Usability and the Bottom Line

George M. Donahue, *Sapient*

There's little debate that usability engineering benefits end users, but its benefit for companies and the people who work for them is less widely known. The author discusses these broader usability benefits and also how to use a cost-benefit analysis to demonstrate the value of usability to your company's bottom line.

Usability engineering benefits end users. Few people disagree with that idea. However, usability's beneficiaries also include system developers and the companies they work for. Improving usability—whether of IT systems, e-commerce Web sites, or shrink-wrapped software—is not only highly cost-effective, but it can also reduce development, support, training, documentation, and maintenance costs. Additionally, it can shorten development time and improve a product's marketability.

Here, I will show how you can use a cost-benefit analysis to sell usability engineering based on the bottom line. I'll then discuss the broader benefits a company can realize from usability engineering.

Cost-benefit analysis

Although usability's broad benefits are impressive, a cost-benefit analysis might be a necessary first step in introducing usability into your organization or a particular project. In usability cost-benefit analyses, the goal is to estimate the costs and benefits of specific usability activities—such as prototyping, usability testing, heuristic evaluation, and so on—and contrast them with the likely costs of not conducting the activities.

The analysis has four steps:

- selecting a usability technique,
- determining the appropriate unit of measurement,
- making a reasonable assumption about the benefit's magnitude, and
- translating the anticipated benefit into a monetary figure.

In a usability cost-benefit analysis, it's also important to focus on the techniques and benefits likely to yield the most value for, and seem most persuasive to, the people you're trying to persuade. As Deborah Mayhew and Marilyn Mantei put it, you should "decide the relevant audience for the analysis and then what the relevant categories of benefits are for that audience, because not all potential benefits are relevant to all audiences."[1] For example, a commercial software company might be more interested in a cost-benefit analysis that focuses on usability's potential for reducing development costs and increasing customer satisfaction

0740-7459/00/$10.00 © 2001 IEEE

In usability cost–benefit analyses, the goal is to estimate the costs and benefits of specific usability activities and contrast them with the likely costs of not conducting the activities.

than in an analysis that focuses on its potential to improve end-user productivity.

To illustrate, I offer the following scenario, based on a method developed by Mayhew and Mantei. While the scenario's anticipated benefit is a productivity improvement on an internally developed IT system, you can use the same methodology to perform usability cost-benefit analyses for other benefits and in different organizations, including e-commerce and commercial software companies.

For this scenario, assume that you work at Pretty Good Systems. You are aware that the human resources department has lodged complaints against PGS's internally developed human resource system, Getting In Good (GIG). You're also aware that development on GIG Version 2 is about to begin. You suspect that improving GIG's usability would not only make GIG users' lives easier, it could save the company money—and PGS management is very interested in reducing costs. Before you broach the subject of improving GIG's usability, you wisely decide to do some preliminary usability work followed by a cost-benefit analysis.

Preliminary work

To start, you interview the HR director and several GIG users. They all say that GIG is too complicated. They can't understand why there's one screen for entering new applicant data, another for entering data about applicants who have been interviewed, another if an applicant is hired, and so on. "There's simply not that much applicant data," a GIG user tells you. "I don't see why we need so many different screens. It takes so long to go through them all." Next you spend an afternoon observing HR staff using the system. You then sketch out some low-fidelity, paper prototypes of how a single GIG data-entry screen might look and try out the prototypes on a few HR staffers in an informal usability test. Your preliminary usability work supports your hypothesis: GIG's user interface is inefficient.

Usability-aware person that you are, you also interview other significant GIG stakeholders—namely, the GIG development manager and some GIG developers. You then discuss the option of entering applicant data on a single screen, as opposed to several. The developers tell you that, from a technical standpoint, it's easier to "modularize" applicant information according to the applicant's status in the hiring process. Nonetheless, you ask if it's possible to use a one-screen-per-applicant approach in GIG's next version. The development manager says that they could do it, but that it would take at least 30 more person-hours.

Estimating costs

You're now ready to do the cost-benefit analysis. First, you estimate how much it costs to process a job application in GIG's current version. Based on your interview with the HR manager, you know that it takes an average of four hours per applicant and that the average loaded salary of a GIG data-entry person is $25 an hour. You multiply $25 by four to get the cost of processing a single application: $100.

On average, PGS receives about 1,000 job applications a year. Multiplying $100 by 1,000 gives you the current average annual cost of processing job applications at PGS: $100,000.

To be on the safe side, you assume that making the programming change to GIG will take 40 hours, rather than the 30 hours estimated by the development manager. You then multiply 40 hours by the loaded average salary of a developer at PGS ($60 an hour) to determine the cost of making the change: $2,400.

Estimating benefits

Your preliminary usability work suggests that adopting a one-screen-per-applicant approach would cut application-processing time in half. This is your unit of measurement for usability. However, to be cautious, you assume a 25-percent reduction in processing time. Given this, on average, a person could process an application in three hours rather than four. Given that the average loaded salary of a data entry person is $25 per hour, you estimate a cost of $75 to process a single job application in the new system.

To estimate the overall savings in the average annual cost of processing job applications at PGS, you multiply $75 by 1,000. The result is $75,000—$25,000 a year less than the current average processing costs. You then factor in the cost of making the changes ($2,400) and subtract it from the anticipated first-year savings, giving you a first-year benefit of $22,600.

However, you're not quite finished. The typical lifespan of a PGS system is three years. Because the cost of making the changes will be incurred once, you need only deduct that cost for one year, whereas the benefit will be realized each year that the new version is used. Thus, you add the first-year benefit ($22,600) to the benefit for the second and third years ($50,000), and you get a total lifetime usability benefit of $72,600.

The overall result is a cost-benefit ratio of 1:301/4. That is, $72,600 ÷ $2,400 = $30.25.

In this example, the benefits to the HR department are different than the benefits to PGS as a whole. Because HR's data-entry process will be streamlined, GIG2's users should be able to get more work done and their main benefits are higher productivity and better job satisfaction. For PGS's development managers (and presumably also its executives and shareholders), the main benefit is decreased costs. Although managers, executives, and shareholders may be happy to hear that usability engineering will improve job satisfaction, your cost-benefit analysis should focus on the cost savings, because that's the most relevant benefit to development managers—and their buy-in is crucial.

Broader usability benefits

According to the above analysis, every dollar spent on usability offers a return of $30.25. That's a nice return-on-investment. Nonetheless, considering usability solely from an ROI perspective does not give it its full due.

Usability costs are typically seen as additional; that is, if development doesn't include any formal usability activities—such as usability testing or prototyping—there won't be any usability costs. That assumption is wrong. Every software product has a user interface, whether it's usability-engineered or not. The interface takes time to build, regardless of whether it's consciously engineered for usability. Time is money. In other words, user-interface costs are inevitable and intrinsic to development. Many systems also have support, documentation, maintenance, and other costs. These are also usability expenditures, and regardless of how such costs appear on the company's books, usability engineering can help manage them.

Although the ROI argument is compelling, your usability case can be bolstered by pointing out that the company always spends money on usability, even though it may not see it this way. In his article, Arnold M. Lund cost-justifies the existence of a permanent usability group within an organization over carrying out discrete usability activities.[2] I'll now examine several broader usability benefits in more detail.

Reduced development and maintenance costs

Software development projects typically overrun their budgets and schedules. Such overruns are often caused by overlooked tasks and similar problems, which techniques such as user analysis and task analysis can address.

When you focus on real user needs and understand the people you're designing for, the result is often fewer design arguments and fewer iterations. Usability techniques, such those described in the sidebar, "Basic Usability Engineering," are also highly effective in helping you detect usability problems early in the development cycle, when they're easiest and least costly to fix. By correcting usability problems in a project's design phase, for example, American Airlines reduced the cost of those fixes by 60 to 90 percent.[1] One frequently referenced study found that correcting a problem once a system is in development costs 10 times as much as fixing the same problem in the design stage. Once a system is released, the cost to fix a problem increases to 100 times that of a design-stage fix. This study also found that 80 percent of software life-cycle costs occur during the maintenance phase; many maintenance costs are associated with user requirements and other problems that usability engineering can prevent.[3]

Whether your company does usability testing or not, your customers will, in effect, usability-test the system. Ultimately, of course, relying on such "usability testing by default" risks angering customers, and, as the studies above show, post-release problems cost much more to fix. A printer manufacturer, for example, released a printer driver that many users had difficulty installing. More than 50,000 users called the support desk for assistance, costing the company nearly $500,000 a month. To correct the situation, the manufacturer sent out letters of apology and patch diskettes (at a cost

Whether your company does usability testing or not, your customers will, in effect, usability-test the system. Ultimately, relying on such "usability testing by default" risks angering customers.

Basic Usability Engineering

Applying usability techniques—even if only in an informal, "guerilla" manner—can offer many usability benefits. Here, I outline a few basic techniques that are fundamental to usability engineering. For more information, see books such as Jakob Nielsen's *Usability Engineering* (Academic Press, Boston, 1993), Ben Shneiderman's *Designing the User Interface* (third edition, Addison Wesley Longman, Reading, Mass., 1997), or Mark Pearrow's *Web Site Usability Handbook* (Charles River Media, Rockland, Mass., 2000). There are also many good usability Web sites, including www.stc.org/pics/usability/topics/index.html, www.useit.com, and www.usableweb.com.

User and task analysis

The focus in this technique is on interviewing the actual or intended users. If the system does not yet exist, you can ask your marketing department for customer profiles and use them to guide your recruitment effort. (Because marketing departments typically think of people as customers, rather than users, it's important that you ask specifically for "customer profiles, as asking for "user profiles" will most likely produce only quizzical looks.) Once you've recruited users, ask them to explain what they use the system for, what they most like about it, what they don't like about it, and so on. If there is no digital system in place, ask users questions about the current, manual process of completing the tasks. Next, observe them using the system (or completing the tasks manually). You can then ask them questions based on your observations, such as, "When you were using the XYZ Screen, you said, 'I always get mixed up here.' Could we go back to that screen now so you can show me exactly where it is you get mixed up?"

Low-fidelity prototyping

With this technique, the focus is on user interaction with designs, screens, or pages. You begin by sketching system screens or site pages, preferably on paper. The less "finished-looking" your designs are, the better. Users are typically more candid with rough, high-level interaction designs that look as if they didn't take much work. Focus your prototypes on the screens or pages that are commonly used or that you think users might find difficult to work with. It's unlikely that you'll be able to prototype and test the entire user interface.

Usability-testing the prototype

Once you've designed your low-fidelity prototype, usability-test it with three to five users or intended users. To do this, you first write a usability test script with tasks for the test participants to perform using the prototyped pages. Although the prototype is low-fidelity, ask users to interact with it as if it were a functional system. For example, if it's a paper prototype, have them use their finger for the mouse and say "I'd click here to display my most recent transactions," and so on. If they have problems completing the task, note what the problems are.

Once you've completed testing, review the findings for patterns and to understand why people had the problems they did. Next, think of alternative designs that might eliminate the problems discovered in testing. Do this as many times as necessary or possible until you have a design that facilitates good user performance. Once you've done low-fidelity prototype usability testing, you might want to conduct usability testing with an interactive, higher fidelity system prototype. However, the main idea is to try to identify usability bugs as soon as possible, which is why low-fidelity prototyping—which you can do prior to coding—is so important.

of $3 each). In all, they spent $900,000 on the problem. The company ran no user testing of the driver before its release. As one researcher put it, "The problem could have been identified and corrected at a fraction of the cost if the product had been subjected to even the simplest of usability testing."[1]

Improved productivity and efficiency

People tend to be more productive using usability-engineered systems. This benefit can be especially important in the context of IT software systems. For example, a major computer company spent $20,700 on usability work to improve the sign-on procedure in a system used by several thousand people. The resulting productivity improvement saved the company $41,700 the first day the redesigned system was used. On a system used by more than 100,000 people, for a usability outlay of $68,000, the same company recognized a benefit of $6,800,000 within the first year of the system's implementation. A cost-benefit analysis of such figures results in a cost-benefit ratio of $1:$100.[1]

Working with systems that have not been usability engineered is often stressful. Alan Cooper, "the father of Visual Basic," worked on a project to improve the usability of an airline in-flight entertainment system. IFEs are devices connected through an onboard local area network that provide movies and music to travelers on transoceanic routes. One airline's IFE was so frustrating for the flight attendants that many of them were bidding to fly shorter, local routes—which are usually considered highly undesirable—to avoid having to learn and use the difficult systems. "For flight attendants to bid for flights from Denver to Dallas just to avoid the IFE indicated a serious morale problem."[4]

When possible, people avoid using stressful systems; if people must use such systems, stress tends to undermine their productivity. And, as Cooper's anecdote illustrates, poor usability can undermine morale.

Reduced training costs

Usability-engineered systems can reduce training needs. When user interface design is informed by usability data and expertise, the resulting interfaces often facilitate and reinforce learning and retention, thereby reducing training time. At one company, only one hour of end-user training was needed on a

usability-engineered internal system, in contrast to the full week of training for a predecessor system that had no usability work. At AT&T, usability improvements saved the company $2.5 million in training expenses.[1]

Lower support costs

Providing telephone support for computer software is estimated to cost companies between $12 and $250 per call, depending on the organization.[5] Such support costs can add significantly to a system's total cost of ownership and erode profits for both the developing company and purchaser alike. When a software product is understandable and easy to learn, users don't need to call support as often. As a result, commercial software companies may need fewer people to work the support lines (and perhaps fewer DJs to entertain those on hold). At Microsoft several years ago, Word for Windows's print-merge feature was generating a lot of lengthy support calls (45 minutes each, on average). As a result of usability testing and other techniques, the user interface for the feature was adjusted. In the next release, support calls dropped dramatically, and Microsoft's savings were substantial.[1]

Reduced documentation costs

Because usability-engineered systems tend to have predictable and consistent interfaces, they are relatively easy to document. As a former technical writer, I can attest that user manuals and online help for such systems are completed more quickly and are less susceptible to inaccuracies than those of difficult-to-document systems. Also, usability-engineered systems often require less documentation, and that documentation tends to cost less to produce than documentation for systems developed without usability engineering. For example, one company saved $40,000 in a single year when usability work eliminated the need to reprint and distribute a manual.[1]

Litigation deterrence

Although software makers aren't necessarily subject to the same sorts of litigation as, for example, a manufacturer of medical equipment might be, poor usability is a potential element in lawsuits and other litigation. For example, in July 2000, a US-based Web consultancy was sued by a client company that accused it of creating a site-component interface that was "unusable."[6] Even if the lawsuit was spurious, as the Web consultancy contends, the situation points to a new liability for software development firms. Although usability engineering may not prevent such lawsuits, companies that can demonstrate that they applied usability-engineering techniques during product development might be less vulnerable should such litigation occur. The US government's recent case against Microsoft hinged on a usability question: Are users well-served when the browser and operating system are closely integrated? Although no usability experts were called in to testify, they are likely to be included in the future as usability awareness increases.

Increased e-commerce potential

Though usability can benefit all development organizations, perhaps nowhere is the relationship between usability and profitability as direct as in e-commerce, as Forrester Research suggests:

> *Usability goals are business goals. Web sites that are hard to use frustrate customers, forfeit revenue, and erode brands. Executives can apply a disciplined approach to improve all aspects of ease-of-use. Start with usability reviews to assess specific flaws and understand their causes. Then fix the right problems through action-driven design practices. Finally, maintain usability with changes in business processes.*[7]

Usability-engineered sites let users be more efficient and productive. However, it's important that you interpret efficiency and productivity in relation to online shopping. Ideally, online shopping should be enjoyable, rather than frustrating: Users should not have to waste time searching for merchandise or figuring out how to buy it; nor should they have any doubt that their credit-card numbers and other personal information are secure. Buying a product or service online should be superior to making a purchase in a brick-and-mortar shop.

More than 44 million people in the US have made online purchases; 37 million more say they expect to do so soon.[8] However, many of these would-be online-shoppers won't succeed in making a Web

When user interface design is informed by usability data and expertise, the resulting interfaces often facilitate and reinforce learning and retention.

Usability is important for all Web sites, but for e-commerce sites, having a competitive edge in usability is crucial.

purchase, because e-commerce sites are, for the most part, too difficult for the average user to navigate. Moreover, the experience of some online shoppers has been so bad that they don't want to buy online again.[3,7]

Finally, online shoppers spend most of their time and money at sites with the best usability.[9] Good navigation and site design make it easier for users to find what they're looking for and to buy it once they've found it. Usability can significantly improve the e-commerce bottom line: According to Jakob Nielsen, usability efforts can increase sales by 100 percent.[10]

Competitive edge

Users always cite ease-of-use as high on their list of software system demands.[1] Thus, giving users usability is giving them what they want. Users appreciate software and Web sites that don't waste their time or try their patience with complicated user interfaces. Building usability into your software tells users that you value their time and don't take them for granted.

Usability is important for all Web sites, but for e-commerce sites, having a competitive edge in usability is crucial. Such sites commonly drive away nearly half of repeat traffic by making it hard for visitors to find what they need.[11] And, repeat customers are the most valuable: New users at one e-commerce site spent an average of $127 per purchase, while repeat users spent nearly twice that.[12]

Usable e-commerce sites also build goodwill. Users recognize the effort put into making their e-commerce experience easy and efficient by returning to usable sites. Moreover, one of the biggest obstacles to e-commerce is trust. Consumers must trust a site before they will disclose the personal and financial information typically required for online purchases. A study of e-commerce trust found that navigation and presentation—both usability concerns—were essential for creating trust.[13]

Advertising advantages

High usability can garner attention for your company's Web site and help distinguish it from other sites. Improved usability can also help differentiate commercial software applications. Compaq, Microsoft, and Lotus have all made usability part of their advertising campaigns, for example.[5] More recently, a Swedish consultancy announced a "usability guarantee" for sites it develops, requiring large-scale projects to undergo testing in a usability lab before they are released. If the project fails the test, the consultancy promises to "improve the solution without any additional costs to the client."[14] As another example, MacroMedia recently issued a press release describing its "usability initiative." Though the initiative seems to consist of posting basic usability tips on its Web site, the fact that a large software company took such action suggests that companies might be realizing the advantage of being perceived as usability-aware.

These examples notwithstanding, and despite its great potential, usability's advertising value remains largely unexploited. This seems especially so in e-commerce, where users are increasingly nontechnical consumers who won't suffer technical difficulties gladly. It may behoove usability proponents to try to increase advertising departments' awareness of usability's value.

Better notices in the media

People in the media have discovered the connections among usability, productivity, and cost-effectiveness, especially on the Internet. Companies are regularly taken to task about usability in business publications and on e-business sites. For example, *CIO Business Web Magazine* pointed out, "On a corporate intranet, poor usability means poor employee productivity; investments in making an intranet easier to use can pay off by a factor of 10 or more, especially at large companies."[15] The question arises: If those in the media see increased productivity and cost-effectiveness, can shareholders be far behind?

In 1993, Nielsen studied the coverage of usability issues in trade press reviews of new software products and found that approximately 18 to 30 percent of the accounts were usability-related.[16] A good review in an industry publication can be worth millions in advertising. Such reviews increasingly include usability as a criterion. One of *Internet Week*'s most popular columns features user-interface design and usability specialists discussing the relative usability of various e-commerce and e-business sites, for example.

Performing usability cost-benefit analyses in your company could be a first step toward introducing usability engineering techniques, and thus the first step toward realizing the benefits I outlined. Working to improve usability can bring significant economic benefits to companies that develop IT applications, e-commerce sites, and commercial software. Of course, users of these systems benefit as well.

References

1. R.G. Bias and D. J. Mayhew, eds., *Cost-Justifying Usability*, Harcourt Brace & Co., Boston, 1994.
2. A.M. Lund, "Another Approach to Justifying the Cost of Usability," *Interactions*, vol. 4, no. 3, May/June 1997, pp. 48–56.
3. R.S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw Hill, New York, 1992.
4. A. Cooper, *The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*, SAMS, Indianapolis, 1999.
5. M.E. Wiklund, *Usability In Practice: How Companies Develop User-Friendly Products*, Academic Press, Boston, 1994.
6. B. Berkowitz and D. Levin, "IAM Sues Razorfish for Poor Design," *The Standard*, 14 July 2000, www.thestandard.com/article/display/0%2C1151%2C16831%2C00.html (current 3 Jan. 2001).
7. H. Manning, "Why Most Web Sites Fail," Forrester Research, Sept. 1998; view an excerpt or purchase the report at www.forrester.com/ER/Research/Report/Excerpt/0,1338,1285,FF.html (current 3 Jan. 2001).
8. S. Wildstrom, "A Computer User's Manifesto," *Business Week*, Sept. 28, 1998.
9. J. Nielsen, "The Web Usage Paradox: Why Do People Use Something This Bad?" *Alertbox*, 9 Aug. 1998; www.useit.com/alertbox/980809.html (current 3 Jan. 2001).
10. J. Nielsen, "Web Research: Believe the Data." *Alertbox*, 11 July 1999; www.useit.com/alertbox/990711.html (current 3 Jan. 2001).
11. H. Manning, "The Right Way To Test Ease-Of-Use." Forrester Research, Jan. 1999; view an excerpt or purchase the report at www.forrester.com/ER/Research/Brief/Excerpt/0,1317,5299,FF.html (current 3 Jan. 2001).
12. J. Nielsen, "Loyalty on the Web," *Alertbox*, 1 Aug. 1997, www.useit.com/alertbox/9708a.html (current 3 Jan. 2001).
13. Cheskin Research and Studio Archetype/Sapient, "E-commerce Trust Study," 1999; www.studioarchetype.com/cheskin/index.html (current 3 Jan. 2001).
14. "Icon Medialab Announces Usability Guarantee," Icon Media Lab, 29 Nov. 2000; www.iconmedialab.se/default/news/press_releases/right.asp?id=310 (current 3 Jan. 2001).
15. S. Kalin, "Mazed and Confused," *CIO Web Business Magazine*, 1 Apr. 1999; www.cio.com/archive/webbusiness/040199_use.html (current 3 Jan. 2001).
16. J. Nielsen, "Is Usability Engineering Really Worth It?" *IEEE Software*, vol. 10, no. 6, Nov. 1993, pp. 90–92.

About the Author

George M. Donahue is a senior experience modeler at Sapient. In addition to conducting scores of usability tests of e-commerce, financial, interactive television, and new media Web sites, he has designed user interfaces and written user interface design guides, user manuals, and online help guides. His current research interests include strategic usability and cross-cultural user interface design. Before joining Sapient, Donahue was a senior usability specialist at the Compuware Corporation. He holds degrees from the University of Delaware and Clemson University and is a member of the ACM Special Interest Group on Computer-Human Interaction. He is also a member of the Usability Professionals' Association. Contact him at Sapient, 250 Williams St., Ste. 1400, Atlanta, GA 30303; gdonahue@sapient.com; www.sapient.com.