

Cost-Justifying Accessibility

Paul Sherman
austin usability.

Introduction

This paper is intended to provide an introduction to software and web accessibility, and describe the costs and benefits of integrating accessibility with the software development process. We propose that accessibility is a crucial part of the software development lifecycle because it can both mitigate certain costs and yield significant benefits to the development organization. In this work we also set forth some guidelines for integrating accessibility into development in a cost-conscious manner.

Detailed guidelines for designing accessible software and websites are beyond the scope of this paper; coverage of this can be found in the Austin Usability white paper, "Designing in Accessibility for Software Applications and the Web."

What is Accessibility?

Software accessibility can be defined as a trait of software or other electronic information sources whereby it is usable by people with physical, cognitive or emotional disabilities. A software, website or other electronic information source is accessible if someone with a disability is able to use the source's data, information, or services as effectively as someone without a disability (Slatin, 2001a).

Disabilities in the US and Worldwide

The US Census Bureau (Macneil, 2001) reports that as of the mid-1990's, about 21% of the population (54 million people) had some level of disability. Disability is defined by the 1990 Americans with Disabilities Act as an impairment that substantially limits one or more of the following activities:

- major life activities such as seeing, hearing, speaking, lifting and carrying, walking, and using stairs,
- daily living activities such as getting around inside the home, getting in or out of a bed or chair, bathing, dressing, eating, and using the toilet, and
- instrumental activities such as leaving the home, keeping track of money or bills, preparing meals, doing light housework, and using the telephone.

In the European Community, 37 million people have a disability. Worldwide, the number of people with some level of disability is estimated at about 500 million people. The number of people impacted by inaccessible computer and software design is difficult to calculate precisely, but is estimated to be over 30 million in the United States alone (Microsoft, 2001). Increasingly, older people - with higher disability rates - are using computers and getting online. (Georgia Institute of Technology, 1998)

Government Initiatives Mandating Accessibility

Over the past decade, the US Federal government has mandated that people with disabilities be afforded access to electronic information sources, including software programs and the World Wide Web.

Title III of the 1990 Americans with Disabilities Act (ADA) mandated that all "public accommodations" must provide reasonable access to persons with disabilities. In 1996, the Department of Justice ruled that Web sites are public accommodations and must therefore offer access to the disabled.

Section 508 of the Workforce Rehabilitation Act requires that, effective June 2001, any Federal agency or vendor that provides software to a Federal agency must ensure that those with disabilities have comparable access to information and data. As of this writing, the US Department of Education is currently deciding whether to require institutions receiving funding to abide by Section 508 regulations.

Requirements on Software, Web Suppliers

This regulatory environment puts requirements on groups with web sites that are accessed by government employees or government contractors, and vendors who sell to a government agency (although current vendors are exempt).

Failure to address accessibility exposes software makers to negative consequences such as noncompliance with Federal guidelines and possible loss of contracts. These in turn can lead to loss of sales, lawsuits, and public relations disasters. A notable example of a lawsuit and bad press is the action brought against AOL by the National Foundation for the Blind (NFB) on behalf of blind users. The suit claimed that AOL's software was not compatible with screen readers, assistive technology typically employed by blind and visually impaired people to read aloud the contents of the computer screen (Smith, 1999). Another PR debacle occurred before the 2000 Summer Olympic Games in Sydney, Australia, when Games organizers conceded that it was not possible to make the official web site accessible to blind people before the opening ceremony, despite an order from the Australian Human Rights and Equal Opportunity Commission (Lebihan, 2000).

Recently, the NFB and the Connecticut attorney general's office reached an agreement with four online tax filing services that were listed as e-filing partners by the IRS, but whose sites were not accessible by users with screen readers. Blind and visually impaired taxpayers could not file returns when using these sites. The four companies agreed to make their sites accessible for the 2000 tax season (Heim, 2000).

Cost-Justifying Accessibility

These consequences may impose considerable costs on software vendors. However, several of these costs are *potential* and not realized costs. Because these costs are not always tangible, some vendors may not feel they need to minimize them, and so may not actively integrate accessibility into the development lifecycle. Vendors may also shy away from integrating accessibility into development because they are unaware of the range of benefits engendered by accessible designs.

We argue here that accessibility is a crucial part of the development effort because it can mitigate certain costs, *and* because it can yield significant benefits for an organization. We also claim that accessibility can be cost-justified in much the same way that usability efforts can.

Table 1: Benefits and Costs of Accessibility Engineering.

Benefits	Costs
<ul style="list-style-type: none"> • Savings from finding problems early. • Decreased need for user support; decreased training costs. • Increased user productivity, lower errors, and greater satisfaction. • Allows entry into markets that may be off-limits if accessibility is not addressed. • Possibly provides a measure of protection from litigation. • Yields possible public relations benefits. 	<ul style="list-style-type: none"> • Additional resources need to be devoted in product planning and design stages. • Compliance must be assessed and documented.

Benefits

It has been demonstrated that adding usability to a software development program can provide up to a one hundred-fold return on investment (Bias & Mayhew, 1994). Benefits to the software organization include savings from finding problems early, decreased need for user support, decreased training costs, and increased sales. Benefits to the users include increased productivity, decreased errors, and greater satisfaction.

Designing for accessibility (both in the physical world and onscreen) invariably brings about a 'spread of benefits.' Accessible designs – be they wheelchair ramps, curb cuts, web sites, or software applications – yield greater usability for all, not just for people with disabilities. For example, wheelchair ramps afford access to parents pushing strollers, as well as to the wheelchair-bound. Similarly, a web site designed to account for visually impaired or blind users will often benefit those with healthy vision, because it will be more legible and have better color contrast – design elements that increase usability for all.

Therefore, attending to accessibility in the software development process provides all the benefits of usability engineering, as well as these benefits:

- Allows entry into markets (such as government or educational institutions) that may have been previously off-limits.
- Might provide a measure of protection from litigation (avoidance of legal costs is a benefit).
- Can yield public relations benefits when the product is released.

Greenfield Scenario: Costs for New Applications, Sites

Without sufficient planning, the costs associated with accessibility efforts during development can exceed the costs of business-as-usual. These costs include the additional resources that must be devoted to accessibility engineering in the product planning and design stages, as well as to the compliance assessment and accessibility process documentation efforts.

However, with up-front planning and good practices, the cost of accessibility can be lowered to negligible levels (Slatin, 2001b). Like usability or internationalization efforts, accessibility is most effective and cheapest when it is addressed at the outset, rather than appended onto a development effort after its inception.

The components of successful up-front planning and practices can be summarized as follows (Sherman, 2001; Slatin, 2001b; W3C, 2000):

- Set specific accessibility goals, based on priorities and knowledge of the disabilities present in the target user group.
- Avoid the temptation to create a separate “accessible” site or software version. As Slatin points out, separate is *not* equal. The cost of maintaining and synchronizing two separate versions of a resource – be it a website or an application – can be prohibitive. When resources are limited, the secondary version always receives less attention.
- Adapt and operationalize guidelines and standards for the organization. Incorporating accessibility guidelines that are specific to the product within the various requirements and test documents that are associated with software development can accomplish this. Providing developers with styleguides that include accessibility guidance is another effective method to consider.
- Assign accessibility tasks to appropriate individuals throughout the team. This entails that responsibility for ensuring accessibility should be spread around, and not limited to a particular department or individual. Technical writers should craft accessible textual elements, software developers should create conforming code, testers should inspect for conformance, etc.
- Use development and authoring tools that support accessibility.

- Test early and use a variety of methods, including automated evaluation tools, end-user assistive technology, professional review, and scenario testing with actual end users.

This last point should be stressed: as in usability engineering, accessibility engineering demands that designs be tested with target users. A design is genuinely accessible only when empirically demonstrated that it is usable by people with disabilities.

Retrofit Scenario: Costs for Existing Applications, Sites

The costs associated with making existing applications and websites accessible are probably greater than the costs of greenfield accessibility efforts, because achieving accessibility may involve significant rework. However, these costs can be mitigated by employing the techniques described in the above section, as well as by following the principles of usability triage (Bias & Keough, 2000). In usability triage, usability issues are assigned priority categories – emergent, urgent, and non-urgent – that depend on the criticality of the usability problem, and the amount of time and resources available to fix them. Issues are then addressed accordingly.

This approach can be followed for accessibility issues as well, particularly for web sites and web-based applications. In a similar vein, the Web Content Accessibility Guidelines provide direction for achieving website accessibility (W3C, 1999). Each guideline has prioritized checkpoints that can be used to measure conformance.

The checkpoints are grouped into three priority levels. Priority 1 checkpoints are “must haves;” nonconformance to these checkpoints means that some people will *not* be able to access the information presented by the site. Priority 2 checkpoints are “should haves.” Nonconformance means some people *will* have difficulty accessing the site’s content. Priority 3 checkpoints are “nice to haves” – some people *may* have difficulty accessing content. Level A conformance denotes that all Priority 1 checkpoints have been satisfied. Level AA conformance indicates that both Priority 1 and Priority 2 checkpoints have been satisfied. Level AAA conformance indicates that all checkpoints are met.

Although this checkpoint approach and the principles of usability triage were created for web-based content, their general principles and emphasis on prioritization apply to traditional software applications as well.

Conclusion

Accessibility is increasingly important to software development organizations. Its presence or absence from a software product or website can have a significant economic impact on the vendor organization. In order to manage these costs and to lower exposure to risk, it is essential that software vendors or website publishers integrate accessibility into their development processes. This can be done more effectively and efficiently if accessibility efforts are well planned, and a triage (or prioritization) approach is taken. If these general guidelines are adhered to, vendor organizations can minimize their costs, lessen their exposure to risk, and realize significant benefits.

References

- Bias, R.G., & Keough, K. (2000). Usability triage for web sites. In *Proceedings of the 6th Conference on Human Factors and the Web*. Austin: SBC TRI.
- Bias, R. G., & Mayhew, D. J. (Eds.) (1994). *Cost-justifying usability*. Boston: Academic Press.
- Georgia Institute of Technology (1998). *GVU Center 10th WWW User Survey, 1998*. Atlanta: GVU. (www.gvu.gatech.edu/user_surveys/)
- Heim, J. (2000). Locking out the disabled. *PC World*, September 2000.
- Lebihan, R. (2000). Olympic site must race for blind accessibility. *ZDNet Australia*, August 29, 2000. (<http://au.computers.yahoo.com/computers/20000829/news/967503600-3020611587.html>)

- Microsoft (2001). *How many people have disabilities?* (www.microsoft.com/enable/microsoft/)
- Sherman, P.J. (2001). Designing in accessibility for software applications and the Web. White paper. Austin: Austin Usability.
- Slatin, J.M. (2001a). *Focus on participation: Toward a more accessible web*. Unpublished presentation.
- Slatin, J.M. (2001b). The art of alt: Toward a more accessible web. In *Computers and Composition: An International Journal for Teachers of Writing*, 18(1-2), 73-82.
- Smith, E (1999). Suing for AOL access. *PCWorld.com*, November 4, 1999. (www.pcworld.com/news/article.asp?aid=13632).
- Macneil, J. (2001). Americans With Disabilities: 1997. In *Household Economic Studies Current Population Reports*, P70-73. Washington: US Census Bureau.
- W3C (2000). *Techniques for accessibility evaluation and repair tools*. W3C Working Draft, 26 April 2000. Cambridge: W3C.
- W3C (1999). *Web content accessibility guidelines 1.0*. W3C Recommendation, 5 May 1999. Cambridge: W3C.