Understanding, Grading, and Judging Computer Science Projects

Updated 7-11-09

The grading and judging of computer science projects can often be a confusing undertaking—especially for those without a background in computer science. This article will help you understand computer science projects and provides a grading/judging rubric for your use.

Contrary to a seemingly popular belief, not all science fair projects follow the scientific method! Three types of projects are presented at science and engineering fairs, each of which follows their own process/method: science projects, which follow the scientific method; engineering projects, which follow the engineering design process; and math projects, which follow the mathematical reasoning/proof process. Each of these types of projects follows a different process, because each of these types of projects has a different end goal.

The "classical" science project, which follows the scientific method, is the best-understood type of project. Nearly everyone who has ever been in a science class has had the scientific method taught to them: Make observations, ask a question, do research, develop a testable hypothesis, test that hypothesis by doing experiments, analyze data, and then draw conclusions based on the collected data.

The two other methods, the engineering design process and the mathematical reasoning/proof process, are different from the scientific method, but engineering projects and math projects are still valid science fair projects, even though these projects often do not involve complete experiments.

The engineering design process, like the scientific method, consists of a series of steps. The following table summarizes the steps of the scientific method and the engineering design process. The mathematical reasoning/proof process is beyond the scope of this article.

The Scientific Method	**The Engineering Process**
1. State your question	1. Define a need
2. Do background research	2. Do background research
3. Formulate your hypothesis, identify variables	3. Establish design criteria
4. Design experiment, establish procedure	4. Prepare preliminary designs
5. Test your hypothesis by doing an experiment	5. Build and test a prototype
6. Analyze your results & draw conclusions	6. Test & redesign as necessary
7. Present results	7. Present results

So What Are Computer Science Projects? Computer science projects are engineering projects and therefore, follow the engineering design process. The difference between computer science projects, in which the product is typically a program or improved functionality, and stereotypical engineering projects, in which the product is usually a device of some sort, is analogous to the difference between an astrophysics project and a behavioral science project. Though the details of how the scientific method is applied during a behavioral science study is different from the exact way in which the scientific method is applied during an astrophysics experiment, the scientific method is in fact applied during both projects. Similarly, the way in which the engineering design process is applied during computer science and "typical" engineering projects is different, but the process is used for both types of projects.

How Does the Engineering Design Process Apply to Computer Science Projects? Let's take a step-by-step look at how the engineering design process works in the specific case of a computer science project. The following discussion assumes familiarity with the engineering design process. If you are not already familiar with the engineering design process, look at the following resources, which explain the engineering design process in detail:

http://sciencebuddies.org/science-fair-projects/project_engineering.shtml
http://synopsys.championship.googlepages.com/winningengineeringprojects
http://synopsys.championship.googlepages.com/engineeringdesignworkshop2

Step 1: State a Design Goal. The first step of the engineering design process is to define a need, which is based on what users desire. This need is then stated as a design goal. For example, the need may be to find a faster way for computers to scan files for virus and spyware infections. The design goal statement for this project might be “The goal of this project is to write, test, and optimize software that decreases the time needed for a computer to scan files for virus and spyware infections.”

Step 2: Background Research. The second step of the engineering design process is to do background research. During this part of the project, the student should seek out information related to his or her general area of study (in the case of our example, this would be antivirus/antispyware software). In addition to becoming intimately familiar with his or her area of study, the student doing a computer science project should seek to understand the needs of their target, research the software that may or may not currently exist to address the student’s stated design goal, and do research that will allow the student to state specific design criteria.

Understanding the needs of the target user will help the student define design criteria (the next step of the engineering design process). For example, a graphical user interface could be important to faster antivirus/spyware scanning if the target user for that product is a typical PC user. Becoming familiar with both the needs of the end user of the product/program and the capabilities of existing products/programs will help the student write meaningful and measureable design criteria.

Step 3: Establish Design Criteria. The third step of the engineering design process is to establish design criteria. Design criteria are requirements that the student specifies that will help the student develop his or her software and determine the extent to which the final product/program meets the stated design goal. For our example, some of the design criteria might be (1) scans 1,000 files in 0.5 seconds with 99.5% detection accuracy, (2) performs a complete system scan in less time than Brand X antivirus/antispyware software, and (3) consumes less than 15 MB of memory while running.

Step 4: Preliminary Designs. The fourth step of the engineering design process is to draw up preliminary designs. In the case of a computer science project, this step usually involves writing the first iteration of the program’s code. Perhaps the student might write two or three (or more) completely different programs that go about reaching the design goal in different ways.

Step 5: Build and Test. The fifth step of the engineering design process is to build and test. For a computer science project, this step really only involves testing because the “building” (i.e. writing the first iteration of code) is done as part of the fourth step of the engineering design process. At this point in the engineering design process, the student does an initial test of his or her code. During this test, the student should note any bugs in the program, slow parts in the code, the best parts of the code, etc. A “test plan” is a key part of the testing process. Using our example, the test plan might include trying a scan of 1,000 files on a specific computer. During the scan, the students might note the duration of the scan, the speed of the scan, how much memory and processor speed the program uses, and where the program slows down/stops.

Step 6: Redesign and Retest. The sixth step of the engineering design process—redesign and retest—is always the longest step of the engineering design process, and for computer science projects, this may be especially true. During this phase of the project, the student works on debugging, rewriting, and optimizing the code. In order to do so, the student should conduct several different tests of the code (remembering to use the test plan developed in during step five of the engineering design process) and use failure analysis, the stated design criteria, and the stated design goal to guide revision of the code.

This step of the engineering design process involves iteration—repeatedly testing and revising the code until you finally reach your stated design criteria and design goal. Keeping an accurate and detailed record of this part of the project is essential to a high-caliber computer science project. Keeping a detailed and neat project notebook is an essential part of any science fair project, and computer science projects are no exception. A detailed project notebook should be kept at all times, but it is critically important that an especially good record be kept of the redesign and retest portion of a computer science project.

In the case of our example, the redesign and retest process could lead to the elimination of all but one of the various initial programming approaches, as repeated testing might show one approach superior to others. Detection and elimination of bugs would be an integral part of the redesign and retest process, as would optimization of code. Rewriting and revision of code is the crux of the project, and the focus of this step of the engineering design process. You must iterate—redesign and retest, debug, optimize, etc.—until your stated design criteria have been reached and your design goal has been clearly fulfilled. (Unless, of course, your student runs out of time because the science fair is next week. In that case, it would be best to stop and move on to step seven without achieving all the design criteria).

Step 7: Present Your Work. The seventh and final step of the engineering design process is to present your work. Your presentation should highlight the final product, its usefulness, and merit *and* outline the process that you went through to reach that final product. All too often students doing engineering and computer science projects only address one of these purposes. They either present only the product without the process, in which case any judge who starts to poke deeper into how the student actually obtained that product may begin to have doubts about the actual merit of the final product and so not give an award to the project, or the student presents only the process, in which case the final product, which may be quite outstanding, isn't appreciated and so that project, too, is not selected as a winner by judges.

A winning computer science project must highlight both the product and the process. The product/program that truly addresses a meaningful need will attract judges' attention and the ability to show a thorough and thoughtful design process will provide convincing evidence of the quality of your science and your engineering abilities, which will separate your project from the other projects that have neat, useful products/programs. The key to a winning computer science/engineering project is to have both product and process.

So How Do I Grade These Projects? With an understanding of the above concepts, it will hopefully be significantly easier for teachers (and judges) to evaluate computer science projects done by students. Below is a grading rubric that you are welcome to use. The grading rubric aims to be very detailed to help those who are unfamiliar with computer science projects be able to come to know what it takes to have a quality computer science project. It is hoped, however, that even seasoned teachers and judges familiar with computer science projects will still find the rubric helpful. If you have suggestions on how to improve the rubric, please email Terik Daly at terik.daly@gmail.com. Updated versions of this rubric (and the article as a whole) will be periodically posted.

Bibliography: In addition to the author's thoughts and ideas, the following articles and websites were consulted as sources in writing this article:

Kawanami, Bruce. "Engineering Design Workshop". PowerPoint presentation available online at http://synopsys.championship.googlepages.com/engineeringdesignworkshop2.

SCVSEFA. "Winning Engineering Projects: Step by Step". http://synopsys.championship.googlepages.com/winningengineeringprojects

Science Buddies. Engineering & Programming Project Tips. http://sciencebuddies.org/science-fair-projects/project_engineering.shtml.

Rubric for Grading Computer Science Projects

Updated 7-11-09

Computer Science projects follow the engineering design process. Just as with a typical science project, following the process is an essential part of the student's project. A high-quality computer science project not only provides an innovative and useful product, it also shows evidence of a logical, structured design process.

Process (42 points)

Did the student state a specific design goal or goals? Does that statement identify the product/program to be developed? Does the design goal identify the target user? The need the product/program will satisfy?

1 2 3 4 5 6

To what extent did the student conduct background research? Did this background research address all important facets of the project (science concepts, mathematical formulas, existing products/programs, etc.)?

1 2 3 4 5 6

To what extent did the student identify meaningful design criteria? Did the student keep the target user in mind when identifying these criteria? Did these design criteria guide the student in building/programming, testing, and revising the product/program?

1 2 3 4 5 6

Did the student evaluate multiple approaches to solving the problem/filling the need? Can the student justify the chosen approach? Did the student prepare preliminary code or initial design schematics?

1 2 3 4 5 6

To what extent did the student develop a test plan for evaluating each iteration of the program/product? Did the student follow this plan when testing the initial program/product design and subsequent designs?

1 2 3 4 5 6

Did the student use information from testing the program/product to improve the product/program? To what extent did the student continue to redesign and retest the product/program until the design goal and design criteria were reached (e.g. through debugging, optimizing, etc.)

1 2 3 4 5 6

Does the student's project notebook, display, and oral presentation provide ample evidence of how the student used the engineering design process throughout the project? Is the project more than just gadgeteering?

1 2 3 4 5 6

Total /42

Comments:

Product (48 points)

In *your* experience, to what extent does the student's product/program represent significant improvements over existing products/programs?

1 2 3 4 5 6 7 8 9 10 11 12

Does the *student* understand the extent to which the product/program represents significant improvements over existing products/programs?

1 2 3 4 5 6 7 8 9 10 11 12

To what extent is the final product useful to the target user? Does the project fill a meaningful need?

1 2 3 4 5 6 7 8 9 10 11 12

Is the program code or product design clear enough that others would be able to replicate the student's work?

1 2 3 4 5 6 7 8 9 10 11 12

Total /48

Comments:

Presentation/Interview* (10 points)

To what extent does the student's presentation/interview communicate both the merits of the final product/program and the process that the student went through to reach that final product/program?

1 2 3 4 5

To what extent can the student communicate effectively about the project? Can the student provide cogent responses to questions? Can the student defend the design choices that s/he made?

1 2 3 4 5

Total /10

Comments:

TOTAL /100

*If the grading is done based solely upon the student's display board, then the points in this section may be based on the display board alone.