Jason McGarr
July 30, 2006
Extra Credit Assignment

Article 1:
http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Heuristic evaluations are the simplest and possibly the cheapest out of the available options for usability testing (a). According to the article, a case study by the Nielson Norman Group put the cost at about $10,500 with expected benefits around $500,000. For this reason it may be the one most of us will use when we create software applications in a business setting. In most of our class projects, usability is barely even a concern. Our focus is on completing the assigned requirements. If the program is full of bugs, hacked together, or barely usable it's not a concern so long as it does not affect our grade. But, for professional products usability equals success and profits and this article gives a decent summary of how to conduct an evaluation. The article suggests that more than one evaluator is key to finding the majority of your usability flaws in a heuristic evaluation. Different evaluators tend to find different problems and so combining several evaluators' results will yield a larger number of usability bugs that are found. According to the research 3 to 5 evaluators will find around 65% to 75% of the total usability problems; however more evaluators can be used where usability is more critical. An added benefit of using the heuristic evaluation is that it can be used to test systems in their early stages, "on paper only", to evaluate the potential usability without wasting time creating an unusable system. For conducting your own heuristic evaluation, generic usability principles listed on a link from the article could be used, but more specific usability heuristics can be determined by evaluating current implementations in the given field (perhaps medical equipment user interfaces) and noting what works and what can be improved. Other points in the article include useful advice like using an observer to summarize individual evaluators' reports rather than have each evaluator submit a report. Usability testing is an important part of developing a new system or product. Better usability can increase the commercial success of your product and usability tests can prevent cases of bad HCI that contributed to the horrible accidents of Therac-25. We will all be professionals one day so understanding the benefits and procedures of usability testing in more detail is useful for all CS students.

a. See Table: http://www.useit.com/papers/guerrilla_hci.html
Estimated cost reduction of a Heuristic evaluation versus the materials needed for a more in depth usability test.

Article 2:
http://www.cio.com/archive/021501/et_pundits.html (Linked on Jakob Nielson site)

The article contains additional reasons why usability testing is very useful practice for programmers who want to their products to have a competitive edge against other software. Design flaws and bugs in programs may not seem too bad. A solution or workaround may be easy to find and if it only costs a user an additional 10 seconds, why do we need to worry? Now imagine that the problem happens "25 times each day for 50 representatives, they'll waste more than 900 hours a year on this single usability issue alone". Its amazing how something so small can cause such a problem in the long-run. The article goes on to say that one study has shown that most programs contain at least 40 design flaws. At a modest 10 seconds each, that sums up to be about 36, 250 hours a year not counting the time costs of breaking a user's concentration. Additionally, if an application is easily usable it cuts costs of training professionals to provide support for your otherwise complicated system. Maintenance costs can also be reduced since little needs to be added to a system that properly addresses the user needs. On the time scale that I use a program which I have coded, "small" bugs do not seem particularly big issues, but when considering how often your program will be used small bugs mean large losses for a company. This article will help remind, particularly those in CS, that this mentality is certainly flawed.

Article 3:

http://www.useit.com/alertbox/20030908.html

The article addresses some misconceptions about usability that keeps many developers from using these systems to direct their design. As pointed out in the first article, benefits certainly exceed the cost, but usability can still seem expensive for professionals in a small company. The article gives practical advice through a link in the article on how a simple usability evaluation can be done on a budget of about $200 dollars. Secondly, usability evaluations are mistakenly assumed to slow down your schedule when, in fact, they may save you time. Usability can be done early in development on paper to discuss user requirements and avoid pointless meetings and design time to figure out what your customer truly wants out of your product. Creativity can be increased rather than degraded since more time can be focused on improving the design. Aesthetics are an important part of functionality (take Donald Normans ATM test from his book Emotion Design: http://www.jnd.org/dn.mss/CH01.pdf). Also, other aspects of usability such as use of standards and conventions allow people to better understand your user interface on first glance, but they do not restrict on your creativeness (consider research paper conventions: they help a person evaluate the paper using a common format, but does not affect the content). User errors and frustration can be minimized as well, by following usability standards such as minimizing the amount of information a person must remember while using an interface. The article gives further examples with usability in web development. Conciseness and simple interfaces are more successful in attracting a casual web surfers long term attention rather than fancy and usually cumbersome flash or pictures (http://blogs.zdnet.com/BTL/?p=2776). Finally, customer feedback is not a good replacement for proper usability design testing. To demo a product with cool 3-D user interface may wow a customer, but using this as a measure for how to design the final interface is a bad idea. Using an application such as this would prove to be very tedious and frustrating in reality when the customer initially agreed it would be a good idea. This article is just a further point to drive home the impact, usefulness, and importance that usability plays in the computer science field. It also helps CS2340 students understand why we need to study this.