such” it is allowed to say “we will do such and such” and instead of “the experiment showed that” it is allowed to say “the experiment will show either this or that.” The final report will necessarily change some, as you carry out the work and discover issues and interesting problems you did not anticipate, but the smaller these changes are, the better.

The project plan has to include the following:

1. The research question(s) you will answer. In other words, what problem(s) are you solving? This should also include motivation for why this question is worth answering.
2. How your solution will solve the problem(s). This should convince the reader that the solution will actually have a good chance at being successful.
3. How will you evaluate your solution. In other words, how will you prove, demonstrate, or argue that your solution solved the problem?

While discussing your plan, you may find it useful to refer to related work. There is no need to repeat what you said in your literature review. If you choose, you may refer to that document directly. For example “As outlined in our literature review, automated testing solutions suffer greatly from the lack of large quantum computers.”

Deliverables

Each group should upload 2 files via [Gradescope](#):

- Submit a .pdf of a single write up, per group. There is no right length minimum or maximum: you should write enough to describe your problem, your solution, why your solution will succeed, and how you will evaluate your solution to know that it succeeded. Just as with the literature review, three pages is probably not enough to write a convincing argument, whereas more than six pages probably means you are not being direct enough in your discussion.
- Submit a powerpoint, keynote, .pdf, or any other reasonable format of the presentation slides you will use in your class presentation. **Focus your presentation on the problem you are solving, and the data or experiments you will use and run.**

How to get started

To get started, write down your research question and have the entire group discuss it until you agree that this is what you will solve. A research question is not a method! “We will combine automated test generation and mutation testing” is not a research question. It is not even a question. An example of a research question is “Can mutations used in mutation testing generate the kinds of bugs observed in real, open-source development?” Nailing down your research question first is critical to organizing the rest of your plan.

Once you settle what the research question is, write a motivation section around it. It is important to motivate why your research question has value. Good ways to do this is to describe a task developers have to do today, and why it is hard. Then explain how answering your research question would simplify performing this task. For example, today developers have to manually write tests for their software. If we knew that mutation testing mutations are representative of real bugs, the developers could reduce the amount of manual testing they’d have to do because they wouldn’t have to worry about writing tests for those types of bugs. Writing a before-and-after scenario can be helpful: describe the use case of what developers do today to accomplish something, and then what the developers will have to do after you are done with your work to accomplish the same task.

Next, focus on how you will know if you succeed. This is the evaluation part of the plan, which will likely come last in your write up. Figure out what you’ll need to demonstrate in order to know that you succeeded. For example, you might succeed if you show that at least 40% of the development versions of some open-source code that fail a test can be produced by taking a version of the same code that passes the test and applying mutations to it. Think hard about your evaluation. What happens if something goes wrong? What if you can’t find enough open-source programs? Come up with a back up plan. Consider that the research question might be answered in the negative and explain how you’ll know if that’s the case.

Finally, describe your solution. This part of the plan (typically the middle) will vary the most among the groups. Some will develop a tool. Some a new technique (that they may implement). Some only an evaluation framework. Describe your solution in enough detail to convince the reader that the solution will actually have a good chance of being successful.

Sections and subsections can help you organize your report and get your points across most effectively.