

# **Evaluation of State-Based Integrated Health Information Systems**

## **Web-Based Systems for the Dissemination of Health-Related Data:**

### **A Guide for Public Health Agencies Developing, Adopting, or Purchasing Interactive Web-based Data Dissemination Systems**

**Contract Number: 200-96-0598, Task 23**

\*This report is provided solely as a resource and does not represent the official positions or policies of the CDC.

*Submitted to:*  
Epidemiology Program Office  
Centers for Disease Control and Prevention

*Submitted by*  
ORC Macro  
3 Corporate Square, Suite 370  
Atlanta, GA 30329

**February 2003**

**ORC** MACRO<sup>SM</sup>

# Table of Contents

	<u>Page</u>
<b>Introduction .....</b>	<b>1</b>
<b>About this Guide .....</b>	<b>1</b>
<b>Task 1: Determine Organizational Readiness and Impact.....</b>	<b>7</b>
Step 1: Identify and Engage Stakeholders .....	8
Step 2: Assess Costs.....	10
Step 3: Assess Staffing.....	10
Step 4: Consider Existing Organizational Policies and Procedures.....	11
<b>Task 2: Define the Project .....</b>	<b>13</b>
Step 1: Define the Purpose of the Site .....	14
Step 2: Identify System Users .....	15
Step 3: Set Goals .....	17
Step 4: Assess the Technical Environment .....	18
Step 5: Establish Data Confidentiality Policies .....	19
Step 6: Establish System Security Policies .....	19
Step 7: Establish Business Rules.....	20
Step 8: Develop System Design Document .....	21
Step 9: Develop a Project Strategy.....	21
<b>Task 3: Select the Approach .....</b>	<b>24</b>
Step 1: Determine the Functional Capability of Each Approach.....	24
Step 2: Identify Staff Resources Necessary to Implement the Approach .....	24
Step 3: Estimate the Cost of Implementing the Approach.....	25
Step 4: Determine Whether the Selected Approach Will Meet Organizational Needs and Constraints .....	26
<b>Task 4: Design Data Presentation Formats .....</b>	<b>28</b>
Step 1: Maintain Adherence to Data Presentation Standards.....	28
Step 2: Organize Data Presentation.....	29

# Table of Contents

Page

<b>Task 5: Implement the System</b> .....	<b>33</b>
<b>Approach 1: Design and Develop a New System</b> .....	<b>34</b>
Step 1: Design Web Pages for Usability .....	35
Step 2: Design Web Pages for Accessibility .....	36
Step 3: Create Graphic Composites .....	38
Step 4: Create User Friendly Interface.....	39
Step 5: Conduct Image Slicing.....	40
Step 6: Create Templates .....	40
Step 7: Utilize Cascading Style Sheets (CSS).....	41
Step 8: Conduct Validation and Test Guidelines .....	41
Step 9: Implement Code Review.....	42
Step 10: Designate File and Directory Naming Standards.....	42
Step 11: Determine File Naming Conventions .....	43
Step 12: Implement Coding Standards.....	44
Step 13: Engage in HTML Production.....	45
Step 14: Create Style Guide .....	45
Step 15: Create User Friendly Query Functions .....	46
Step 16: Design Usable Outputs .....	47
<b>Approach 2: Adopt or Purchase an Existing System</b> .....	<b>48</b>
Step 1: Organize a Team to Facilitate the Adoption or Purchase .....	48
Step 2: Assess the System .....	48
Step 3: Determine System Compatibility.....	51
Step 4: Establish Data Presentation Standards.....	51
Step 5: Review Confidentiality Constraints .....	52
Step 6: Consider Data Availability .....	52
Step 7: Modify User Interface as Necessary.....	52
Step 8: Modify Query Functions as Necessary .....	54
Step 9: Modify Outputs as Necessary .....	54
Step 10: Concept Prototyping and Design .....	55
Step 11: Modify Web Pages for Usability .....	55
Step 12: Modify Web Pages for Accessibility .....	56
Step 13: Review and Update the Style Guide .....	58

# Table of Contents

	<u>Page</u>
<b>Task 6: Test the System .....</b>	<b>60</b>
Step 1: Test the System .....	60
Step 2: Publicize the Launch Date .....	60
Step 3: Promote Training on the System.....	61
<b>Task 7: Evaluate the System .....</b>	<b>63</b>
Step 1: Perform Log Analysis and Monitor System Use .....	65
Step 2: Obtain User Feedback.....	67
<b>Task 8: Maintain the System .....</b>	<b>69</b>
<b>Summary .....</b>	<b>70</b>

## Appendices

- Appendix A: Checklist of System Considerations and Best Practices in System Design Features
- Appendix B: References and Resources
- Appendix C: User Analysis
- Appendix D: Goals and Measurement
- Appendix E: Technical Specifications
- Appendix F: Discount Usability Testing
- Appendix G: Accessibility Checklist
- Appendix H: Design Guidelines
- Appendix I: Model Discussion Guide
- Appendix J: Glossary

## Introduction

Increasingly, public health agencies are using the Internet to disseminate data. Many of these agencies are in various stages of developing, adopting, or purchasing systems that will better enable them to combine data from different sources, improve the timeliness, accuracy, and quality of the data, and present the data in a way that is useful to their constituents.

The increased activity and interest by state and local health agencies have created a need for uniform guidance to address the key organizational, technical, and system design issues involved in developing, adopting, or purchasing a system for Web-based data dissemination.

To help address this need, the Centers for Disease Control and Prevention (CDC), through a contract with ORC Macro, undertook a three-year evaluation of current public health agency Web-based data dissemination systems and practices.<sup>1</sup>

The results of this evaluation were used to develop this guide as a practical method for public health agencies to plan for the development, adoption, or purchase of an interactive Web-based data dissemination system. The principles and best practices discussed in this guide are based on lessons learned from the actual experience of public health agencies that have been through the process. The guidelines incorporate best practices and industry standards in software development and are consistent with federal guidelines on accessibility based on Section 508 of the U.S. Rehabilitation Act<sup>2</sup> and CDC interface<sup>3</sup> guidelines.

## About this Guide

### Content

This document provides a consolidated list of processes, standards, and checklists that public health agencies and state and local health authorities can use to successfully develop, adopt, or purchase a queryable Web-based dissemination tool. This guide may also be used by public health agencies that have existing systems, but desire to make modifications or enhancements.

This document has eight main sections. Each represents a specific major task that agencies should engage in if wishing to develop, adopt, or purchase a queryable Web-based tool. Throughout the remainder of the document the terms “section” and “task” are used interchangeably. Each of the tasks is further broken down into individual steps that are required to complete that task.

---

<sup>1</sup> Additional information on the evaluations leading up to the development of this guide, including reports from all earlier phases of the project, can be reviewed at <http://www.macoint.com/~atl/dphsi/>.

<sup>2</sup> These guidelines can be found at <http://www.access-board.gov/sec508/508standards.htm>. Additional information about these guidelines can be found at <http://www.section508.gov>.

<sup>3</sup> The images, commands, display formats, and other devices provided by a Web site to let the user communicate with and use the Web.

The guidelines provide state and local health agencies a project management framework that ensures all relevant factors are identified and analyzed at the outset of the project. It highlights potential pitfalls and areas that have historically proven problematic when undertaking a project of this kind. The project management framework is primarily aimed at assisting organizational decision makers, project managers and program staff within public health agencies to make informed decisions about the implementation of a system.

The guidelines also provide information on best practices to ensure that the system is designed to optimize usability<sup>4</sup>. This information is aimed mainly at designers, developers or purchasers and focuses on best practices that maximize user experience. Accessibility guidelines are also provided and they can be interpreted by each state or local health agency depending on its accessibility policy.

As a technical reference, the guidelines are limited to the user interface<sup>5</sup> and do not describe or prescribe approaches to the process of designing an entire Web application. Specifically issues related to data storage, acquisition, and management are described briefly, and only to provide context for the project-planning framework. The National Association for Public Health Statistics and Information Systems (<http://www.naphsis.org>) and the National Association of Health Data Organizations (<http://www.nahdo.org>) provide more guidance on data related issues.

## Organization

The first task, **Determining Organizational Readiness and Impact**, focuses on organizational level issues. It provides a framework to conduct a full analysis of all of the implications and costs associated with the development, adoption, or purchase of a queryable Web-based data dissemination tool, to help an agency make an informed decision about how to implement this tool.

Tasks two, three, four, and five **Define the Project, Select the Approach, Design the Data Presentation Format, and Implement the System** provide brief descriptions of activities that should be conducted during the individual stages of development, adoption, or purchase and implementation of the system.

Tasks six, seven, and eight, **Test the System, Evaluate the System, and Maintain the System**, include the steps to make the system operational and provide ongoing maintenance.

---

<sup>4</sup> The ease with which users can perform tasks (e.g., searching for information, submitting data).

<sup>5</sup> The images, commands, display formats, and other devices provided by a Web site to let the user communicate with and use the Web.

The sequence of steps described above is widely accepted as an industry standard for information technology<sup>6</sup> (IT) projects. Employing a structured project management methodology will help to:

- Clarify and refine project objectives
- Define scope, scale, and budget
- Identify technical requirements
- Identify key project activities and organize resources
- Develop and communicate a timeline
- Define accountability for activities and resources
- Provide a documented framework for management, control, and measurement of project progress

The process can be used for the development, adoption, or purchase of a tool, although it will need to be adapted to the particular characteristics of an individual project. Not all of these steps may be necessary for every project.

*Developing* a system encompasses the entire process of designing, implementing, and maintaining a software system, either in-house or through an outside contractor. All of the sections of this report are relevant for developing a system.

*Adopting* a system involves obtaining source code<sup>7</sup> (i.e., the original programming code that drives the system) from another state/agency and modifying it for use and maintenance within your state/agency.

*Purchasing* a system means licensing a commercial-off-the-shelf (COTS) software system. In this situation, the software has already been designed, coded, and tested, so sections of the Guide addressing these tasks will not be applicable if purchasing a COTS system.

The specific project guidelines focus on interface design and development, adoption, or purchase of the Web application. References to the development or adoption of a back-end database<sup>8</sup> are limited; the approach an organization would take depends on factors pertinent to them (e.g., types of databases, data definitions used, aggregated vs. raw data). However, we have included information on establishing “business rules,” because this is a critical step in the successful development, adoption, or purchase of a queryable Web-based data dissemination system.

The details contained in these guidelines are summarized in two checklists (Appendix A). The first checklist includes the most important system features to consider when determining the type of system to be developed, adopted, or purchased. The second checklist contains best practices

---

<sup>6</sup> The computer systems in a business or other enterprise.

<sup>7</sup> The form in which a programmer writes a computer program in a formal programming language.

<sup>8</sup> This term has two meanings:

A database that performs tasks that the user is not aware of.

A database whose content is accessed through a Web interface.

in design features. Both checklists can also be used as a tool for evaluating already existing systems.

Appendix B provides a listing of the references and resources that are cited throughout the guidelines. Appendices C through I provide tools for implementing the tasks and steps outlined in the guidelines. The guidelines and appendices contain many technical terms that are defined when they are first used and are also summarized in a glossary, contained in Appendix J.

## **Audience**

The guide is designed to be all-encompassing in terms of the issues associated with developing, adopting, or purchasing a queriable Web-based data dissemination system, from organizational level considerations to coding standards. However, the individual sections are stand-alone pieces that can be used by individuals with particular project roles.

The primary audiences for the first three tasks, **Determine Organizational Readiness**, **Define the Project**, and **Select the Approach**, include senior managers and administrators and information technology managers. Information presented will help this audience understand the full scope of the project, how it fits with organizational goals and existing information technology, and the resources required for the project.

Tasks four and five, **Design the Data Presentation Format** and **Implement the System**, provide detailed, technical information geared toward information technology staff who will be responsible for such roles as:

- Information architecture<sup>9</sup>
- Usability testing
- Interface design
- Accessibility testing
- Graphic design<sup>10</sup>
- Content management
- Programming
- Hypertext markup language<sup>11</sup> (HTML) coding
- System testing/quality assurance

Task five, **Implement the System**, presents two approaches. Approach one is targeted to organizations that are designing and developing a system, Approach two is targeted to organizations that are adopting or purchasing a system.

---

<sup>9</sup> The basic conceptual structure of the system, broken down by content categories and user tasks. How large collections of information are organized in a Web site so that users can conduct searches or carry out tasks easily and successfully.

<sup>10</sup> Commercial art, including that produced for Web sites (such as the buttons, logos, and other images).

<sup>11</sup> The code used to create Web pages.



The final three tasks, **Test the System**, **Evaluate the System**, and **Maintain the System** also provide technical instructions geared towards information technology staff. However, senior managers and administrators should become familiar with the general purpose of these tasks as well, since they provide important feedback.

### **How to Access this Guide**

The guidelines document and a non-deployable prototype<sup>12</sup> that demonstrates the best practices described in these guidelines are available at <http://www.macoint.com/~atl/dphsi/>.

---

<sup>12</sup> Representation of a system for testing purposes and as a way of understanding the potential limitations of design as well as features that work well.

## **Task 1**

**Determine Organizational Readiness and Impact**

## Task 1: Determine Organizational Readiness and Impact

---

Undertaking the design and development, adoption, or purchase of a queriable Web-based data dissemination system has strategic implications for public health agencies. Making the decision to implement a system will have an impact on information technology strategies and infrastructure, budgeting and planning, personnel and staffing, and policy. The decision may also have an impact on how the agency sees its role as a provider of data and how it chooses to service its constituents and develop relationships with them. The strategic impact of such a project will vary depending on the size and structure of the agency and its existing resources.

The decision to embark on the implementation of a system and whether the system should be developed, adopted, or purchased, depends on a number of interrelated factors. Identifying the best approach depends on analyzing the characteristics and objectives of each individual public health agency to identify the positive and negative effects of development, adoption, or purchase. The following guidelines are designed to provide a decision-making framework for each public health agency. No specific recommendations are made here. However, by reviewing the information contained in these guidelines an organization will be much better positioned to make an informed choice about which option is most appropriate for them.

### Key considerations:

- Direct and indirect effects on agency mission, structure, staffing
- Cost
- Compatibility with agency policies and procedures

Addressing the elements listed in the following checklists will allow agencies to build a profile of their capacity to implement a queriable Web-based data dissemination tool. Agencies can then build capacity where it does not exist or is insufficient. Such areas may include:

- Staffing
- System support
- System security
- Training
- Organizational support for technology
- System compatibility
- Standards and confidentiality
- Funding
- Data issues
- Users

Steps for identifying areas for strengthening include conducting a stakeholder analysis, documenting costs, assessing staffing, and examining implications for organizational policies and processes.

## Step 1: Identify and Engage Stakeholders

To assess the impact of developing, adopting, or purchasing a queryable Web-based data dissemination system it is necessary to identify its direct or indirect effects on the organization, during development and once it is implemented. Key stakeholders (e.g., internal staff, end users, etc.) need to be identified and included in the initial decision-making process.

Experience shows that organizational support is a critical element in the development, adoption, or purchase of a system. The project team is likely to include members from different parts of the organization, and creating an environment of inter-departmental cooperation is essential to success. Public health agencies that have already developed, adopted, or purchased such systems emphasized the importance of engaging the IT department at the earliest stages of discussion.

High-level administrative support can help assure that adequate resources are dedicated to the project and backing from the IT department can help ensure compatibility and technical support for implementation and ongoing maintenance.

Public health agencies also may want to involve external stakeholders such as local health departments or other constituents. This will depend on existing relationships and the extent to which the agency prefers to keep decision-making an internal process at this early stage. The degree to which people are affected or involved will vary, but understanding their roles, requirements, motivations, and commitment is important.

Potential issues of concern to explore with each stakeholder group are summarized below.

### Issues to explore with senior management and administrators:

- What is driving the demand for a queryable Web-based data dissemination system?
- Who needs the system and its data? Who are the potential users?
- Is its priority high enough to warrant investment in the project?
- What are the present organizational goals?
- Will the implementation of a system further the organizational goals?
- Do organizational goals need to be reassessed or reprioritized?
- Will the implementation of a data dissemination system fulfill the needs of internal and external users of the data and reduce the number of queries, freeing staff to focus on other activities?
- How will a Web-based approach help fulfill the needs of internal and external consumers?
- Is there an alternative or better technical approach than a Web-based tool?
- What are the policy issues associated with conducting this project and how substantial/problematic are they?
- How will this project move the organization forward in achieving its goals?
- How can users' needs be institutionalized in designing and implementing the system?

#### **Issues to explore with information technology managers and staff:**

- Is there a need for a queryable Web-based data dissemination system?
- What are the issues or concerns that would need to be addressed related to existing hardware, operating system software, and Web server applications?
- How will the project interface with existing system functions (e.g., Web applications, databases, etc.)?
- Are any changes anticipated for existing systems, applications, and functions that would have implications for a queryable Web-based data dissemination system?
- Who should have the decision-making authority and responsibility for content, datasets, technical development, and maintenance?
- How well do current data dissemination efforts meet users' needs? What additional needs do users have?
- Does the IT department have the staff capacity and expertise to undertake the acquisition or development of a new dissemination system?
- What are the implications for IT department collaborations with other departments in the agency? Who else would need to be involved?

#### **Issues to explore with data users and providers:**

- Do current data dissemination efforts meet needs? What works well? What are the gaps?
- What additional types of data are available/needed? What level of data?
- Would it be an appropriate focus for the agency to provide access to a queryable Web-based data dissemination system?
- Are there other constituent needs that should have a higher priority?
- Is the agency perceived to have the necessary expertise to support this type of technology?
- How should users be involved in the design and implementation of the system?

#### **Issues to explore with budget and accounting staff:**

- Given the current financial environment, is funding to support the development or acquisition of a system feasible?
- What is the potential for cost savings or cost offset?
- What level of funding is available to support indirect costs related to making the system operational (e.g., software upgrades, personnel costs such as training fees, staff time spent in training)?
- What funding is available to support ongoing maintenance costs?
- If additional full-time equivalents (FTEs) are required can they be funded?
- Will additional funding be needed? What are the potential sources?

Given the answers to these questions, an organization can quantify more clearly the true benefits to be gained from undertaking an effort to implement a queryable Web-based data dissemination system. More detailed assessments of user needs are described in Task 2.

## **Step 2: Assess Costs**

It is important to identify the costs involved in the current data dissemination efforts in order to assess the cost-benefit of a queryable Web-based data dissemination system. To assess these costs, consider the resource expenditures in dollars and staff time to:

- Produce printed reports
- Respond to telephone inquiries
- Develop static tables or reports for posting on a Web site
- Maintain current hardware, operating system software, Web server applications, and system functions

The costs associated with the current level of effort should be compared with the estimated costs associated with whichever approach the organization wants to deploy, once that decision has been made.

Information about cost implications of developing, adopting, or purchasing a Web-based data dissemination system will be discussed in Task 3.

## **Step 3: Assess Staffing**

There are several issues to explore regarding IT staff needs for system development, adoption, or purchase. It is important to assess whether the IT department has the staff capacity and expertise to undertake the development or acquisition of a new dissemination system.

Does staff capacity include:

- Technical support staff to resolve server or system issues as they arise?
- Sufficient staff to perform all needed development and maintenance work for the system?

Does staff expertise include adequate experience in:

- Maintaining the specific server (e.g., Apache Web server), database (e.g., SQL Server), and additional software (e.g., statistical analysis software) that comprise the system?
- Establishing security protocol and backup plans for ensuring protection against data loss or system interruption?
- Programming in the languages selected for the database and application?
- Identifying data confidentiality and data integrity issues related to the system?
- Planning for adequate server resources to support the number of anticipated users?
- Planning for adequate server support for the amount of data in the system?

#### **Step 4: Consider Existing Organizational Policies and Procedures**

Organizations should also consider their organizational policies and procedures and assess how these may impact the management and progress of the project. General considerations include:

- What organizational channels must be navigated to move the project forward?
- How best can this be achieved?
- What are the approval processes?

Conversely, organizations also need to consider whether the technical approach they choose will conflict with organizational policy. This is an area of enormous importance, especially when adopting or purchasing a queriable Web-based data dissemination system. The specifications of the system must be examined thoroughly to ensure that it meets organizational policy on data presentation standards, confidentiality, security, etc. A thorough requirements definition (refer to Task 2) is necessary to avoid incompatibility with organizational policy and extra costs incurred in retrofitting the system for compliance.

#### **Issues to explore in identifying current organizational needs and constraints include:**

- What statutes and rules govern data collection and dissemination activities?
- Are there written policies and guidelines that need to be reviewed?
- What is the approval process for determining what data are made available and in what formats?
- Who are the internal decision-makers? What departments need to be involved?
- Are there external partners who are involved in decisions?

## **Task 2**

**Define the Project**



## Task 2: Define the Project

---

Whether the organization is developing, adopting, or purchasing a system, the second major task is to define the project. Key components in this section are defining the system requirements and developing a project strategy.

To define system requirements:

- Gather background information and related materials.
- Understand users and their tasks to guide or assess interface design.
- Establish business rules to guide database design and system programming.
- Create a requirements definition.

### Gather Background Information

Whether a system is developed, adopted, or purchased, it is important to gather as much background information and documentation as possible. An organization that adopts a system from another public health organization or purchases a system from a developer must be familiar with the tool's goals, design, development and performance to ensure it is obtaining a tested and sound system that meets its data dissemination needs.

An organization that develops its own system should gather background information on tasks and data needs from its staff, any documented inquiries or requests from the general public, and documentation from other similar systems.

### Understand Users' Needs

Understanding users and their tasks will enable the agency to develop or select an appropriate system. Examples of user tasks might be “Generate reports on disease trends by county” or “Find geographic data on disease outbreaks in my area.” When developing a system, understanding and prioritizing user tasks is the key to good interface design. When adopting or purchasing a system, clearly articulating user tasks will ensure that the organization selects a system that is designed to meet their users' needs.

### Establish Business Rules

Business rules define processes, data, and constraints. Data validation—programming designed to identify errors in data—typically enforces the business rules. For example, a business rule might be, “All users must enter a unique email address to register.” To enforce that rule, the system might be programmed to check that something was entered in the “email” field; that what was entered conforms to rules about email addresses (e.g., username@domain.com); and that the email address entered is not already stored in the database.

## Create a Requirements Definition

Developing a project strategy involves construction of a project plan and system design documentation for the database and Web application. Not all of the detailed steps may be relevant to every project. When a system is adopted or purchased, for example, system documentation might be acquired rather than written. However, some project management steps, such as requirements definition, are necessary to every process to determine whether a system is suitable for adoption or purchase. It is important to go through all steps thoroughly and ensure that the relevant activities are undertaken.

Requirements definition is the first and most important step in establishing a framework that guides the development, adoption, or purchase of a system that meets the true needs of the project. System requirements will reflect the needs of stakeholders and users and they should be analyzed closely so that appropriate technical solutions can be developed.

The project is likely to begin with a core set of requirements that will form the basis for project planning and design and development activity. However, the initial set of requirements is likely to be modified and the changes need to be managed.

At each stage of development, the requirements definition document should be consulted to make sure the project plan reflects the requirements. After the system is deployed, its success will be gauged by the system's ability to meet stated requirements.

This significant undertaking is important whether developing, adopting, or purchasing a system because the definition will inform the criteria used to determine which system to adopt or purchase, or how to develop it.

The following sections describe the key components of the system that should be considered in the requirements definition.

### Step 1: Define the Purpose of the Site

The purpose of the site must be clear from the outset so that the project team can maintain its focus through the various stages of planning and development.

- Develop an overall statement of intent.
- Clarify the purpose of the site to make it easier to develop project goals and eventually measure the effectiveness of the tool.

#### Questions to answer:

- What is the site's main purpose?  
[Make sure this question is answered "narrowly." That is, a useful answer is not, "to make getting data easier," but, "to make it possible for researchers to create and save their own sets of data."]
- What are the site's secondary purposes?

## Step 2: Identify System Users

A system's users are the individuals who will actually use the queryable Web-based data dissemination system to accomplish tasks. System users must be identified before beginning a design or evaluation process. The types of users determine a great deal about the system (e.g., the platform,<sup>13</sup> data required, best way to present the data). Representative users may include technical support staff, administrators, managers, and customers—the entire community of individuals who will use the system or its products (e.g., the general public, public health practitioners, the research community, etc.).

User inquiry, also called user and task analysis, is a collection of methods designed to gather information about how people accomplish tasks. Project teams often make assumptions about users that may or may not be accurate. In order to assess user needs, it is necessary to involve users directly. User and task information can be gathered before design begins or throughout the development process. It can also provide important information for assessing systems for adoption or purchase.

### User and task analysis may yield information about:

- User goals (what they are attempting to accomplish with the task)
- What they do to accomplish their goals
- Characteristics they bring to the task (who they are and where they came from)
- Environment in which they accomplish goals and tasks
- Their skills and experience
- Their needs

Gather information from users directly to allow the project team to create realistic scenarios that reflect real needs.

Steps in user inquiry can include:

- Assemble a team that regularly interacts with system users (if the system already exists), or is involved in the design effort (if it is a new system or Web site).
- Use brainstorming techniques to identify known and potential users.
- List characteristics of individual users and groups of users. Use this information to create task and user characteristic matrices to model the anticipated user community. Field-study and other user and task analysis techniques should either support or refute these models.
- Test the design team's assumptions through interaction with users.

---

<sup>13</sup> The type of hardware and operating system in a computer, such as Macintosh and IBM compatible.

### Questions to answer:

- Who is your target audience? [Choose the typical users and profile each in detail. Include information like: occupation, technical expertise, education level, age range, gender, online frequency, online activities, what kind of equipment they have, and what kind of Internet connection they have. Do as many profiles as you need. Use Web logs from an existing site to get information about your users.]
- What are the typical tasks the users might perform with the tool? [For example, register, log on, search for information on maternal and child health, make a query, look for help, contact the organization. Server log data from an existing system can be used to support or refute some of your task analysis. For example, you can tell which pages were visited, for how long, and how frequently.]

Appendix C contains more detailed information about methods to gather user and task data. (Also see Appendix F, which offers information on easy-to-perform, lower-cost usability testing).

The CDC has identified three main user groups for data dissemination systems maintained by public health agencies, 1) the general public, 2) public health practitioners, and 3) the research community.

### General Public

Members of the general public who use public health agency data dissemination systems are more likely to be legislators, legislative aides, or representatives of the media than residents of the state, county, or town served by the data system. They are more likely to want to access static HTML tables of statistical data or conduct simple queries than they are to use a complex query interface. They will need assistance to construct effective queries.

### When members of the general public use data dissemination tools, they must be able to:

- Find descriptive information on a disease or topic
- Find basic statistics in simple language (e.g., how many people die from a specific disease or cause each year? How many people get the disease each year in a specific area? Does the incidence of the disease differ by race or other factors?)
- Find out which group is affected by a disease or condition
- View community profiles

## Public Health Practitioners

Public health practitioners include state, county and city health officials who need to look up information to do their day-to-day work. Members of this group usually have a bachelor's or a master's degree in public health. They may also have doctoral or medical degrees.

### When public health practitioners use data dissemination tools, they must be able to:

- Run an effective query and produce a table
- Print query results
- Save old queries
- Look at community profiles or create profiles
- Compare county/city information with other counties/cities or state or national statistics

## Research Community

The research community includes individuals at nonprofit, state, and community public health organizations who need specific, detailed information on rates of disease in specific groups.

### When research community members use data dissemination tools, they must be able to:

- Run a sophisticated query (i.e., of six or more variables such as age, race, sex, ethnicity, year, disease/condition)
- Retrieve high-end statistics, including weighted percentages and age-adjusted rates
- Compare results across states or communities
- Print tables or other results of queries
- Save datasets as Microsoft (MS) Excel, comma-delimited, or other data format, or otherwise be able to manipulate the data for reports

## Step 3: Set Goals

Determine the system's goals to help identify which functions are priorities and need to be imported or developed first. Whether developing, adopting, or purchasing a system, add features incrementally to keep the initial work manageable and to allow for testing and consolidation of the system at each stage. Define both short-term (6 months to a year) and long-term (longer than a year) goals.

The type and number of an organization's development goals will determine how they are tracked, measured and prioritized. For example, it might be a long-term organizational goal to move a client-server-based system to the Web. This larger goal would have its own set of composite goals, each with its own measurements and priorities. Immediate priorities within that goal might include gathering user data about the old system or identifying design specifications for the new Web interface.

## Project Goals Should be Measurable

An example of a measurable goal is to specify the number of people who will use the system within a defined period of time. For example, “there will be 200 queries to the data system per month.” This goal could be quantitatively measured by the number of people registered to use the tool, or a measurement obtained from the system database.

### Measurements could include:

- Percentage increase in total unique visitors
- Decrease in the amount of requests for paper copies of publications
- Decrease in the amount of phone time spent finding data for requestors

Quantitative measurements can also be obtained from Web server logs. Server logs are text files that list requests made to a Web server. New products continue to enter and leave the marketplace for this type of software, but current products include WebTrends<sup>14</sup> and WebQA<sup>TM</sup>.<sup>15</sup> Analyzing the logs before adding a new system will provide a benchmark for measuring the success of the system. Check the configuration of the Web site’s servers to determine whether they will need to be reconfigured to gather more or different types of data. The logs can be difficult to read but can be analyzed by commercial software that produces statistics.

Measurements for goal achievement will change based on the goal. Success of a system redesign might be measured by an increased number of registered users or by an increased number of “page views” (“page views” refer to the number of times a Web page is accessed as a whole; this can be measured by log analysis software). The organization should establish methods for measurement and achievable standards for success in meeting its goals. A goals and measurements worksheet is included in Appendix D.

## Step 4: Assess the Technical Environment

An assessment should be made of existing technical expertise and equipment to answer specific questions.

If adopting or purchasing a tool, consider the following:

- Will it work on the current servers and network?
- Will current staff be able to support and maintain it?
- If current staff cannot support it, how will the system be supported?
- How much maintenance, if any, is required?

---

<sup>14</sup> <http://www.netiq.com/WebTrends/>

<sup>15</sup> <http://www.watchfire.com/>

If designing a tool, consider the following:

- Can current staff build it?
- Will current staff be able to support and maintain it?
- Will a consultant be hired to build it? If so, can current staff support it once delivered?
- How much maintenance, if any, is required?

Appendix E contains a checklist of questions that can be used to establish a detailed understanding of the technical environment.

### **Step 5: Establish Data Confidentiality Policies**

Data confidentiality policies will need to be consistent with federal privacy regulations and state legislation pertaining to data privacy and the release of data, which differs from state to state. In addition, it is important to involve data stewards<sup>16</sup> in identifying confidentiality issues relative to the data sets for which they are responsible, and in establishing dissemination policies for those data sets. Information on the Health Insurance Portability and Accountability Act (HIPAA) Data Privacy Rule is available at <http://aspe.hhs.gov/admsimp/index.shtml>. Additional sources for information on data privacy include:

- HHS Privacy Committee (<http://aspe.hhs.gov/datacncl/privcmte.htm#goal>)
- Confidentiality and Data Access Committee of the Federal Committee on Statistical Methodology (<http://www.fcsm.gov/cdac/index.html>)
- CDC National Center for Health Statistics (<http://www.cdc.gov/nchs/>)
- National Association for Public Health Statistics and Information Systems (<http://www.naphsis.org/>)
- National Association of Health Data Organizations (<http://www.nahdo.org/>)

### **Step 6: Establish System Security Policies**

Web site security should be a primary concern when hosting and managing a system that uses the Web to disseminate data. This may include restricting access to:

- Data
- User information and profiles
- Server logs
- Search logs
- Email documentation at the organizational level

Every Web application should have a defined security policy—even if this policy allows unrestricted access by the general public. Develop a security policy that enables the owners of the system and the data to make conscious decisions about the availability of the data and how an attack by hackers would be handled.

---

<sup>16</sup> A trained staff member who ensures that certain procedures regarding the data in the system (typically procedures related to security and confidentiality) are followed appropriately.

Answer the following questions when establishing security policies:

- Should users register in order to receive data?
- Once users are registered, should data be password-protected so that only a unique identification number and password allow access to the records?
- Should users enter their user ID and password for each session?
- How should database security be handled?
- Which firewall software should be used?
- Should different levels of user access be predetermined and the data delivered to them customized based on their level of access?
- Should time-out features<sup>17</sup> be included?
- How should an attack by hackers be handled?
- How should the system be backed-up?

Establishment of security policies and practices should also include a means of auditing and monitoring to ensure that system security is maintained. This could include creating checklists that are followed by technical support and project staff on a regular basis, analyzing backup tapes, or periodically updating virus and hacker protections.

CDC Secure Data Network Standards and Procedures can be accessed for more information at: <http://www.cdc.gov/nedss/Security/index.html>. Also refer to *For the Record: Protecting Electronic Health Information* (<http://www.nap.edu/readingroom/books/for/>).

### **Step 7: Establish Business Rules**

As data confidentiality and security (steps 5 and 6, respectively) are basic requirements to developing, adopting, or purchasing a system, they set the stage for establishing business rules. If a rule affects data confidentiality or security, those issues and the business rule will need to be brought into alignment.

A business rule is a statement that sets data integrity constraints that apply to a field in a database or a relationship between tables in that database. Business rules are based on how an organization views and uses data, which reflect the way the organization functions. The goal of a data integrity constraint is to enforce a business rule within a database or application.

---

<sup>17</sup> Often used as a security device, time-out features assume that an error has occurred in specified situations and shut down a program. For example, a user's failing to use a program for more than 30 minutes might trigger the time-out feature.



Building a Web application to disseminate public health data involves making choices about those data – what types of data will be disseminated, how they will be disseminated, and how they will be reflected in the interface that the user sees. To establish the choices that need to be made during database design, the organization should compose a formal set of business rules. The rules will influence data selection, how relationships between tables in a database are constructed, and what reports the database produces. The rules may also affect security and confidentiality issues.

For example, Public Health Agency A may choose to gather hospitalization data on children using two age categories: 5-9 and 10-14, while Public Health Agency B may simply gather hospitalization data for children using a single age category: 5-14. (The business rule for Public Health Agency A might state, “Hospitalization data will be gathered for the following age ranges: 0-1, 2-4, 5-9, 10-14, etc.”) This type of decision will impose constraints on how each public health agency’s hospitalization data is gathered, stored, and displayed. The reasons for these data decisions would derive from the way the organization functions – Public Health Agency A might track data on smaller, more specific age ranges for funding reasons.

### **Step 8: Develop System Design Document**

After completing the relevant information-gathering activities described in steps 1 through 7, develop a system design document that synthesizes all of the requirements and information into a clear statement comprising the functional and organizational requirements of the system. A system design document should be developed detailing a technical approach that ensures the development, adoption, or purchase of a system that:

- Meets the high-level goals of the organization and the goals of the project
- Provides identified users the ability to perform the required tasks

The system design document should also:

- Identify software design parameters (Web application and business rules and database design)
- Identify software and hardware needs

For agencies that are developing a system, the system design document serves as the guidance for the development process; for agencies that are adopting or purchasing a system, the system design document serves as a tool for identifying systems that meet requirements.

### **Step 9: Develop a Project Strategy**

The project strategy transforms the requirements definition into a plan of action. The project strategy interprets the information contained in the definition and describes how it will shape the dissemination system and the steps needed to complete the project.

The organization should complete a project plan whether it is developing, adopting, or purchasing a system. All of these approaches will affect current technical environments, workflow, and staffing. This should be addressed in the project plan so that the integration of the system into the agency's operations will be smooth.

The project plan should document what the system is expected to do (defined requirements) and how this will be achieved. The plan should include:

- Activities
- Schedule and sequencing
- Critical path
- Milestones
- Resources and budget
- Project staffing and management information

All staff working on the project should agree on the project plan. Progress can then be monitored, any changes to the scope of the project can be factored in, and the impact of the project can be measured. The plan provides a vehicle to communicate with those involved in the pre-planning stages. It can also be used to coordinate changes and as an overall measure of the progress of the project. Keep a copy of the project baseline on paper and/or electronically so you can compare progress.

The project baseline consists of the first project plan's milestone descriptions and dates. With most projects, deadlines shift, requirements change, and unforeseen events occur. It is helpful to keep a copy of the original timeline as a baseline to compare with the actual development progress.

Managing and coordinating these activities is time consuming and management time needs to be factored into the overall cost of the project.

The development of a project plan is relatively straightforward if designing and developing a system. If adoption is the approach, then resource planning must consider the activities that need to be conducted by the donating public health agencies or in conjunction with donating public health agencies. A thorough dialogue on this issue needs to take place between public health agencies that are working together to identify all activities (e.g., customization, training, coordination, system integration), who will conduct them, and when tasks will be performed. Similarly, purchasing organizations will need to clarify with the vendor expectations for milestones and timelines, and specific activities that are included in the purchase agreement (e.g., customization and training).

### **Revisit the Requirements**

Once a project plan has been developed it is useful to review the requirements definition statement. Staff should conduct this review to make sure that the system is still the same as was originally defined and that it can be realistically implemented using the project plan.

### **Task 3**

**Select the Approach**

## Task 3: Select the Approach

---

Engaging in the following steps will help determine whether an agency should develop its own system, or adopt or purchase an existing system.

### Step 1: Determine the Functional Capability of Each Approach

Once the system requirements are defined and a project strategy is developed, it is time to assess the functional capability of the proposed approach to meet requirements. If acquiring an existing system, either through adoption or purchase, it should be assessed to determine:

- Can it meet the primary and secondary purposes defined for the site?
- Is it suitable for our target audience?
- Does it support the user tasks identified?
- Is it compatible with existing platforms?
- Does it meet requirements for confidentiality?
- Is comprehensive documentation available?

### Step 2: Identify Staff Resources Necessary to Implement the Approach

The cost of resources and their availability has been a major concern to organizations that have already developed or adopted Web-based data dissemination tools. They expressed concerns about the difficulty of staffing their initiatives and in recruiting staff with the appropriate technical skills. Organizations should consider their capacity in this area and establish how an implementation project may be supported.

#### Questions to consider to identify staff resources needed:

- What are the short- and long-term staffing requirements for the project and can they be supported?
- What will be the impact on IT support functions and how will project staff interact with the IT department?
- Do staff have the skills to meet programming requirements for the new tool?
- Are staff available in-house to work on the project and can they be diverted from their present tasks?
- Do staff have the appropriate technical skills and depth of technical knowledge?
- Can the organization compete with the private sector on salary and compensation to hire appropriate technical people? If not, how will the right staff be recruited?
- Are there other sources of suitable labor (e.g., university students, interns)?
- Do consultants need to be hired?
- Are sufficient training resources available or will they need to be acquired?
- Can the organization still function effectively and achieve organizational goals, or will this project have an impact on other activities?

An additional consideration is training users in the public health agency and associated field staff to use the system to run queries. It is also important to implement “live” system support to answer questions and provide technical assistance in running queries on an ongoing basis.

If the agency is developing a system, project staff must evaluate the impact of creating training and help mechanisms as well as their associated costs.

If adopting or purchasing a system, the project staff must determine whether the donating/licensing entity will provide training support and technical assistance. If not, the adopting agency will need to be prepared to develop these services. Issues to explore include:

- Does the donating/licensing organization provide initial training and technical assistance?
- Does it provide ongoing technical assistance?
- Is on-line help available?
- Are there fees for training/technical assistance?

### **Step 3: Estimate the Cost of Implementing the Approach**

To identify costs accurately, the following items must be considered:

- **Proprietary Systems**—Assess all of the costs associated with the adoption or purchase of a system (e.g., purchase costs, annual fees, licenses). Conduct thorough research on systems or technologies under consideration and obtain system documentation. Identify any compatibility issues and the costs associated with integration and customization.
- **Hardware**—A new system may put pressure on existing servers, need additional network resources, or require personal computers for all staff. Identify the load on existing hardware including servers, desktops, laptops, and network hardware, and any additional hardware requirements.
- **Software**—Assess the system’s capability with existing applications, databases, proprietary software, network software, operating systems, statistical software, and geographic information system (GIS) software.
- **Security**—Identify any additional costs to ensure system and data security.
- **Licenses**—Consider fees for upgrades, and long-term licensing.
- **Maintenance**—Software, database, network
- **Customization costs**—May be needed, if adopting a system
- **Staffing support**—Costs associated with implementing the selected approach

Once these costs have been defined and estimated, they should be compared with the costs associated with the current level of effort for data dissemination activities. This comparison will help ensure that the new approach is cost-effective, or identify that the approach is possibly not realistic for the organization.

**Step 4: Determine Whether the Selected Approach Will Meet Organizational Needs and Constraints**

The steps taken in Task 1, **Determine Organizational Readiness and Impact**, will have laid the groundwork for determining organizational needs and constraints. At this juncture, revisit the steps in the decision making process to determine if the plan meets the original goals.

## **Task 4**

### **Design Data Presentation Formats**

## Task 4: Design Data Presentation Formats

---

Regardless of how a Web-based queryable data system is brought to an organization, it must consider standards for data presentation. First, public health agencies must ensure that the system adheres to the data presentation standards, including confidentiality, that are in place. Second, they must decide how best to display information on the computer screen.

### Step 1: Maintain Adherence to Data Presentation Standards

As will be discussed in Task 5, the data presentation standards in place must be considered. Prior to launching a system, determine the established rules protecting confidentiality of data and make sure the appropriate personnel know them. If a system is being developed or purchased, ensure the programmers know these standards and program the system accordingly. If a system is being adopted, determine whether the system will need to be modified so that the standards can be followed.

Include documentation of data element standards<sup>18</sup> for users of the system. This documentation should be contained in a separate section that is clearly marked. In addition, it is helpful to include messages in the query output explaining standards when they are enforced and marking these data with special characters. For example, in the case of cell suppression<sup>19</sup> providing a message that explains why the data were not returned will help the user understand that the query was not faulty.

A few common examples of instances where standards for statistical calculation and data presentation should be developed and adhered to, include:

- Cell suppression criteria
- Age adjustment methodology
- Confidence intervals
- Data smoothing methodology
- Geographic units of analysis (relative to a mapping component)
- Methodology for handling unknown or missing data
- Race/ethnicity categories (relative to Census 2000 data)<sup>20</sup>
- ICD coding (relative to ICD-10 conversion)
- Denominators for rate calculation

---

<sup>18</sup> A rationale by which system developers and managers label each category of data that reside in the database; among other things, standards should provide guidance on how to name each data element, and where those elements should reside in the database structure.

<sup>19</sup> A technique that can be used to maintain the confidentiality of data about individuals (persons or establishments). The technique "suppresses" (i.e., does not show) data for one or more cells in a table, when the number of individuals in a cell is small enough (e.g., <5) to possibly allow identification of an individual.

<sup>20</sup> Office of Management and Budget (OMB) October 1997 Changes to the Statistical Policy Directive No. 15, Race and Ethnicity Standards for Federal Statistics and Administrative Reporting. <http://www.whitehouse.gov/omb/fedreg/ombdir15.html>



**Information and guidance on standards for statistical calculation and data presentation can be obtained from:**

- CDC's National Center for Health Statistics (<http://www.cdc.gov/nchs/>)
- The Federal Committee on Statistical Methodology (<http://www.fcs.gov>)
- The National Association for Public Health Statistics and Information Systems (<http://www.naphsis.org>)
- The National Association of Health Data Organizations (<http://www.nahdo.org>)

## **Step 2: Organize Data Presentation**

The presentation of the query options and output on the computer screen are critical features of a user-friendly system. In presenting data, usability principles should be followed. Task 5 describes usability testing procedures for use before, during and after system development. In addition to those procedures, other methods, such as heuristic evaluation<sup>21</sup>, may be used for ensuring system usability. By adhering to basic “rules of thumb” or heuristics, public health agencies can ensure that their system has user-friendly query interfaces and output screens.

Heuristic evaluations are one of the most informal methods for usability inspection in a cost-efficient and timely manner. Two of the best-known lists of usability heuristics were published by Jakob Nielsen and Larry Constantine (see Appendix F). Application of these heuristics to the public health agencies' data dissemination systems yields specific recommendations about the design of the user interfaces, queries and outputs.

Below are general recommendations for 1) Query page layout 2) Results 3) Help and error messages and 4) Color, fonts and images. More specific guidelines are provided in Task 5, **Implement the System, Approach 1: Design and Develop a New System** (see Steps 14, 15, and 16), and **Approach 2: Adopt or Purchase an Existing System** (see steps 7,8, and 9).

---

<sup>21</sup> Evaluation of a Web site that uses best practices as the standard for assessing the site. The best practices usually include consistency, flexibility, efficiency, simplicity, attractive design, online help, etc.

## Query Page Layout

Some best practices vary depending on the target audience of the system:

- For infrequent or inexperienced users, a series of questions or prompts is easy to use and may help prevent many errors. Quick links to static sets of commonly-requested statistics that don't require querying also are useful to this group.
- For expert users, include all variables on a single scrollable screen to allow them quicker access to query options.

For all users:

- Use the same interface design elements for different datasets in the same system. For example, navigation menus should not change from query to query.
- When fewer than four options are available, use checkboxes\* or radio buttons\*\* instead of drop-down\*\*\* lists and multiple select lists. This will make options easier to see and select.
- Place a single "Submit" button close to the form elements,\*\*\*\* toward the bottom of the form. Be consistent with Web conventions, so users know what to expect.
- Do not place other buttons (e.g., Cancel) at the bottom right of the form. Users expect to find the Submit button there and may click another button without reading, creating unexpected results.
- Include consistent navigation links on all query pages so that users do not have to rely on the Back button. However, ensure that use of the Back button will not cause an error.

\* Clicking in this small box on the screen is equivalent to making a checkmark in a box on a paper form.

\*\* An HTML form element, often presented in a list, that allows the user only one selection at a time.

\*\*\* A list from which the user can choose at least one item. The current choice is visible in a small rectangle and when the user clicks on it, a list of items is revealed below it. Also known as pull-down list or menu.

\*\*\*\* Elements in a Web form such as checkboxes, drop-down lists, radio buttons, and text boxes (an area on a Web screen into which the user can type information). Form element tags include <INPUT>, <SELECT>, and <TEXTAREA>.

## Opportunity to Manipulate Results

The more users can manipulate their results, the better. Options may include:

- Sorting within a table
- Saving in alternate formats, e.g., comma- or tab-delimited files
- Saving queries/results for later retrieval
- Choosing presentation format, such as graphs or tables
- Generating maps, allowing zoom in and zoom out
- Selecting colors in map and graph output
- Producing a clean, printable version suitable for a report

Again, include consistent navigation links on all pages so that users don't have to rely on the Back button, and ensure that use of the Back button will not cause an error.

### **Help and Error Messages**

Help should be available at all stages of the query and table generation process (i.e., for both query and results).

- If possible, provide context-specific help with one click.
- Indicate clearly to users that the help shown is context-specific, keyed to where they are in the system.
- Provide a glossary of statistical terms that the general public may not be familiar with.

Prevent errors as much as possible. If an error occurs:

- Clearly explain the error in clear, understandable terms.
- Clearly explain how to resolve the error.

Pop-up dialog boxes\* that describe the error and prompt for corrected information work well.

\* A window that opens on the screen, asking the user to supply information or choose options.

### **Colors, Fonts, and Images**

The display should be as clear, fast loading, and high-contrast as possible.

- Where possible, use relative font sizes so users with impaired vision may enlarge text in their browsers.
- Select foreground and background color combinations to maximize contrast. This will improve readability and printing results.
- Limit number and size of images to decrease page download time.
- Do not use animated GIFs. They are distracting to all users and difficult to view for users with impaired vision and users with some cognitive disabilities.

## **Task 5**

**Implement the System**

## Task 5: Implement the System

---

There are a variety of steps that must be taken when implementing the chosen system. While there is some overlap between implementation steps for designing a system and adopting and/or purchasing a system, important distinctions must be made. Therefore the following section is broken down into two main components:

- Designing and Developing a New System
- Adopting or Purchasing an Existing System

## **Approach 1: Design and Develop a New System**

Once requirements definition has taken place and information gathered has been translated into a project strategy, initial system design and prototyping can begin. During this phase, prototype information architecture is developed and tested, graphical user interface<sup>22</sup> (GUI) prototypes are developed and tested, and finally, the database and Web application are developed to defined coding standards.

In this section, steps 1-3 relate to design, steps 4-16 relate to development.

### **Concept Prototyping and Design**

Concept prototyping should be conducted before any programming begins to produce a more user-centered design. In the prototyping stage, usability pitfalls and problems are anticipated, identified, and fixed inexpensively.

Once the prototypes have been tested and approved, they can be made “production ready,” meaning that the files are ready for the programmers. The programming stage is less problematic if the prototypes have clean, commented,<sup>23</sup> accessible HTML code.<sup>24</sup>

#### **Prototyping typically entails:**

- Conducting user testing with the preliminary prototypes on paper or in HTML, when changes can be easily made.
- Assessing the usability of the system, and testing different approaches.
- Creating a look that reflects the organization and gives the user the best information to complete tasks.
- Reviewing the prototypes for potential accessibility problems.

---

<sup>22</sup> The use of images in addition to words on the screen (e.g., icons, buttons, dialog boxes) to provide a picture-oriented way to interact with the Web site.

<sup>23</sup> Written comments placed by developers in computer code that do not appear on the interface, but are viewable when reading the code itself. Comments are helpful because they help new development project staff read through the code, identify development history, and see context.

<sup>24</sup> Code that meets the Section 508 requirements of the U.S. Rehabilitation Act.

## Design Stages

The basic design process may involve the following:

- Produce an initial set of Web pages called a “wireframe.”<sup>25</sup> The wireframe will demonstrate how the site might be organized (e.g., navigation, browsing screens, search results) and suggest content for the pages. The wireframe should not show actual content, the site’s look-and-feel (e.g., colors, fonts), or a functional back-end<sup>26</sup> (e.g., database). The wireframe will help the project team visualize the possibilities for the Web-based query system interface.
- Produce “graphic composites”<sup>27</sup> (flat electronic images). The composites will illustrate ideas for how the site could look, including colors, fonts, logos, etc.

The design of the graphic user interface and the information architecture will be dictated by the information gained during requirements definition and by input from stakeholders and end-users regarding the aesthetic preferences of the organization. This part of the process is an iterative one and the design of the user interface should be tested with users to ensure optimal usability. Testing and redesign may need to be done a number of times until the most usable design is identified. Thus, the wireframe should be kept as simple as possible, so that it can be developed and changed quickly. The design can then be built out.

Following are the critical steps and elements in designing a user-centered interface and an effective and easily maintainable Web application.

### Step 1: Design Web Pages for Usability

Usability is the measure of the quality of a user’s interaction with an interface.

#### Usability tests frequently answer questions like these:

- Can the users achieve their goals with this site?
- Can they accomplish tasks successfully?
- How long does it take them to complete a task?
- How long did it take them to get there? What steps did they take?
- Do they understand what the site’s capabilities are?
- What do they want to do that they cannot?
- What do they find confusing or frustrating?

<sup>25</sup> A plain-looking graphic that shows a Web page layout (the content, navigation, etc.) It does not show colors or actual images.

<sup>26</sup> In a network application, the software that performs a task not visible to the user. For example, the back-end of the system handles security.

<sup>27</sup> A rapidly drawn but high-quality sketch intended for presentation. Also known as “graphic comp.”

Usability testing is often seen as an expensive undertaking that calls for labs with two-way mirrors and video equipment, carefully screened subject groups, and expert analysis. The perceived expense and difficulty of the testing process often discourages people from conducting any usability tests.

In his article, “Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier,”<sup>28</sup> usability expert Jakob Nielsen argues that a few simple user-testing methods can be used to gather excellent information on the usability of system screens.

**Nielsen recommends a three-tiered approach involving:**

1. Scenarios
2. Simplified “thinking out loud”
3. Heuristic evaluations

Appendix F describes discount usability testing techniques in greater detail.

## **Step 2: Design Web Pages for Accessibility**

Designing for accessibility means designing pages and systems that can be accessed by people with a variety of disabilities. Nearly one-fifth of all Americans have a severe or functional disability. Although people with disabilities can derive great benefit from having easy access to goods and services online, many Web pages are not accessible.

Though a number of automated tools for assessing accessibility have been developed, accessibility evaluations cannot be completely automated. Comprehensive accessibility reviews may incorporate a combination of automated tests, guideline reviews, user tests, browser tests, and reviews with assistive technology. Assistive technology is software or hardware designed to assist people with disabilities in carrying out daily activities. A knowledgeable evaluator should analyze the results.

Public health agencies may have their own accessibility guidelines and requirements. Employees at public health agencies should check with their organization’s IT staff to learn about any state or local requirements that may apply to Web development efforts. If no accessibility guidelines are in place, Section 508 standards or World Wide Web Consortium (W3C) guidelines may be applied. The official Section 508 Web site is <http://www.section508.gov>.

---

<sup>28</sup> [http://www.useit.com/papers/guerrilla\\_hci.html](http://www.useit.com/papers/guerrilla_hci.html)



**Three U.S. laws apply to software and telecommunications accessibility:**

- The Americans with Disabilities Act (ADA) requires that businesses that employ 15 or more people make reasonable accommodations for employees or potential employees with disabilities. Title II of the ADA applies specifically to state and local governments. No Web development guidelines or standards were developed for ADA compliance.
- The Rehabilitation Act Amendments of 1998, Section 508, mandate accessibility for technology purchased by the Federal government and organizations receiving Federal funding. Standards for accessible Web development were published in the Section 508 final rule in 2001. All Federal Web sites must comply with Section 508 standards.
- Section 255 of the Telecommunications Act requires hardware and software manufacturers to develop products that can be used by individuals with disabilities, or that are compatible with assistive technology used by individuals with disabilities.

*Consider accessibility early.* Accessibility issues are much easier to correct if diagnosed early. They are even easier to correct when accessibility is considered during site or system design. During early design meetings, project teams should:

- Involve staff responsible for accessibility evaluations
- Identify the guidelines or standards to be applied, and establish a target conformance level
- Agree on a review schedule
- Distribute training materials on accessible HTML, so developers can implement accessibility correctly and learn to conduct spot reviews of their work

The following Web site has a good list of evaluation software products: <http://www.w3.org/WAI/ER/existingtools.html>. It also lists guidelines for conducting accessibility evaluations: <http://www.w3.org/WAI/eval/>.

Once a site has launched, every content update must comply with accessibility guidelines and achieve the conformance level selected during the design process. To accomplish this, project teams should:

- Articulate the conformance level to be maintained during future updates
- Determine who monitors and updates the site, and train them to test updates for accessibility
- Establish a schedule for follow-up evaluations of the complete site or system

- Purchase or download software to facilitate evaluation and make the software available to whoever maintains the system
- Provide a mechanism for user feedback (e.g., email, email form, mail-to link, or an address)

Refer to Appendix G for more details about assessing accessibility.

### **Step 3: Create Graphic Composites**

To get an idea of how similar tools look, make a list of sites that are appealing. Identify elements that are useful or attractive. Consolidate ideas and give this information to the graphic designer, along with any existing logos, graphics, or other elements that must be part of the queriable system. It may be useful if the designer provides two or three design alternatives.

The sample design is usually a layered, digital file that shows the elements of the page: logos, navigation, titles, example illustrations, and some mock body text. It should provide an idea of the color scheme, the grid (where elements will exist on the screen), the fonts, and the tone or feel of the page. In its finished form, the design will indicate all the information necessary to create the page in HTML.

Once the main look and feel have been agreed upon, the designer should create mockups of the lower-level pages and how the elements change from section to section. For example, a home page may look different from a second-level page, which in turn looks different from a third-level page. There should be a design for each of these showing the differences.

For more detail about design best practices, see Appendix H.

### **Development**

Development should be approached methodically and carefully. Each step is proposed because it ensures the success of the next step. The use of templates,<sup>29</sup> validation, testing, code reviews, file standards, coding standards, and the style guide are part of the process because each has been found to benefit the development process and the final product.

At the end of programming, the system will still be an ongoing project. There will be maintenance, additions, and changes; the care with which it was initially programmed will help or hinder this process. The following steps are recommended for development.

---

<sup>29</sup> The shell structure, coded in HTML, ColdFusion, or another language, into which individual page content is inserted. Templates represent a working model of a system in development. They often include a fully functioning GUI and a partially functioning back-end.

## Step 4: Create User Friendly Interface

The system interface should be clean and easy to follow.

### Consider the following when creating the system's interface:

- Ensure that links and buttons are clearly defined so users know what to expect when clicking on them.
- Make error messages easy to understand so that users know what caused the error, how to fix the error, and how they can prevent such errors in the future.
- If users are required to register for use of the database, explain why they must do so.
- Specify whether or not payment is required for using the system.
- Keep text simple, preferably at a 6th- to 8th-grade reading level.\* If content is aimed at an audience with a specialized vocabulary, be sure to use terms familiar to that audience.
- Provide definitions and help for users.
- Make help links context specific, e.g., the “help” link on a query page should provide information on querying, while the “help” link on an output page should provide information on tasks that can be performed in the output screen.
- Allow users to accomplish tasks with the least number of steps possible.
- Use the correct form element for the questions. For example, if the user can select more than one option in a brief list, use checkboxes. If the list is quite long, consider using a drop-down list or breaking up the list in some way. Try to limit the number of form elements that are present in a single form. For more information on designing forms for usability, see <http://formsthatwork.com>.
- Avoid causing multiple browser windows to open because users are sometimes unaware that they are opened and can be unsure how to close them.
- Show contact information clearly on every page.
- Use fonts that are 11-point or larger. Where possible, use relative font sizes to allow users to control the size of the text in their browser.
- Keep the color scheme uniform throughout the site.
- Select foreground and background colors to achieve high contrast (e.g., don't use light yellow on a white background). This will increase the readability of the site.
- Test the Web site using multiple browsers and browser versions because the appearance of the Web site may change depending on what browser is used to view it. Set a browser compatibility standard and require compliance.

\* Although the users for these systems are generally well educated professionals, content on the Web—due to the great reduction in the resolution of the typeface—is most easily read when it is brief and simply written.

## Step 5: Conduct Image Slicing<sup>30</sup>

Once the designs are finalized, they have to be “sliced” into images that are placed in table cells so that the images look seamless. At the same time, the slicing allows the changing elements to be swapped as the user moves to different sections. After you decide where to cut the images, a software program optimizes the image (compresses the images into as small a file size as possible), slices the image, and creates the HTML slice code to put the slices back together in a table.

## Step 6: Create Templates<sup>31</sup>

The sliced code can then be inserted into a complete HTML file (with no page-specific content), creating a template. The template matches the page design visually as closely as it can. All functionality should be added to the template at this point, including dynamic element codes, META tags<sup>32</sup>, rollovers, forms, etc. User feedback via observation and testing can be incorporated quickly across the site by updating a template. This encourages continual improvement.

### Templates offer many advantages, they:

- Provide tools to page developers to streamline the development process
- Generally include required HTML elements such as META tags, Cascading Style Sheets (CSS), and predefined layout strategies that allow developers to focus on content rather than recurring presentation elements
- Encourage consistency between pages
- Provide consistent tools and strategies in the template to encourage the creation of pages that contain common elements presented in the same way
- Encourage consistent application of the site’s architecture
- Provide a means to centralize site navigation, which discourages developers from extending the site’s architecture without considering its effect on the whole site
- Greatly reduce the cost of iterative design, which improves usability

---

<sup>30</sup> The process by which a Web developer cuts up or crops the graphic composite file into the requisite pieces of the Web page. Each piece is rendered to the user in its proper location within the interface.

<sup>31</sup> The shell structure, coded in HTML, ColdFusion, or another language, into which individual page content is inserted. Templates represent a working model of a system in development. They often include a fully functioning GUI and a partially functioning back-end.

<sup>32</sup> Labels that are included in the code of the system so that industry search engines (e.g., Google) can find the system when users enter related keywords.

## Step 7: Utilize Cascading Style Sheets (CSS)<sup>33</sup>

Cascading style sheets (CSS) can be used for adding style (e.g., fonts, colors, spacing) to Web documents. Image slices, part of the HTML templates, combine with style sheets to form the foundation or shell of the system. The content for each page, which may include text, additional images, or data, is incorporated into this shell. By attaching style sheets to Web documents, developers can change the format without adding new HTML tags<sup>34</sup> Users can override stylistic decisions that prevent them from accessing a site's content by either turning styles off, or substituting their own style sheets.

### Cascading style sheets offer many advantages, they:

- Separate content from presentation
- Provide many benefits to users, such as consistency, usability, and accessibility, while improving the development process
- Support future technologies and diverse users
- Allow the presentation of information in a format appropriate for any device or user with the use of alternate style sheets
- Improve accessibility by encouraging the correct use of HTML elements
- Reduce the need for "tag misuse" to gain control of presentation elements, e.g., using TABLE as a presentation element rather than a structuring element
- Improve usability by rendering pages faster and more consistently
- Reduce the complexity of page code, eliminating tags like FONT and BOLD, nested layout tables, and spacer images. This reduction in presentation code produces pages that are compliant with upcoming HTML standards, and backwards-compatible in browsers that do not support certain features
- Override presentation elements that users have trouble perceiving

## Step 8: Conduct Validation and Test Guidelines

Once the templates look correct, they should be validated and tested according to the accepted guidelines of the project. Refer to the organization's policy for validation and testing guidelines particularly for accessibility.

HTML and CSS validation is fairly simple now because the W3C has posted validators<sup>35</sup> online. HTML documents can be submitted at <http://validator.w3.org/>, and CSS files at <http://jigsaw.w3.org/css-validator/> for immediate feedback. The CSS validator can also be downloaded.

---

<sup>33</sup> Additional, supporting files that tell a browser what a document should look like in terms of colors, headings, etc. Because they are not linked to a specific type of computer or software but are "platform independent," style sheets let developers design pages that can be viewed as intended on all computers.

<sup>34</sup> The means by which programmers create the features or elements of a Web page.

<sup>35</sup> A computer program that examines HTML documents for syntax errors.

## Step 9: Implement Code Review

The main reason for validating and reviewing the queryable Web-based data dissemination tool's code is to be sure your users see what you want them to see on their browser. The two main browsers, Netscape Communicator and Microsoft (MS) Internet Explorer, do not agree on how certain codes appear on their browsers, so Web developers have to write code that works on both. The W3C has guidelines ([www.w3c.org](http://www.w3c.org)) about how to write HTML and CSS code that works on both browsers.

## Step 10: Designate File and Directory Naming Standards

Once the code within the system has been validated and reviewed, attention must be given to organizing the discrete files that comprise the system, including how files are named and grouped for easy maintenance and updating.

The queryable Web-based data dissemination system should use logical and consistent directory structure and file naming conventions to make maintenance easier and to reduce errors. Conventions should conform to existing agency standards.

If the structure is intuitive, maintaining the site will be easier. Keep files neatly organized in multiple subdirectories based on the system's information architecture. Include a subdirectory for images and a subdirectory for dynamic elements,<sup>36</sup> like ColdFusion<sup>37</sup> (CF) labels and queries.

For example, a site divided into sections called About the Organization, News, Statistics, and Links might have a directory structure like this:

- root directory (contains home page and other top-level pages)
  - about/
  - news/
  - stats/
  - links/
  - images/

Within each subdirectory that represents a section, the section's home or default page should be named "index" or another name conventionally accepted as a default. Sections that feature many pages in a series could have files "index," "index2," "index3," etc. Sections that feature interactive tools may benefit from file naming that reflects the function of the page, for example, "email," "validate," etc.

Ensure that the site's physical structure (where the HTML files are, where the images are, how the folders are nested) is based on the information architecture. If the structure is intuitive, maintaining the site will be easier.

---

<sup>36</sup> Elements in a Web page that are updated every time someone requests the page. Examples are the results of a search, or a button that reads Login or Logout, depending on whether the user has already logged in or not.

<sup>37</sup> Software used to develop database applications.

## Step 11: Determine File Naming Conventions

Naming conventions can contribute greatly to the readability and maintainability of code. These guidelines will help you choose meaningful names for your code elements.

- Set a project standard for naming files, variables, and objects (e.g., project name\_requirements.doc).
- Use variable names that clearly indicate the content of each variable (e.g., “first\_name”).
- Use familiar vocabulary (e.g., use “last\_name” rather than “surname”).
- Match form field names and variable names to the corresponding database field name.
- Use naming conventions that conform to the syntax of the programming language.
- Use lowercase file names in 8.3 format (i.e., eight characters for the name and three for the appropriate extension).

Although many platforms now allow for long, case-insensitive file names, programmers should strive to maintain the lowercase, 8.3 format for file names. There are two primary reasons for this approach. First, the 8.3 format can be used on all of the platforms currently in use and ensures that the file name will be readable. Second, in certain cases, Windows has demonstrated difficulties with restoring long file names from back up. Therefore, by using long file names, the programmer is introducing instability into procedures to back up the application.

There may be circumstances in which the programmer cannot use this 8.3 format due to constraints of the specific programming language or the server environment. For example, the Macromedia ColdFusion application architecture requires the existence of a file named Application.cfm. The uppercase “A” is required if the application is deployed on a UNIX server. Exceptions to this standard should be based on compelling language- or platform-specific requirements.

Characters that should **not** be used in file names under any circumstances include:

Spaces

[/]

[\\]

[\*]

[:]

[.]

## Step 12: Implement Coding Standards

Using comments<sup>38</sup> in your code provides valuable information, both to yourself and to other programmers who may be unfamiliar with your code. For the best coding, follow these guidelines:

- Include standard information (name, date, file name, purpose, update).
- Break up sections of the code using comments.
- Comment on exceptions to the standards.

Someone unfamiliar with the code needs to know who worked on it last and when, to determine whether that person is still available to provide guidance.

Following is a sample comment that could be inserted before a template:

```
<!-- CalcBen.tem: Calculate and display Life Benefit given user's
pre-entered data.
    Created by: rroe@qrc.com, November 2000.
    Libraries referenced: Mathrout.tlb, Header.tlb,
System.tlb
    Modification History:
    December 2001: Fixed bug which caused Javascript execution error
in display table using IE 5.0 - rroe
-->
```

Too often the comments at the top of a file remain unchanged, despite major revisions to the code. It is best to note all significant modifications at the top: then the next programmer does not have to search through the code. If applicable, include the version number and notes about the history of the file.

Divisions should be clear to the reader so that the different sections of the code can be identified easily. Use asterisks or a full line of a repeated symbol to differentiate sections. Some people also find it helpful to always begin their comments flush left, to provide a visual contrast to the indented code around them.

---

<sup>38</sup> Written comments placed by developers in computer code that do not appear on the interface, but are viewable when reading the code itself. Comments are helpful because they help new development project staff read through the code, identify development history, and see context.



### **Step 13: Engage in HTML Production**

Production starts after the templates have been validated and tested. That means formatting the content, placing it into the templates, and saving it as new files. In production it is important for the HTML coders to use the original templates and a style guide to make sure the pages are consistent across the site.

### **Step 14: Create Style Guide**

The style guide defines how each recurring element should be formatted and used. The guide should explain which fonts to use; how titles, headings, subheadings, footnotes, and captions should look; how to prepare pictures; and where elements go. It should show everything that a graphics standards manual would show for a magazine, for example. In addition, it should detail how to code elements, what ALT tags<sup>39</sup> to use for images, what comments to use in certain sections, and how to use the cascading style sheets. Using a guide will make production quicker and more consistent. After production, the guide will be used for creating new pages.

For production and maintenance, the guide should include:

- How to modify and use the templates
- How to add or modify graphics; where the original graphics files are located
- How the Web server is set up and where the files are located
- Naming conventions for:
  - CSS files and codes
  - Files and folders
  - Images
- HTML code conventions
- Other coding conventions as needed (e.g., Perl,<sup>40</sup> Common Gateway Interface<sup>41</sup> (CGI) scripts, ColdFusion includes<sup>42</sup>)

---

<sup>39</sup> Used in HTML coding to describe in words any graphic images on a Web page. Assistive technology for persons with disabilities like JAWS for Windows or PowerBraille can read these described images out loud.

<sup>40</sup> A programming language that is popular among Web developers, especially for sites that involve the processing of forms.

<sup>41</sup> A category of interfaces that presents the system's features and functions graphically rather than with a simple prompt. It is the standard for running programs from a Web server.

<sup>42</sup> A simple Web server technology that allows one to include, or insert, a second file into an existing file; using includes makes updates and changes to a template very easy.

## Step 15: Create User Friendly Query Functions

The system's query functions should allow for data searches to be conducted with ease.

### Consider the following when creating the system's query functions:

- Create different query interfaces for users with different levels of experience—for example, one for beginning users who want a simple query, and one for advanced users who want to conduct complex queries.
- By default, have users start with a simple query interface. Make it easy for more expert users to skip over the beginner interface.
- Write the accompanying text and instructions for simple queries at a 6th- to 8th-grade reading level. For all queries, write text and instructions in the simplest language possible.
- Use tables rather than body text to allow users to construct queries quickly. Users will more easily select items from a bulleted list than a paragraph to build a query.
- Describe the content of the datasets, so that users can determine if the data are relevant to their query.
- Clearly state disclaimers and limitations of the data, either on the query page or on a link from the query page.
- Allow users flexibility in selecting age range and year variables.
- Allow users to query death by name of primary diagnosis as well as by ICD-9 or ICD-10 code.
- When possible, allow users flexibility in defining areas of geographic analysis (e.g., building neighborhoods by aggregating zip codes).
- Identify which variables cannot be used in combination because they will produce illogical or circular queries, will result in person-identifiable results, or are not supported by the system.
- If it does not violate confidentiality standards, allow users to download acceptable data sets to perform more sophisticated analyses.
- Provide a clear way to submit data by way of a submit button, in an easy to find location. For long pages, a submit button at the top and bottom of the screen is acceptable, but both should perform exactly the same function and be identifiable as such.
- Provide a query progress bar during processing of the query, so that users can see that the system is actively generating results from the query. This is especially important for large datasets, when the query may take some time to process. This information will only be provided while the system is building the results screen, so easier-to-build results will mean the information is more quickly presented and departed from the screen.
- Maximize system performance to increase speed of queries. Usability expert Jakob Nielsen recommends that pages take no longer than 10 seconds to load, based on the most prevalent connection speeds of system users (e.g., cable modem, T-1 line, or dial-up modem).\*

\* Jakob Nielsen. *Designing Web Usability*. New Riders Publishing, 2000; p. 44.

## Step 16: Design Usable Outputs

The system's output functions should provide users with data that are clearly distinguishable and well organized. The output functions should also be flexible to meet users data reporting needs.

### Consider the following when designing the system's output functions:

- Identify the query variables on the output pages.
- Use colors or other mechanisms such as size or position to differentiate between the output and the query variables.
- Allow users to select multiple output formats simultaneously (e.g., allowing users to create a table and a graph from the same query).
- Allow users to easily manipulate output after the query is complete (e.g., changing colors or labels on graphs or otherwise modifying the appearance of graphs, charts, tables, and maps).
- Allow users to further query their outputs (e.g., creating nested tables).
- Ensure that outputs can be downloaded and printed with ease and integrity.
- Allow outputs to be downloaded in multiple formats and clearly define those formats.
- Provide surrounding language, interpretation of the output, and instructions for additional steps and options (e.g., printing and downloading) on the output page.

## **Approach 2: Adopt or Purchase an Existing System**

Once requirements have been defined, and a project strategy developed, the adoption or purchase process can begin.

Many of the steps for purchasing a system are the same as for adopting one. However, purchasing a system was not explored in depth during the earlier phases of the research for these guidelines. Agencies that wish to purchase a system may refer to the steps in the adoption process, but are also encouraged to conduct additional research.

Adoption or purchase should be approached methodically and carefully. Public health agencies acquiring an existing system should work carefully with the donating agency or vendor to ensure clear transfer of knowledge and history of the system. The success of the technology transfer may depend on clear communication about a variety of issues.

### **Step 1: Organize a Team to Facilitate the Adoption or Purchase**

While it may benefit the project if a single “champion” promotes the adoption or purchase of the system, having a diverse team of dedicated staff members will lead to an easier transfer of the technology to the adopting agency. Include members on the team who are stakeholders and end-users, as well as those who will serve as managers, programmers, data analysts, information technology specialists, and public health scientists (e.g., epidemiologists, statisticians, etc.). As described in Task 1, it is particularly important to enlist the participation of information technology staff as early in the process as possible because it will be essential to obtain their support for the project’s success. This team may receive hands-on training, or communicate electronically and/or via telephone. The team should establish mechanisms for providing feedback and asking for technical assistance for the future.

### **Step 2: Assess the System**

One of the team’s first assignments is to assess the system to be adopted or purchased and gain an in-depth understanding of the current technology used by the vendor or donating agency. This assessment will highlight compatibility issues and reveal the components and structures required to adopt or purchase the system. It is also recommended that the assessment include a review of the vendor’s or public health agency’s findings from its experience with users of its system, because this may point out similarities or differences in the audience or in the functionality that the application supports. Adopting agencies must investigate the following critical aspects of the system being donated or purchased before proceeding with the technology transfer or purchase:

- Organization and architecture
- Server hardware and operating system
- Web software and server software used with the system
- Connectivity of the server housing the system
- Dynamic content of the system
- Data requirements

### **Questions to Answer When Assessing the Organization and Architecture:**

- How many servers are involved? What functions – applications, databases, Web, etc. – do those servers support?
- What are the Internet Protocol (IP)\* address(es) of the server(s) on which queriable data reside?
- Where are queriable data housed? (On the Web server or on a separate server?)
- How are the data accessed by the system? Are the data housed in the program that provides them, or are data housed elsewhere in such a way that the program points or links to them?
- How many datasets are there?
- Who is primarily responsible for determining what content should be added to (or deleted from) this system?
- Who is responsible for Web server hardware and/or operating system software?
- Who is responsible for Web server software?
- Who is responsible for networking and connectivity to the Web server?
- Who is responsible for publishing dynamic Web content (such as multimedia, Common Gateway Interface (CGI)\*\* -scripted, database- or program-generated pages)?
- Who is primarily responsible for providing scripting or programming support for Web-based queriable data?
- Does the queriable system meet Federal requirements for accessibility? If not, does the system use a technology or design that precludes it from being modified to meet the requirements?

\* The method for sending data from one computer to another on the Internet. IP also designates data by location, as in IP address.

\*\* A category of interfaces that presents the system's features and functions graphically rather than with a simple prompt. It is the standard for running programs from a Web server.

### **Questions to Answer When Assessing the Hardware and Operating System:**

- What is the platform/operating system that runs the database server?
- How much RAM is installed on the server?
- What is the hardware architecture of the server?
- How much disk space is installed on the server?
- How often is the server backed up?
- If backups are made, what is the primary backup medium?
- How many staff members, either full-time or part-time, support the hardware and operating system for this server?
- Approximately how many hours per month do all of these staff members spend supporting the hardware and operating system for this server?
- What physical security options are in place for the server (e.g., is the server in a locked room)?
- What security options are in place for data on the server (e.g., is a firewall in place)?
- What are the benefits of the hardware used?
- What are the limitations of the hardware used?

### **Questions to answer when assessing the Web and server software:**

- For each Web server on which queriable data reside, what Web server software is used?
- What database servers (e.g., a state's health department database) interact with or are installed on each Web server on which queriable data reside?
- What application environments are supported on this Web server?
- Does this Web server support secure data transmission (via HTTPS, IPSEC, or other secure protocols)?
- How many staff hours are spent each month in support of the Web server software?
- What support/security systems are available for the queriable data system(s)?
- What are the benefits of the software used?
- What are the limitations of the software used?
- What analytic software do you use to support your queries?—Web and server software
- Does the system accommodate disability requirements?—Web and server software

### **Questions to answer when assessing server connectivity:**

- How is each server connected to the LAN?
- How is the server (or the entire LAN) connected to the Internet?
- Do any firewalls protect the server (or the entire LAN)?
- Is your connection reliable (i.e., do you experience any interruptions in service)? What problems do you have maintaining connectivity?

### **Questions to answer when assessing dynamic content:**

- What database structures are supported by the application?—dynamic content
- How many staff members, full-time or part-time, publish content for this Web site?
- Approximately how many hours per month do these staff members spend working on publishing the content for this Web site?
- Is this Web site validated or tested to ensure that it is accessible to all users?
- What standards for accessible Web design are followed when designing or implementing this Web site?
- Which technologies does this Web site use (SSIs, DHTML, XML, Other)?
- What types of dynamic server-side applications are used in this Web site?
- If CGI is used in this Web site, in what languages are CGI programs written?
- What types of multimedia content are used in this Web site, if any?
- What tools are used to create this dynamic content?
- Are META tags\* added to the (static or dynamically generated) HTML pages of this Web site?
- What types of META tags are added to the HTML pages of this Web site?
- What browsers are recommended for optimum viewing of this Web site?
- What browser plug-ins are recommended for optimum viewing of this Web site?

\* Labels that are included in the code of the system so that industry search engines (e.g., Google) can find the system when users enter related keyword.

**Questions to answer when assessing data requirements:**

- How are data validated to ensure that correct information is returned to the user?
- Are screens in place to prevent certain runs? How do these screens operate?
- How much Web server space\* is needed for the program?
- How much Web server space is needed for the database?
- How do the data requirements benefit the system's interface or functionality?
- How do the data requirements limit the system's interface or functionality?

\* One judges Web server space by taking the average size of a page in the system, multiplying that by the number of anticipated total pages, and then adding roughly 15–20 percent for growth.

Please refer to Appendix I to review a model discussion guide for assessing queryable systems.

**Step 3: Determine System Compatibility**

It is important for the donating/licensing agency/commercial developer and the adopting agency to determine the compatibility of the queryable system with the adopting agency's information technology systems. It is critical for an information technology staff member to be involved in the donation discussions.

**Step 4: Establish Data Presentation Standards**

Establishing, data confidentiality policies and system security have been mentioned in Task 2, Define the Project, and establishing statistical data presentation standards has been mentioned in Task 4, because these steps apply whether the agency plans to design and develop a new system (Approach 1) or to adopt or purchase an existing system (Approach 2).

The additional concerns about statistical data presentation standards and data confidentiality policies mentioned in Approach 2 are important because adopting agencies need to evaluate the compatibility of the organizational standards and requirements and product specifications.

Adopting agencies will need to take the guidelines for data standards created in Tasks 2 and 4 compare these with the specifications for any product considered for adoption or purchase. For example, a very common data standard is for cell suppression. Although each donating agency and commercial developer has a different standard, there is a universal recognition of the need for cell suppression standards. Other standards in place for some systems include standards for how statistics are calculated, how fields and data elements are included in the system, and for data quality standards.

**Examples of instances where standards for statistical calculation and data presentation should be developed and adhered to, include:**

- Cell suppression criteria
- Age adjustment methodology
- Confidence intervals
- Data smoothing methodology
- Geographic units of analysis (relative to a mapping component)
- Methodology for handling unknown or missing data
- Race/ethnicity categories (relative to Census 2000 data)\*
- ICD coding (relative to ICD-10 conversion)
- Denominators for rate calculation

\* Office of Management and Budget (OMB) October 1997 changes to the Statistical Policy Directive No. 15, Race and Ethnicity Standards for Federal Statistics and Administrative Reporting.

Additional information and guidance on standards for statistical calculation and data presentation can be obtained from the following organizations:

- CDC's National Center for Health Statistics (<http://www.cdc.gov/nchs/>)
- The Federal Committee on Statistical Methodology (<http://www.fcsm.gov>)
- The National Association for Public Health Statistics and Information Systems (<http://www.naphsis.org>)
- The National Association of Health Data Organizations (<http://www.nahdo.org>)

### **Step 5: Review Confidentiality Constraints**

Donating agencies and commercial developers have some confidentiality measures in place. These measures often range from cell suppression to procedures (e.g., restricted access to data), software (e.g., firewalls), and hardware security (e.g., restricted access to hardware). These should be reviewed for consistency with the confidentiality policies established in Task 2.

### **Step 6: Consider Data Availability**

It is critical for the adopting agency to decide which data to make available. User feedback can be examined to determine the data that are most often requested.

### **Step 7: Modify User Interface as Necessary**

When designing and developing a new system (Approach 1), usability issues are the first to be addressed. However, when adopting or purchasing a system (Approach 2), usability concerns are addressed after issues related to user interface, query function, and output modification for adopting agencies, because potential products will already have a user interface. Products should conform to general usability principles; therefore, adopting agencies should focus first on the ability of any system or product to support user tasks and data requirements. The system interface should be clean and easy to follow.



**Consider the following when reviewing and modifying the system's interface:**

- Ensure that links and buttons are clearly defined so users know what to expect when clicking on them.
- Make error messages easy to understand so that users know what caused the error, how to fix the error, and how they can prevent making such errors in the future.
- If users are required to register for use of the database, explain why they must do so.
- Specify whether or not payment is required for using the system.
- Keep text simple, preferably at a 6th- to 8th-grade reading level.\* If content is aimed at an audience with a specialized vocabulary, be sure to use terms familiar to that audience.
- Provide definitions and help for users.
- Make help links context specific, e.g., the “help” link on a query page should provide information on querying, while the “help” link on an output page should provide information on tasks that can be performed in the output screen.
- Allow users to accomplish tasks with the least number of steps possible.
- Use the correct form element for the questions. For example, if the user can select more than one option in a brief list, use checkboxes. If the list is quite long, consider using a drop-down list or breaking up the list in some way. Try to limit the number of form elements that are present in a single form. For more information on designing forms for usability, see <http://formsthatwork.com>.
- Avoid causing multiple browser windows to open because users are sometimes unaware that they are opened and can be unsure how to close them.
- Show contact information clearly on every page.
- Use fonts that are 11-point or larger. Where possible, use relative font sizes to allow users to control the size of the text in their browser.
- Keep the color scheme uniform throughout the site.
- Select foreground and background colors to achieve high contrast (e.g., don't use light yellow on a white background). This will increase the readability of the site.
- Test the Web site using multiple browsers and browser versions because the appearance of the Web site may change depending on what browser is used to view it.

\* Although the users for these systems are generally well educated professionals, content on the Web—due to the great reduction in the resolution of the typeface—is most easily read when it is brief and simply written.

## Step 8: Modify Query Functions as Necessary

The system's query functions should allow for data searches to be conducted with ease.

### Consider the following when reviewing and modifying the system's query functions:

- Create different query interfaces for users with different levels of experience—for example, one for beginning users who want a simple query, and one for advanced users who want to conduct complex queries.
- By default, have users start with a simple query interface. Make it easy for more expert users to skip over the beginner interface.
- Write the accompanying text and instructions for simple queries at a 6th- to 8th-grade reading level.\* For all queries, write text and instructions in the simplest language possible.
- Use tables rather than body text to allow users to construct queries quickly. Users will more easily select items from a bulleted list than a paragraph to build a query.
- Describe the content of the datasets, so that users can determine if the data are relevant to their query.
- Clearly state disclaimers and limitations of the data, either on the query page or on a link from the query page.

\* Although the users for these systems are generally well educated professionals, content on the Web—due to the great reduction in the resolution of the typeface—is most easily read when it is brief and simply written.

## Step 9: Modify Outputs as Necessary

The system's output functions should provide users with data that are clearly distinguishable and well organized. The output functions should also be flexible to meet users' data reporting needs.

### Consider the following when modifying the system's output functions:

- Identify the query variables on the output pages.
- Use colors and other mechanisms to differentiate between the output and the query variables.
- Allow users to select multiple output formats simultaneously (e.g., allowing users to create a table and a graph from the same query).
- Allow users to easily manipulate output after the query is complete (e.g., changing colors or labels on graphs or otherwise modifying the appearance of graphs, charts, tables, and maps).
- Allow users to further query their outputs (e.g., creating nested tables).
- Ensure that outputs can be downloaded and printed with ease and integrity.
- Allow outputs to be downloaded in multiple formats and clearly define those formats.
- Provide surrounding language, interpretation of the output, and instructions for additional steps and options (e.g., printing and downloading) on the output page.

## Step 10: Concept Prototyping and Design

Concept prototyping should be conducted before any programming begins to produce a more user-centered design. In the prototyping stage, usability pitfalls and problems are anticipated, identified, and fixed inexpensively.

### Prototyping typically entails:

- Conducting user testing with the preliminary prototypes on paper or HTML, when changes can be easily made.
- Assessing the usability of the system, and testing different approaches.
- Creating a look that reflects the organization and gives the user the best information to complete tasks.
- Reviewing the prototypes for potential accessibility problems.

Once the prototypes have been tested and approved, they can be made “production ready,” meaning that the files are ready for the programmers. The programming stage is less problematic if the prototypes have clean, commented,<sup>43</sup> accessible HTML code.<sup>44</sup>

## Step 11: Modify Web Pages for Usability

Usability is the measure of the quality of a user’s interaction with an interface.

### Usability tests frequently answer questions like these:

- Can the users achieve their goals with this site?
- Can they accomplish tasks successfully?
- How long does it take them to complete a task?
- How long did it take them to get there? What steps did they take?
- Do they understand what the site’s capabilities are?
- What do they want to do that they cannot?
- What do they find confusing or frustrating?

Usability testing is often seen as an expensive undertaking that calls for labs with two-way mirrors and video equipment, carefully screened subject groups, and expert analysis. The perceived expense and difficulty of the testing process often discourages people from conducting any usability tests.

---

<sup>43</sup> Written comments placed by developers in computer code that do not appear on the interface, but are viewable when reading the code itself. Comments are helpful because they help new development project staff read through the code, identify development history, and see context.

<sup>44</sup> Code that meets the Section 508 requirements of the U.S. Rehabilitation Act.

In his article, “Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier,” usability expert Jakob Nielsen argues that a few simple user-testing methods can be used to gather excellent information on the usability of system screens. (See Appendix F for more detail on discount usability testing.)

**Nielsen recommends a three-tiered approach involving:**

1. Scenarios
2. Simplified “thinking out loud”
3. Heuristic evaluations

## **Step 12: Modify Web Pages for Accessibility**

Web pages should be accessible by people with a variety of disabilities. Nearly one-fifth of all Americans have a severe or functional disability. Although people with disabilities can derive great benefit from having easy access to goods and services online, many Web pages are not accessible.

Though a number of automated tools for assessing accessibility have been developed, accessibility evaluations cannot be completely automated. Comprehensive accessibility reviews may incorporate a combination of automated tests, guideline reviews, user tests, browser tests, and reviews with assistive technology. Assistive technology is software or hardware designed to assist people with disabilities in carrying out daily activities. A knowledgeable evaluator should analyze the results.

Public health agencies may have their own accessibility guidelines and requirements. Employees at public health agencies should check with their organization’s IT staff to learn about any state or local requirements that may apply to system adoption or purchase efforts. If no accessibility guidelines are in place, Section 508 standards or W3C guidelines may be applied. The official Section 508 Web site is <http://www.section508.gov>.

**Three U.S. laws apply to software and telecommunications accessibility:**

- The Americans with Disabilities Act (ADA) requires that businesses that employ 15 or more people make reasonable accommodations for employees or potential employees with disabilities. Title II of the ADA applies specifically to state and local governments. No Web development guidelines or standards were developed for ADA compliance.
- The Rehabilitation Act Amendments of 1998, Section 508, mandate accessibility for technology purchased by the Federal government and organizations receiving Federal funding. Standards for accessible Web development were published in the Section 508 final rule in 2001. All Federal Web sites must comply with Section 508 standards.
- Section 255 of the Telecommunications Act requires hardware and software manufacturers to develop products that can be used by individuals with disabilities, or that are compatible with assistive technology used by individuals with disabilities.

*Consider accessibility early.* Accessibility issues are much easier to correct if diagnosed early. They are even easier to correct when accessibility is considered during initial site review or system design. During early adoption or purchase meetings, project teams should:

- Involve staff responsible for accessibility evaluations
- Identify the guidelines or standards to be applied, and establish a target conformance level
- Agree on a review schedule
- Distribute training materials on accessible HTML, so developers can implement accessibility correctly and learn to conduct spot reviews of their work

The following Web site has a good list of evaluation software products: <http://www.w3.org/WAI/ER/existingtools.html>. It also lists guidelines for conducting accessibility evaluations: <http://www.w3.org/WAI/eval/>.

Once a site has launched, every content update must comply with accessibility guidelines and achieve the conformance level selected during the design process. To accomplish this, project teams should:

- Articulate the conformance level to be maintained during future updates
- Determine who monitors and updates the site, and train them to test updates for accessibility
- Establish a schedule for follow-up evaluations of the complete site or system
- Purchase or download software to facilitate evaluation and make the software available to whoever maintains the system
- Provide a mechanism for user feedback (e.g., email, email form, mail-to link, or an address)

The donating public health agency or vendor should be able to provide information about the accessibility of the tool. An assessment should be conducted to establish if a system conforms to the adopting public health agency's policies on the implementation of accessibility standards. (See Appendix G for more details on assessing accessibility.)

### Step 13: Review and Update the Style Guide

The style guide defines how each recurring element should be formatted and used. The guide should explain which fonts to use; how titles, headings, subheadings, footnotes, and captions should look; how to prepare pictures; and where elements go. It should show everything that a graphics standards manual would show for a magazine, for example. In addition, it should detail how to code elements, what ALT tags<sup>45</sup> to use for images, what comments to use in certain sections, and how to use the cascading style sheets.<sup>46</sup> Using a guide will make production quicker and more consistent. After adoption, the guide will be used for creating new pages.

For production and maintenance, the guide should include:

- How to modify and use the templates<sup>47</sup>
- How to add or modify graphics; where the original graphics files are located
- How the Web server is set up and where the files are located
- Naming conventions for:
  - CSS files and codes
  - Files and folders
  - Images
- HTML code conventions
- Other coding conventions as needed (e.g., Perl, CGI scripts, ColdFusion includes<sup>48</sup>)

When adopting a tool, make sure the developers have the style guide to customize it.

---

<sup>45</sup> Used in HTML coding to describe in words any graphic images on a Web page. Assistive technology for persons with disabilities like JAWS for Windows or PowerBraille can read these described images out loud.

<sup>46</sup> Additional, supporting files that tell a browser what a document should look like in terms of colors, headings, etc. Because they are not linked to a specific type of computer or software but are “platform independent,” style sheets let developers design pages that can be viewed as intended on all computers.

<sup>47</sup> The shell structure, coded in HTML, ColdFusion, or another language, into which individual page content is inserted. Templates represent a working model of a system in development. They often include a fully functioning GUI and a partially functioning back-end.

<sup>48</sup> A simple Web server technology that allows one to include, or insert, a second file into an existing file; using includes makes updates and changes to a template very easy.

## **Task 6**

**Test the System**

## Task 6: Test the System

---

Testing the system is a necessary step in the system development, adoption, or purchase process. Testing and quality assurance focus on functionality of the site, quality and accuracy of query output, and accessibility.

The testing plan should have been considered in the project plan, but it should be reviewed before starting the formal test.

### Step 1: Test the System

Testing helps programmers fix problems before the system has real users and data. Any system, no matter how well coded and standardized, will have “bugs” that only testing can catch. Testing conducted by a designated system tester, will find problems that programmers and others familiar with the system might overlook. Testing should be done across the common user environments defined in the requirements definition, in the following order:

- System testing
  - In defined user environments (browsers, platforms)
  - With live data
  - With scenarios
- Content check
- Fixing testing failures
- Test fixes

### Step 2: Publicize the Launch Date

When the system and site are ready to go live, let the users know. Tell them when it will be live and where it is, and solicit their feedback. Send out email, publish notices, and post notices on Web sites that users will be likely to see. If applicable, inform the public relations department. Also enlist the help of stakeholders and end-users to publicize the launch date.

Ensure that information about the new system is added to your existing site’s META information<sup>49</sup> (key words, phrases, description paragraphs). Then re-register with the search engines that refer users to the site. Referring search engine statistics can be found through the Web logs.

---

<sup>49</sup> Labels that are included in the code of the system so that industry search engines (e.g., Google) can find the system when users enter related keywords.



### **Step 3: Promote Training on the System**

The need for training should be addressed whether developing, adopting, or purchasing a system; implications for each approach are described below.

***Developing a System.*** A large part of developing a system is developing training and documentation. A Web content specialist or technical writer should walk through the system with the interface designer and programmers to write the documentation. Writing the “Help” section can begin as soon as the overall functionality is planned and some pieces have been programmed. As each section is finished, the writer should add to the online help system or printed manual, and the entire documentation should be reviewed after testing and deployment, to ensure that it is consistent with the final system.

***Adopting a System.*** One of the benefits of adopting a system is that a similar organization already has experience with it; the best training might be from current users. It might be possible to arrange for staff from the donating organization to come on-site to provide training and technical assistance. Another alternative would be to have staff from the adopting organization observe system users. The system builders may also have documentation that would be helpful for training purposes. At a minimum, they should be able to recount their own “lessons learned” in using the system.

***Purchasing a System.*** Frequently software designers and vendors offer training on their software. Most programs also have online help and other documentation. The vendors might include on-site training for a fee or include it with the purchase. This should be a factor in deciding which system to buy.

Beyond initial training, developing a plan for training that reaches local health departments and other users statewide is essential. It is important that the training be provided on an ongoing basis, particularly considering staff turnover. Some states have addressed the need for training through “train the trainer” programs so that local or regional trainers are available. Also enlist the help of stakeholders and end-users to promote training on the system. In addition to training on the use of the Web-based data dissemination system, training for local public health users should also address proper use and interpretation of health statistics.

## **Task 7**

**Evaluate the System**

## Task 7: Evaluate the System

---

Evaluation of the system applies to developers, adopters, and purchasers and should encompass a series of elements/phases:

- Usability/user testing
- Accessibility and assistive technology testing
- Validation and browser testing

Additionally, stakeholders and end-users should be involved in the evaluation process, including preparing recommendations as a result of the evaluation.

### Developing a System

The process of evaluation will depend on budget, time limits, organizational support, whether or not a contractor is involved and the contractor budget, personnel, and a variety of other possible issues. Evaluation should be conducted during development, to establish that the design meets user needs and does not break down in required environments. (In other words, once the operating system and Web server environments have been decided, the system must function normally within them.) Standards for success need to be established prior to testing, preferably before the development process is initiated.

#### Questions to answer:

- Are accessibility tests being conducted using federal standards or W3C guidelines? Does the organization have a separate set of standards or guidelines in place?
- If a contractor is involved, is the contractor aware of state or local requirements for Web content? Is the contractor experienced and competent enough to design and evaluate its work in order to meet those requirements?
- What are the goals for site visits and user satisfaction?
- Are data about users being collected prior to design, and if so, how?

Whoever manages development should establish a realistic timeline and budget for specific types of evaluation. Both accessibility testing and usability testing should be conducted early and often during the development process (the process of gathering user data is part of the usability engineering lifecycle). Browser testing and validation should occur after development, but before final accessibility compliance audit (which should be conducted just prior to launch). It is important to educate staff that evaluation occurs throughout the project and not only at the end of the process.

### Contracting versus In-House Evaluation

If a contractor has been used to build the Web system the contractor might conduct its own set of evaluations. However, unless the state or local health agency asks for user testing or accessibility reviews, they are probably not going to happen. Usability and accessibility testing still are not standard steps in most development processes. An extra fee may be charged for evaluation services, and it is one more area of expertise to assess before hiring a contractor.

If the project manager wants to handle evaluation externally, he or she should know which evaluation methods are desired (accessibility, user analysis and testing, log analysis, browser testing/validation, etc.) and select contractors who have that expertise. Samples of the contractor's work should be obtained to assess whether the contractor has the necessary experience and expertise to evaluate a Web system.

The organization may also have a contractor design and build the system, but choose to conduct certain types of tests internally. In that situation, the project manager will need to know when during the development process evaluation should occur, and coordinate that with the contractor. The benefit of internal usability testing is greater control within the organization. Most health agencies also have a ready source of test subjects in their project staff. Internal user testing requires organizational support and funding, which can sometimes be a barrier.

Most Web contractors regularly conduct browser tests. The organization should establish a standard set of browsers for which testing is required (e.g., Microsoft Internet Explorer versions 2-6, Netscape version 3-6, AOL versions 6-8, etc.), and decide what features should be tested (e.g., JavaScript<sup>50</sup> disabled, style sheets disabled, text magnification, etc.). A standard for success or failure can thus be established.

**Questions to answer:**

- Who is managing the testing process? If it is the contractor, when and what kinds of testing does the contractor plan to conduct?
- Does the contractor understand the system users and tasks?
- Does the contractor have experience with accessibility testing? Can they describe their testing procedures and evaluation tools in detail? Is their work for other clients accessible?
- If testing is being conducted internally, does upper management understand that user testing is necessary? How can user tests be justified?
- On what browsers and browser versions should the system run?

Most Web contractors are not experts on accessibility and usability. However, many can conduct simple tests that provide useful feedback about the usability of the system. If evaluation is handled externally then it is important to determine whether the contractor has experience with that type of evaluation, how often and at what points in the process they intend to test, and what methodologies they intend to use. This is also helpful for keeping track of the development budget. It will be difficult to set up a usability lab with a minimal budget (e.g., \$2,000) for user testing. If a contractor sets up log analysis software to gather information about site visits, that will only gather certain types of information. It is important to know what kind of data is needed to justify the expense of the system to organizational decision makers.

---

<sup>50</sup> A simple, cross-platform World Wide Web scripting language (only vaguely related to Java).

## **Adopting a System**

If a system is adopted, usability and accessibility testing should have been conducted by the organization that developed the system. Adopting organizations should inquire as to the testing conducted and ask for documentation if it exists. Before deciding on a system to adopt, it may be useful to have experts conduct a review of the system to assess the level of usability. An accessibility review should also be conducted. Another important question to consider is the degree to which an adopted system can be modified in order to overcome usability issues or to bring it into compliance with accessibility standards. This has cost implications and should be considered carefully.

## **Purchasing a System**

If a system is purchased, usability and accessibility testing should have occurred prior to purchase. The company that produced the application should have conducted user testing. Some browser testing will probably be required once the site or system is populated with the purchasing agency's data to test the data (this assumes that the system is viewable in a commercial Web browser).

Accessibility testing should be the responsibility of the developer, since this type of solution is supposed to be delivered ready for installation, like a commercial software product. Managers should ask about implementation of state or local accessibility requirements before purchasing a product, since it is still the buyer's responsibility to ensure access for users with disabilities.

If the product does not meet stated guidelines, the buyer should consider purchasing another product or hiring a contractor instead. If the purchased system is still their preferred solution, then they should work out the cost of additional development with the product developer, who should recognize that the accessibility modifications will be desirable to other consumers and adjust fees accordingly.

## **Evaluation Techniques**

The following are basic techniques that can be used to monitor the performance of the deployed system. They can be used for developed, adopted or purchased systems.

### **Step 1: Perform Log Analysis and Monitor System Use**

Server log files record activity on a Web server. They provide information on where visitors to the site are coming from, which pages they visit, how long they stay, and which browsers they use. What data are collected depends on how the logs are set up by IT staff.

The project manager and technical staff should work together to select log analysis software (a review of log analysis software can be found at <http://webdesign.about.com/cs/traffic/>). If a contractor is involved, discuss which tools they use for server log analysis (if any), and how the reports will be accessed. Responsibility for log analysis may depend on where the system or site is hosted (e.g., Will it be hosted on the contractor's server or the health agency's server?). Some contractors provide access to reports from their Web servers on an agreed-upon schedule.

### Server logs should collect data on:

- Who visits the Web site
- How users move through site pages—how the user goes from one page to the next (using what page element – link, button, etc.)
- How much time users spend on a given page (e.g., If they spend a lot of time in one place, either the content meets their needs, or they are very confused. If many visitors spend lots of time on one page, that content may need to be moved up to a higher level within the site architecture, so visitors can find it quickly and easily.)
- The location from which visitors leave the site or system (e.g., If users are hitting the home page and leaving again immediately, there's a problem.)
- Task completion—logs can tell you what files were requested and whether the request was successfully filled. (This is not the same thing as user testing; logs can tell you whether someone got what they asked for, not whether they wanted something that was not there in the first place or whether what they asked for turned out to be the wrong thing.)

It is important to recognize that server log analysis is not entirely accurate. Some requests are never logged and some requests should not be counted. For example, if the same person visits the site multiple times, is each visit counted? Determine whether the organization wants to know how many unique visitors accessed the pages, or how many times the pages were accessed.

Issues to be aware of when using log analysis are described below:

**Caching.** Web browsers store pages in their cache so that the user does not have to load the pages anew every time one is requested. If a page is in the cache, the browser calls it up from the cache instead of requesting it from the server. Thus, no new visit is recorded.

**Proxy Servers.** Internet Service Providers (ISPs) often use proxy servers (servers that intercede between a Web browser and a Web server). Proxy servers also cache pages. If the proxy server has a page in its cache, it will load it instead of requesting it from the Web server.

**IP Address Confusion.** Some software equates a unique IP address/browser with a unique user. This may not be accurate because: 1) Internet Service Providers may assign a new IP address each time a user connects, resulting in an overestimate of users, or 2) Some users may share IP addresses, resulting in an underestimate of users (e.g., when a number of users are behind a firewall, they may all show up as the same IP address on Web server logs of sites outside the firewall).

**Robots.** Robots are computer programs that visit Web sites to catalog them for search engines. It is possible to filter out visits from robots, but the filter will have to be updated to catch new robots.

## **Step 2: Obtain User Feedback**

Methods of obtaining user feedback include:

- Email
- Customer service calls
- Interviews
- Surveys
- Usability tests
- Group discussions/focus groups
- A prominent contact phone number presented on every page of the site

All these methods can be effective, and different methods can be used together to achieve maximum effect.

## **Task 8**

**Maintain the System**



## Task 8: Maintain the System

---

Even a well-designed and programmed site will need maintenance. Users will find errors that were not foreseen. Software may be added to a server that creates conflicts with the system. The managers will want to see whether the site is achieving the goals defined in the requirements definition stage. They will also need to check and adjust how search engines are finding and indexing the site.

Eventually, any site will have things added to it. New modules must be tested for errors, and should be tested with users. Can users find the information? Do the new modules make things easier or introduce new problems? All this should be considered in the maintenance plan.

A Web site that does not have a well conceived maintenance plan will soon become confusing. The maintenance plan is part of the requirements definition but it should be reviewed when the site is implemented because during development issues may have come up that were not considered in the original plan. After about a year, the maintenance plan should be reviewed again to see that it is still on target, and changes should be made, if needed.

Following deployment, ongoing maintenance will be required and a maintenance plan should be implemented that includes:

- Schedule for running automated accessibility evaluation software and link-checking software, at a frequency determined by the number of changes made to the site in a given period, to ensure Section 508 compliance (if necessary) and prevent dead links
- Schedule for running log-analyzer software and running online surveys to acquire user feedback
- Schedule for optimizing strategic keywords (META tags and HTML)
- Incorporation of answers to questions from users into the site content
- Periodic user testing to ensure site upgrades do not diminish user experience
- Checking of site search capability at least weekly—site searches often become disabled without being noticed unless they are checked
- Examination of technical support logs and application server logs for Web server errors, and performance times
- Database backup plan
- Periodic database optimization including consistency checks, index maintenance, and analysis of queries for efficiency

For systems that are adopted or purchased, these issues may or may not be relevant depending on the system parameters and the types of maintenance features that have been built into the system by the donating or licensing organization.

## Summary

The information presented in these guidelines provides a framework for making an informed decision about whether to develop, adopt, or purchase a Web-based data dissemination system. The guidelines also outline the key tasks and steps within each task required for implementation, with attention to best practices in usability and accessibility of Web-based systems. The guidelines focus on the user interface of a system; they are not intended to address the complete process of designing an entire Web application. By following the steps contained in these guidelines, public health agencies will have:

- conducted a full analysis of implications and costs associated with their choice of option
- completed the individual stages of implementing the developed, adopted or purchased system
- tested and evaluated the system, and
- developed an ongoing maintenance plan.

Incorporating the best practices and procedures addressed in the many references and resources provided throughout the guidelines, and using the hands-on tools provided in the appendices, will help ensure that the developed, adopted, or purchased system is consistent with industry standards, federal guidelines, and the specific needs of the public health agency within the environment and context in which it operates.

## **Appendix A**

### **Checklist of System Considerations and Best Practices in System Design Features**

## Appendix A

### Checklist of System Considerations and Best Practices in System Design Features

---

#### System Considerations

The following checklist can be used as a tool to ensure that key system features are considered as the system is being designed or developed, or to ensure that systems being assessed for adoption or purchase have the necessary features. The first column lists the key considerations—those listed are not intended to be exhaustive, rather they represent common considerations. Others may also be relevant depending on the agency’s specific needs and circumstances. The second column lists the page numbers in this document where the topic is addressed; the third column lists the staff who need to be involved in addressing these considerations; the fourth column provides space for adding notes; and the fifth column provides space to document that the topic has been addressed.

Considerations	Page to Refer to for More Information	Lead Staff Responsible	Notes	Completed
Is the system compatible with present and future organizational goals and overall system goals?	Pp 7-9 Pp 17-18 Appendix D	Administrators Budget staff IT staff		
Is the system compatible with organizational standards and policies? Consider: <ul style="list-style-type: none"> <li>• system security</li> <li>• data confidentiality</li> <li>• accessibility</li> <li>• statistical calculations</li> <li>• data presentation</li> </ul>	P 11 Pp 19-20 Pp 28-31 Pp 36-37 Pp 51-52 Pp 56-57 Appendix G	Administrators IT staff		

Considerations	Page to Refer to for More Information	Lead Staff Responsible	Notes	Completed
<p>Can the system be adequately supported by agency staff? Consider:</p> <ul style="list-style-type: none"> <li>• current/future staffing capacity</li> <li>• technical skills</li> <li>• training needs</li> </ul>	<p>P 10 Pp 24-25 Pp 49-50 P 61 Appendix I</p>	<p>Administrators Budget staff IT staff</p>		
<p>Does the system meet the needs of identified users? Consider:</p> <ul style="list-style-type: none"> <li>• characteristics of target audience</li> <li>• common user tasks</li> <li>• data requirements</li> <li>• user technical/analytic capabilities</li> <li>• technical environments in which they work</li> </ul>	<p>Pp 8-9 Pp 15-17 Pp 35-36 Pp 55-56 Appendix C Appendix F</p>	<p>Administrators IT staff Data providers</p>		
<p>Is the system compatible with, or easily adaptable to fit, the current technical environment and future plans for emerging technologies? Consider:</p> <ul style="list-style-type: none"> <li>• hardware</li> <li>• operating system software</li> <li>• Web application</li> <li>• database servers</li> <li>• programming language</li> </ul>	<p>Pp 9 Pp 18-19 Pp 48-51 Appendix E Appendix I</p>	<p>IT staff</p>		
<p>Does the system fit within cost constraints? Consider:</p> <ul style="list-style-type: none"> <li>• start-up costs</li> <li>• staffing support requirements</li> <li>• ongoing maintenance costs and license fees</li> <li>• indirect costs associated with system/software upgrades, customization, staff training</li> </ul>	<p>P 10 P 25-26</p>	<p>Administrators Budget staff IT staff</p>		

Considerations	Page to Refer to for More Information	Lead Staff Responsible	Notes	Completed
<p>Will the system integrate with geographic information system, statistical, and other software currently used or projected for future use?</p>	<p>P 25 P 50</p>	<p>Administrators IT staff Data providers</p>		
<p>Does the system have the ability to import/process in a timely manner a variety of data sets relevant to the agency's mission and operations (e.g., vital statistics, hospitalization data, BRFSS data, Medicaid data, other program-specific and registry-based data)? Consider needs for aggregate data sets and individual record-based data sets</p>	<p>P 9 P 25 P 50 Appendix E</p>	<p>Administrators IT staff Data providers</p>		
<p>Has the system been sufficiently tested and documented? Consider:</p> <ul style="list-style-type: none"> <li>• Usability</li> <li>• Accessibility</li> <li>• Templates</li> <li>• Cascading Style Sheets</li> <li>• Coding standards and comments</li> <li>• File and directory naming standards</li> <li>• Style guides</li> </ul>	<p>Pp 21-22 Pp 34-36 P 40-45 P 55-56 P 58 Pp 60-67</p>	<p>IT staff</p>		
<p>Will it be necessary to make modifications to system functions (e.g., interface, query and output), templates, graphics, etc.? Consider:</p> <ul style="list-style-type: none"> <li>• Ease and cost of making modifications</li> <li>• Who will be responsible for making the modifications</li> </ul>	<p>Pp 52-57</p>	<p>Administrators IT staff</p>		

## Best Practices in System Design Features

This checklist can be used as a tool to monitor whether particular design features have been incorporated into systems as they are being developed, or are present in systems being considered for adoption or purchase. These features are discussed on pages 39, 46-47, and 53-54 in this guidelines document, and in Appendix H.

System Interface	Present	Notes/Comments
Links and buttons are clearly defined		
Error messages are easily understandable		
Explanations for user registration are provided (if applicable)		
Usage fees are specified (if applicable)		
Text is at a 6 <sup>th</sup> to 8 <sup>th</sup> grade reading level		
Definitions are provided		
Help links are provided and are context specific		
Tasks can be accomplished in as few steps as possible		
Form elements are appropriate for the task (e.g., checkboxes for 2-3 options versus Drop-down list for multiple options)		
Contact information appears on every page		
The number of browser windows that can be open at one time is limited		
Font size is 11 point or larger; relative font sizes are used when possible		
The color scheme is uniform		
High contrast between foreground and background		
Web site has been tested using multiple browsers and browser versions		

Query Functions	Present	Notes/Comments
At least two query interfaces (one for beginners, one for advanced)		
A simple query interface is the default		
Text is written at a 6 <sup>th</sup> to 8 <sup>th</sup> grade level		
A tabular format <sup>51</sup> is used		
Content of data sets is described		
Disclaimers and limitations of data are clearly stated		
Users have flexibility in selecting age range and year variables		
Users can query cause of death alphabetically and by ICD codes		
Users can compare geographic regions		
Variables that cannot be used in combination are identified		
Users can download datasets (within confidentiality standards)		
A clear way to submit data is provided (e.g., submit button) in a easy to find location		
A query summary page is provided		
Users are informed that the query is being processed		
Speed of queries is maximized (no longer than 10 seconds)		

<sup>51</sup> Information presented in tables rather than narrative text.



Outputs	Present	Notes/Comments
Query variables are identified		
Output and query variables are differentiated (by colors, etc.)		
Users can select multiple formats (e.g., tables, graphs)		
Users can easily manipulate output (change colors, labels, etc.)		
Users can conduct further queries		
Outputs can be downloaded and easily printed		
Users can download in multiple formats that are clearly defined		
The output page includes interpretation and instructions for additional steps and options		

## **Appendix B**

### **References and Resources**

## Appendix B References and Resources

Introduction – References and Resources	Description
<a href="http://www.macoint.com/~atl/dphsi/">http://www.macoint.com/~atl/dphsi/</a>	Additional information on the evaluations leading up to the development of this guide, including reports from all earlier phases of the project  Information about the tutorial and the non-deployable prototype
<a href="http://www.access-board.gov/sec508/508standards.htm">http://www.access-board.gov/sec508/508standards.htm</a>	Federal guidelines on accessibility based on Section 508 of the U.S. Rehabilitation Act
<a href="http://www.nahdo.org/">http://www.nahdo.org/</a>	National Association of Health Data Organizations
<a href="http://www.naphsis.org/">http://www.naphsis.org/</a>	National Association for Public Health Statistics and Information Systems

Task 2 – References and Resources	Description
<a href="http://www.netiq.com/WebTrends/">http://www.netiq.com/WebTrends/</a>	WebTrends, a server log analysis program
<a href="http://www.watchfire.com/">http://www.watchfire.com/</a>	WebQA™, a server log analysis program
<a href="http://aspe.hhs.gov/admsimp/index.htm">http://aspe.hhs.gov/admsimp/index.htm</a>	Information on the Health Insurance Portability and Accountability Act (HIPAA) Data Privacy Rule
1) <a href="http://aspe.hhs.gov/datacncl/privcmte.htm#goal">http://aspe.hhs.gov/datacncl/privcmte.htm#goal</a> 2) <a href="http://www.fcs.m.gov/cdac/index.html">http://www.fcs.m.gov/cdac/index.html</a> 3) <a href="http://www.cdc.gov/nchs/">http://www.cdc.gov/nchs/</a> 4) <a href="http://www.naphsis.org/">http://www.naphsis.org/</a> 5) <a href="http://www.nahdo.org/">http://www.nahdo.org/</a>	Sources for information on data privacy: 1) HHS Privacy Committee 2) Confidentiality and Data Access Committee of the Federal Committee on Statistical Methodology 3) CDC National Center for Health Statistics 4) National Association for Public Health Statistics and Information Systems 5) National Association of Health Data Organizations
<a href="http://www.cdc.gov/nedss/Security/index.html/">http://www.cdc.gov/nedss/Security/index.html/</a>	CDC Secure Data Network Standards and Procedures
<i>For the Record: Protecting Electronic Health Information</i> <a href="http://www.nap.edu/readingroom/books/for/">http://www.nap.edu/readingroom/books/for/</a>	Information about security policies and practices

Task 4 – References and Resources	Description
Office of Management and Budget (OMB) October 1997 Changes to the Statistical Policy Directive No. 15, Race and Ethnicity Standards for Federal Statistics and Administrative Reporting. <a href="http://www.whitehouse.gov/omb/fedreg/ombdir15.htm">http://www.whitehouse.gov/omb/fedreg/ombdir15.htm</a>	Information about race/ethnicity standards
1) <a href="http://www.cdc.gov/nchs/">http://www.cdc.gov/nchs/</a> 2) <a href="http://www.fcs.gov">http://www.fcs.gov</a> 3) <a href="http://www.naphsis.org">http://www.naphsis.org</a> 4) <a href="http://www.nahdo.org">http://www.nahdo.org</a>	Information and Guidance on Standards for Statistical Calculation and Data Presentation Can be Obtained from:  1) CDC's National Center for Health Statistics 2) The Federal Committee on Statistical Methodology 3) The National Association for Public Health Statistics and Information Systems 4) The National Association of Health Data Organizations

Task 5 – References and Resources	Description
<a href="http://www.useit.com/papers/guerrilla_hci.html">http://www.useit.com/papers/guerrilla_hci.html</a>	Jakob Nielsen's article, "Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier," in which the usability expert argues that a few simple user-testing methods can be used to gather excellent information on the usability of system screens
<a href="http://www.section508.gov">http://www.section508.gov</a>	The official Section 508 Web site for accessibility guidelines
<a href="http://www.w3.org/WAI/ER/existingtools.html">http://www.w3.org/WAI/ER/existingtools.html</a>	List of evaluation software products
<a href="http://www.w3.org/WAI/eval/">http://www.w3.org/WAI/eval/</a>	Lists guidelines for conducting accessibility evaluations
<a href="http://validator.w3.org/">http://validator.w3.org/</a>	An online validator for HTML documents
<a href="http://jigsaw.w3.org/css-validator/">http://jigsaw.w3.org/css-validator/</a>	An online validator for CSS files
1) <a href="http://www.cdc.gov/nchs/">http://www.cdc.gov/nchs/</a> 2) <a href="http://www.fcs.gov">http://www.fcs.gov</a> 3) <a href="http://www.naphsis.org">http://www.naphsis.org</a> 4) <a href="http://www.nahdo.org">http://www.nahdo.org</a>	Information and Guidance on Standards for Statistical Calculation and Data Presentation Can be Obtained from:  1) CDC's National Center for Health Statistics 2) The Federal Committee on Statistical Methodology 3) The National Association for Public Health Statistics and Information Systems 4) The National Association of Health Data Organizations

Task 7 – References and Resources	Description
<a href="http://webdesign.about.com/cs/traffic/">http://webdesign.about.com/cs/traffic/</a>	A review of log analysis software

Appendices – References and Resources	Description
<p>Holsti, O.R. <i>Content Analysis for the Social Sciences and the Humanities</i>. Addison-Wesley, 1969.</p> <p>Krippendorf, K. <i>Content Analysis: An Introduction to Its Methodology</i>. Sage Publications, 1980.</p>	Standard references on content analysis methodology
<a href="http://www.useit.com/papers/heuristic/heuristic_list.html">http://www.useit.com/papers/heuristic/heuristic_list.html</a>	Jakob Nielsen’s Usability Heuristics

**Appendix C**  
**User Analysis**

## Appendix C User Analysis

---

User analysis tells the project team the problems to be solved, the tools the users need to achieve their goals, how best to organize the system to meet user needs, and the technical limitations the users have. User analysis should be done before any system design so that when the interface and database designers start their work, they have the needs of the users in mind. The system should exist to serve the users, and doing a user analysis keeps project teams focused.

### 1. Document Analysis

A review of any related documents describing the mission, purpose, or users of the site can help a Web site evaluator understand the broad conceptualization of the system, site, stakeholders who should be interviewed or surveyed, documents that will be published on the site, and the target audience(s) for the site.

### 2. Feedback/Support Request Analysis

Email and phone messages that contain comments, questions, and/or feedback on a Web site can indicate users' informational needs and the types of problems they encounter. Support request analysis categorizes and analyzes textual information, such as that found in email messages.<sup>52</sup>

### 3. Surveys Using Questionnaires

Questionnaires are an excellent way of obtaining either quantitative or qualitative information (depending on the questionnaire design), since user responses are written and can be tallied to illustrate user preferences. Questionnaires, however, are only able to assess users' opinions about the user interface; they cannot assess actual behavior while using it. For this reason testing data that illustrate the actual behavior of users should be obtained and weighed more heavily than users' statements on questionnaires or in interviews. (See item #5 below.)

- **Benefits**

One of the benefits of questionnaires is that they can be administered without an evaluator present; forms can be distributed to users. Another benefit is that questionnaires can be distributed to large groups or geographically dispersed populations. In fact, a questionnaire could be distributed to every user of a particular system. This comprehensive coverage increases the opportunity to find differences between user groups and identify specialized needs of smaller user groups. Surveys that use questionnaires are typically conducted among a randomly selected sample ranging from 50-1,000 users.

---

<sup>52</sup> Standard references on content analysis methodology include: Holsti, O.R. *Content Analysis for the Social Sciences and the Humanities*. Addison-Wesley, 1969, and Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, 1980.

- **Questionnaire Content**

It is often effective to ask users to document specific, critical incidents that occurred while using the system. Recording when the system performed poorly can help designers avoid worst-case issues in future redesigns.

One of the drawbacks of using questionnaires is that questions cannot be rephrased as they can during verbal interviews. Questionnaires should therefore be subjected to pilot testing and iterative design before they are distributed to users. As a questionnaire is really a user interface in its own right, usability principles should be in force. Questionnaires that are too long, hard to understand, or unprofessional will often get a low response rate. Revised questionnaires can be used later in the system's evolution to measure changes in user response.

Users frequently have difficulty responding to open-ended questions (e.g., how using a site or product made them feel) and may simply ignore the question or answer it cryptically. Since these types of responses are difficult to interpret, most questionnaires use closed-ended questions where users simply respond with a single, easily quantifiable fact (e.g., the number of times they visited a Web site). Questionnaires may also use checklists or ratings scales to obtain specific, easily tabulated responses.

Users are more likely to respond to a short questionnaire than a long one. Questionnaire or survey designers should limit questions to those directly related to the success of the project at hand.

- **Response Bias**

With tools like questionnaires and interviews users may say they do one thing but in fact do another. This tendency is more evident when interviews are conducted in person. Online surveys or questionnaires administered via email may garner more accurate responses than verbal interviews when the topical matter is sensitive and users might avoid responding out of embarrassment.

#### **4. Online Surveys**

Online surveys are a cost-effective way to gather Web site information from a large and/or dispersed population.

Placement of the online survey is likely to affect response rates and the types of participants. Selection of questionnaire respondents is non-random (self-selected), regardless of where a questionnaire is placed on a Web site. Demographic information collected by the survey helps to characterize respondents, but care should be taken in generalizing responses to all Web site visitors.

The questionnaire results can provide information about the variety and characteristics of users participating in the survey, how they learned about the Web site, and how frequently they have used it. Questions can be designed to elicit user responses to content and design, and to identify valuable new services and features users may want.



The process for conducting an online survey includes the following steps:

- **Development.** Questionnaire development begins with thoughtful and appropriate research questions. These research questions may be formatted using a Web-enabled survey software package. Placement of the survey should be based on level of traffic through the site (which can be determined through use of Web server log analysis software), as well as the objectives of the survey.
- **Implementation.** A link inviting participation in the survey should be placed on the Web site and responses monitored daily. Placement of the link may be adjusted based on the number of responses received from a given location. Pilot testing can help determine the best placement. The link and the data collection instrument should remain available until the desired number of responses is obtained.
- **Reporting.** Following the completion of data collection and removal of the survey from the Web site, data should be stored in database records for analysis. Closed-ended questions (e.g., scales and pick lists) are tabulated and summarized by category of respondent, while open-ended questions are coded and categorized.

Off-the-shelf survey software can be used to format and conduct the pilot test survey. To set up a questionnaire and conduct a survey, software should allow the developer to do the following:

- Easily format and set up a survey comprised of several types of questions
- Link the survey from the Internet or make it accessible via an email package
- Easily create reports and/or export the data to other software for analysis and reporting

## 5. Analyzing Users in their Environment

It is extremely important that design teams observe users in their working environments. Site visits allow developers to understand all the ways users are interacting with the system, many of which cannot be anticipated during the design process.

In a site visit, the designer or evaluator visits a user or user group on-site. There, he or she watches the users work and takes notes or videotapes them at work (if the users agree). The evaluator may also conduct contextual interviews to gather information about the users' work and tasks.

The observer should be as unobtrusive as possible. If he or she requires an explanation of user behavior, it is generally best to note the action and wait to see if it recurs. At the end of the site visit, the users can be interviewed and questions answered.

Users often have questions for observers, especially if the observer is there as a representative of the system design group. But the goal of the visit is to gather information, not to provide instruction, so observers should politely decline to answer questions until they have as much information as possible about the system's use.

## **User Analysis Worksheet**

Document analysis, feedback analysis, questionnaire, online survey and/or user environment analysis will provide the information needed to fill out this worksheet.

### **Audience/Users**

Define groups of users, identifying their occupation, age range, gender, online frequency, online activities, and anything else that characterizes that user group.

### **Rank**

Rank the importance of the user group from 1 to 10, 1 being the least important. This is to help prioritize the needs of the different user groups. Some user groups may have competing needs making decisions based on rankings necessary.

### **Benefits of the Worksheet**

It is best to summarize the user analysis so managers can quickly see who the end users are and what their tasks are. Don't forget to include technical limitations of the user groups. For instance, can JavaScript be used? May cookies be set? Do the users have slow Internet connections?

Once the data are gathered, consider organizational needs and goals for the Web site and how they might differ from or relate to user goals. Consider how conflicts might be resolved.

## User Analysis Worksheet with Examples

Audience/Users	Rank	Technical environment	List the three most important information needs of this audience	List three (or more) important tasks that these users need to accomplish
Example 1: Public Health Researcher (state level)	6	PC Windows 98SE T-1 line Uses SQL Online daily Fairly comfortable with technology	<ol style="list-style-type: none"> <li>1) Maternal and Child Health data</li> <li>2) State census data</li> <li>3) State hospitalization data</li> </ol>	<ul style="list-style-type: none"> <li>• Ability to download datasets for manipulation using desktop analysis software</li> <li>• Ability to modify age ranges at will</li> <li>• Ability to age-adjust data</li> </ul>
Example 2: General Public	8	PC Windows 56K modem On line once a month for health issues Has not used statistical software Can navigate on line, but not much more technical background High school degree	<ol style="list-style-type: none"> <li>1) Information on cause of a specific disease or injury</li> <li>2) Information on who the disease or injury affects</li> <li>3) Prevalence of the disease or injury in their community</li> </ol>	<ul style="list-style-type: none"> <li>• Find basic statistics on disease or injury and who it affects (in layman's terms)</li> <li>• Understand cause of disease or injury and contributing factors</li> <li>• Ability to print information</li> </ul>
Example 3: Researcher at academic institution	4	PC Windows T-1line On line daily MA or PhD Proficient in conducting statistical analyses Comfortable with technology	<ol style="list-style-type: none"> <li>1) Epidemiological information</li> <li>2) Weighted percentages, age adjusted rates across multiple variables</li> <li>3) Technical information on how data were collected</li> </ol>	<ul style="list-style-type: none"> <li>• Ability to customize queries for data needs</li> <li>• Ability to make comparisons across states</li> <li>• Ability to save results</li> </ul>

## **Appendix D**

### **Goals and Measurement**

## Appendix D Goals and Measurement

---

Identifying and prioritizing site goals is important to ensure that all who are involved clearly understand the project. It is important also to measure to what extent these goals are achieved on an ongoing basis. The chart on page D-2 offers a means of doing this.

### How to Use this Chart

The following chart provides a format for documenting and ranking goals and describing measurement criteria.

1. Decide on the Web site goals. These should be based on the ultimate goal of the site. Thus, if the reason for creating the site is to give health professionals greater access to current public health data, possible goals might be to increase the number of health professionals accessing the Web site, to keep the system continually updated with new data, and to increase the system's ease-of-use for the health professionals.
2. Assign specific measurement criteria to each goal. Use numbers or percentages whenever possible. Each measure should also have a timeframe. For instance, one measure could be to have at least 500 users per week from a .gov or .org organization. Or, if users must register to use the tool, additional measures could be to get at least 50 new registrations per week, or achieve 1,500 registrations in six months. If measures are specific, then comparisons can be made at designated time intervals to see if goals have been reached. By tracking these measures the project team can see how the system is actually being used and determine how to budget future work.
3. Rank the measures by importance, 1-4, in the Rank column, 1 being lowest. Setting a rank for system goals will help you establish and monitor development priorities — the higher the rank, the higher the priority. Some measures may be in conflict with each other, so it is crucial to decide early in the process what the relative priorities are.

For example, one goal's measure may be 50 new registrations each week, while another goal's measure may be decreasing the time it takes for users to access data. Because registering for a system increases the time it takes users to access data, this is a goal conflict. The organization must decide whether it is more important to obtain new registrations or speed up users' work in the system. The tool's design or selection will be informed by these priorities.

4. Establish at least one measure for each goal. If a goal of the site is hard to quantify, be more specific. For example, measures for the goal, "Help health professionals with their work by giving them easier access to state-wide data," could be to increase the number of users per month by 10%, or increase the length of an average user session from 10 minutes to 20 minutes.

## Goals and Measurement Chart with Examples

Description of Goal	Measurement Criteria	Rank 1 to 4
<p><b><u>Example 1</u></b></p> <p>Increase the number of county health professionals using the software.</p>	In 6 mos., have 200 more county health professionals registered for the software.	1
<p><b><u>Example 2</u></b></p> <p>Obtain and use feedback from users to improve product quality.</p>	Talk to 10 users 3 months after pilot is available	2
<p><b><u>Example 3</u></b></p> <p>Make the query interface simple to use so a broader and potentially underserved population can generate accurate results.</p>	Check user log and site visit analysis to determine user traffic for targeted populations.	4

## **Appendix E**

### **Technical Specifications**

## Appendix E Technical Specifications

Whether developing, adopting, or purchasing a Web-based dissemination system, it is important to analyze the organization's current technical environment. For state and local health agencies developing a system, the process will help determine which software should be used to develop the system and how the system will be incorporated into the present IT environment. For adoption or purchase, this information is important for assessing compatibility and the ease with which a system can be integrated into the present IT environment. The process will highlight potential technical barriers to successful integration of adopted or purchased systems and will provide a basis for costing additional hardware or software that must be purchased to achieve this. The analysis will also help determine how much staff training is required to support development, adoption, or purchase of a system.

<b>General Requirements</b>
Are there any security issues?
Are there other technical issues or limitations (licenses, database, and middleware <sup>53</sup> choices)?
Will the site be database-driven or static? If static, do you need to build in database-driven capabilities for the future?
<b>Web Host Environment</b>
If a vendor, give name, contact information, and point of contact:
Details about the production (live) environment:
Web server hardware and software with version numbers:
Database server software:
What is the access to the Web server?
What is the training level of the staff?
If using a vendor, what are our responsibilities at launch and what are theirs? If an in-house development effort, who has what responsibilities at launch?
How much maintenance, if any, is required?
Who is responsible for maintenance?

<sup>53</sup> Software that manages the interaction between an application and a network.



Testing Environment		Live Environment	
Database name:	Access authorized for:	Database name:	Access authorized for:
Location:		Location:	
Web server name:		Web server name:	
Web server operating system and version:		Web server operating system and version:	
Web server software:		Web server software:	
Server-side scripting languages supported:		Server-side scripting languages supported:	
Database server name:		Database server name:	
Database server version:		Database server version:	
Database(s) name(s):		Database(s) name(s):	
Who needs access to which databases?		Who needs access to which databases?	

## **Appendix F**

### **Discount Usability Testing**

## **Appendix F**

### **Discount Usability Testing**

---

#### **How To Use Usability Testing:**

Usability testing should be done early in the process and repeated as often as possible. Simple user tests can be conducted during the Concept Prototype stage of the process. The design, information architecture, and early prototypes can all benefit from early user involvement. At early stages in development, design usability can be tested quickly and inexpensively, using the methods described below.

As the project team works, it often becomes involved in solving programming problems and loses sight of issues that would be significant to an end user. Because of this inevitable loss of objectivity, much can be learned from testing with representative end users. Even one hour of a test before the system is programmed can save the many hours of programming it might take to fix a serious usability flaw discovered later.

These guidelines were written as a primer for conducting informal user tests for diagnostic purposes. If a more scientifically rigorous approach is required, the organization should consider hiring a consultant with training and experience in Human-Computer Interaction (HCI) or a consulting company that specializes in usability testing. Formal quantitative testing will require a larger participant population to achieve statistically valid results.

Following are descriptions for three types of usability testing: scenarios, simplified thinking out loud, and heuristic evaluation. See Appendix D of the Transference Report at <http://www.macoint.com/~atl/dphsi/> for additional information on user and task analysis.

#### **1. Scenarios**

A scenario is a type of prototype. The goal of prototyping is to limit the complexity of the test system by building out only select sections of the planned system. Prototypes allow designers to see how well a design meets users' needs before investing time and money in building out the full site or system. Scenarios simulate a specific path through the system, reducing the functionality of the screens even further. Because of their limitations, scenarios can be quickly changed and are cheap to implement. Paper mockups, HTML screens, or presentation slides may all be used as scenarios in simple user tests.

## **2. Simplified Thinking Out Loud**

The second method in discount user testing is bringing in representative users and asking them to think aloud while performing a series of tasks. Thinking out loud tests can be carried out with as few as three users per test (the smaller testing group allows the project team to invest in more frequent tests). If only one round of testing is planned, a larger user group will yield more comprehensive results.

### **Facilitating the Test**

Anyone can learn to conduct a simple user test. The person who designed the system or site should not conduct the evaluation. The facilitator should be someone who is a good listener and does not have a stake in the results.

The facilitator's role is to provide instruction and encourage users to think out loud by describing what they are doing and why. The facilitator should listen carefully to what they say and avoid interpreting it. Facilitators should not discuss the site or system with subjects before testing begins.

Prior to testing, the facilitator should take the test to evaluate whether it can be completed in the time allotted and whether the tasks require more domain knowledge or training. He or she should check to be sure that tasks and questions are clearly articulated. Equipment should be tested to make sure everything works and that no additional items are required.

The facilitator should not give participants hints about what to do. If they are unsure about how to achieve a task, the facilitator may ask them questions to encourage them to find the solution themselves. Instructions should be clear and simple, and be repeated as often as necessary. The facilitator should be prepared to modify the testing script or scenario in response to user feedback.

During interviews, the facilitator should maintain a professional, neutral persona. He or she should be careful not to offend participants or make them uncomfortable. The facilitator should avoid biasing participants by showing his or her opinions of the system or participants' knowledge. Participants should not be asked how to fix design problems. It is often helpful for the facilitator to summarize for the participant what has been heard or observed at key points in testing. This will give the participant an opportunity to rephrase statements that are inaccurate or offer more detail.

When each interview is complete, the facilitator should prepare a written summary of notes and observations to provide to the project team when testing is complete.

### **Observing the Test**

Even if the thinking out loud test is videotaped or audiotaped, it is a good idea to have a second person in the room to observe the test and take notes. The facilitator is often busy asking questions and is unable to take notes during the test. The observer should watch closely and silently, and take notes without interpreting the users' responses.

## Equipment

To conduct thinking out loud tests, the evaluation or project team needs an office or conference room with two chairs, a computer, an Internet connection (if testing a live Web site), a camcorder, and a tripod. The camcorder records what the user says and does, and what user and facilitator say to each other.

Videotaping the interviews may not always be necessary. Facilitators should write down their notes and observations as soon as a test is complete. To prepare a report more quickly, facilitators can compare notes, rather than recording and analyzing videotaped interviews, which can be a laborious process. Reducing the amount of time and money spent on analysis will allow the evaluation team to invest more time in testing different iterations of the design and modifying test scenarios based on user responses. The goal of testing, in most cases, is to identify problems so they can be corrected. Frequent testing that facilitates quick modification of design screens is the best way of accomplishing this.

## Participants

In simplified thinking out loud tests, users should be selected who are familiar with the Web or the system in question. They do not have to be expert users, but they do have to be able to move around in the screens. Ideally, they should reflect the target user group of the system under development. There will be some differences between the people recruited and those the system is meant to serve. Facilitators should factor in differences when analyzing test results.

A more specific user group or groups should be recruited if:

- Site or system will be used by one type of user only
- Audience is split between clearly defined groups with divergent needs
- Using the site or system requires specific domain knowledge

If the site has different user groups (e.g., scientists and artists), or groups with different levels of technical experience, it is a good idea to include users who represent that range. Including a broader representation of users may increase the number of subjects needed for testing.

## Interpreting Results

After each round of tests, the project team should review its observations and decide on solutions and next steps. This discussion will result in changes to the scenario screens, which can then be used for additional user testing to evaluate the design solutions.

Facilitators should look for patterns in user responses, frustrations or observations that are repeated across interviews or groups. If most users had a similar negative experience with a particular feature, that will indicate that a change should be made.

### **3. Heuristic Evaluation**

In a heuristic evaluation, one or more usability professionals evaluate a Web site or system based on recognized rules of thumb (also called “heuristics”). Typically the emphasis is not on comprehensively examining the functionality of the site. The review is usually conducted in the context of typical user tasks, to provide feedback to the site’s developers on the extent the interface is likely to be compatible with the intended users’ needs and preferences.

#### **Personnel**

**Number of reviewers.** Heuristic evaluations are typically conducted by one or a small number of reviewers. It is difficult for any one reviewer, no matter how knowledgeable, to anticipate the full range of usability issues that a system’s users may encounter. Three to five reviewers are advisable, but meaningful reviews can be accomplished with fewer.

**Qualifications of reviewers.** Because heuristic evaluations focus on the design of the user interface, it is best if they are conducted by reviewers who are knowledgeable about industry best practices and current thinking in designing for ease of use. Experience in performing such evaluations is probably a better predictor of competence than any academic credentials.

Heuristic evaluations are best accomplished by individuals other than those who created the interface that is under review. While prior domain knowledge about the content of the Web site is helpful, it is not critical. It is useful for the reviewer to consider the goals of the Web site, the nature of other commercial or government systems similar to the Web site, and the constraints under which the organization responsible for the Web site is operating. It is critical, however, for the reviewer to examine the Web site from the perspective of a user who may not have prior domain knowledge.

#### **Time Required to Conduct the Evaluation**

Most heuristic evaluations can be accomplished in a matter of days. The time required varies with the size of the Web site, its complexity, the purpose of the review, the nature of the usability issues that arise in the review, and the competence of the reviewers.

The time requirements include not only a visual inspection of the site, but also an understanding of the design objectives, the range of users the site is intended to accommodate, and typical tasks. Time is also required to document usability concerns and formulate design recommendations, as required.

The Web site’s development may also affect the time required. A cursory review of an early stage prototype to assure the developers that they are on the right track can be done quickly, while a more comprehensive review of a fully developed site may take longer.

#### **When to Evaluate a Design**

Heuristic evaluations can be conducted on very early prototypes, including paper mockups, as well as later stage electronic prototypes, with or without all of the back-end functionality implemented.

## Conducting the Evaluation

Planning for a heuristic evaluation involves acquainting the reviewers with the Web site or application, specifying usability objectives, identifying the characteristics of typical users, and delineating use cases (i.e., typical task scenarios). Information on problems that may have surfaced from help-desk inquiries, user email comments, or professional critiques by media or industry reviewers should be incorporated into preparation for the evaluation.

After gathering background information on site objectives, user characteristics, and user tasks, the reviewer can proceed with a systematic examination of the site. If more than one reviewer is involved, each should work independently. The reviewer should make two passes: one to become acquainted with the flow of the interface screens, and another to focus on individual design elements or functionality.

## Nielsen's Usability Heuristics

Heuristics are also known as “design rules of thumb.” Usability specialists use these design standards to identify potential problems with a user interface. There are many sets of usability design heuristics. These are not mutually exclusive and cover many of the same aspects of interface design. Two of the best-known lists of usability heuristics were published by Jakob Nielsen and Larry Constantine. These are summarized below:

### Nielsen's Usability Heuristics<sup>54</sup>

- **Visibility of system status**—The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **Match between system and real world**—The system should speak the user's language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. It should follow real-world conventions, making information appear in a natural and logical order.
- **User control and freedom**—Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted location without having to go through an extended dialogue. The system should support undo and redo.
- **Consistency and standards**—The system should follow platform conventions. Users should not have to wonder whether different words, situations, or actions mean the same thing.
- **Error prevention**—Even better than a good error message is a careful design that prevents a problem from occurring in the first place.
- **Recognition rather than recall**—Make objects, actions, and options visible, so the user does not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

---

<sup>54</sup> [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)

- **Flexibility and efficiency of use**—Accelerators—unseen by the novice user—may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users and allow users to tailor frequent actions.
- **Aesthetic and minimalist design**—Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- **Help users recognize, diagnose, and recover from errors**—Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Help and documentation**—The ideal system can be used without documentation, but it may often be necessary to provide help and documentation. Any such information should be easy to search, task focused, list concrete steps to be carried out, and not be too large.

### Constantine's Usability Principles<sup>55</sup>

- **Structure Principle**—Organize the user interface purposefully, in meaningful and useful ways that put related things together and separate unrelated things based on clear, consistent models that are apparent and recognizable to others.
- **Simplicity Principle**—Make simple, common tasks easy to do, communicating simply in the user's own language and providing good shortcuts that are meaningfully related to longer procedures.
- **Visibility Principle**—Keep visible needed options and materials for a given task without distracting the user with extraneous or redundant information.
- **Feedback Principle**—Keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions using clear, concise, and unambiguous language familiar to users.
- **Tolerance Principle**—Be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing, at the same time preventing errors wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions reasonably.
- **Reuse Principle**—Reduce the need for users to rethink and remember by reusing internal and external components and behaviors, maintaining consistency with purpose rather than merely arbitrary consistency.

---

<sup>55</sup> From Constantine, L.L. (1994). Collaborative Usability Inspections for Software. Software Development '94 Proceedings, San Francisco: Miller Freeman.



**Appendix G**  
**Accessibility Checklist**

## **Appendix G**

### **Accessibility Checklist**

---

Designing for accessibility means designing pages and systems that can be accessed by people with a variety of disabilities. Accessibility reviews are conducted to identify general problems with a site or system. This type of review is not an exhaustive test of compliance, but an assessment of representative accessibility barriers on a select series of system screens.

#### **When to Review**

If a system is being developed internally, preliminary accessibility reviews should be conducted as soon as page templates have been developed. If accessibility problems are caught at this stage, it will save the developer from having to fix extensive accessibility problems after the system has already been programmed, when modifications will be expensive. If the system is being adopted or purchased, a preliminary review can be conducted before the system is received. This should establish the breadth and severity of any accessibility barriers in the system. If the system is being adopted, and the preliminary review indicates that the system has significant accessibility problems, then the adoptee should consider those errors during implementation and consider altering the design to correct them. If the system is being purchased, any accessibility barriers and fixes should be discussed with the vendor.

Conformance evaluation is conducted when programming is finished to ensure that the final system is compliant with the appropriate accessibility guidelines or standards. This evaluation is necessary to see that accessibility problems have not been introduced during development or modification and to verify that the system has met its requirements.

All federal government agencies have adopted accessibility standards. Section 508 of the Federal Rehabilitation Act states that federal agencies must provide people with disabilities with access to information that is comparable to the access afforded other members of the public. Section 508 standards for Web technology are included in this appendix for reference. Because many organizations have developed their own standards, it is recommended that the project team research other standards that might apply to their system.

#### **Preliminary Review**

Preliminary reviews can identify major barriers early in development, so they can be fixed before resources are invested in an inaccessible design. A preliminary review or consultation may also take place prior to a redesign or retrofitting effort, to help guide developers to areas in need of attention.

These early reviews combine a manual check of representative pages or templates with automated tests using accessibility software. The reviewer should be familiar with hypertext markup language (HTML) and the set of guidelines being enforced. He or she should be assuming the role of user with a disability or disabilities, and for that reason should have a basic familiarity with accessibility barriers common to the Web.

The project team should select a series of pages or features that are representative of the site that is under development. Test pages may be simple navigation templates, or they may be a set of complete screens. Any major features or functionality should be reviewed. If the feature in question is not yet available, the accessibility reviewer should be made aware of the purpose and intended functionality of the feature in question.

The selected screens should then be reviewed using a graphical user interface (GUI) browser such as Microsoft (MS) Internet Explorer, Netscape, or Opera. The reviewer should:

- Turn off images, sound, style sheets, and scripting to see whether changing those settings interferes with access to site content.
- Review use of color and color contrast to be certain that color is not necessary for understanding site content and that color contrast is sufficient for persons with vision impairments.
- Use the tab key to move through the screen to assess whether the site can be used by someone who cannot use a mouse.
- Review link text, to make sure that link text is clear and meaningful, and that it communicates the target of the link.

A preliminary review should include reading selected pages with a screen reader (e.g., JAWS) or voice browser (e.g., IBM Home Page Reader). Using assistive technology will help the reviewer understand access issues more completely. A screen reader or voice browser will also help illustrate the severity of an accessibility barrier.

A text-only browser (e.g., Lynx) may be used to assess whether text equivalents for images, image maps, and frames are available. Text-only browsers also demonstrate clearly whether site content will be rendered in a logical reading order when accessed using assistive technology.

The reviewer may also use automated evaluation tools. There are a number of tools available on the commercial market. It is important to note, however, that no accessibility testing software is foolproof, and all require that the user be familiar with accessible design principles and guidelines. A list of automated accessibility tools can be found at:  
<http://www.w3.org/WAI/ER/existingtools.html>.

Once the reviewer has reviewed the representative pages for accessibility barriers using these methods, he or she will summarize problems encountered and suggest solutions. Suggested solutions should be accompanied by clear examples.

### **Conformance Evaluation**

In a conformance evaluation, an accessibility expert assesses whether a site or system complies with set guidelines to meet a predetermined level of conformance. Federal Web sites must comply with standards set by Section 508 of the Federal Rehabilitation Act. Private companies and state and local health agencies often require compliance with the W3C Web Content Accessibility Guidelines.

Conformance evaluation includes all the steps used in preliminary evaluation, but requires a comprehensive, page-by-page assessment of accessibility issues within a site or system. Reviewers must be familiar with Web markup languages. Ideally, a conformance evaluation would also include user testing with individuals with a variety of disabilities. While reviewers can assay the role of a person with a disability, they cannot completely reproduce the user's experience with the site or system.

In a conformance evaluation, an reviewer checks the following items for compliance with guidelines:

- Images and image maps
- Link text
- Text (paragraphs, phrases, punctuation, and symbols)
- Color use and coordination
- Lists and headings
- Multimedia
- Forms
- Cascading Style Sheets (CSS)
- Tables
- Frames
- Applets and scripts

A conformance evaluation should take place before a site is launched, to verify that the final version of the site or system is compliant with guidelines.

In addition to the evaluation methods used in preliminary review, a conformance review includes the following steps (applied to the full site rather than selected pages):

- Code validation with an automated tool such as the HTML Validation service offered in the W3C Web site
- Checking with at least one automated accessibility tool
- Manual evaluation against the guidelines or standards selected for compliance
- Browser review with at least three different GUI browsers, such as MS Internet Explorer, Netscape, and Opera, in at least two versions. (The project team or Web development group should establish a base level of browser compatibility prior to development.) Browser review should include testing with scripting, CSS, and images turned off, as well as testing without a mouse.

The reviewer should then summarize problems and best practices for each page or section, clearly indicating which problems are global, and the severity of those problems. The report should include recommended steps, including accessibility fixes and any additional testing. It may also recommend steps for ongoing maintenance.

## Section 508 Standards for Web Accessibility

Although non-federal public health agencies are not required to comply with federal standards for Web accessibility, many state and local public health agencies do receive significant federal funding, and compliance with federal standards will ensure that their Web sites and systems meet requirements at the federal level.

- (A) A text equivalent for every non-text element shall be provided (e.g., via “alt,” “longdesc,” or in element content).**

Nontext elements include, but are not limited to: images, graphical representations of text (including symbols), image maps and image map hotspots, animation (e.g., animated GIFs), applets and programmatic objects (including Java<sup>56</sup>), ASCII art, frames, scripts, images used as list bullets, spacers and line images, graphical buttons, stand-alone audio files, audio tracks of video, and video. Text equivalents may take the form of ALT text, longdesc text, d-links, captioning, text transcripts, or audio description.

The most important thing developers can do to increase the accessibility of their work is to use text equivalents consistently and appropriately. ALT text should be meaningful and communicate the purpose of the image in the context of the page’s content (as opposed to the image’s appearance). For example, a green arrow image described as ALT=“green arrow” will not help visually disabled users, but ALT=“Go to Next Section” will help them understand what the image does. When an image or button indicates a navigational action, the ALT text corresponding to the image must communicate the purpose of the image rather than its appearance.

- (B) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.**

For those who are deaf or hard of hearing, audio content is not accessible. Videos without audio descriptions are not accessible to the blind. In both cases, the information needs to be provided in an alternative format. Important information contained in audio or video also needs to be available in text.

- (C) Web pages shall be designed so that all information conveyed with color is also available without color, for example, from context or markup.**

Information must not be conveyed through color alone. For example, do not write, “Please select an item from those listed in green.” Instead, make sure information is available through other style effects (e.g., **bold** or *italic*) and through context (e.g., comprehensive text links).

---

<sup>56</sup> A programming language for the Internet, typically written in the form of short scripts that are platform-independent.

Color alone should not be used to indicate an action (e.g., “Push the green button to begin”). A colored button or icon that is used to initiate action must also include an appropriate text description.

**(D) Documents must be organized so pages are readable without requiring browser support for style sheets.**

When style sheets are used in a document, content must still be accessible when style sheets are turned *off* or are not supported. Style sheets should not be used to generate dynamic content or create styles that could cause content to be illegible if style sheets are disabled or not supported.

CSS describes how documents are presented—on screens, in print, in synthesized speech, or in Braille. By attaching style sheets to HTML documents on the Web, authors and users can both choose how documents are presented.

**(E) Redundant text links must be included for each active region of a server-side image map.**

Server-side image maps are not accessible to anyone using a text browser or a browser with images turned off. When a user clicks on a location in a server-side image map, the map only specifies the *coordinates* in the image at the moment the mouse is clicked. The server must decipher which link has been selected. This means that no text description can be used at the HTML code level. When a server-side image map is used to illustrate a series of options or a navigation menu, the user’s browser cannot tell him or her which URL will be followed when an area of the map is clicked. So redundant text links (i.e., text links that repeat the links in the image map) are used to provide access to the page for users who cannot see the map or use a mouse to click on it. The image itself should also include a text equivalent.

**(F) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.**

With a client-side map, each active region in the image is assigned its own link. Because individual links are assigned to specific areas of the image, ALT text can be used to provide a text equivalent for the active areas in the map. So client-side image maps are preferred to server-side maps for purposes of accessibility. All linked areas within the image map must include text equivalents.

**(G) Row and column headers shall be identified for data tables.**

**(H) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.**

If a table is used to lay out non-tabular content (i.e., as a layout table), some assistive technology will not be able to read the content. There are currently no workable alternatives to using tables for layout, however. Provided that structural markup is not used for formatting purposes (i.e., using TH to achieve a formatting effect) and the tables are structured correctly, tables can be used for layout.

There are four primary rules for creating accessible layout and data tables:

- Do not use tables for layout unless the table makes sense when linearized (presented in a *linear* format). If the table does not make sense when linearized, the content must be presented in an alternate format.
- If a table is used for layout, do not use structural markup (such as TH) to achieve visual *formatting* effects.
- In *data* tables, identify row and column headers using the TH element.
- If your data table has two or more levels of row or column headers, make sure headers and table cell content are associated in the table markup.

**(I) Frames shall be titled with text that facilitates frame identification and navigation.**

If clearly labeled, frames can be helpful to users of screen readers and other assistive technologies. Frames should be labeled using the “title” attribute in the FRAME element. Frame titles like “top,” “bottom,” and “left” do not tell the user anything about the content in the frame. Instead, use titles like “navigation bar” or “main content” to help the user identify the content of the frame and where it is located.

**(J) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 59 Hz.**

Users with cognitive or visual disabilities may not be able to read text that is moving quickly. Individuals who use screen readers cannot read moving text at all. For someone with a cognitive disability, movement may be so distracting that the entire Web page becomes incomprehensible.

People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 2 to 59 flashes per second (Hertz) range, with a peak sensitivity at 20 flashes per second, or by quick changes from dark to light (like strobe lights). This can be a problem when certain animated effects are used. Avoid flickering images or offer users two versions of the same content. Never use the BLINK or MARQUEE elements.

- (K) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.**

Content developers should resort to alternative pages only when other solutions fail, because alternative pages are generally updated less often than primary pages. An out-of-date page may be as frustrating as one that is inaccessible. Automatically generating alternative pages may lead to more frequent updates, but content developers must be careful to ensure that generated pages always make sense and that users are able to navigate a site by following links on primary pages, alternative pages, or *both*. Before resorting to an alternative page, reconsider the design of the original page; making it accessible is likely to improve its usability for all users. Text-only pages may increase accessibility for users with visual disabilities, but they may not increase accessibility for users with motor, cognitive, or hearing disabilities.

- (L) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.**

Scripts must be readable by assistive technologies. If functional text is not included with a script, a screen reader may read the text of the script itself as a mixture of numbers and letters that will make no sense to users. Text must be provided that communicates what the script is displaying and its purpose in the page.

Some of the functionality provided by scripts can improve usability or, at least, not negatively affect *accessibility*. For example, JavaScript mouseovers<sup>57</sup> used for visual effects, such as highlighting, are not an issue for accessibility for the blind and may improve usability for the learning-impaired.

- (M) When a web page requires that an applet, plug-in, or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with software provisions (a) through (l).**

When a user must have a plug-in or applet installed on their system in order to read a file, the required plug-in must meet software accessibility guidelines. If the plug-in is not accessible, then the file should be available in an alternate format.

Software provisions are part of Section 508, Sub-part B, 1194.21 (a) through (l) of the Federal Rehabilitation Act, and they require the following:

- (a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

---

<sup>57</sup> Effects to a button or link such that when the user hovers the mouse over the item, it changes (e.g., shifts color). Also known as rollovers.



- (b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.
- (c) A well defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.
- (d) Sufficient information about a user interface element including the identity, operation, and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.
- (e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.
- (f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.
- (g) Applications shall not override user-selected contrast and color selections and other individual display attributes.
- (h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.
- (i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
- (j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.
- (k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.
- (l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

- (N) When electronic forms are designed to be completed online, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.**

Screen readers have trouble interpreting forms when labels and controls are placed in different table cells, or labels are located at a distance from the controls themselves. Form labels and controls should be explicitly associated in markup.

This provision does not forbid the use of scripts or plug-ins, but the developer is responsible for providing accessible versions of any scripts or plug-ins he or she uses.

- (O) A method shall be provided that permits users to skip repetitive navigation links.**

When a navigation bar appears at the top of a page and/or down the left side of the page, users who are reading the page with speech synthesis must listen to all the navigation links before they hear the main page content. After a few pages, this becomes very repetitive. Users who are sighted can ignore the links and go right to the main content. It is therefore helpful if a link is provided to skip the user over the navigation links.

- (P) When a timed response is required, the user shall be alerted and given sufficient time to indicate that additional time is necessary.**

Some users with motor disabilities may not be able to respond within a time limit. Some users with visual disabilities may also take longer to respond due to difficulty with page layout or other concerns. The system should alert users when time is running out through an error screen or dialog box. The user should be given an opportunity to respond and tell the system that additional time is required.

**Appendix H**  
**Design Guidelines**

## Appendix H Design Guidelines

---

The following design guidelines are most useful when designing the architecture and the look of the site, and should be taken into consideration as early as possible.

The table below describes which sections should be considered at each stage of development:

Section	Web Development Stage
<b>Navigation</b>	User and Task Analysis Wireframes Concept Prototyping Graphic Composites
<b>Menus and Toolbars</b>	User and Task Analysis Wireframes Concept Prototyping Graphic Composites
<b>Fonts</b>	Graphic Composites HTML Templates HTML Production
<b>Form Elements and Controls</b>	Concept Prototyping HTML Templates
<b>Mechanisms for Feedback</b>	Graphic Composites HTML Templates Programming
<b>Color</b>	Graphic Composites HTML Templates
<b>Web Style</b>	Concept Prototyping Graphic Composites HTML Templates HTML Production Programming
<b>Graphics</b>	Graphic Composites HTML Templates Programming

## Navigation

- **Make navigation easily learned and consistent.** The user will give up if he or she must spend too much time learning to navigate the site. Make the navigation easily learnable and clear, and then apply it consistently throughout the site.
- **Navigation must demonstrate its context.** On the Web, navigation must clearly indicate where users are, where they have been, and how to get back where they started. An example of this would be a grayed-out button in a navigation bar, which is often used to illustrate a user's present location.
- **Do not make the user go through too many steps to get to content.** If users must click through two or three pages before arriving at the information they need, they may give up. Too many levels and sub-levels may be confusing. Users should not have to click through more than three screens to reach their goal.
- If a series of data entry forms is used, **clearly indicate the user's present location** in the series, and how many screens the user must complete before data entry is complete.
- **Provide feedback.** Page navigation and controls must demonstrate that the user's situation has changed. The user needs feedback to understand what he or she has done, and what effect it has had.
- **Always provide a link back to the home page.**
- **Provide different means of navigating the site.** In addition to top and side navigation menus, provide a site map or index. If possible, provide a "breadcrumb trail" (a series of links at the top of a Web page that demonstrates how users arrived at their current location).
- **Provide an email contact link on every page of the site.** The link may lead to a feedback form, or may simply be a mailto link. Users must have contact information on every page of the site.

## Menus and Toolbars

- **Use succinct and clear menu titles.** Choose titles that clearly indicate page content.
- **Limit the number of items in a global navigation menu.** Try to keep menu items to nine or less. A number of subheadings may fit under each item in the menu.
- **Use drop-down list sparingly.** Drop-down lists are used on the Web for a variety of purposes, including allowing users to fill in forms and providing lengthy drop-down lists for navigational purposes. Users often find interactive drop-downs that change as they make selections confusing.

- **Disable menu items when they are unavailable.**
- **Do not use navigation menus to jump users to another site or open an email link.** Global navigation links should open pages within the site.
- **Order menu items appropriately in terms of importance to the user.** Organize related menu items into groups.
- **All navigation menu items must be accessible by keyboard.** The user must be able to use the Tab key to access menu items. This does not mean that JavaScript menus may not be used, only that the menu items must also be available elsewhere in HTML.
- **Establish a tab order from top to bottom and left to right of the page.** This is rarely a problem on the Web, but pay attention to the structure of layout tables so the tab order of menu links makes sense, and groupings of links are not lost.
- **Use standard navigation menus on all pages.** The top-level navigation menu should be available from any page on the site. This will brand your content, and provide users with a consistent means of moving about within the site.

## Fonts

- **Avoid overspecific use of fonts in Web site or system design.** It is often best to accept a default font that will work on all users' machines. If a specific font is used, list alternatives to increase the probability that one of the fonts will be available on the user's machine. Limit the number of typefaces used in an interface. Too many fonts can contribute to visual clutter.
- **Avoid using uppercase text unless it is in reference to a database variable or HTML code.** On the Web, uppercase text implies emphasis.
- **Do not override the user's font selections.** Use style sheets to format text wherever possible. Cascading Style Sheets (CSS) make it possible for the developer to separate content from presentation. If style sheets are used, the text must still be legible even if they are turned off.
- **Avoid using bold and italic.** Use bold and italic text occasionally for emphasis. Use it rarely and consistently.

## Form Elements And Controls

- **Establish consistent system or site navigation**, with a link to the main menu on all pages.
- **Include the organization or department name on all pages.**
- **Include brief directions** (if possible, fewer than 100 words) on query use on the first screen of the system. Educate the user through hypertext links and context-sensitive help.
- **Allow users to refine results without initiating a new query.**
- **Preserve the user's work** (especially with pick lists and text entry boxes) so users do not have to start over or review past entries.
- **Use the appropriate form element for a task.**
  - Use checkboxes for toggling between two choices. If the checkbox is checked, the option is selected. If it is not checked, the option is not selected.
  - Use drop-down combo boxes when users should make one selection from a series of options. A single selection drop-down box is appropriate when the user may only select one item from a preset list, and the user cannot add new items to the list.
  - Use multiple selection checkboxes when more than one selection is available, and the user may not add new items to the list.
  - Do not use input and submission methods that cannot be used with older model browsers and assistive technology, or provide an accessible alternative.
- **Provide accelerators for expert users and simplified elements and controls for novices (e.g., provide complex query construction for researchers and static tables for the general public).**
- **Use standard controls. Maintain system conventions when creating or using controls.**

When a series of buttons is presented in a screen or form (e.g., Add, Cancel, and Apply), it should be presented in the same order on every screen on which they appear. Consistency in presentation will increase the usability of system forms. Use consistent capitalization and justification on all submit image buttons.

In Web-based forms, explicitly associate form controls with their labels using HTML markup. This can be achieved using the LABEL element.

- **Required fields must be indicated visually through an image or character.** For example, required fields could be marked with a red asterisk. The “required field” indicator must also be available to users who cannot see the page.

## Mechanisms For Feedback

Minimize error messages by preventing errors whenever possible. To minimize user errors:

- Disable invalid menu items and form controls, and make the inaccessibility of the item or control obvious to the user.
- Inform the user if incorrect data entry has occurred, so he or she can correct the error.
- Where possible, present lists of options to prevent data entry errors.

If an error occurs, provide a clear and simple error message that explains how the problem may be fixed. Good error messages tell the user what the problem is and how to fix it. They also provide information on who to contact if the error cannot be corrected. Keep it brief and to the point.

- **Do not use humorous or sarcastic language in error messages.** Do not imply that the error is the user's fault. Do not use system error language or error numbers.
- **Feedback on status.** Provide visual feedback for available and unavailable selections. If a form field is unavailable for data entry, the system should indicate that.
- **Let users know when they are about to be timed out of a system.** Give them the option of requesting additional time.

## Color

- **Do not override users' color selections.** This is especially important for users with impaired vision, who may need to view pages in high contrast.
- **Make sure background colors and text provide sufficient contrast for legibility.** There must be a 90 percent contrast difference to read text clearly. A light background with dark text is easiest to read.
- **Use color to emphasize important items and draw attention to them. Use it sparingly.** Do not use color alone to create the emphasis or draw attention. Do not use color on unimportant background items, as it may distract users.
- **Use color to group similar items and differentiate between dissimilar items.** Contrasting colors can emphasize difference, while using the same color on a series of pages can demonstrate that they are related. Also use some means other than color (e.g., text) to indicate the relationships.
- **Limit use of colors in color-coding. Do not attach meaning to more than three colors,** as users will not be able to remember the meanings. The more colors used, the more distracting and less effective they are.
- **Test colors** on different monitors and different browsers, with different settings, to check legibility and effectiveness.



- **Use browser-safe colors.** The browser-safe palette contains 216 colors out of a possible 256. The palette is useful for flat-color graphics and logos, and areas in an image with a lot of a single color.

## Web Style

- **Titles and headings.** Use titles, page headings, and subheadings consistently in Web pages and systems. Structure articles with two or three levels of subheadings, since subheadings help assistive technology interpret the page content. Consistency in headings and titles will also help users navigate and provide a meaningful title for bookmark pages.
- **Use page titles.** Page titles are included in the TITLE element of an HTML document. The page title is generally the first thing users see when the page loads, and the title will show up as the text of any bookmark the user sets.
- **Page titles should not be longer than 40 to 60 characters.** Page titles should not all begin with the same word. Different pages should use different titles; the same title should not be used for every page of a site or a section of a site. Eliminate articles (a, an, and the) from page titles.
- **Include the name of the organization or Web site in each page title.** The title should be a concise and plainly worded description of page contents. A unique title should be used on each page. New information should precede old information (e.g., the title of the page would appear before the name of the site, if both were included).
- **Heading text has to make sense when the rest of the text is unavailable.** Clearly explain the content of the page or article. Using different levels of subheadings will increase usability for individuals with disabilities.
- **Users don't like to read online.** If the article is longer than half a page, users may print it out rather than reading it on the computer screen. Documents written for reading on the Web must be concise and formatted to facilitate scanning. The "inverted pyramid" writing style used in journalism, where important information is placed at the beginning of the first sentence so it can be easily found, works well. Developers can also highlight keywords, use meaningful subheadings, use bulleted lists, limit themselves to one idea per paragraph, and load their writing by beginning with the conclusion.
- For Web publishing, usability experts recommend **writing only 50 percent of the content you would for a print publication.**
- **Limit jargon.** Figure out who will read your pages, and target the level of reading difficulty accordingly. Write clearly and succinctly. Avoid hyperbole and marketing language. Avoid metaphor.
- **Avoid using too much markup in paragraphs.** Too many styles of typeface can make the page appear busy.

- **Use hypertext to split up long information into multiple pages.**
- **Do not construct sentences around link phrases like “click here.”** Avoid linking the text “click here”; instead choose meaningful text that indicates the target of the link. For example, “ICD codes include several new revisions.”
- **Inserting multiple links in a paragraph can distract users** from the page text. Only the most germane and useful links should be included in the body of the page text. All other supporting links can be included in groups of footnote links at the bottom of the document.
- If you are not using default link colors, **try to use link colors that match the text color.**
- **Use Spell Check.** Poor spelling or grammar will make users unwilling to trust your content. Be sure text and text equivalents are spellchecked and copyedited.
- **Provide a search feature on documentation pages.** After all, users only access documentation when they are unable to figure something out.
- **Provide examples** to help users implement instructions. Any instructions should be task oriented and demonstrate the steps required to complete the task.
- **Use language familiar to the user** to describe actions or events (e.g., use language from the Windows system screens to describe actions taken within the Windows environment).
- **Repetition will provide users with clear expectations for page layout and behavior.** It will also “brand” the site with a consistent identity. Consistent layout and navigation will help users find what they need on the site, and avoid them having to decipher a new navigation design every few pages.
- **Reuse controls, buttons, and navigation elements** when designing new pages and prototypes. A consistent look-and-feel will increase user comfort with your content.
- **Place logos and navigation menus in a consistent location on the page,** and make them a consistent size throughout the site.
- **Use color consistently.** Make sure there is sufficient contrast between text and background colors so that pages are legible.

English readers read from left to right, and from the top of the page to the bottom. This is the basis for most graphic design of print publications. On Web pages, the top of the page is the most important “real estate” on the screen. The top four or five inches of the page are all that is visible on the computer monitor. Important content should be visible in that space.

## Graphics

- **Graphics should support the content, data, and/or purpose of the system or site.** Do not use graphics for the sake of adding images.
- **Do not use graphics that have negative cultural connotations.** Review and test graphics to be certain that you are not using images that may have different meaning to someone of a different culture. For example, in some cultures a black box around a picture means that the person pictured is dead. The Web is a global medium. Be aware of the messages you are sending.
- Place information on the screen, including graphics, based on a **hierarchy of information**. Ask yourself, “What does the user want to do first? Second? Third?” Place content accordingly.
- **Use an appropriate image format for Web graphics**, either GIF or JPG. Choose the smallest image you can. Keep graphics small; under 24K in size is ideal.
- **Use the fewest images with the smallest byte size that is appropriate to achieve an acceptable download speed.** Large images slow Web page downloads, especially for users with slow Internet connections. Breaking a large image into several smaller images will decrease download times, but if there are too many small images on a page, download times will still be problematic.
- **All Web graphics should be optimized.** Optimizing reduces the bit depth of the image, resulting in a smaller byte size.
- **Specify height and width attributes with the IMG element in HTML.** If you do this, placeholder text will appear in the browser window while the image loads. This allows the user to scan page content and complete tasks, whether the images have finished loading or not.
- **Provide a text version of each image using the ALT attribute in HTML.** ALT text communicates the meaning or purpose of an image to users who cannot see the page or cannot see it well. ALT text should not be used to create ToolTip popups in MS Internet Explorer. It is more important that the ALT text describe what the image does than how it appears. If the button is used for navigation purposes (i.e., it is a link), then the ALT text should indicate the target of the link.
- **Avoid using animated GIFs.** Animated GIFs can distract users from important page content.
- **Do not import graphic elements from another Web site or print publication.** The images may not be in an appropriate format, and what works in print may not work on the Web.

## **Appendix I**

### **Model Discussion Guide**

# Appendix I

## Model Discussion Guide

---

### Part I: Contact Information

The first few questions are about contact information. This information will not be distributed to anyone, and will be used only to contact you if we need to follow up for clarification.

1. Name
2. Email Address
3. Work phone
4. Determine organizational affiliation within [institute or agency name]? Are you:
  - An [institute or agency name] employee (specify branch, division, and/or office)
  - A contractor with [institute or agency name] (specify company)
  - Other affiliation

### Part II: Organizational and Architectural Information

The following questions are to get a better idea of the architecture of your system and network.

1. Can you describe the architecture of your system to provide queryable data to the public and public health practitioners? (e.g., How many servers are involved? What functions – applications, databases, Web, etc. – do those servers support?)
2. What are the IP address(es) of the server(s) on which queryable data reside?
3. Where are queryable data housed (on the Web or on a separate server)?
4. How are the data accessed by the system? In other words, is the dataset itself physically housed in the program that provides it, or is it housed elsewhere and the program points or links to it?
5. How many datasets are there?

The following questions are to get an idea of what groups provide support for the system within your organization.

6. Are you primarily responsible for determining what content should be added to (or deleted from) (SYSTEM NAME)? If not, who is? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

7. Are you responsible for Web server hardware and/or operating system software? If not, then who is responsible? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

8. Are you responsible for Web server software? If not, then who is responsible? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

9. Are you responsible for networking or connectivity to the Web server? If not, then who is responsible? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

10. Are you responsible for publishing dynamic Web content (such as multimedia, CGI-scripted, or database- or program-generated pages)? If not, then who is responsible? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

11. Are you responsible for providing scripting or programming support for Web-based queriable data? If not, then who is responsible? (Get Name, Title, Address, Phone)

Can you give me a brief description of their job responsibilities?

### Part III: Web Server Hardware and Operating System

The next few questions are about the server(s), which constitute the system for providing queriable data. This may include more than one server if your system uses a multi-tiered architecture:

1. What is the hardware architecture of the server?

- Intel/AMD/Cyrix PC-compatible
- Mac
- Sun
- SGI
- Non PC-Compatible IBM
- Non PC-Compatible HP
- Non PC-Compatible Digital/Compaq
- Other, please specify: \_\_\_\_\_
- Don't know

2. What is the platform/operating system that runs the server?

- Windows NT 3.51
- Windows NT 4.0
- Windows 2000
- Windows 95/98
- MacOS
- Linux
- Solaris
- Irix
- Other, please specify: \_\_\_\_\_
- Don't know

3. How much RAM is installed on the server?

- Less than 32MB
- 64MB
- 128MB
- 256MB
- 512MB
- 1024MB
- Greater than 1024MB
- Don't know

4. How much disk space is installed on the server?

- Under 2GB
- 2-4GB
- 5-6GB
- 7-10GB
- 11-16GB
- 17-26GB
- 27-42GB
- 43-68GB
- 69-80GB
- More than 80GB
- Don't know

5. Does the server use Redundant Array of Inexpensive Devices (RAID) drives?

- RAID level 0 (striping)
- RAID level 1 (mirroring)
- RAID level 2
- RAID level 3
- RAID level 4
- RAID level 5
- Does not use RAID
- Don't know

6. How often is the server backed up?

- More frequently than daily
- Daily
- Weekly
- Several times a month
- Monthly
- Less often than monthly
- Never
- Don't know



7. If backups are made, what is the primary backup medium?

- Local disk
- Remote disk
- Removable disk [including Zip and Jaz]
- CD-R
- CD-RW
- Other optical
- QIC tape [not including Travan]
- Travan tape
- DAT
- DLT
- Other tape, please specify: \_\_\_\_\_
- Other type, please specify: \_\_\_\_\_
- Don't know

8. How many staff members, either full-time or part-time, support the hardware and operating system for this server?

- Staff members
- Don't know

9. Approximately how many hours per month do all of these staff members spend supporting the hardware and operating system for this server?

- Hours per month for all staff members
- Hours per month per staff member
- Don't know

10. What physical security options are in place for the server (e.g., the server is in a locked room)?

11. What security options are in place for data on the server (e.g., a firewall is in place)?

12. What are the benefits of the hardware used?

13. What are the limitations of the hardware used?

## Part IV: Web Server Software

For one Web server on which queriable data reside (if multiple servers are used, these questions may be repeated):

1. What Web server software is used?

- Apache (or derivatives)
- Microsoft IIS (any version)
- Netscape FastTrack or Enterprise
- O'Reilly Web site Pro
- WebStar
- Other, please specify: \_\_\_\_\_
- Don't know

2. What other servers are installed?

- Real Networks streaming media server
- Chat server
- No other servers are installed
- Other, please specify: \_\_\_\_\_
- Don't know

3. What application environments are supported on this Web server?

- CGI
- Java
- ColdFusion
- StoryServer
- Other, please specify: \_\_\_\_\_
- Don't know

4. Does this Web server support secure data transmission (via HTTPS, IPSEC, or other secure protocols)?

- Yes
- No
- Don't know

IF YES: Via what mechanism?

- HTTPS
- IPSEC
- Other secure protocols, please specify: \_\_\_\_\_
- Don't know

5. How many staff members, full-time or part-time, support this Web server software?

- Number of staff members
- Don't know

6. Approximately how many hours per month do these staff members spend supporting this Web server software?

- Hours per month for all staff members
- Hours per month per staff member
- Don't know

7. What support/security systems are available for the queryable data system(s)?

*PROBE: Are there protection screens in place to ensure that responses to queries do not reveal individual persons or their identities? (Describe how they work.)*

8. What are the benefits of the software used?
9. What are the limitations of the software choices used?

### **Part V: Connectivity**

Please answer the following questions about the server on which the data reside:

1. How is the server connected to the LAN?

- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- Token-Ring
- Other, please specify: \_\_\_\_\_
- No LAN
- Don't know

2. How is the server (or the entire LAN) connected to the Internet?

- Dial-up
- T-1
- T-3
- Other, please specify method and speed  
(For example: Frame Relay, 256Kbp): \_\_\_\_\_
- Don't know

3. Do any firewalls protect the server (or the entire LAN)?

- Yes
- Please specify type or vendor: \_\_\_\_\_
- No
- Don't know

4. Is your connection reliable? Do you experience any interruptions in service? What problems do you have maintaining connectivity?

## Part VI: Dynamic Content

1. For the Web site datasets you are responsible for, how many hours per month do you spend working on publishing this content?

\_\_\_\_\_ Hours per month

2. How many staff members, full-time or part-time, publish content for this Web site?

\_\_\_\_\_ Number of full-time staff  
\_\_\_\_\_ Number of part-time staff  
\_\_\_\_\_ Don't know

3. Approximately how many hours per month do these staff members spend working on publishing the content for this Web site?

\_\_\_\_\_ Hours per month for all staff members  
\_\_\_\_\_ Hours per month per staff member  
\_\_\_\_\_ Don't know

4. Do you validate or test this Web site to ensure that it is accessible to all users?

\_\_\_\_\_ Yes  
\_\_\_\_\_ No. SKIP TO QUESTION 7.  
\_\_\_\_\_ Don't know

5. What tools do you use to validate or test the accessibility of this Web site?

\_\_\_\_\_ Bobby automated validation tool  
\_\_\_\_\_ WWW HTML Accessibility Tool (WHAT)  
\_\_\_\_\_ Text-based browser (such as Lynx)  
\_\_\_\_\_ Assistive technology (screen reader, etc.)  
\_\_\_\_\_ Other, please specify: \_\_\_\_\_  
\_\_\_\_\_ Site is not validated or tested for accessibility issues  
\_\_\_\_\_ Don't know

6. What standards for accessible Web design do you follow when designing or implementing this Web site?

- W3C/WAI Web Content Accessibility Guidelines 1.0 (May 5, 1999)
- Earlier version of W3C/WAI recommendation
- Trace Wisconsin accessibility guidelines
- Other, please specify: \_\_\_\_\_
- Do not use any specific accessibility standards
- Don't know

7. Which of the following technologies does this Web site use?

- SSIs
- DHTML
- XML
- VRML
- Other, please specify: \_\_\_\_\_
- None of these technologies
- Don't know

8. What types of dynamic server-side applications are used in this Web site?

- ASP
- ColdFusion
- JavaScript
- VBScript
- CGI
- Java Servlets
- Other, please specify: \_\_\_\_\_
- None of these applications
- Don't know

9. If CGI is used in this Web site, in what languages are CGI programs written?

- Perl
- C
- C++
- Java
- Visual Basic
- Other, please specify: \_\_\_\_\_
- CGI language is not used
- Don't know

10. What types of multimedia content are used in this Web site?

- Adobe Acrobat PDF
- Flash/Shockwave
- QuickTime
- Real Media (streaming video)
- Real Media (streaming audio)
- .WAV, .AIFF, or other non-streaming audio formats
- Other, please specify: \_\_\_\_\_
- None
- Don't know

11. What tools are used to create this dynamic content?

- Adobe Acrobat Exchange
- Macromedia Flash
- Macromedia Director
- Macromedia Authorware
- QuickTime
- Real Media Encoder
- Sound editing utilities
- Other, please specify: \_\_\_\_\_
- None
- Don't know

12. Are META tags added to the (static or dynamically-generated) HTML pages of this Web site?

- Yes
- No. SKIP TO QUESTION 15.
- Don't know

13. For what purposes are META tags added to the HTML pages of this Web site?

- Authoring tool adds them automatically
- Standard site development policy
- Site search engine uses META tags to display search results
- To improve ranking in Web search engines
- To refresh or redirect a page
- To set a session or permanent cookie
- Other, please specify: \_\_\_\_\_
- Don't know

14. What types of META tags are added to the HTML pages of this Web site?

- Keyword
- Description
- Expires
- Generator
- Set-cookie
- Other, please specify: \_\_\_\_\_
- Don't know

15. What browsers are recommended for optimum viewing of this Web site?

- Lynx
- Netscape Navigator 2+
- Netscape Navigator 3+
- Netscape Navigator 4+
- Internet Explorer 3+
- Internet Explorer 4+
- Internet Explorer 5+
- Other, please specify: \_\_\_\_\_
- None
- Don't know

16. What browser plug-ins are recommended for optimum viewing of this Web site?

- Macromedia Flash
- Adobe Acrobat Reader
- Other, please specify: \_\_\_\_\_
- None
- Don't know



## **Part VII: Data Requirements**

The next several questions are about data formatting issues:

1. How much recoding is needed?
2. How are data validated to ensure that correct information is returned to the user?
3. Are screens in place to prevent certain runs?
4. How do they operate?
5. How much Web space is needed for the program?
6. For the database?
7. Benefits of this setup?
8. Limitations of this setup?

## **Part VIII: Conclusion**

1. What do you believe would be the ideal way to disseminate health data on the Internet? (Hardware, sever software, data management software, user interface)
2. Do you have any system diagrams or documentation you'd be willing to share with us to help us better understand your tool? Are they in electronic format? Would you be willing to email them to us?

## **Appendix J**

### **Glossary**

## Appendix J Glossary

---

### **Accessible HTML Code**

Code that meets the Section 508 requirements of the U.S. Rehabilitation Act.

### **ALT tags**

Used in HTML coding to describe in words any graphic images on a Web page. Assistive technology for persons with disabilities like JAWS for Windows or PowerBraille can read these described images out loud.

### **Back-end**

In a network application, the software that performs a task not visible to the user. For example, the back-end of the system handles security.

### **Back-end database**

This term has two meanings:

A database that performs tasks that the user is not aware of.

A database whose content is accessed through a Web interface.

### **Cascading style sheets (CSS)**

Additional, supporting files that tell a browser what a document should look like in terms of colors, headings, etc. Because they are not linked to a specific type of computer or software but are “platform independent,” style sheets let developers design pages that can be viewed as intended on all computers.

### **Cell suppression**

A technique that can be used to maintain the confidentiality of data about individuals (persons or establishments). The technique “suppresses” (i.e., does not show) data for one or more cells in a table, when the number of individuals in a cell is small enough (e.g., <5) to possibly allow identification of an individual.

### **Checkbox**

Clicking in this small box on the screen is equivalent to making a checkmark in a box on a paper form.

### **ColdFusion (CF)**

Software used to develop database applications.

### **Comments**

Written comments placed by developers in computer code that do not appear on the interface, but are viewable when reading the code itself. Comments are helpful because they help new development project staff read through the code, identify development history, and see context.

## **Common Gateway Interface (CGI)**

A category of interfaces that presents the system's features and functions graphically rather than with a simple prompt. It is the standard for running programs from a Web server.

## **Data element standards**

A rationale by which system developers and managers label each category of data that reside in the database; among other things, standards should provide guidance on how to name each data element, and where those elements should reside in the database structure.

## **Data stewards**

A trained staff member who ensures that certain procedures regarding the data in the system (typically procedures related to security and confidentiality) are followed appropriately.

## **Dialog box**

A window that opens on the screen, asking the user to supply information or choose options.

## **Drop-down list**

A list from which the user can choose at least one item. The current choice is visible in a small rectangle and when the user clicks on it, a list of items is revealed below it. Also known as pull-down list or menu.

## **Dynamic elements**

Elements in a Web page that are updated every time someone requests the page. Examples are the results of a search, or a button that reads Login or Logout, depending on whether the user has already logged in or not.

## **Form elements**

Elements in a Web form such as checkboxes, drop-down list, radio buttons, and text boxes. Form element tags include <INPUT>, <SELECT>, and <TEXTAREA>.

## **Graphic composite**

A rapidly drawn but high-quality sketch intended for presentation. Also known as "graphic comp."

## **Graphic design**

Commercial art, including that produced for Web sites (such as the buttons, logos, and other images).

## **Graphical user interface (GUI)**

The use of images in addition to words on the screen (e.g., icons, buttons, dialog boxes) to provide a picture-oriented way to interact with the Web site.

## **Heuristic evaluation**

Evaluation of a Web site that uses best practices as the standard for assessing the site. The best practices usually include consistency, flexibility, efficiency, simplicity, attractive design, online help, etc.

**Hypertext markup language (HTML)**

The code used to create Web pages.

**HTML Tags**

The means by which programmers create the features or elements of a Web page.

**Image Slicing**

The process by which a Web developer cuts up or crops the graphic composite file into the requisite pieces of the Web page. Each piece is rendered to the user in its proper location within the interface.

**Includes (noun)**

A simple Web server technology that allows one to include, or insert, a second file into an existing file; using includes makes updates and changes to a template very easy.

**Information architecture**

The basic conceptual structure of the system, broken down by content categories and user tasks. How large collections of information are organized in a Web site so that users can conduct searches or carry out tasks easily and successfully.

**Information technology (IT)**

The computer systems in a business or other enterprise.

**Interface**

The images, commands, display formats, and other devices provided by a Web site to let the user communicate with and use the Web.

**Internet Protocol (IP)**

The method by which data are sent across the Internet from one computer to another. IP also designates data by location, as in IP address.

**Java**

A programming language for the Internet, typically written in the form of short scripts that are platform-independent.

**JavaScript**

A simple, cross-platform World Wide Web scripting language (only vaguely related to Java).

**META information/META tags**

Labels that are included in the code of the system so that industry search engines (e.g., Google) can find the system when users enter related keywords.

**Middleware**

Software that manages the interaction between an application and a network.

**Mouseovers**

Effects to a button or link such that when the user hovers the mouse over the item, it changes (e.g., shifts color). Also known as rollovers.

**Perl**

A programming language that is popular among Web developers, especially for sites that involve the processing of forms.

**Platform**

The type of hardware and operating system in a computer, such as Macintosh and IBM compatible.

**Prototype**

Representation of a system for testing purposes and as a way of understanding the potential limitations of design as well as features that work well.

**Radio button**

An HTML form element, often presented in a list, that allows the user only one selection at a time.

**Source code**

The form in which a programmer writes a computer program in a formal programming language.

**Tabular format**

Information presented in tables rather than narrative text.

**Templates**

The shell structure, coded in HTML, ColdFusion, or another language, into which individual page content is inserted. Templates represent a working model of a system in development. They often include a fully functioning GUI and a partially functioning back-end.

**Text box**

An area on a Web screen into which the user can type information.

**Time-out features**

Often used as a security device, time-out features assume that an error has occurred in specified situations and shut down a program. For example, a user's failing to use a program for more than 30 minutes might trigger the time-out feature.

**Usability**

The ease with which users can perform tasks (e.g., searching for information, submitting data).

**Validator**

A computer program that examines HTML documents for syntax errors.

**Wireframe**

A plain-looking graphic that shows a Web page layout (the content, navigation, etc.) It does not show colors or actual images.

*Presented by*

Lela N. Baughman, M.S.W.  
Christopher P. Garrett, M.B.A.  
Amy B. Harris, M.P.A.  
Denise M. Koschmann, M.A.  
Jenifer L. Maroon  
Lawrence E. Scholl, M.P.H.

**ORC** MACRO<sup>SM</sup>

3 Corporate Square, Suite 370  
Atlanta, GA 30329  
Phone: (404) 321-3211  
Fax: (404) 321-3688  
Web Site: [www.macoint.com](http://www.macoint.com)