Term Project  
(Feb. 13, 2020)

Important Dates

- Topic selection due: Feb. 20, 2020  
- Paper due: Mar. 12, 2020

Project Description

The term project is a written paper. This document is a preliminary description of possible topics, to help you in thinking about your paper as the course proceeds.

The first thing that must be said is that the paper is to be your original work, and you must be very careful about acknowledging your sources. The Internet will be an important resource, but it is all too easy to abuse it. Whenever you copy exact wording or graphics, or take an idea, even to modify it, from any external source, you must cite that source, and use quotation marks to clearly indicate what you have not written by yourself. I suggest that as you collect material, you maintain files that include the detailed citation along with the material itself, so that when you come to use it, the source cannot be forgotten. Material that is not your own is expected to be a minor part of your paper. If you have any doubts about the appropriate way to reference material, or any question about the (serious) penalties for plagiarism, please ask your instructor.

The paper is to be printed single-spaced in a standard page format, and ten pages is the rough size, but no rigid limits are set. Both hard copy and email copy are required. Please email your report to the instructor before coming to the class on the due day and turn in the hard copy in class. The deadline is firm. Please plan accordingly.

The paper will be graded on its content and organization, but not on its English (please use a spell-checker, and do the best job you can, but so long as I can understand what you say, grammar will not be graded). The primary criteria for grading are:

1. Does the paper make clear assertions of your opinions, rather than just reiterating the statements of others?
2. Does it give convincing arguments in support of your opinions?
3. Does it display good understanding of software engineering, as presented in CS 4/554, in texts and in class discussion?

A list of standard topics are described below. You MUST select a topic by Feb. 20, 2020.

1. Comparison of open-source development and traditional software development.  
2. DevOps.  
3. Software development involving the application of a formal method.

Report your selection to the instructor by e-mail on Feb. 20; no approval is required.
The first topic asks you to compare two methodologies for software development. If you choose this topic, your paper should be organized to give a brief description of the two methodologies, and then to compare and contrast them. Present their strengths and weaknesses, and describe the situations in which those strengths and weaknesses are most important. Be critical of both methodologies. In particular, you are requested to point out when the strengths of one methodology clearly address the weakness of the other methodology.

The second topic asks you to research on an emerging methodology for software development. If you choose this topic, your paper should be organized to give a brief description of this methodology, present its strengths and weaknesses, and describe the situations in which those strengths and weaknesses are most important. In particular, you are requested to point out when this methodology should be adopted.

The last topic is more demanding. The textbook gives an introduction to formal methods, but for the paper you must select a particular method, explain it, and discuss its particular advantages and disadvantages. Most of the effort for this topic will go into understanding that methods. If you are interested in this topic, below is a list of possible methods:

1. Symbolic execution
2. Model checking
3. Theorem proving

Choose a particular method, find a real-world project that uses the method, and discuss the method in the context of the project. If you are interested in symbolic execution, I recommend the KLEE project at Stanford University, which has been widely used by many people in both industry and academia. If you are interested in model checking, I recommend the SLAM/SDV project at Microsoft Research, which is a very high-impact project of practical model checking. If you are interested in theorem proving, I recommend the ACL2 project, which has many successful industrial applications.