

Chapter 12. Moving into Implementation



- Managing the programming process.
- Testing.
- Developing documentation.

- The **implementation phase** consists of developing and testing the system's software, documentation, and new operating procedures.
- **Managing the programming process** is the major task of the systems analyst in this phase.
- While programmers work on programming, the systems analyst design a variety of **tests** to ensure that the system does what it was designed to do.
- During this phase, the systems analysts finalize the **system documentation** and develop the **user documentation**.

MANAGING THE PROGRAMMING PROCESS

- The project manager's tasks during the process of programming:
 - assigning programming tasks,
 - coordinating the activities, and
 - managing the programming schedule.

Assigning Programming Tasks

- The project manager first groups together modules that are related, and then assigns the groups of modules to programmers on the basis of their experience and skill level.
- The project manager must deal with a mismatch between the available programming skills and the programming skills that are needed for the project.
- The best size of programming team is the smallest feasible one that can function as independent as possible.

- Coordination can be done through both high-tech and low-tech means.
- The simplest approach is to have a weekly project meeting.
- Another important method is to create and follow standards.
- Many project teams set up three “areas” for programmers:
 - development area,
 - testing area, and
 - production area.

- Implement ***change control*** techniques:
 - keeping files and programs in different places according to completion status,
 - using a ***program log*** to keep track of program changes.
- Many CASE tools are set up to track the status of programs and help manage programmers.

- The initial time estimates must be refined as the project progresses during construction.
- One of the most common causes for schedule problems is **scope creep**.
- Another common cause is unnoticed day-by-day **slippages** in the schedule.
- Typically, a project manager will create risk assessment that tracks potential risks that have an impact on the schedule and costs.

- Classic Implementation Mistakes:
 1. Research-oriented development.
 2. Using “low-cost” personnel.
 3. Lack of code control.
 4. Inadequate testing.

- The attention paid to testing is justified by the high costs associated with downtime and failures caused by software bugs.

Brokerage Service	\$6.4 million
Energy	2.8 million
Telecom	2.0 million
Manufacturing	1.6 million
Retail	1.1 million
Health Care	636,000
Media	90,000
"Assessing the Financial Impact of Downtime," Vision Solutions, 2008, www.visionsolutions.com	

- Testing starts with the tester's developing a **test plan** that defines a series of tests that will be conducted.
- A test plan describes a set of very specific **test cases** to examine, and defines the expected results.
- The tester develops a series of test cases to ensure that the quality of programs is validated.
- There are four general stages of tests: unit tests, integration tests, system tests, and

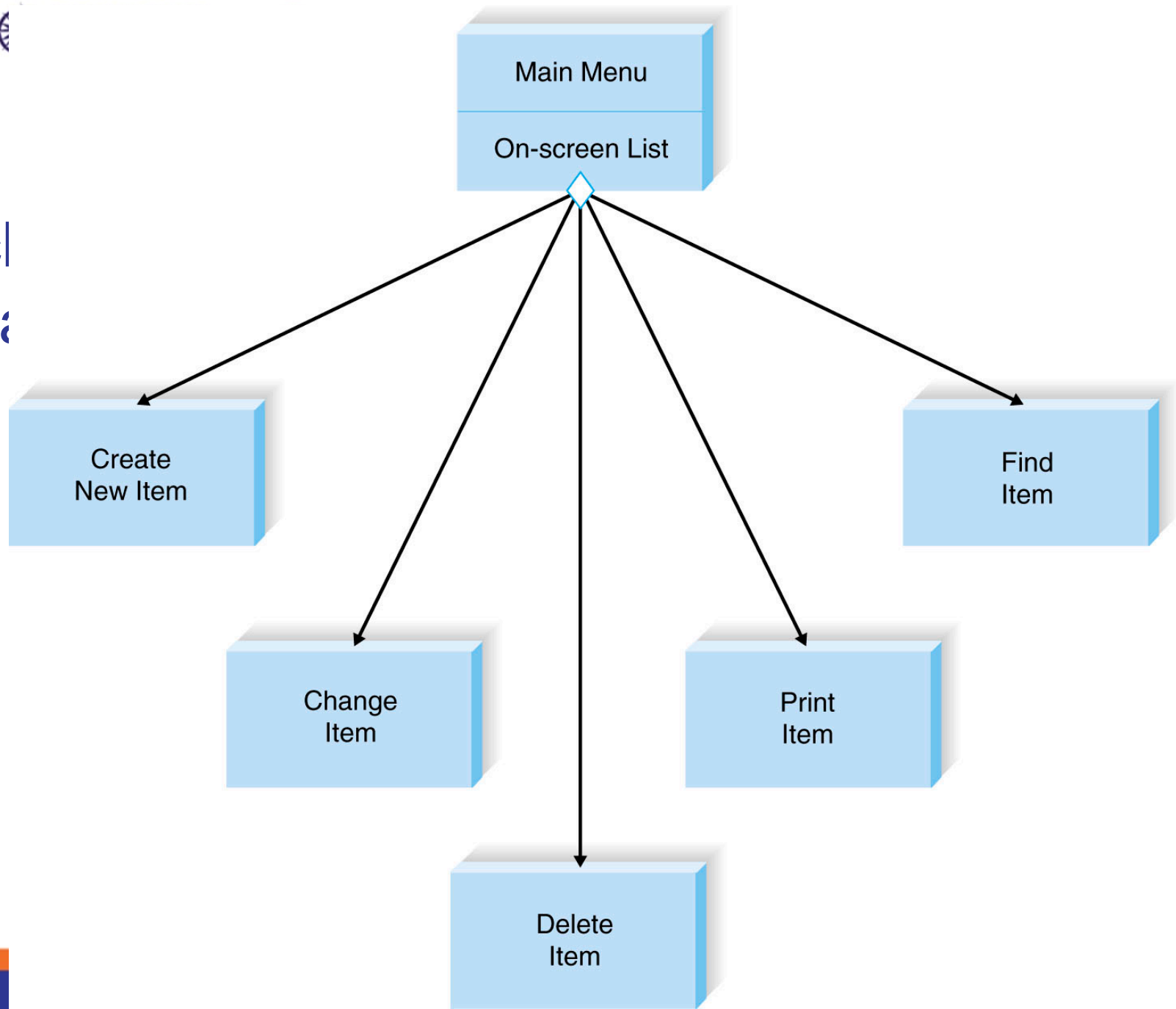
Test plan

Test Plan			Page ____ of ____
Program ID: _____		Version number: _____	
Tester: _____		Date designed: _____	Date conducted: _____
Results: <input type="checkbox"/> Passed <input type="checkbox"/> Open items: _____			
Test ID: _____		Requirement addressed: _____	
Objective: _____			
Test cases			
Interface ID	Data Field	Value Entered	
1. _____	_____	_____	
2. _____	_____	_____	
3. _____	_____	_____	
4. _____	_____	_____	
5. _____	_____	_____	
6. _____	_____	_____	
Script			

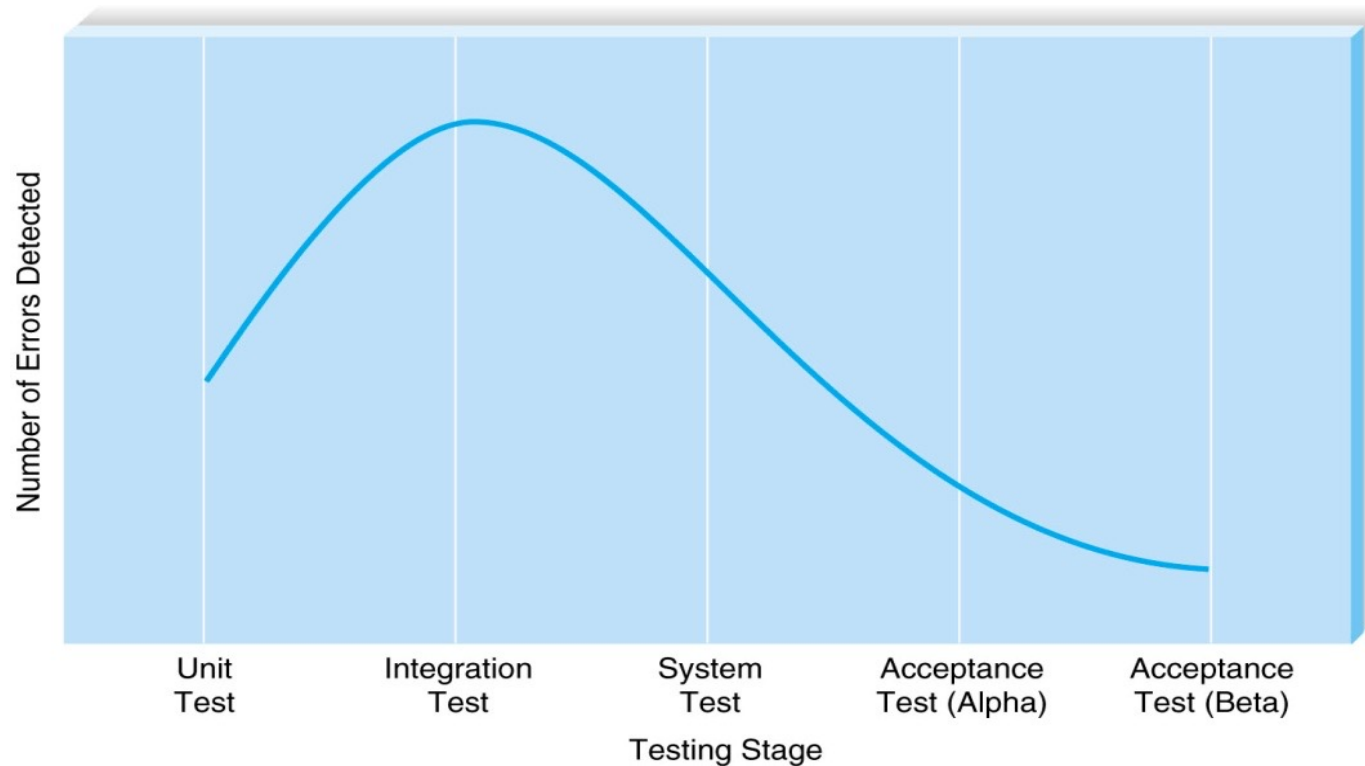
Expected results/notes			

Actual results/notes			

Each
tha



Error Discovery Rates for Different Stages of Tests



Stage	Types of Tests	Test Plan Source	When to Use	Notes
Unit Testing	Black-box testing: treats program as black box.	Program specifications	For normal unit testing	The tester focuses on whether the unit meets the requirements stated in the program specifications.
	White-box testing: looks inside the program to test its major elements.	Program source code	When complexity is high	By looking inside the unit to review the code itself, the tester may discover errors or assumptions not immediately obvious to someone treating the unit as a black box.
Integration Testing	User interface testing: The tester tests each interface function.	Interface design	For normal integration testing	Testing is done by moving through each and every menu item in the interface either in a top-down or bottom-up manner.
	Use scenario testing: The tester tests each use scenario.	Use scenario	When the user interface is important	Testing is done by moving through each use scenario to ensure that it works correctly. Use scenario testing is usually combined with user interface testing because it does not test all interfaces.
	Data flow testing: Tests each process in a step-by-step fashion.	Physical DFDs	When the system performs data processing	The entire system begins as a set of stubs. Each unit is added in turn, and the results of the unit are compared with the correct result from the test data; when a unit passes, the next unit is added and the test is rerun.
	System interface testing: tests the exchange of data with other systems.	Physical DFDs	When the system exchanges data	Because data transfers between systems are often automated and not monitored directly by the users, it is critical to design tests to ensure that they are being done correctly.
System Testing	Requirements testing: tests whether original business requirements are met.	System design, unit tests, and integration tests	For normal system testing	This test ensures that changes made as a result of integration testing did not create new errors. Testers often pretend to be uninformed users and perform improper actions to ensure that the system is immune to invalid actions (e.g., adding blank records).
	Usability testing: tests how convenient the system is to use.	Interface design and use scenarios	When user interface is important	This test is often done by analysts with experience in how users think and in good interface design. This test sometimes uses the formal usability testing procedures discussed in Chapter 9.
	Security testing: tests disaster recovery and unauthorized access.	Infrastructure design	When the system is important	Security testing is a complex task, usually done by an infrastructure analyst assigned to the project. In extreme cases, a professional firm may be hired.
	Performance testing: examines the ability to perform under high loads.	System proposal and infrastructure design	When the system is important	High volumes of transactions are generated and given to the system. This test is often done by the use of special-purpose testing software.
	Documentation testing: tests the accuracy of the documentation.	Help system, procedures, tutorials	For normal system testing	Analysts spot-check or check every item on every page in all documentation to ensure that the documentation items and examples work properly.
Acceptance Testing	Alpha testing: conducted by users to ensure that they accept the system.	System tests	For normal acceptance testing	Alpha tests often repeat previous tests, but are conducted by users themselves to ensure that they accept the system.
	Beta testing: uses real data, not test data.	No plan	When the system is important	Users closely monitor the system for errors or useful improvements.

DFD = data flow diagram.

- **Unit tests** focus on one unit – a program or a program module that performs a specific function that can be tested.
- There are two approaches to unit testing:
 - **Black-box testing**
 - The test plan is developed directly from the program specification.
 - **White-box testing**
 - The tester reviews the actual program code.

- ***Integration tests*** assess whether a set of modules or programs that must work together do so without error.
- There are four approaches to integration testing:
 - user interface testing,
 - use scenario testing,
 - data flow testing, and
 - system interface testing.

- **System tests** are usually conducted by the systems analysts to ensure that all modules and programs work together without error.
- System tests examine
 - how well the system meets business requirements,
 - usability,
 - security,
 - performance under heavy load, and
 - system's documentation.

- **Acceptance tests** are done primarily by the users.
- The goal of acceptance tests is to confirm that the system is complete, meets the business needs, and is acceptable to the users.
- Acceptance testing is done in two stages:
 - **alpha testing** - users test the system using made-up data,
 - **beta testing** – users begin to use the system with real data and carefully monitor the system for errors.

- There are two fundamentally different types of documentation:
 - ***system documentation*** is intended to help programmers and systems analysts understand the system and enable them to build it or maintain it;
 - ***user documentation*** is designed to help the user operate the system.

- User documentation should not be left until the end of the project.
- Time required to develop and test user documentation should be built into project plan.
- On-line documentation is becoming the predominant form.

- There are four key strengths of online documentation:
 - Searching for information is simpler.
 - The same information can be presented in many different formats.
 - It enables the user to interact with the documentation in many new ways.
 - It is significantly less expensive than paper documentation.

- There are three fundamentally different types of user documentation:
 - ***Reference Documents*** are designed to be used when the user needs to learn how to perform a specific function.
 - ***Procedural Manuals*** describe how to perform business task.
 - ***Tutorials*** teach people how to use major components of the system.

- The general structure used in online documentation is to develop a set of ***documentation navigation controls*** that lead the user to ***documentation topics***.
- There are five general types of navigation controls for topics: table of contents, index, text search, intelligent agent, and Web-like links.

Organizing O

Navigation Controls

Contents

Introduction
Basic Features
Finding Albums

Index

Finding
Finding Albums
Finding Artists
Finding Tunes

Text Search

Albums
Announcements
Articles
Artists

Agent Search

Enter a question



Topics

Tasks

How to...

How to...

How to...

Commands

Move...

Copy...

Delete...

Definitions

Full
Search

Music
Category

Credit
Card

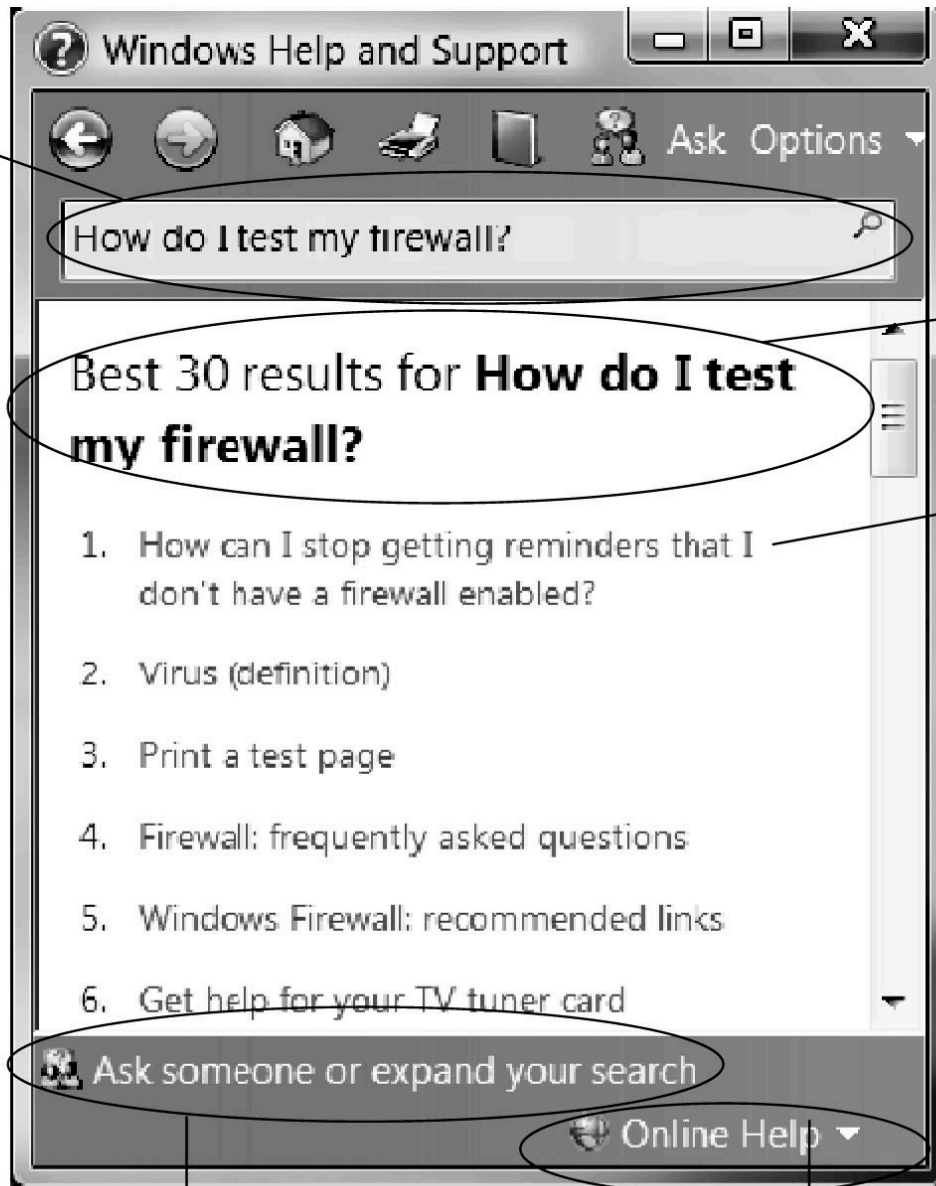
Links

nts

- Topics start with clear titles, followed by introductory text that define the topic, and then provide detailed, step-by-step instructions.
- Many topics include screen images and “show me” examples.
- Most also include navigation controls to enable movement among topics.
- Some also have links to related topics.

AI

Natural language
text search



Search
results

Links to
help topics

Expanded
Search
Resources

Extended
Search on
Web



Guideline Topics

Guideline	Before the Guideline	After the Guideline
Use the active voice: The active voice creates more active and readable text by putting the subject at the start of the sentence, the verb in the middle, and the object at the end.	Finding tunes is done by using the tune title, the artist's name, or a genre of music.	Find a tune by the tune title, the artist's name, or a music genre.
Use e-prime style: E-prime style creates more active writing by omitting all forms of the verb <i>to be</i> .	The text you want to copy must be selected before you click on the copy button.	Select the text you want to copy before you click on the copy button.
Use consistent terms: Always use the same term to refer to the same items, rather than switching among synonyms (e.g., change, modify, update).	Select the text you want to copy. Pressing the copy button will copy the marked text to the new location.	Select the text you want to copy. Press the copy button to copy the selected text. Press the paste button to place the text into the new location.
Use simple language: Always use the simplest language possible to accurately convey the meaning. This does not mean that you should "dumb down" the text, but that you should avoid artificially inflating its complexity. Avoid separating subjects and verbs and try to use the fewest words possible. (When you encounter a complex piece of text, try eliminating words; you may be surprised at how few words are really needed to convey meaning.)	The Georgia Statewide Academic and Medical System (GSAMS) is a cooperative and collaborative distance learning network in the state of Georgia. The organization in Atlanta that administers and manages the technical and overall operations of the currently more than 300 interactive audio and video teleconferencing classrooms throughout the Georgia system is the Department of Administrative Service (DOAS). (56 words)	The Department of Administrative Service (DOAS) in Atlanta manages the Georgia Statewide Academic and Medical System (GSAMS), a distance learning network with more than 300 teleconferencing classrooms throughout Georgia. (29 words)
Use friendly language: Too often, documentation is cold and sterile because it is written in a very formal manner. Remember, you are writing for a person, not a computer.	Blank disks have been provided to you by the operations department. It is suggested that you make backup copies of all essential data to ensure that your data are not lost.	Make a backup copy of all data that is important to you. If you need more diskettes, contact the operations department.
Use parallel grammatical structures: Parallel grammatical structures indicate the similarity among items in lists and help the reader understand content.	Opening files Saving a document How to delete files	Opening a file Saving a file Deleting a file
Use steps correctly: Novices often intersperse actions and the results of actions when describing a step-by-step process. Steps are always actions.	1. Press the <i>customer</i> button. 2. The customer dialogue box will appear. 3. Type the customer ID and press the <i>submit</i> button and the customer record will appear.	1. Press the <i>customer</i> button. 2. Type the customer ID in the customer dialogue box when it appears. 3. Press the <i>submit</i> button to view the customer record for this customer.
Use short paragraphs: Readers of documentation usually quickly scan text to find the information they need, so the text in the middle of long paragraphs is often overlooked. Use numerous separate paragraphs to help readers find information more quickly.		

Source: Adapted from *Writing Software Documentation*, Boston: Aliyn & Bacon, 1998, by T. T. Barker.

Identifying Navigation Terms

- ***Navigation terms*** are used to help users find topics.
- Terms for the index and search engine can come from four distinct sources:
 - The set of the commands in the user interface;
 - The set of major concepts in the system (e.g., data entities);
 - The set of business tasks the user performs;
 - The set of synonyms for the three sets of items mentioned above.

- **Managing programming**
 - The project manager assigns tasks to the programmers, coordinates the program development, manages the schedule, and adjusts the schedule.
- **Testing**
 - A **test plan** contains several tests.
 - A test specifies several **test cases**.
 - Four types of tests: **unit tests**, **integration tests**, **system tests**, and **acceptance tests**.
- **Documentation**
 - **User documentation** and **system documentation** are moving towards online documentation.
 - Three types of user documentation: **reference documents**, **manuals**, and **tutorials**.