

**CAREER  
PATHS**

Virginia Evans  
Jenny Dooley  
Enrico Pontelli



# SOFTWARE ENGINEERING



**Express Publishing**

Published by Express Publishing

Liberty House, Greenham Business Park, Newbury,  
Berkshire RG19 6HW

Tel.: (0044) 1635 817 363

Fax: (0044) 1635 817 463

e-mail: [inquiries@expresspublishing.co.uk](mailto:inquiries@expresspublishing.co.uk)

<http://www.expresspublishing.co.uk>

© Express Publishing, 2014

Design and Illustration © Express Publishing, 2014

First published 2014

Made in EU

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means, electronic, photocopying, or otherwise, without the prior written permission of the publishers.

This book is not meant to be changed in any way.

ISBN 978-1-4715-1930-7

## Acknowledgements

### Authors' Acknowledgements

We would like to thank all the staff at Express Publishing who have contributed their skills to producing this book. Thanks for their support and patience are due in particular to: Alex Newton (Editor in Chief); Sean Todd (senior editor); Steve Miller (editorial assistant); Richard White (senior production controller); the Express design team; Sweetspot (recording producers). We would also like to thank those institutions and teachers who piloted the manuscript, and whose comments and feedback were invaluable in the production of the book.

**Every effort has been made to trace all the copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.**

**CAREER  
PATHS**

# SOFTWARE ENGINEERING

Book

**1**

Virginia Evans  
Jenny Dooley  
Enrico Pontelli



**Express Publishing**

## Scope and Sequence

Unit	Topic	Reading context	Vocabulary	Function
1	The Software Engineer	Course Description	artifact, design, develop, evaluate, install, investigate, programming- in the-large, programming-in-the small, software, test, write	Expressing enthusiasm
2	Types of Computers	Journal Article	computer, computing cluster, desktop, embedded computer, laptop, notebook, PC, server, tablet, workstation	Making plans
3	Accessories and Peripherals	Email	flash drive, flat panel, inkjet printer, keyboard, laser printer monitor, optical mouse, peripheral, scroll wheel, scanner, wireless	Apologizing for an error
4	Inside the Computer	Brochure	case, CD/ DVD drive, cover, fan, hard drive, heat sink, motherboard, port, power supply, processor	Offering advice
5	System Software 1	Textbook chapter	BIOS, control, device driver, firmware, hardware, manually, operate, operating system, system software, window system	Giving a reminder
6	System Software 2	Webpage	antivirus software, deny, firewall, malware, permit, quarantine, removal, security software, spyware	Describing consequences
7	Programming Software	Textbook Chapter	compiler, debugger, IDE, interpreter, linker, program, programming language, programming software, source code editor, text editor	Expressing confusion
8	Application Software 1	Advertisement	accounting, application software, desktop publishing, enterprise, image editing, office suite, spreadsheet, video editing, web browsing, word processing	Politely disagreeing
9	Application Software 2	Journal Article	bioinformatics, cost analysis, data management, digital assistant, mobile app, multimedia player, payroll, route planning, satellite navigation, simulation	Asking for more information
10	The Desktop and GUI	Manual	cursor, desktop, dropdown menu, folder, GUI, icon, open, right-click, run, select	Giving instructions
11	Basic Numbers and Math	Chart	add, equal, divide by, hundred, less, minus, multiply by, over, subtract, times	Making a realization
12	Analyzing Numbers and Quantities	Textbook Chapter	convert, decimal number, denominator, fraction, numerator, out of, percent, percentage, point, reduce	Describing progress
13	Describing Change	Magazine Article	decline, decrease, double, expand, fluctuate, increase, rise, stabilize, steady, trend	Expressing confidence
14	Presentations and Communication	Email	body language, eye contact, handout, note card, presentation, project, review, signpost, summary, visual aid	Giving constructive criticism
15	Education	Webpage	Bachelor's degree, calculus, circuit analysis, computer architecture, computer engineering, control system, electronics, foundation, linear algebra, programming	Describing order of events

# Table of Contents

<b>Unit 1 – The Software Engineer</b> .....	4
<b>Unit 2 – Types of Computers</b> .....	6
<b>Unit 3 – Accessories and Peripherals</b> .....	8
<b>Unit 4 – Inside the Computer</b> .....	10
<b>Unit 5 – System Software 1</b> .....	12
<b>Unit 6 – System Software 2</b> .....	14
<b>Unit 7 – Programming Software</b> .....	16
<b>Unit 8 – Application Software 1</b> .....	18
<b>Unit 9 – Application Software 2</b> .....	20
<b>Unit 10 – The Desktop and GUI</b> .....	22
<b>Unit 11 – Basic Numbers and Math</b> .....	24
<b>Unit 12 – Analyzing Numbers and Quantities</b> .....	26
<b>Unit 13 – Describing Change</b> .....	28
<b>Unit 14 – Presentations and Communication</b> .....	30
<b>Unit 15 – Education</b> .....	32
<b>Glossary</b> .....	34

## Hullward University: Software Engineering Department

### Software Engineering 101: Course Outcomes

This class focuses on computer **software**. It covers various elements of development and programming. The students will learn to **design and develop** programs. The objective is to **write** useful computer software.

Small groups of students will complete several short projects. These focus on **programming-in-the-small**. The whole class will work together on **programming-in-the-large**. This project runs throughout the entire semester.

The students will also **install and test** their own software **artifacts**. This is an

opportunity to **investigate** any software development problems. Finally, students will **evaluate** the correctness of each other's software. Student reviews are part of the final grade.



### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some steps in the process of creating software?
- 2 What are some responsibilities of a software engineer?

### Reading

2 Read the course description. Then, choose the correct answers.

- 1 What is NOT included in the course?
  - A how to write software
  - B steps for investigating problems
  - C the history of software development
  - D testing other students' software
- 2 What will the students do for each other?
  - A adjust development plans
  - B recommend career paths
  - C install software
  - D evaluate performance
- 3 What is true of the programming-in-the-small project?
  - A It is the first step in writing a program.
  - B It involves small groups of students.
  - C It deals with the main framework of a program.
  - D It is used to install programs.



## Vocabulary

3 Match the words (1-8) with the definitions (A-H).

- |                  |              |
|------------------|--------------|
| 1 __ evaluate    | 5 __ design  |
| 2 __ software    | 6 __ develop |
| 3 __ investigate | 7 __ install |
| 4 __ write       | 8 __ test    |

- A to form letters and words into sentences or instructions
- B to plan the way that something will be created
- C to bring something from initial conception to action or implementation
- D to carefully study something and assess its qualities
- E to operate something to see whether it works
- F to put something into the place where it will function
- G to get more information about something
- H the programs that perform particular functions on a computer

4 Choose the sentence that uses the underlined part correctly.

- 1 A Programming-in-the-small often creates less complex software.
- B Students must develop problems in order to repair the program.
- 2 A The teacher will install the software's performance.
- B Students are working on programming-in-the-large to create a program with many levels and functions.

5 Listen and read the course description again. What is the difference between programming-in-the-large and programming-in-the-small?

## Listening

6 Listen to a conversation between a student and an instructor. Mark the following statements as true (T) or false (F).

- 1 \_\_ The woman recommends programming-in-the-large.
- 2 \_\_ The man enjoys investigating problems.
- 3 \_\_ The man is nervous about working in groups.

7 Listen again and complete the conversation.

**Student:** Professor Wendell? I'm really interested in 1 \_\_\_\_\_ . But is it a good career choice?

**Instructor:** I think so. You are a good leader. You'd enjoy 2 \_\_\_\_\_ .

**Student:** I agree. I like working in groups.

**Instructor:** You like to 3 \_\_\_\_\_ , right?

**Student:** Yes, I do. But software engineering seems like it could 4 \_\_\_\_\_ .

**Instructor:** It's sometimes challenging when others 5 \_\_\_\_\_ your work. But if you are patient, it is very rewarding.

**Student:** That 6 \_\_\_\_\_ something I can do.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I'm interested in ... / You're a good ...  
It can be ...*

**Student A:** You are a student. Talk to Student B about:

- a career in software engineering
- how it is rewarding
- how it is challenging

**Student B:** You are an instructor. Talk to Student A about a career in software engineering.

## Writing

9 Use the conversation from Task 8 to complete a career advice webpage.

### Is Software Engineering Right for You?

#### Rewards

- Engineers can \_\_\_\_\_ .
- The job comes with opportunities to \_\_\_\_\_ .

#### Challenges

- It can be hard to \_\_\_\_\_ .
- Engineers must \_\_\_\_\_ .

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What kinds of computers are typical for personal use?
- 2 What kinds of computers are typical for business use?

## Reading

2 Read the journal article. Then, choose the correct answers.

- 1 What is the main idea of the article?
  - A recommendations for computer purchases
  - B the challenges of today's software engineering industry
  - C the equipment that a computer company manufactures
  - D technology arising from advances in software development
- 2 According to the article, which of the following is NOT something that software engineers do?
  - A create programs for individual use on PCs
  - B develop complex software to run on government computing clusters
  - C connect many computers to large corporate servers
  - D increase the size of desktops for homes and businesses
- 3 What opinion does the article express about software engineering?
  - A It is expanding more quickly each year.
  - B It is responsible for improving many areas of people's lives.
  - C It is a good area in which to start a successful career.
  - D It is a subject that everyone should be educated about.

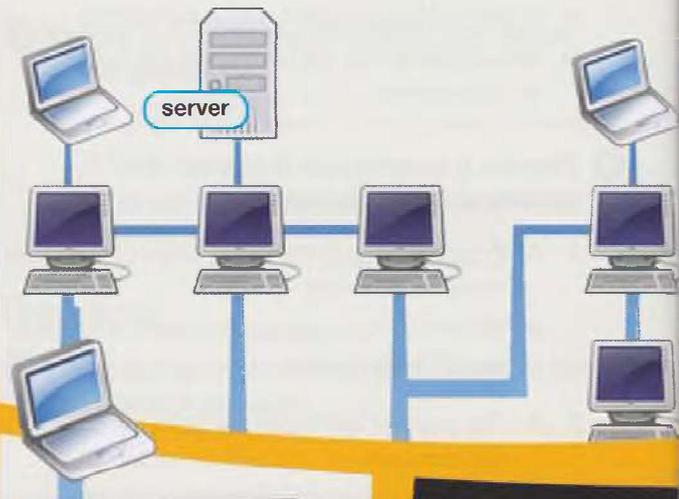
## The Weekly Techie

LET'S FACE IT:  
WE RELY ON SOFTWARE ENGINEERS

They create programs for our home **PCs**. They also develop advanced software for government **computing clusters**. They help us connect multiple **workstations** to massive corporate **servers**.

Home **computers** were a luxury just a short time ago. People felt lucky to have bulky **desktops** in homes and businesses. Now these computers are smaller and more powerful. Many people use **laptops** or **notebooks** instead. **Tablets** provide added mobility. And now people can install **embedded computers** just about anywhere.

This is all possible because of software engineers. The software development industry strives to make life easier. So from all of us at The Weekly Techie: thanks, software engineers!



## Vocabulary

3 Match the words (1-5) with the definitions (A-E).

- 1 \_\_ PC                      3 \_\_ laptop                      5 \_\_ workstation  
2 \_\_ tablet                      4 \_\_ desktop

- A a very small computer that typically does not have a keyboard  
B a hinged computer that is easy to transport  
C a computer that is intended for personal use  
D a powerful computer that processes advanced tasks  
E a computer that is intended for use in one location

4 Read the sentences and choose the correct words.

- The student carried a **desktop / notebook** to class every day.
- The company connected all of its computers to the same **PC / server**.
- Early **computers / laptops** were so large that they occupied entire rooms.
- A **tablet / computing cluster** is more powerful than most other types of computers.
- The company installed **embedded computers / workstations** in employees' cars.

5 Listen and read the journal article again. What is a benefit of using a tablet?

## Listening

6 Listen to a conversation between two engineers. Mark the following statements as true (T) or false (F).

- The woman finished developing a program for desktops.
- The man recommends creating another application for laptops.
- The woman plans to make the program work with a touch screen.

7 Listen again and complete the conversation.

Engineer 1: Hey, Grace. What are you 1 \_\_\_\_\_ ?

Engineer 2: I'm still developing the home banking application.

Engineer 1: Wait, didn't you 2 \_\_\_\_\_ already?

Engineer 2: Well, sort of. I finished a version for 3 \_\_\_\_\_

Engineer 1: So what are you doing now?

Engineer 2: Next, I'm going to create an application for 4 \_\_\_\_\_.

Engineer 1: Oh, that's a good idea. 5 \_\_\_\_\_ carry tablets nowadays.

Engineer 2: Right. That's why 6 \_\_\_\_\_ needs to work well with a touch screen.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

- Didn't you finish ...*  
*What are you doing ...*  
*Next, I'm going to ...*

**Student A:** You are an engineer. Talk to Student B about:

- a program that he or she is developing
- the types of computers that the program currently works on
- the types of computers that the program will work on

**Student B:** You are an engineer. Talk to Student A about a program that you are developing.

## Writing

9 Use the conversation from Task 8 to complete the project extension request.

**Brown & Steele  
Software Development:**

### Project Extension Request Form

Project: \_\_\_\_\_

Current Progress: So far, I developed the program for use on \_\_\_\_\_.

Reason for Extension: I would like to develop the program for use on \_\_\_\_\_ because \_\_\_\_\_.

An important feature of the new version will be \_\_\_\_\_.



To: l.carmichael@rjdtechcorp.com

From: p.rossini@worldwidecompsupply.com

Subject: Re: Your Order (#25841)

Dear Ms. Carmichael,

Thank you for choosing **peripherals** from Worldwide Comp Supply. Please ensure the following information is correct:

Category	Quantity	Description
Monitors	24	XR60 <b>flat panel</b> screen (17-inch)
	12	XR90 flat panel screen (24-inch)
Attachments	36	S740 QWERTY <b>keyboard</b>
	30	SL90 <b>optical mouse</b> with <b>scroll wheel</b>
	6	SL90W <b>wireless</b> optical mouse with scroll wheel
Printers	4	P1070 black-and-white <b>laser printer</b>
	4	PI66 photo-quality <b>inkjet printer</b> with a built-in <b>scanner</b>

Congratulations! This order qualifies for eight FREE storage devices. A package of ShurStore 4GB **flash drives** is included in your shipment. Thank you for your business!

Sincerely,  
Paul Rossini  
Worldwide Comp Supply

### Get ready!

- Before you read the passage, talk about these questions.
  - What computer accessories are used to input information?
  - What computer accessories are used to display information?

### Reading

- Read the email. Then, mark the following statements as true (T) or false (F).
  - The order includes monitors in two different sizes.
  - Some of the optical mice are not wireless.
  - The customer requested an extra package of flash drives.

### Vocabulary

- Fill in the blanks with the correct words and phrases from the word bank.

#### Word BANK

wireless flat panel laser printer  
peripherals keyboard scroll wheel

- The mouse is \_\_\_\_\_, so the user doesn't have to plug it in.
- If the \_\_\_\_\_ breaks, it's difficult to move up and down on the screen.
- Today's \_\_\_\_\_ monitors are much more popular than the old, rounded ones.
- The student types fifty words per minute on his \_\_\_\_\_.
- The new \_\_\_\_\_ produces clear, precise text on each page.
- A monitor and a mouse are types of \_\_\_\_\_.



4 Place the correct words and phrases from the word bank under the correct headings.

**word BANK**

inkjet printer    optical mouse  
flash drive    monitor    scanner

Displaying information	Inputting information	Storing information
_____	_____	_____
_____	_____	_____
_____	_____	_____

5 Listen and read the email again. What are some different features that an optical mouse might have?

**Listening**

6 Listen to a conversation between a representative and a customer. Choose the correct answers.

- What is the main idea of the conversation?
  - A placing a new accessory order
  - B correcting an error in an accessory order
  - C confirming the details of a recent accessory order
  - D reviewing the costs of an accessory order
- What product will the woman receive for free?
  - A extra wireless mice
  - B packages of flash drives
  - C a QWERTY keyboard
  - D a laser printer

7 Listen again and complete the conversation.

**Representative:** Thanks for calling Worldwide Comp Supply. How 1 \_\_\_\_\_ you?

**Customer:** Hi, I'm calling about order number 25841. Some of the order 2 \_\_\_\_\_.

**Representative:** 3 \_\_\_\_\_ to hear that. What's the problem?

**Customer:** I ordered thirty-six 4 \_\_\_\_\_. Thirty were regular, and six were wireless.

**Representative:** I see that on your order confirmation, 5 \_\_\_\_\_ were in the shipment?

**Customer:** Well, I received all thirty-six mice. But 6 \_\_\_\_\_ are wireless.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I'm sorry about ...*  
*I ordered ... but ...*  
*I'll throw in ...*

**Student A:** You are a representative. Talk to Student B about:

- his or her accessory order
- an error in the shipment
- how you will resolve the problem

**Student B:** You are a customer. Talk to Student A about an error in your accessory order.

**Writing**

9 Use the conversation from Task 8 to complete the order correction request.

**Order Correction Request**

Customer: \_\_\_\_\_

Order Number: \_\_\_\_\_

Description of Problem:  
The customer was supposed to receive \_\_\_\_\_  
Instead, she received \_\_\_\_\_

Correction Needed:  
We need to \_\_\_\_\_

Please describe any special offers you made to the customer: \_\_\_\_\_



Thank you for purchasing the Spirtec X900 Desktop Computer! Take a moment to learn about your new computer:

### Exterior

- The X900 features a durable metal **case**. This protects it from typical wear and damage.
- Every **port** has a fitted rubber **cover**. These prevent interior dust buildup.

### Interior

- Your computer's **power supply** connects to the **motherboard**. This is what drives the X900's powerful **processor**!
- All that power produces heat. That's why your X900 has an efficient **fan** and **heat sink**. These keep internal temperatures at a safe, low level.
- You'll find lots of data storage on the **hard drive**. Also, transfer files with the built-in **CD/DVD drive**.

## Get ready!

- ① Before you read the passage, talk about these questions.

- 1 What features protect the interior of a computer?
- 2 What parts of a computer allow it to function?

## Reading

- ② Read the brochure. Then, mark the following statements as true (T) or false (F).

- 1  The computer's ports are protected from interior dust buildup.
- 2  The computer has a display that shows its internal temperature.
- 3  A CD/DVD drive can be purchased separately.

## Vocabulary

- ② Read the sentence pairs. Choose which word or phrase best fits each blank.

1 cover / processor

- A A \_\_\_\_\_ prevents a computer from getting dirty.  
 B The new \_\_\_\_\_ is faster than the old one.

2 port / case

- A The computer's \_\_\_\_\_ protects it from damage.  
 B The cable plugs into the \_\_\_\_\_.

4 Place the correct words and phrases from the word bank under the correct headings.

**word BANK**

heat sink    hard drive    power supply  
motherboard    CD/DVD drive    fan

Cooling	Routing Energy	Accessing Data

5 Listen and read the brochure again. What parts of a computer prevent it from overheating?

**Listening**

6 Listen to a conversation between two engineers. Check (✓) the problems with the computer that the engineers identify.

- 1  slow processor
- 2  warm case
- 3  malfunctioning motherboard
- 4  loud fan
- 5  defective heat sink

7 Listen again and complete the conversation.

Engineer 1: Hey, Greta. Have you used the computer in the conference room?  
 Engineer 2: No. Is there a 1 \_\_\_\_\_ with it?  
 Engineer 1: Yeah. I think it's 2 \_\_\_\_\_.  
 Engineer 2: Oh, that's not good. What's it doing?  
 Engineer 1: Well, the processor is working 3 \_\_\_\_\_. And the case feels unusually warm.  
 Engineer 2: Does it 4 \_\_\_\_\_ of noise?  
 Engineer 1: Now that you mention it, 5 \_\_\_\_\_ did seem really loud.  
 Engineer 2: It sounds like you're right. If I were you, I'd replace the 6 \_\_\_\_\_.  
 Engineer 1: Yeah, it's probably defective. I'd better give that a try.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Is there a problem ...*  
*If I were you ...*  
*I'd better ...*

**Student A:** You are an engineer. Talk to Student B about:

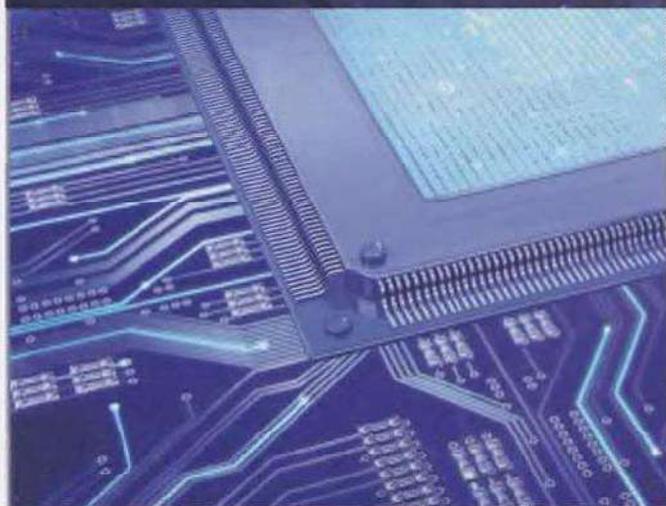
- a possible problem with a computer
- why you think the problem exists
- how to solve the problem

**Student B:** You are an engineer. Talk to Student A about solving a problem with a computer.

**Writing**

9 Use the conversation from Task 8 to complete the repair request.

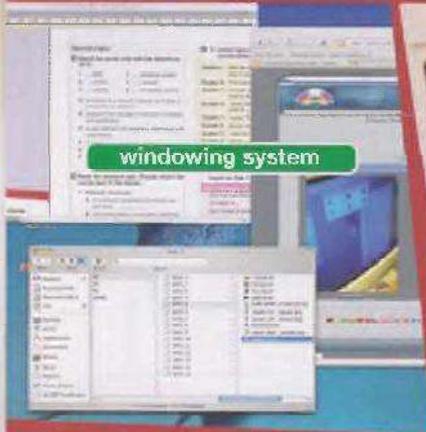
**Computer Repair Request**



Machine: \_\_\_\_\_  
 Describe problem: I noticed that the computer \_\_\_\_\_  
 \_\_\_\_\_ and \_\_\_\_\_.  
 I think the problem is \_\_\_\_\_.  
 Request: The computer needs \_\_\_\_\_.

## Get ready!

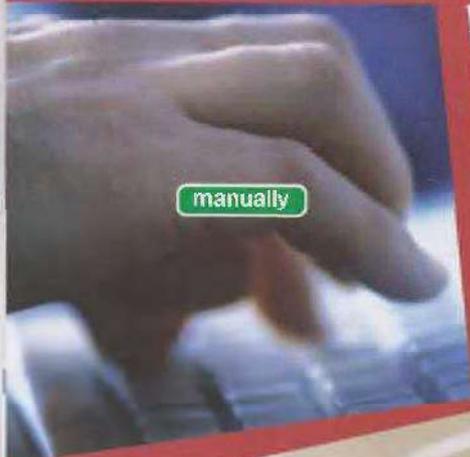
- 1 Before you read the passage, talk about these questions.
- 1 What types of software allow computers to operate?
  - 2 How do users manage a computer's hardware and software?



windowing system



BIOS



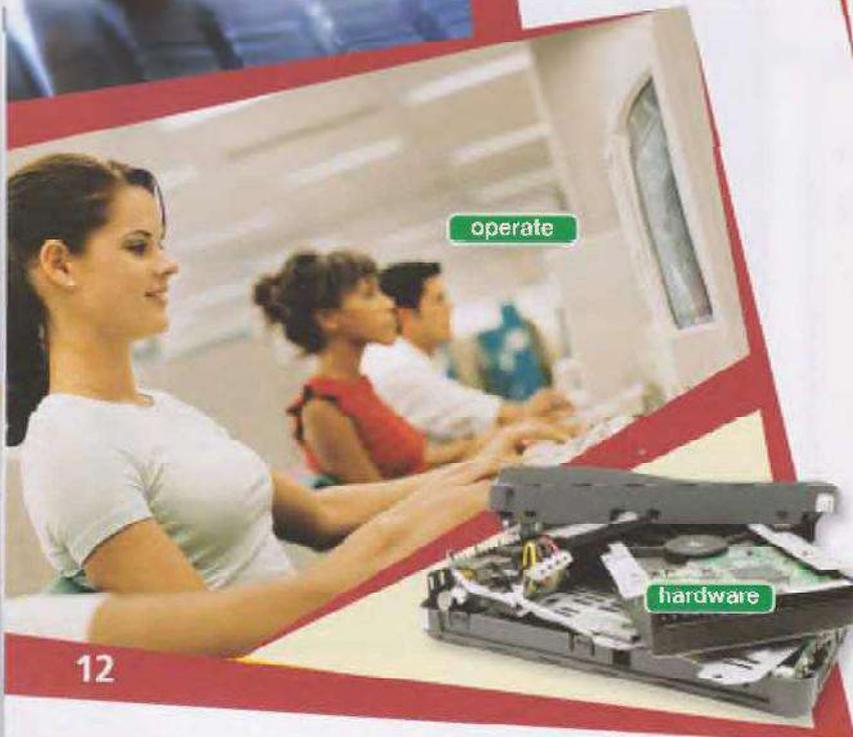
manually



operating systems



Mac OS X



operate

hardware

## Chapter 3

## System Software

A system software is any program that allows computers to perform basic operations. The earliest computers **operated** with only basic software support. Users **manually** entered commands into a computer. This required specialized knowledge and lots of patience.

Now, computers come with sophisticated **operating systems**. These systems manage both the **hardware** and software of a computer. Users **control** their computer's operations easily with **windowing systems**. These allow even beginners to perform complicated operations.

Most modern computers come with pre-installed **device drivers**. These control the unit's operating system. Other components ensure that peripherals work with the operating system. Sophisticated **BIOS** in the **firmware** performs this function.

## Reading

- 2 Read the textbook chapter. Then, choose the correct answers.

- 1 What is the main idea of the chapter?
  - A differences between old and new system software
  - B costs of developing different types of system software
  - C methods for developing system software
  - D types of jobs available for system software developers
- 2 Which of the following is a system software NOT used for?
  - A managing a computer's hardware
  - B allowing a user to enter commands into a computer
  - C letting a computer run applications
  - D installing firmware on a computer
- 3 What is the benefit of a windowing system?
  - A It manages a computer's device drivers.
  - B It simplifies computer operations.
  - C It costs less than basic support software.
  - D It allows users to enter commands manually.

## Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- 1 \_\_\_ BIOS                      4 \_\_\_ operating system  
 2 \_\_\_ control                 5 \_\_\_ manually  
 3 \_\_\_ operate                 6 \_\_\_ windowing system

- A to function in a specific manner according to instructions or software  
 B programs that manage a computer's hardware and applications  
 C a user interface that organizes information into visual boxes  
 D a set of instructions in firmware  
 E to have power over the way something functions  
 F done directly by a person, without automatic functions

3 Read the sentence pairs. Choose which word or phrase best fits each blank.

- 1 **firmware / hardware**  
 A A computer's keyboard and monitor are part of its \_\_\_\_\_.  
 B Information about a computer's operating system is stored in the \_\_\_\_\_.
- 2 **system software / device driver**  
 A The \_\_\_\_\_ dictates how a computer interacts with peripherals.  
 B A \_\_\_\_\_ allows a user to run additional programs on a computer.

5 Listen and read the textbook chapter again. What do operating systems do?

## Listening

6 Listen to a conversation between two students. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The students are discussing scores on a recent quiz.  
 2 \_\_\_ The students completed an assignment on early system software.  
 3 \_\_\_ The man thinks modern operating systems are easier to use than manual systems.

7 Listen again and complete the conversation.

**Student 1:** What did you think of that assignment about early 1 \_\_\_\_\_ ?  
**Student 2:** That was a really 2 \_\_\_\_\_ !  
**Student 1:** I thought so, too. I can't believe people used to enter everything manually.  
**Student 2:** I know what you mean. I'm ready to study 3 \_\_\_\_\_ .  
**Student 1:** I agree. They're so much easier to use.  
**Student 2:** Are you ready for the 4 \_\_\_\_\_ ?  
**Student 1:** I need to review that 5 \_\_\_\_\_ .  
**Student 2:** Don't 6 \_\_\_\_\_ about device drivers.  
**Student 1:** I'm not too worried about that.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*What did you think of ...*  
*I'm ready to ... / Don't forget to study about ...*

**Student A:** You are a student. Talk to Student B about:

- what assignments you found difficult
- what subjects you enjoy studying
- what you will study for an upcoming quiz

**Student B:** You are a student. Talk to Student A about your assignments and an upcoming quiz.

## Writing

9 Use the conversation from Task 8 to complete the quiz on system software.

Write the function of each type of software in the space provided.

Software	Function
operating system	manages hardware, allows a user to perform functions
_____	allows a computer to interact with additional devices
windowing system	_____

QUIZ ON SYSTEM SOFTWARE

HOME

ABOUT US

SERVICES

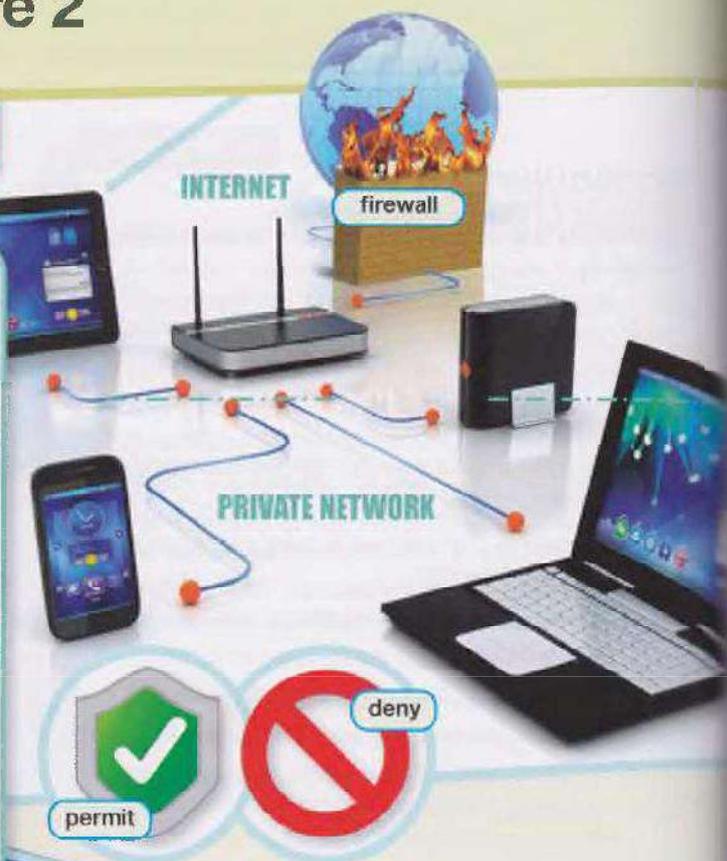
CONTACT

## SHARP ALERT COMPUTING SYSTEMS

Security threats can cause serious damage to your computer. SharpAlert offers excellent **security software** to keep your computer safe.

**Antivirus Software** – Don't let unauthorized parties steal your personal information with **spyware**! And stop **viruses** before they destroy your computer. The SharpAlert Exviro package protects against all types of **malware**. It **quarantines** infected files. Then **threat removal** just takes one click.

**Firewalls** – Remember to update network security programs regularly. SharpAlert's Steelbar firewall **permits** access only to authorized users. It will **deny** access to anyone without proper credentials. Make sure your confidential information stays confidential!



security software

## SHARP ALERT COMPUTING SYSTEMS

### Scan & Clean

Current Progress

Hyper Terminal Icon Ext

removal

Name	Status	Comments
malware		

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some threats to computer security?
- 2 How can users protect computers from security threats?

## Reading

2 Read the webpage. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ The company's Exviro package includes protection against spyware.
- 2 \_\_\_ The antivirus software destroys files that are infected by malware.
- 3 \_\_\_ The webpage recommends using antivirus software instead of firewalls.

## Vocabulary

3 Write a word that is similar in meaning to the underlined part.

- 1 The purpose of the software is to not allow access to unauthorized users.  
\_ e \_ \_
- 2 The system allows access only after users enter their passwords.  
\_ \_ r \_ i \_ s
- 3 The engineer installed a program that protects a computer from various threats on the company's systems.  
s \_ \_ \_ r \_ \_ \_ of \_ \_ \_ r \_ \_

6 Place the correct words and phrases from the word bank under the correct headings.

**Word BANK**

removal malware virus spyware  
quarantine firewall antivirus software

Security threats	Security programs	Protective actions
_____	_____	_____
_____	_____	_____
_____	_____	_____

7 Listen and read the webpage again. What is the difference between spyware and viruses?

**Listening**

8 Listen to a conversation between a company manager and a software engineer. Choose the correct answers.

- What is the main idea of the conversation?
  - A how much damage was caused by a virus
  - B which information was stolen by a spyware program
  - C why the company should update its antivirus software
  - D what caused a failure in the company's firewall
- What prediction does the woman make?
  - A Unauthorized users will attempt to access the network again.
  - B The company's systems will be damaged by a virus.
  - C A new firewall will probably not be effective.
  - D The company's files will need to be quarantined.

9 Listen again and complete the conversation.

Engineer: Well, Mr. Clay, I 1 \_\_\_\_\_ from your company's system.

Manager: Oh, thank you! Our systems contain a lot of 2 \_\_\_\_\_.

Engineer: Yes, I can see that. If you don't update your 3 \_\_\_\_\_, you could really be in trouble.

Manager: Really? You think this will 4 \_\_\_\_\_?

Engineer: Whoever wanted your information is likely to 5 \_\_\_\_\_.

Manager: I guess we'd 6 \_\_\_\_\_ the company's antivirus software, then.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

If you don't ... you could ...  
You think this will ...  
I guess we'd better ...

**Student A:** You are an engineer. Talk to Student B about:

- security threats to his or her system
- consequences of the threats
- ways to prevent security problems

**Student B:** You are a company manager. Talk to Student A about security threats to your system.

**Writing**

9 Use the conversation from Task 8 to complete the email from a software engineer.

Dear Mr. Greene,

I am concerned about your company's computer security. When I inspected your systems, I discovered \_\_\_\_\_.

This is dangerous because \_\_\_\_\_.

To keep your systems safe, I recommend \_\_\_\_\_.

This will help because \_\_\_\_\_.

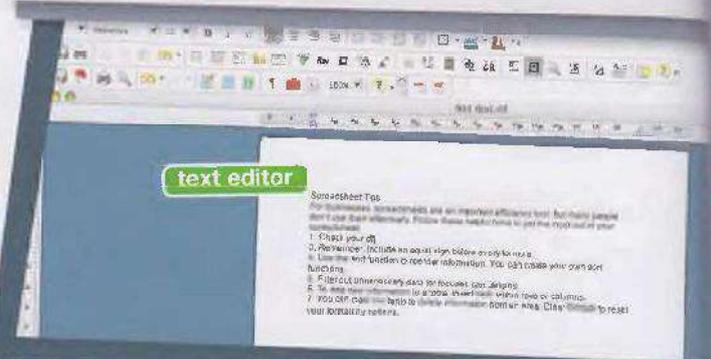
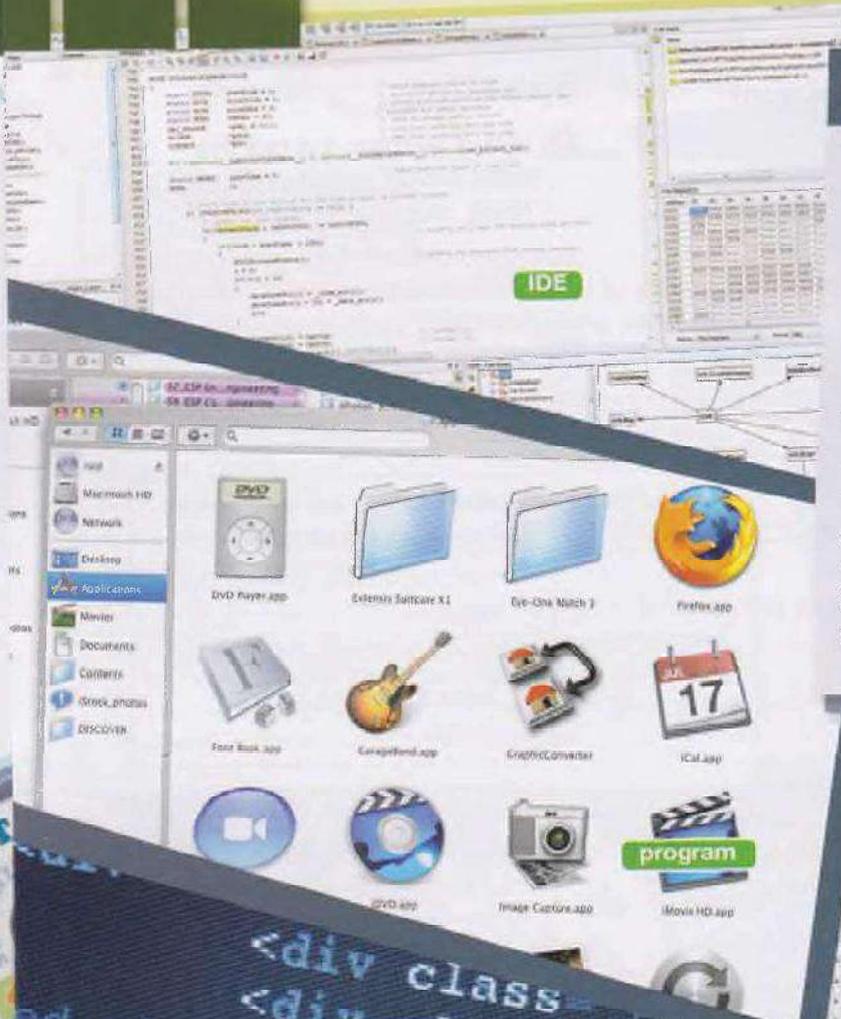
Let's meet to discuss this further.

Danielle Corwin  
SharpAlert Computing Systems

## Chapter 7 Programming Software

Programming software allows programmers to develop new **programs**. The most basic programming software is a **source code editor**. These programs are usually basic **text editors**. Programmers use them to enter lines of code into a computer. Lines of code are written in a **programming language**. Programmers use this to write series of commands. When executed *together*, these commands form a complete program.

Other **programming software** are more complex. An **IDE** (integrated development environment) provides tools for writing programs. These usually include a text editor and a **debugger**. Other tools may include a **linker**, a **compiler**, or an **interpreter**.



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 How do programmers create new programs?
- 2 What kinds of software helps programmers write new programs?

## Reading

2 Read the textbook chapter. Then, mark the following statements as true (T) or false (F).

- 1  Programmers use text editors to interpret commands.
- 2  Text editors are a type of source code editor.
- 3  An IDE is the most basic type of programming software.

## Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- 1  source code editor
- 2  program
- 3  programming language
- 4  programming software
- 5  interpreter
- 6  compiler

- A any software that supports the development of new applications
- B a series of operations that control the functions of a computer
- C codes used to write commands to a computer
- D an application that decodes instructions written in other languages
- E something that reads and executes other programs
- F software used to enter lines of coded text

6 Read the sentence pairs. Choose which word or phrase best fits each blank.

1 text editor / IDE

- A A(n) \_\_\_\_\_ usually provides programmers with various tools for writing programs.  
 B A(n) \_\_\_\_\_ is a basic program for entering commands and code into a computer.

2 debugger / linker

- A A \_\_\_\_\_ provides links to additional information needed for programs to run.  
 B A \_\_\_\_\_ finds and corrects errors in code.

7 Listen and read the textbook chapter again. What tools does an IDE provide?

## Listening

8 Listen to a conversation between two students. Choose the correct answers.

- 1 What is the conversation mostly about?  
 A how to interpret instructions in programming code  
 B the difference between two types of computer programs  
 C techniques for entering code into a text editor  
 D the importance of learning different programming languages
- 2 According to the woman, what is sometimes needed to run programs correctly?  
 A using a linker and a compiler together  
 B allowing programs to connect to the internet  
 C writing new instructions in a programming language  
 D translating lines of code into a different programming language

9 Listen again and complete the conversation.

Student 1: Can you help me with something?

Student 2: Sure. What is it?

Student 1: I'm confused 1 \_\_\_\_\_ between linkers and compilers.

Student 2: Okay. Well, 2 \_\_\_\_\_ a program to information it needs from other programs.

Student 1: Wait, I thought an 3 \_\_\_\_\_ did that.

Student 2: No. An interpreter just 4 \_\_\_\_\_ so the program can run.

Student 1: Okay. So then a compiler is 5 \_\_\_\_\_ program instructions.

Student 2: Exactly. It translates information from different 6 \_\_\_\_\_.

Student 1: I get it now. Thanks.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

- Can you help me with ...  
 I'm having some trouble ...  
 I was under the impression ...

**Student A:** You are a student. Talk to Student B about:

- terms you are confused about
- what you thought the terms meant
- what the terms actually mean

**Student B:** You are a student. Talk to Student A about terms that he or she is confused about.

## Writing

9 Use the conversation from Task 8 to complete the workbook exercise.

Explain the difference between each set of terms:

1 Debugger / \_\_\_\_\_:

A debugger is used for finding and correcting errors in code.

A(n) \_\_\_\_\_ is used for \_\_\_\_\_.

2 \_\_\_\_\_ / text editor:

A(n) \_\_\_\_\_ is used for \_\_\_\_\_.

A text editor is used to enter lines of code and create programs.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What types of application software do businesses typically use?
- 2 What types of application software do people typically use at home?

## ShorSoft Software Developers

We are the most trusted name in **application software**. Our software will enhance any project.

Our **office suite** comes standard with **word processing software**. You can also add our optional **accounting software**. Our **spreadsheets** will keep your household or small business organized.

Do you own a large company? Get ShorSoft Deluxe. This **enterprise software** has everything you need to manage your business.

ShorSoft's **desktop publishing software** is very user-friendly. It allows users to create prints and web layouts. It even optimizes documents for different **web browsers**.

We also offer **video editing** and **image editing software**. Preserve your family memories with high-quality videos and photos.

## Reading

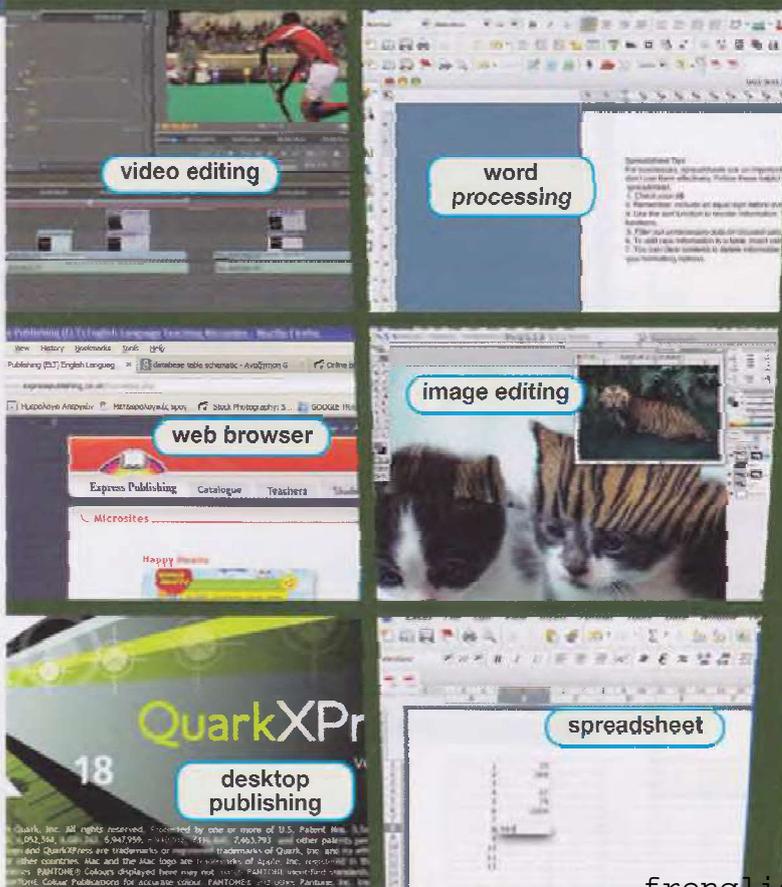
2 Read the advertisement. Then, choose the correct answers.

- 1 What kind of software is available to add on to the office suite?
  - A video editing software
  - B word processing software
  - C accounting software
  - D desktop publishing software
- 2 Which of the following is NOT offered by the company?
  - A web browsers
  - B image editors
  - C accounting software
  - D word processing programs
- 3 What is a feature of the company's desktop publishing software?
  - A It keeps track of household information.
  - B It comes standard with word processing software.
  - C It makes documents easy to read on webpages.
  - D It allows high-quality photo and video preservation.

## Vocabulary

3 Choose the sentence that uses the underlined part correctly.

- 1 A Application software can mean many different types of software.
  - B Desktop publishing software is used to keep track of purchases and expenses.
- 2 A Users can edit videos and photos with word processing software.
  - B Books and brochures are designed with desktop publishing software.
- 3 A Application software allows users to view web pages.
  - B Text documents are produced with word processing software.
- 4 A Video editing software lets users create movies on their computers.
  - B Spreadsheets can be played on computers or other consoles.



4 Fill in the blanks with the correct words and phrases from the word bank.

**Word BANK**

accounting office suite enterprise software  
image editing web browser spreadsheet

- \_\_\_\_\_ software records and manages transactions.
- A(n) \_\_\_\_\_ usually includes a word processor.
- Many large corporations use \_\_\_\_\_ to maintain consistency in all their systems.
- \_\_\_\_\_ software can be used to retouch photographs.
- Users often check their email using a(n) \_\_\_\_\_.
- Large sets of data can be organized into a \_\_\_\_\_.

5 Listen and read the advertisement again. What types of application software does the company make?

**Listening**

6 Listen to a conversation between two software engineers. Mark the following statements as true (T) or false (F).

- \_\_\_ The man is excited about the accounting software.
- \_\_\_ The accounting software will be included in the office suite.
- \_\_\_ The engineers will develop the accounting software before they update the word processor.

7 Listen again and complete the conversation.

Engineer 1: Did you see 1 \_\_\_\_\_ from the CEO?  
 Engineer 2: Do you mean the one about the 2 \_\_\_\_\_ ?  
 Engineer 1: Yeah. They want a whole new program for 3 \_\_\_\_\_. We're supposed to develop it by the end of the month.  
 Engineer 2: I think that's a good idea. A lot of our customers probably use accounting anyway.  
 Engineer 1: Yeah, but I'm worried. What will that mean for the 4 \_\_\_\_\_ in our office suite?  
 Engineer 2: What do you mean?  
 Engineer 1: Well, our word processor needs to 5 \_\_\_\_\_. I'm afraid that a new program will take away from that.  
 Engineer 2: I 6 \_\_\_\_\_, but I think it'll be fine. Another team is scheduled to complete that update this week.

**Speaking**

9 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Did you see ... / I think ...  
 I see what you mean, but ...*

**Student A:** You are an engineer. Talk to Student B about:

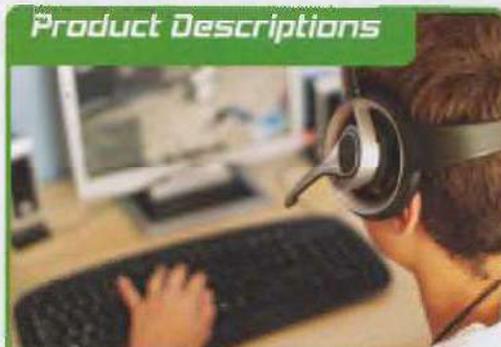
- developments at your software company
- your concerns about the developments

**Student B:** You are an engineer. Talk to Student A about new developments at your software company.

**Writing**

8 Use the conversation from Task 8 to complete the company's software product index.

**Product Descriptions**



\_\_\_\_\_ : Use this software to perfect your photos and preserve your family memories.

Office suite: \_\_\_\_\_

\_\_\_\_\_ : This software will help you keep track of your budget on a monthly or weekly basis. It's a good program to have when tax season comes.

Video editing software: \_\_\_\_\_

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some purposes of software?
- 2 Why must software engineers learn about many different industries?



## Technology Today

Software engineers might not be familiar with a particular industry. Nonetheless, they must be prepared to write the software. This makes the job challenging, but also interesting and rewarding.

Software engineers write many types of programs. They create simple **mobile apps** for the casual user. But they also design complex programs for fields like **bioinformatics**.

Companies use software for many different kinds of **data management**. Managers use programs to organize **payroll** or perform **cost analyses**. Some programs **simulate** actions and events, like changes in financial markets.

Individuals use personal devices like **multimedia players** and **digital assistants**. Travelers benefit from software like **satellite navigation** and **route planning**.



## Reading

2 Read the journal article. Then, choose the correct answers.

- 1 What is the main idea of the article?
  - A where engineers learn to design software
  - B why engineers must learn about various industries
  - C a guide to develop software for a particular purpose
  - D the differences between various types of software
- 2 According to the article, which is a form of data management?
  - A route planning
  - B payroll
  - C satellite navigation
  - D bioinformatics
- 3 What is NOT true of software engineers?
  - A They typically work in different industries before becoming engineers.
  - B They usually learn to design many different types of software.
  - C They create products for both commercial and individual use.
  - D They may be unfamiliar with subjects that they are writing software for.

## Vocabulary

3 Match the words (1-8) with the definitions (A-H).

- |                     |                           |
|---------------------|---------------------------|
| 1 __ payroll        | 5 __ multimedia player    |
| 2 __ simulation     | 6 __ digital assistant    |
| 3 __ cost analysis  | 7 __ data management      |
| 4 __ bioinformatics | 8 __ satellite navigation |

- A a realistic representation of something
- B the ability to track and evaluate information
- C a company's list of employees and how much they are paid
- D a report that explains expenses
- E the application of computer software to the field of biology
- F the process of determining a location based on electronic information
- G a device that can play audio, video, and other files
- H a small, handheld computer that typically works as a mobile phone

**4** Choose the sentence that uses the underlined part correctly.

- 1 **A** Many software apps aid in the process of route planning.
- B** You can use payroll to download whatever app you want.
- 2 **A** Software Engineers must create satellite navigation in order to help biologists.
- B** New phones download mobile apps really quickly.

**5** Listen and read the journal article again. What types of software might a businessperson use?

### Listening

**6** Listen to a conversation between a company owner and an engineer. Mark the following statements as true (T) or false (F).

- 1  The newest project is a route planning application.
- 2  The woman asks to see the mobile app.
- 3  The new program must be compatible with digital assistants.

**7** Listen again and complete the conversation.

**Owner:** Owen, where are we on 1 \_\_\_\_\_?

**Engineer:** We're 2 \_\_\_\_\_ on a lot of new things, Ms. Thompson.

**Owner:** Last I heard, you were finishing a mobile app for 3 \_\_\_\_\_.

**Engineer:** We finished that. Now we're working on a program for 4 \_\_\_\_\_.

**Owner:** Can you tell me more about that?

**Engineer:** We want to 5 \_\_\_\_\_ a satellite navigation program. But ours will be much smaller.

**Owner:** Why is that?

**Engineer:** It needs to be compatible with 6 \_\_\_\_\_.

### Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

Where are we on ...  
 Can you tell me more ...  
 We started ...

**Student A:** You are a company owner. Talk to Student B about:

- software engineering projects
- recent and current developments
- how the software will be used

**Student B:** You are an engineer. Talk to Student A about software engineering.

### Writing

**9** Use the conversation from Task 8 to write an email on software developments to the owner of a company.



Dear Ms. Thompson,

We recently finished \_\_\_\_\_.

Another project is \_\_\_\_\_.

So far, we \_\_\_\_\_.

Since \_\_\_\_\_ is so popular, we will begin writing a new program for it.

Most of these programs will be accessible on \_\_\_\_\_.

Sincerely,  
 Owen

# 10 The Desktop and GUI

## Starting Mailbag Deluxe



right-click

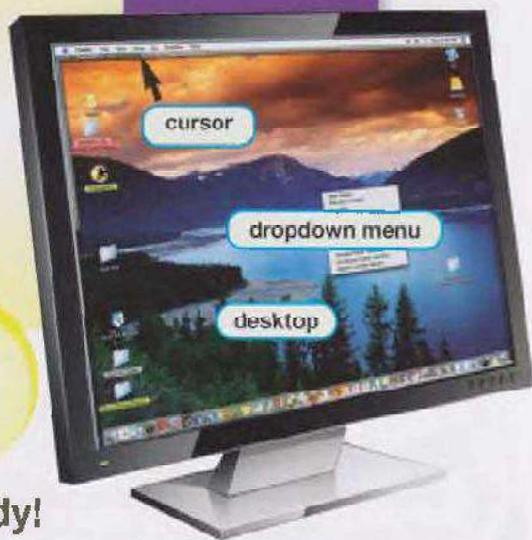


icon

select

Follow these instructions to start Mailbag Deluxe after installation. This process applies to all standard GUIs.

1. Find the shortcut to Mailbag Deluxe on your **desktop**. The **icon** is a **folder** with a star in the corner.
2. **Right-click** the icon to show the **dropdown menu**.
3. **Select** the option that says "Open Mailbag."
4. The setup guide will open. Click "Next."
5. Type your name and email address. Make sure the **cursor** appears in the correct box.
6. Click "Finish Setup." This step may take a few moments.
7. Click "Run Mailbag." You are ready to use Mailbag Deluxe!



cursor

dropdown menu

desktop

### Get ready!

- 1 Before you read the passage, talk about these questions.
  - 1 What features are commonly found on a computer desktop?
  - 2 How do users typically start programs through a GUI?

### Reading

- 2 Read the manual. Then, mark the following statements as true (T) or false (F).
  - 1 \_\_\_ The instructions help users install a program.
  - 2 \_\_\_ "Open Mailbag" is an option from the dropdown menu.
  - 3 \_\_\_ The cursor takes a few moments to appear in the correct box.

### Vocabulary

- 3 Match the words and phrases (1-7) with the definitions (A-G).

- |              |                     |
|--------------|---------------------|
| 1 ___ icon   | 5 ___ cursor        |
| 2 ___ GUI    | 6 ___ right-click   |
| 3 ___ open   | 7 ___ dropdown menu |
| 4 ___ select |                     |

- A to reveal the contents of something in order to use or edit them
- B a list of options that appears when an item is clicked
- C a feature on a computer screen that indicates where input will appear
- D to mark something for a particular operation
- E a visual way of interacting with a computer screen
- F a symbol that indicates a file or program
- G to bring up options using a particular mouse button

- 4 Choose the sentence that uses the underlined part correctly.

- 1 A The user clicked on the cursor to start the web browser.  
B Some people display pictures on their desktops.
- 2 A When a user opens a program, the computer should shut down.  
B The folder contains the engineer's important documents.
- 3 A Clicking on a menu option will often run a program.  
B Use the icon to scroll down to the bottom of the page.

5 Listen and read the manual again. How does a user bring up the dropdown menu?

## Listening

6 Listen to a conversation between a tech support representative and a customer. Choose the correct answers.

- What is the main idea of the conversation?
  - A a defect in the company's software
  - B a program that the man cannot open
  - C compatibility between a program and the man's GUI
  - D instructions for removing icons from a desktop
- What is the cause of the man's problem?
  - A The company installed the wrong version of the program.
  - B The man did not click an icon correctly.
  - C The man forgot to save the program to his desktop.
  - D The company gave the man the wrong instructions.

7 Listen again and complete the conversation.

Representative: Thanks for calling Mailbag Support. How 1 \_\_\_\_\_ you?

Customer: Hi, I just installed the new Mailbag Deluxe. But I can't 2 \_\_\_\_\_.

Representative: I'm sorry to hear you're having trouble. Do you see the icon on 3 \_\_\_\_\_?

Customer: Yes, it's there. But 4 \_\_\_\_\_ when I click it.

Representative: Did you 5 \_\_\_\_\_ the icon?

Customer: No. Should 6 \_\_\_\_\_?

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I can't ...*  
*I'm sorry to hear ...*  
*Go ahead and ...*

**Student A:** You are a tech support representative. Talk to Student B about:

- a problem that he or she is having with a program
- what actions he or she already tried
- what actions he or she should take

**Student B:** You are a customer. Talk to Student A about a problem that you are having with a program.

## Writing

9 Use the conversation from Task 8 to complete the troubleshooting guide.

Mailbag Deluxe:

### Solutions to Common Problems

## Troubleshooting Guide

**Problem:** The program \_\_\_\_\_ when \_\_\_\_\_.

**Solution:** \_\_\_\_\_

**Problem:** The program \_\_\_\_\_ when \_\_\_\_\_.

**Solution:** \_\_\_\_\_

$$2 + 3 = 5$$

plus

$$3 - 2 = 1$$

minus

$$2 \times 3 = 6$$

times

$$6 \div 2 = 3$$

divided by

$$4 + 3 = 7$$

equal

1,400

hundred

## How do they say it?

Symbol	Interpretation	Examples
=	equals	$\frac{1}{4} = 0.25$ One fourth equals zero point two five.
+	plus, add	$2 + 5 = 7$ Two plus five equals seven. Add two and five to get seven.
-	minus, less, subtract	$5 - 2 = 3$ Five minus two equals three. Subtract two from five to get three.
$\times$	multiplied by, times	$2 \times 5 = 10$ Two multiplied by five is ten. Two times five equals ten.
$\div, /$	divided by, over	$10 \div 5 = 2$ $10 / 5 = 2$ Ten divided by five equals two. Ten over five is two.
2,500	two thousand five hundred or twenty-five hundred	We spent two thousand five hundred dollars on office supplies.

## Get ready!

3 Before you read the passage, talk about these questions.

- 1 What are some mathematical operations that combine numbers?
- 2 What are some mathematical operations that split numbers apart?

## Reading

2 Read the chart. Then, mark the following statements as true (T) or false (F).

- 1  Three times ten equals ten minus three.
- 2  Four plus four equals eight.
- 3  Two multiplied by three is the same as two times three.

## Vocabulary

3 Read the sentences and choose the correct words.

1 times / over

A Eight \_\_\_\_\_ four equals two.

B Seven \_\_\_\_\_ two is fourteen.

2 add / subtract

A \_\_\_\_\_ two amounts of a substance to get a larger amount.

B When you \_\_\_\_\_ one number from another, you get the difference between them.

4 Place the correct words and phrases from the word bank under the correct headings.

**word** BANK

equal plus    less divided by    multiplied by minus

Combining amounts	Splitting amounts	Expressing results

5 Listen and read the chart again. What is the result if someone accidentally subtracts instead of adding?

**Listening**

6 Listen to a conversation between two engineers. Choose the correct answers.

- What is the main idea of the conversation?
  - A a mathematical error in a program
  - B a calculation that the woman needs help with
  - C how many extra engineers are needed for a project
  - D when a program will be completed
- What operation does the man use to calculate the total lines per day?
  - A division
  - B addition
  - C subtraction
  - D multiplication

7 Listen again and complete the conversation.

Engineer 1: What's the status on the 1 \_\_\_\_\_ ?  
 Engineer 2: We're right on 2 \_\_\_\_\_ .  
 Engineer 1: Great! Will 3 \_\_\_\_\_ it by next Friday?  
 Engineer 2: That's ten days away, right? And we need about 4 \_\_\_\_\_ more lines of code.  
 Engineer 1: Fifteen hundred 5 \_\_\_\_\_ ten is one hundred fifty lines per day. Can you do it?  
 Engineer 2: I think so. We have 6 \_\_\_\_\_ working on it.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

What's the status ...  
 We need about ...  
 ... times ... equals ...

**Student A:** You are an engineer. Talk to Student B about:

- the status of a project
- how many engineers are working on the project
- how much work each engineer must complete

**Student B:** You are an engineer. Talk to Student A about the status of a project.

**Writing**

9 Use the conversation from Task 8 to complete the email.

Hi Grant,

The development of our new program is right on schedule. We have \_\_\_\_\_ engineers working on the project. According to the schedule, we have \_\_\_\_\_ days left to complete \_\_\_\_\_ lines of code. If each engineer completes \_\_\_\_\_ lines per day, we can complete the project on time. Here are my calculations: \_\_\_\_\_

Let me know if you have any questions.

Susan

## 7.3 Quantities Expressed in Engineering

Numbers appear in multiple forms. Some quantities are expressed in **fractions**.

Example: 3 **out of** every 4 software engineers in a region have bachelor's degrees. Therefore,  $\frac{3}{4}$  of software engineers have degrees.

The same number can be a **percentage**.

Example: 75 **percent** of software engineers have degrees.

Percentages sometimes appear as **decimal numbers**. In this case, the percentage comes after a zero and a **point**.

Example: The rate of software engineers with degrees is 0.75.

It is easy to **convert** a percentage to a fraction. Simply use the percentage as a **numerator** and 100 as the **denominator**. Then, **reduce** the fraction to its lowest form.

Example: 75 over 100 equals 3 over 4.

$\frac{1}{4}$

fraction

$\frac{2}{7}$

denominator

$\frac{3}{5}$

numerator

25%

percentage

0.25

decimal number

0.40

point

### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some different ways to express parts of a whole?
- 2 How can someone convert a percentage into a fraction?

### Reading

2 Read the textbook chapter. Then, mark the following statements as true (T) or false (F).

- 1  Fractions sometimes contain decimals.
- 2  Percentages can be expressed as whole or decimal numbers.
- 3  When a percentage is converted to a fraction, the denominator is 100.

### Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- |                                       |  |
|---------------------------------------|--|
| 1 <input type="checkbox"/> point      | 4 <input type="checkbox"/> numerator   |
| 2 <input type="checkbox"/> percentage | 5 <input type="checkbox"/> denominator |
| 3 <input type="checkbox"/> -out of-   | 6 <input type="checkbox"/> reduce      |

- A the number above the line on a fraction
- B the number below the line on a fraction
- C describing the ratio between actual and potential quantities
- D a quantity measured in terms of 100 units
- E a dot that separates whole and partial numbers
- F to change a fraction into a form containing the lowest possible numbers

**4** Read the sentences and choose the correct words.

- 1 A **decimal number** / **numerator** always contains a point.
- 2 The student **converted** / **reduced** several fractions into percentages.
- 3 A **denominator** / **fraction** can be expressed as x over y.
- 4 Fifty **percent** / **point** is often called one half.

**5** Listen and read the textbook chapter again. What is an appropriate way to express twelve out of fifteen?

### Listening

**6** Listen to a conversation between an engineer and a manager. Choose the correct answers.

- 1 What is the conversation mostly about?
  - A the number of software packages that clients purchased
  - B the number of people who were affected by a system malfunction
  - C the number of engineers who are working on a project
  - D the number of software errors that an engineering team resolved
- 2 What does the man ask the woman to do?
  - A release the new program right away
  - B double-check the issues that were already fixed
  - C focus on one particular problem with the program
  - D inform clients on the progress of the software update

**7** Listen again and complete the conversation.

**Engineer:** Hi, Mr. Lennox. You 1 \_\_\_\_\_ see me?

**Manager:** Yes, Jenna. Our clients are asking about the new 2 \_\_\_\_\_.

**Engineer:** We're working on it. We already fixed twelve 3 \_\_\_\_\_ the fifteen problems.

**Manager:** Well, that's 4 \_\_\_\_\_ done. What about the other three problems?

**Engineer:** They're tricky. The sudden shut-downs are the biggest issue.

**Manager:** I want you to 5 \_\_\_\_\_ fixing that. We'll release the update as soon as that's done.

**Engineer:** So you want us to spend 6 \_\_\_\_\_ of our time on it?

**Manager:** Yes. The other fixes can wait until the next update.

### Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*We're working on ...*  
*What about the ...*  
*So you want us to ...*

**Student A:** You are an engineer. Talk to Student B about:

- your progress on a project
- how much work remains on the project
- his or her instructions

**Student B:** You are a manager.

Talk to Student A about his or her progress on a project.

### Writing

**9** Use the conversation from Task 8 to complete the progress report.

## Progress Report Form

Project: \_\_\_\_\_

Update: We have finished \_\_\_\_\_ out of \_\_\_\_\_ items. That means the project is \_\_\_\_\_ complete. To complete the project as quickly as possible, we will spend \_\_\_\_\_ of our time on \_\_\_\_\_. We will spend \_\_\_\_\_ of our time on \_\_\_\_\_.

# 13 Describing Change

## Get ready!

1 Before you read the passage, talk about these questions.

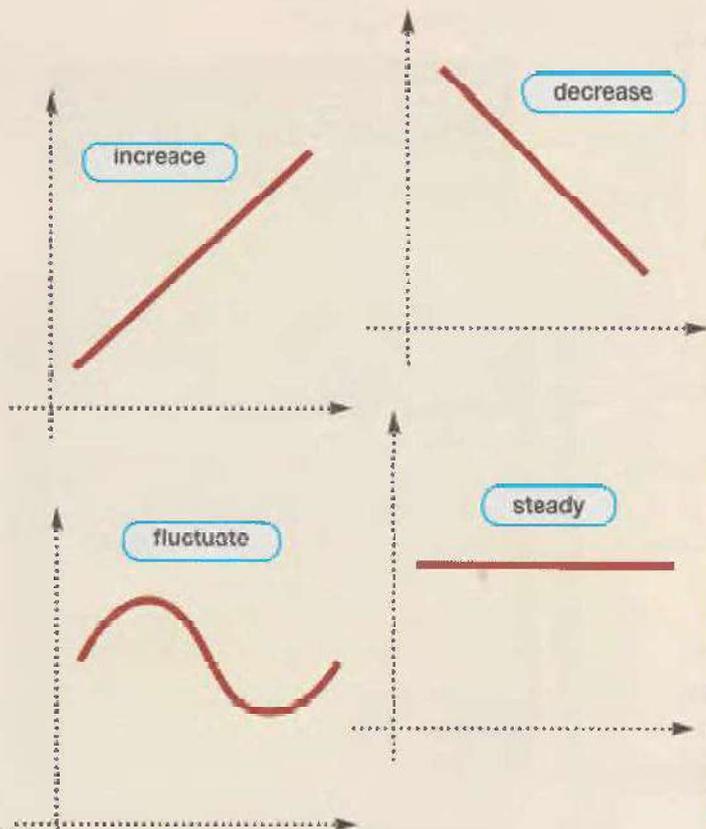
- 1 What are some ways to describe changes in a market?
- 2 What changes are typically desirable in the software market?

## All About Computers: A Growing Industry

Demand for computer technology grows constantly. This **trend** shows no signs of slowing down. As the cost of new technologies **decreases**, the demand **increases**. This is an exciting time for software engineers.

A few years ago, the demand for computers **fluctuated**. In fact, some people even predicted a **decline** in computer popularity. Fortunately, however, the market **stabilized** and sales recovered.

Today, the software development industry is on the **rise**. The job market for skilled software engineers is **expanding**. In the last few years, software companies **doubled** their hiring efforts. Now, experts expect growth to continue at a **steady** pace.



## Reading

2 Read the magazine article. Then, choose the correct answers.

- 1 What is the main idea of the article?
  - A characteristics of the software market
  - B ways to predict the changes in the software market
  - C a guide on how to invest in the software market
  - D products that drive current software market trends
- 2 What is true of the job market for software engineers?
  - A It is in fast decline.
  - B It is slowing down slightly.
  - C It is expanding steadily.
  - D It is constantly fluctuating.
- 3 Which of the following is NOT a reason to hire new software engineers?
  - A to continue the steady pace of software development
  - B to maintain the production of older technologies
  - C to meet increasing demands for new software
  - D to fill the expanding number of software development jobs

## Vocabulary

2 Match the words (1-8) with the definitions (A-H).

- |              |                |
|--------------|----------------|
| 1 __ trend   | 5 __ double    |
| 2 __ steady  | 6 __ rise      |
| 3 __ expand  | 7 __ fluctuate |
| 4 __ decline | 8 __ stabilize |

- A the process of becoming worse or smaller in amount
- B to make twice as large or abundant
- C to become larger in size
- D to change regularly
- E the process of increasing in a quality or amount
- F to reach a state in which changes are infrequent
- G not changing, or changing at a slow and constant rate
- H a consistent change or development

**1 Choose the sentence that uses the underlined part correctly.**

- 1 **A** Manufacturers hope the demand for new computer technology will increase.  
**B** The company is not making money because sales are on the rise.
- 2 **A** The market for cassette players has expanded until it is almost non-existent.  
**B** The popularity of old technology is decreasing.

**5 Listen and read the magazine article again. What are some ways to say that something is lower than it was previously?**

**Listening**

**6 Listen to a conversation between two engineers. Mark the following statements as true (T) or false (F).**

- 1  The demand for the route planning program is in steady decline.  
 2  The route planning program was most successful during the summer.  
 3  The man is confident that the budget mobile app will do well.

**7 Listen again and complete the conversation.**

**Engineer 1:** Hey, Mary. How did the **1** \_\_\_\_\_ do on the market?  
**Engineer 2:** The demand for the route has planning program really **2** \_\_\_\_\_.  
**Engineer 1:** That's strange. Why do you think that is?  
**Engineer 2:** It seems to **3** \_\_\_\_\_ in the summer. That's when people travel the most.  
**Engineer 1:** Oh, that makes sense. What about the budget mobile app?  
**Engineer 2:** That did better. The demand increased at a **4** \_\_\_\_\_.  
**Engineer 1:** How do **5** \_\_\_\_\_ it will do in the future?  
**Engineer 2:** I have **6** \_\_\_\_\_ it will do well.

**Speaking**

**8 With a partner, act out the roles below based on Task 7. Then, switch roles.**

**USE LANGUAGE SUCH AS:**

*How did ... / The demand for ...*

*I have no doubt ...*

**Student A:** You are an engineer. Talk to Student B about:

- a new program on the market
- the reasons for its demand
- the changes you expect to see in the future

**Student B:** You are an engineer. Talk to Student A about a new software on the market.

**Writing**

**9 Use the conversation from Task 8 to write an email to a business associate about a new software on the market.**



Dear Mr. Billings,

We have just released several new programs.

The first software is a bill paying program. Sales are really \_\_\_\_\_ . I believe this is because \_\_\_\_\_

The other is a mobile app. It is not as \_\_\_\_\_ as we thought it would be. The sales are \_\_\_\_\_

Overall, the sales of our programs are \_\_\_\_\_

Sincerely,  
Michelle

# 14 Presentations and Communication



●●●

To:	r.nielson@compspecs.com
From:	k.fielding@compspecs.com
Subject:	Your Presentation

Hi Robert,

I was impressed with your software development plan. However, we need to **review** your **presentation** this morning. Some directors were confused about the plan.

Your **body language** did not demonstrate confidence. Stand up straight, but stay comfortable and relaxed. I was glad that you used **note cards**. Unfortunately, you looked at them too much. It prevented you from making **eye contact** with the audience. Also, **project** your voice. Your audience needs to hear you.

Your audience needed better guidance. **Signpost** often to remind them where you are. And always **summarize** the important points at the end.

Finally, your presentation would be much clearer with **visual aids**. Use **handouts** or posters next time.

I hope this helps.

-Kayla



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some effective ways to communicate when giving presentations?
- 2 Why do engineers need good presentation skills?

## Reading

2 Read the email. Then, choose the correct answers.

- 1 What is the main idea of the email?
  - A a plan for an upcoming presentation
  - B problems with an engineer's communication style
  - C concerns about a software development plan
  - D ways to improve company policies on sharing ideas
- 2 Which of the following is NOT suggested in the email?
  - A Read note cards more carefully.
  - B Remember to look at the audience.
  - C Repeat important ideas at the end.
  - D Give the audience handouts.
- 3 According to the email, what shows confidence?
  - A using note cards
  - B signposting often
  - C summarizing clearly
  - D standing up straight

## Vocabulary

2 Match the words and phrases (1-5) with the definitions (A-E).

- |                 |                    |
|-----------------|--------------------|
| 1 __ project    | 4 __ summarize     |
| 2 __ signpost   | 5 __ body language |
| 3 __ visual aid |                    |

- A a physical representation to demonstrate ideas in a presentation
- B to guide listeners with clear directional wording
- C to briefly present the main points of something
- D communication that is expressed physically, without words
- E to speak loudly so that one's voice is heard at a distance

4 Fill in the blanks with the correct words and phrases from the word bank.

**word BANK**

note card presentation  
review eye contact handout

- Each audience member received a full-page \_\_\_\_\_ listing additional information and references.
- After the speech, the engineers met to the information in detail.
- The speaker couldn't remember what to say, so she checked her \_\_\_\_\_.
- The engineer planned a \_\_\_\_\_ to introduce his new software design.
- Try to make \_\_\_\_\_ with as many audience members as possible.

5 Listen and read the email again. What is a potential problem with using note cards?

**Listening**

6 Listen to a conversation between an engineer and a manager. Mark the following statements as true (T) or false (F).

- The man was impressed with the information that the woman presented.
- The woman was too loud during her presentation.
- According to the man, the note cards should contain more detailed information.

7 Listen again and complete the conversation.

**Engineer:** What did you think of my presentation?  
**Manager:** I 1 \_\_\_\_\_ your ideas. But you need to work on your presentation skills.  
**Engineer:** What do you mean?  
**Manager:** For one, the audience 2 \_\_\_\_\_ towards the back.  
**Engineer:** Oh, I didn't realize that. So I need to 3 \_\_\_\_\_ next time.  
**Manager:** Exactly. And let's work on using fewer 4 \_\_\_\_\_ next time.  
**Engineer:** Why? What's 5 \_\_\_\_\_ them?  
**Manager:** You looked at them too much. You need to 6 \_\_\_\_\_ with your audience.  
**Engineer:** I see. I guess I should practice before my next presentation.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**  
*What did you think of ...*  
*For one ...*  
*Let's work on ...*

**Student A:** You are an engineer. Talk to Student B about:

- the results of your presentation
- problems with your communication style
- how you can improve

**Student B:** You are a manager. Talk to Student A about the results of his or her presentation.

**Writing**

9 Use the conversation from Task 8 to complete the presentation feedback form.

**Presentation Feedback Form**

Presenter: \_\_\_\_\_

Please rate the presentation between 1 (low) and 5 (high): 1 2 3 4 5

**What did the presenter do well?**

I liked that the presenter \_\_\_\_\_ because \_\_\_\_\_.

**How could the presenter improve?**

Next time, the presenter should \_\_\_\_\_ because \_\_\_\_\_.

HOME

ABOUT US

SERVICES

CONTACT

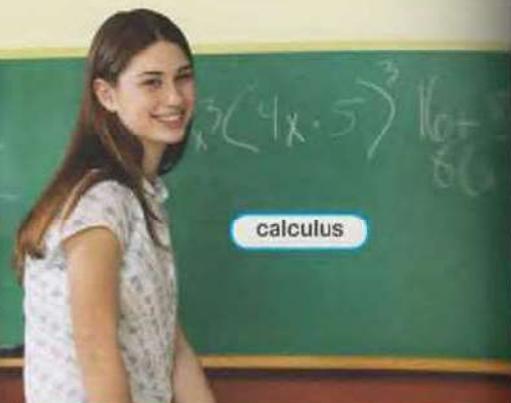
## Ferdinand Technical Institute (FTI) > Degree Programs > Software Engineering

A **bachelor's degree** from FTI is a great way to begin your software engineering career. We have one of the finest programs in the country.

During the **foundation** year, students learn engineering principles. They also take mathematics courses, including **linear algebra** and **calculus**.

Software engineers must understand hardware components, too. Fundamental courses in **computer engineering** teach the basics of **computer architecture**. They also cover physical properties of general **electronics**. This includes the application of electrical principles in advanced **circuit analysis**.

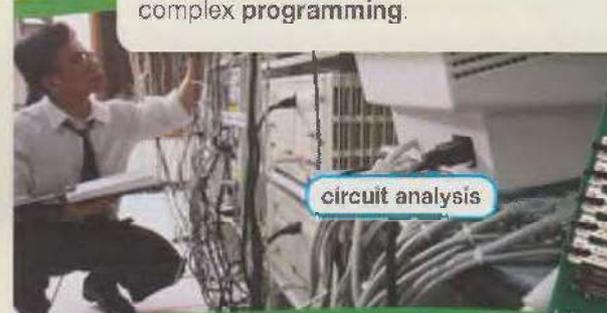
After the basics, students take upper-level software engineering courses. These cover implementation of **control systems** and complex **programming**.



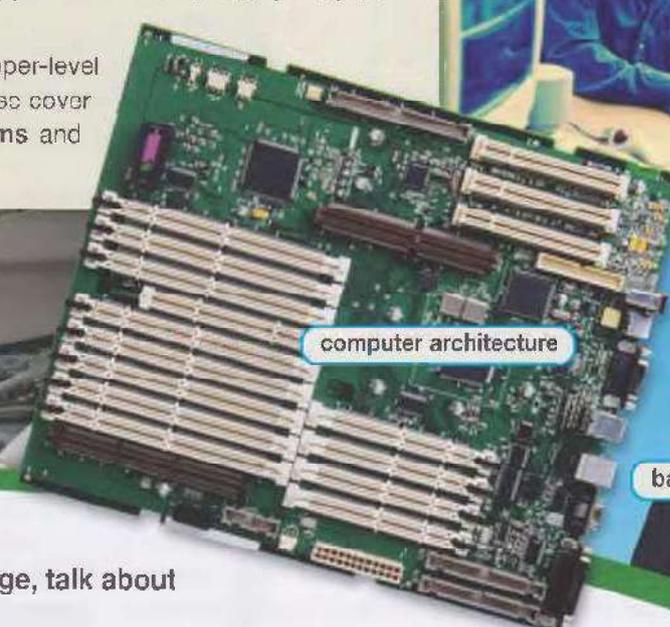
calculus



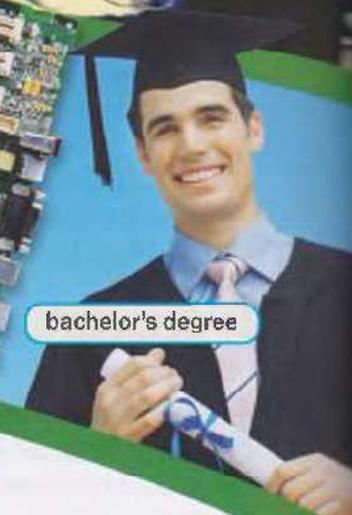
programming



circuit analysis



computer architecture



bachelor's degree

### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What subjects do software engineering students study?
- 2 What are the educational requirements for software engineers in your country?

### Reading

2 Read the webpage. Then, complete the table.

Area of study	Subjects covered
Foundation year	1 _____ _____
Computer Engineering	2 _____ _____
3 _____ _____	implementation of control systems, complex programming

### Vocabulary

3 Match the words and phrases (1-6) with the definitions (A-F).

- |                   |                             |
|-------------------|-----------------------------|
| 1 ___ calculus    | 4 ___ programming           |
| 2 ___ electronics | 5 ___ control system        |
| 3 ___ foundation  | 6 ___ computer architecture |

- A a device that regulates the actions of other devices
- B the physical configuration of computer hardware components
- C a general course of study that a student takes before a degree program
- D a branch of mathematics that analyzes complex physical properties
- E machines that are powered by electrical systems
- F the process of writing computer software

3 Read the sentence pairs. Choose which phrase best fits each blank.

1 bachelor's degree / computer engineering

- A The engineer has a \_\_\_\_\_ in computer sciences.
- B The student majored in \_\_\_\_\_ in college.

2 circuit analysis / linear algebra

- A Students in \_\_\_\_\_ need strong knowledge of electrical principles.
- B The professor's lecture in \_\_\_\_\_ introduced the principles of lines and planes.

5 Listen and read the webpage again. What courses do software engineering students take before upper level courses?

### Listening

6 Listen to a conversation between an advisor and a student. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The woman expresses concern about the man's academic performance.
- 2 \_\_\_ The man recently completed a calculus course.
- 3 \_\_\_ The man cannot enroll in upper-level courses yet.

7 Listen again and complete the conversation.

Advisor: So, Ben. Let's talk about next semester.

Student: Okay. Can I enroll in 1 \_\_\_\_\_ yet?

Advisor: Well, not quite. You still have some 2 \_\_\_\_\_ work to finish.

Student: Really? Like what?

Advisor: First, you need 3 \_\_\_\_\_ . Then you can enroll in upper-level courses.

Student: I guess I'll do that, then. What else 4 \_\_\_\_\_ next semester?

Advisor: Let's get you started on some basic 5 \_\_\_\_\_ courses.

Student: Yes, I think I'm ready for that.

Advisor: I'll sign you up for 6 \_\_\_\_\_ .

### Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

#### USE LANGUAGE SUCH AS:

- First, you need to ...
- I guess I'll ...
- Let's get you started on ...

**Student A:** You are an advisor. Talk to Student B about:

- his or her academic progress
- which courses he or she is not ready to take yet
- which courses you recommend enrolling in

**Student B:** You are a student. Talk to Student A about which courses to enroll in.

### Writing

9 Use the conversation from Task 8 to complete the academic progress report form.

## Academic Progress Report Form

Student: \_\_\_\_\_

Advisor: \_\_\_\_\_

What courses did the student take already?

What courses do you recommend for next semester?

How will these courses help the student meet his or her long-term goals?

# Glossary

- accounting** [N-UNCOUNT-U8] **Accounting** is the process of recording and managing financial transactions.
- add** [V-T-U11] To **add** a number to another number is to increase it by that amount.
- anti-virus software** [N-UNCOUNT-U6] **Anti-virus software** is a type of security software that removes malware, or prevents its installation.
- application software** [N-UNCOUNT-U8] **Application software** is any software that is used to perform a single task, or perform multiple tasks that are related to each other.
- artifact** [N-COUNT-U8] An **artifact** is a feature of software that determines its type or function.
- bachelor's degree** [N-COUNT-U15] A **bachelor's degree** is a certificate indicating that someone has completed an educational program, usually after four years of study, and is qualified to practice a particular profession.
- bioinformatics** [N-UNCOUNT-U9] **Bioinformatics** is the application of computer software to the field of biology.
- BIOS** [N-COUNT-U5] A **BIOS** (Basic Input Output System) is a set of instructions in firmware that controls a device's input and output operations.
- body language** [N-UNCOUNT-U14] **Body language** is communication that is expressed with positions of the body instead of words, including hand gestures and facial expressions.
- calculus** [N-UNCOUNT-U15] **Calculus** is a complex branch of mathematics that deals with rates of change and advanced measurements of physical properties.
- case** [N-COUNT-U6] A **case** is a protective enclosure that contains the parts of something.
- CD/DVD drive** [N-COUNT-U6] A **CD/DVD drive** is a device that reads and writes data on compact discs and digital versatile discs.
- circuit analysis** [N-UNCOUNT-U15] **Circuit analysis** is the study of how electrical components conduct currents.
- compiler** [N-COUNT-U7] A **compiler** is a program that decodes instructions written in a higher order language.
- computer** [N-COUNT-U2] A **computer** is an electronic instrument for storing data and performing various electronic tasks and functions.
- computer architecture** [N-UNCOUNT-U15] **Computer architecture** is the physical configuration of computers from hardware components.
- computer engineering** [N-UNCOUNT-U15] **Computer engineering** is a branch of engineering that includes computer science and electrical engineering, and usually involves designing both hardware and software components for computers.
- computing cluster** [N-COUNT-U2] A **computing cluster** is an extremely powerful computer designed to process large quantities of data.
- control** [V-T-U5] To **control** something is to have power over its actions or functions.
- control system** [N-COUNT-U15] A **control system** is a device or set of devices that regulates the actions of other devices.
- convert** [V-T-U12] To **convert** something is to change it into a different form or system of measurement.
- cost analysis** [N-COUNT-U9] A **cost analysis** is a report that explains expenses.
- cover** [N-COUNT-U6] A **cover** is something that is placed over something else for protection.
- cursor** [N-COUNT-U10] A **cursor** is a movable icon on a computer screen that indicates the point where user input will appear.
- data management** [N-UNCOUNT-U9] **Data management** is the ability to track and evaluate information.
- debugger** [N-COUNT-U7] A **debugger** is a computer program that detects and corrects errors in other computer programs.
- decimal number** [N-COUNT-U12] A **decimal number** is a value in a numbering system based on the number 10, with numbers on both sides of the decimal point.
- decline** [N-COUNT-U13] A **decline** is the process of becoming worse or smaller in amount.
- decrease** [V-I-U13] To **decrease** is to become smaller.
- denominator** [N-COUNT-U12] A **denominator** is the number that is below the line in a fraction. In the fraction  $1/2$ , the denominator is 2.

- deny** [V-T-U6] To **deny** something is to refuse it.
- design** [V-T-U1] To **design** something is to plan the way that it will be created.
- desktop** [N-COUNT-U2] A **desktop** is a personal computer intended to be used at a single location, such as a desk.
- desktop** [N-COUNT-U10] A **desktop** is the working area of a computer screen, usually containing icons that represent files, programs, and other features.
- desktop publishing** [N-UNCOUNT-U8] **Desktop publishing** is the process integrating text, images, and other media into a layout which can be published electronically, usually in a home or small business environment.
- develop** [V-T-U1] To **develop** something is to bring it from the initial stages of conception to action or implementation.
- device driver** [N-COUNT-U5] A **device driver** is a program that allows a computer to interact with additional devices.
- digital assistant** [N-COUNT-U9] A **digital assistant** is a small, handheld computer that typically works as a mobile phone as well.
- divide by** [V PHRASE-U11] To **divide** a number (A) **by** another number (B) is to split number A evenly into B number of parts.
- double** [V-T-U13] To **double** something is to make something twice as large or abundant.
- dropdown menu** [N-COUNT-U10] A **dropdown menu** is a list of options that appears below an item when a user clicks the item.
- electronics** [N-COUNT-U15] **Electronics** are machines and communication methods that are powered by electrical systems.
- embedded computer** [N-COUNT-U2] An **embedded computer** is a computer that is a part of a larger product.
- enterprise software** [N-UNCOUNT-U8] **Enterprise software** is a set of programs that is used to organize and control data for a large company or other organization.
- equal** [V-T-U11] To **equal** something is to be precisely the same number or amount as something.
- evaluate** [V-T-U1] To **evaluate** something is to carefully study it and assess its qualities.
- expand** [V-I-U13] To **expand** is to become larger in size.
- eye contact** [N-UNCOUNT-U14] **Eye contact** is the act of looking directly into the eyes of another person.
- fan** [N-COUNT-U6] A **fan** is a device that makes something cooler by moving air.
- firewall** [N-COUNT-U6] A **firewall** is a type of security software that screens network transmissions to prevent unauthorized access to a system.
- firmware** [N-UNCOUNT-U5] **Firmware** is a fixed data structure or program used to control an electronic device.
- flash drive** [N-COUNT-U3] A **flash drive** is a data storage device containing a type of memory that can be erased and reprogrammed with new information.
- flat panel** [ADJ-U3] If a monitor has a **flat panel**, it is much lighter and thinner than a monitor that uses a CRT.
- fluctuate** [V-I-U13] To **fluctuate** is to change regularly.
- folder** [N-COUNT-U10] A **folder** is an icon on a computer screen that is used to access sets of related documents or files.
- foundation** [N-COUNT-U15] A **foundation** is a general course of study that students take before starting a degree program.
- fraction** [N-COUNT-U12] A **fraction** is a part of a whole number, such as 1/2, or one half.
- GUI** [N-COUNT-U10] A **GUI** (graphical user interface) is a visual way of interacting with a computer using menus, icons and windows.
- handout** [N-COUNT-U14] A **handout** is a document that is distributed to a group of people, and is often used to help audiences follow a presentation.
- hard drive** [N-COUNT-U5] A **hard drive** is the main data storage device that is built into a computer.
- hardware** [N-COUNT-U5] **Hardware** is the physical components of a computer.
- heat sink** [N-COUNT-U6] A **heat sink** is an electronic component that transfers heat to the air.

# Glossary

- hundred** [N-COUNT-U11] **-Hundred** is a way of expressing numbers in the thousands by counting how many times 100 goes into the number. For example, the number 1,400 could be expressed as "fourteen hundred."
- icon** [N-COUNT-U10] An **icon** is a graphic symbol that indicates a particular file or program.
- IDE** [N-COUNT-U7] An **IDE** (integrated development environment) is a software application that provides computer programmers with a source code editor, a debugger, and other tools to help them write new programs.
- image editing** [N-UNCOUNT-U8] **Image editing** is the process of manipulating images on a computer.
- increase** [V-I-U13] To **increase** is to grow larger in amount or numbers.
- inkjet printer** [N-COUNT-U3] An **inkjet printer** is a printer that produces images by spraying ink on paper.
- install** [V-T-U1] To **install** something is to put it into the place where it will function.
- interpreter** [N-COUNT-U7] An **interpreter** is a program that reads and executes programs.
- investigate** [V-T-U1] To **investigate** something is to get more information about it.
- keyboard** [N-COUNT-U3] A **keyboard** is a panel of buttons for entering data into a computer.
- laptop** [N-COUNT-U2] A **laptop** is a small computer that can be easily carried and used in many locations, and usually has a screen and keyboard that fold together on a hinge.
- laser printer** [N-COUNT-U3] A **laser printer** is a printer that produces images with a strong, narrow beam of light.
- less** [PREP-U11] If a number is **less** than another number, the second number is subtracted or taken away from the first number.
- linear algebra** [N-UNCOUNT-U15] **Linear algebra** is a branch of mathematics that deals with properties of lines and planes.
- linker** [N-COUNT-U7] A **linker** is a program that provides links to the libraries needed for another program to run.
- malware** [N-UNCOUNT-U6] **Malware** is computer software that is designed to disrupt computer functions or cause harm to the computer's user.
- manually** [ADV-U5] If something is done **manually**, it is done directly by a person, without using automatic functions.
- minus** [PREP-U11] If a number is **minus** another number, the second number is subtracted or taken away from the first number.
- mobile app** [N-COUNT-U9] A **mobile app** is a program that can be downloaded on a mobile phone or other handheld device.
- monitor** [N-COUNT-U3] A **monitor** is an electronic device that is used to display computer signals.
- motherboard** [N-COUNT-U6] A **motherboard** is the central circuit board of a computer.
- multimedia player** [N-COUNT-U9] A **multimedia player** is a device that can play audio, video, and other files.
- multiply by** [V PHRASE-U11] To **multiply** a number (A) **by** another number (B) is to add number A to itself B number of times.
- note card** [N-COUNT-U14] A **note card** is a small piece of paper that reminds a speaker what to say during a speech.
- notebook** [N-COUNT-U2] A **notebook** is a mobile computer that is typically smaller than a laptop, and often has fewer features or functions.
- numerator** [N-COUNT-U12] A **numerator** is a number that appears above the line in a fraction. In the fraction  $1/2$ , it is the 1.
- office suite** [N-COUNT-U8] An **office suite** is a group of programs that generally includes a word processor, a spreadsheet program, and a presentation program.
- open** [V-T-U10] To **open** something on a computer is to reveal its contents, usually for the purpose of using or editing it.
- operate** [V-I-U5] To **operate** is to function in a specific manner according to specifications of operating systems or other software.
- operating system** [N-COUNT-U5] An **operating system** is a type of software that manages a computer's hardware and allows the user to perform basic operations on a computer.

- optical mouse** [N-COUNT-U3] An **optical mouse** is a mouse that uses LEDs to track hand movements relative to a surface.
- out of** [ADJ-U12] If a quantity is **out of** y, it has x parts per every y parts possible.
- over** [PREP-U11] If a number is **over** another number, it is divided by that number.
- payroll** [N-COUNT-U9] A **payroll** is a list or database of a company's employees and how much they are paid.
- PC** [N-COUNT-U2] A **PC** (personal computer) is a computer that is intended for individual use. It is sometimes used specifically to refer to a personal computer with the Microsoft Windows® operating system.
- percent** [N-COUNT-U12] A **percent** is a part of 100 that is usually represented with the “%” symbol.
- percentage** [N-COUNT-U12] A **percentage** is the rate at which something occurs, measured per 100 units.
- peripheral** [N-COUNT-U3] A **peripheral** is a device that is connected to a computer but is not built into it.
- permit** [V-T-U6] To **permit** something is to allow it.
- plus** [PREP-U11] If a number is **plus** another number, the two numbers are added together.
- point** [N-COUNT-U12] A **point** is a dot placed after a whole unit in a decimal number.
- port** [N-COUNT-U6] A **port** is an outlet that allows an electronic device to communicate to other devices using a plug or cable.
- power supply** [N-COUNT-U6] A **power supply** is a device that provides energy to something.
- presentation** [N-COUNT-U14] A **presentation** is a process of formally introducing or demonstrating an idea to a group of people.
- processor** [N-COUNT-U6] A **processor** is a computer part that allows programs to be interpreted and run.
- program** [N-COUNT-U7] A **program** is a set of coded software that controls the operations of a computer or other electronic device.
- programming** [N-UNCOUNT-U15] **Programming** is the process of writing computer software.
- programming language** [N-COUNT-U7] A **programming language** is an artificial language used to express functions that can be performed by a computer or other machine.
- programming software** [N-UNCOUNT-U7] **Programming software** is an application that programmers use to create, fix, or support other programs or applications.
- programming-in-the-large** [N-UNCOUNT-U1] **Programming-in-the-large** is an approach to writing complicated software that involves the work of many people over a long period of time.
- programming-in-the-small** [N-UNCOUNT-U1] **Programming-in-the-small** is an approach to writing simple software that involves individuals or small groups of people.
- project** [V-T-U14] To **project** one's voice is to speak loudly so that people can hear it from a distance.
- quarantine** [V-T-U6] To **quarantine** a computer file is to separate it from others in order to prevent the spread of a virus.
- reduce** [V-T-U12] To **reduce** a fraction is to change it to a form with the lowest possible whole numbers.
- removal** [N-UNCOUNT-U6] **Removal** is the act of taking something out of or away from somewhere.
- review** [V-T-U14] To **review** something is to go over it closely and analyze or discuss its qualities.
- right-click** [V-T-U10] To **right-click** something is to bring up available actions by clicking the button on the right side of the mouse.
- rise** [N-COUNT-U13] A **rise** is the process of increasing in a quality or amount.
- route planning** [N-UNCOUNT-U9] **Route planning** is the process of determining how to reach a destination.
- run** [V-T-U10] To **run** something on a computer is to use a program or let it perform an action.
- satellite navigation** [N-UNCOUNT-U9] **Satellite navigation** is the process of determining a location using a map that receives information from satellites.
- scanner** [N-COUNT-U3] A **scanner** is an electronic device that copies images or documents and transfers them into a computer.

# Glossary

- scroll wheel** [N-COUNT-U3] A **scroll wheel** is a device on a mouse that allows the user to scroll.
- security software** [N-UNCOUNT-U6] **Security software** is a computer program that is designed to protect a computer from threats such as viruses and unauthorized access.
- select** [V-T-U10] To **select** something is to use keystrokes or a mouse to mark something on a computer screen for a particular operation.
- server** [N-COUNT-U2] A **server** is a central computer that provides services or stores data for many other computers.
- signpost** [V-I-U14] To **signpost** is to guide listeners through a sequence of ideas using clear directional wording.
- simulation** [N-COUNT-U9] A **simulation** is a representation of something that is designed to be very realistic.
- software** [N-UNCOUNT-U1] **Software** is a program or set of programs that perform particular functions or series of functions on a computer.
- source code editor** [N-COUNT-U7] A **source code editor** is a text editor program that changes the source code of other computer programs.
- spreadsheet** [N-COUNT-U8] A **spreadsheet** is a computer program that organizes information into columns and rows, and is often used to make calculations.
- spyware** [N-UNCOUNT-U6] **Spyware** is a type of malware that records personal information about a user, usually for the purpose of reporting it to an unauthorized party.
- stabilize** [V-I-U13] To **stabilize** is to reach a state in which changes are infrequent.
- steady** [ADJ-U13] If something is **steady**, it doesn't change, or changes at a slow and constant rate.
- subtract** [V-T-U11] To **subtract** one number from another number is to reduce it by that amount.
- summarize** [V-T-U14] To **summarize** something is to briefly present its main points.
- system software** [N-UNCOUNT-U5] **System software** is software that allows a user to operate a computer and run additional software on a computer.
- tablet** [N-COUNT-U2] A **tablet** is a portable computer with a flat touch screen that is smaller than a laptop and usually does not have a physical keyboard.
- test** [V-T-U1] To **test** something is to do or operate it to see what it does or whether it works properly.
- text editor** [N-COUNT-U7] A **text editor** is a program that allows users to change and input information in a computer.
- times** [PREP-U11] If a number is **times** another number, it is multiplied by that number.
- trend** [N-COUNT-U13] A **trend** is a consistent change or development.
- video editing** [N-UNCOUNT-U8] **Video editing** is the process of editing video sequences on a computer.
- virus** [N-COUNT-U6] A **virus** is a type of malware that usually damages a computer, and is designed to replicate itself and spread to other machines.
- visual aid** [N-COUNT-U14] A **visual aid** is a physical representation of an important part of a presentation, designed to support the speaker's ideas.
- web browser** [N-COUNT-U8] A **web browser** is a program that is used to navigate the internet by displaying webpages and following hyperlinks.
- windowing system** [N-COUNT-U5] A **windowing system** is a user interface in which a computer's running applications are organized into visual boxes that a user can easily navigate.
- wireless** [ADJ-U3] If a network is **wireless**, it does not use wires to connect computers.
- word processing** [N-UNCOUNT-U8] **Word processing** is the process of entering and editing text for the production of text documents.
- workstation** [N-COUNT-U2] A **workstation** is a powerful computer that can process complex tasks that typical desktops cannot handle, such as advanced graphics design.
- write** [V-T-U1] To **write** something is to form letters and words into sentences or instructions.

**CAREER  
PATHS**

# SOFTWARE ENGINEERING

Book  
**2**

Virginia Evans  
Jenny Dooley  
Enrico Pontelli



**Express Publishing**

## Scope and Sequence

Unit	Topic	Reading context	Vocabulary	Function
1	Traits of a Software Engineer	Webpage	ability, commitment, critical thinking, curious, dedicated, expertise, focus, goal-orientated, innovative, logical, outside the box, team player	Expressing a preference
2	Problem Solving	Email	address, analysis, application, approach, iterative, iteration, problem identification, procedure, problem solving, redefine, solution, synthesis	Asking about progress
3	Accounting	Textbook Excerpt	closed system, consumption, equation, extensive quantity, final, generation, initial, intensive quantity, input, open system, output, system, universal accounting	Making comparisons
4	Requirements Engineering	Webpage	customer-driven, elicitation, functional hierarchy, market-driven, mode, object, requirement engineering, response, specification, user class, user-friendly, validation, verification	Expressing relief
5	Software Architecture	Agenda	application framework, conceptual view, deployment view, design pattern, idiom, implementation view, module, process view, programming plan, software architecture	Setting a deadline
6	Architectural Styles	Textbook Excerpt	abstract data type, component, connector, control structure, DSSA, implicit invocation, layered, main program with subordinates, pipes and filters, repository, system model	Disagreeing with an opinion
7	Design Considerations	Blog	abstraction, call graph, cohesion, complexity, coupling, information hiding, inter-modular attributes, simplify, stopping rule, system structure, wicked problem	Expressing confusion
8	Design Methods 1	Course Description	bottom-up design, decompose, design method, functional decomposition, idealistic, philosophy, primitive, rational, stepwise refinement, subfunction, top-down design	Discussing pros and cons
9	Design Methods 2	Textbook Excerpt	data flow design, DFD, implementation stage, JSD, JSP, modeling stage, network stage, SA, schematic logic, structure chart, SD, structure diagram	Making a polite request
10	Design Methods 3	Online encyclopedia article	attribute, Booch method, class, class diagram, collaboration diagram, fusion method, interaction diagram, object-oriented, OMT, property, sequence diagram, state, state diagram	Confirming information
11	Software Testing Objectives	Webpage	compare, demonstration model, destruction model, error, evaluation model, expected, error, failure, fault, fault detection, fault prevention, oracle, prevention model, satisfy, test criterion	Making a prediction
12	Software Testing Techniques	Journal Article	black-box testing, coverage-based testing, dynamic analysis, error-based testing, Fagan inspection, fault-based testing, peer review, proof of correctness, static analysis, scenario-based evaluation, stepwise abstraction, white-box testing	Delivering bad news
13	Test Adequacy Criteria	Handbook	anticomposition property, antidecomposition property, antiextensionality property, applicability property, complexity property, general multiple change property, inadequate empty set, monotonicity property, non-exhausting applicability property, renaming property, statement coverage property, test adequacy criteria	Giving advice
14	Software Maintenance 1	Memo	adaptive maintenance, corrective maintenance, enhance, insufficient, law of continuing change, law of increasing complexity, perfective maintenance, preventive maintenance, release, repair, software maintenance, unstructured code	Describing order of events
15	Software Maintenance 2	Journal Article	design recovery, functional equivalence, legacy system, modernize, redocumentation, reengineering, renovation, restructuring, revamping, reverse engineering, web-based	Describing degree

# Table of Contents

<b>Unit 1 – Traits of a Software Engineer</b> .....	4
<b>Unit 2 – Problem Solving</b> .....	6
<b>Unit 3 – Accounting</b> .....	8
<b>Unit 4 – Requirements Engineering</b> .....	10
<b>Unit 5 – Software Architecture</b> .....	12
<b>Unit 6 – Architectural Styles</b> .....	14
<b>Unit 7 – Design Considerations</b> .....	16
<b>Unit 8 – Design Methods 1</b> .....	18
<b>Unit 9 – Design Methods 2</b> .....	20
<b>Unit 10 – Design: Methods 3</b> .....	22
<b>Unit 11 – Software Testing Objectives</b> .....	24
<b>Unit 12 – Software Testing Techniques</b> .....	26
<b>Unit 13 – Test Adequacy Criteria</b> .....	28
<b>Unit 14 – Software Maintenance 1</b> .....	30
<b>Unit 15 – Software Maintenance 2</b> .....	32
<b>Glossary</b> .....	34

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What qualities are important when working closely with other people?
- 2 How do engineers benefit from being both logical and creative?

## Reading

2 Read the webpage. Then, choose the correct answers.

- 1 What is the main idea of the webpage?
  - A descriptions of people who work at an engineering firm
  - B the types of projects that an engineering firm handles
  - C clients' experiences with an engineering firm
  - D job opportunities at an engineering firm
- 2 Which of the following is NOT true of the engineering firm?
  - A It attracts talented engineers.
  - B It won awards for its creative projects.
  - C It was started by two engineers.
  - D It is a profitable company.
- 3 According to the webpage, how are the two owners similar?
  - A They are passionate about their jobs.
  - B They can focus for long periods.
  - C They have highly creative minds.
  - D They are naturally curious people.

### MeshSoft Software Engineering & Development > About Us > Meet the Staff

**Fiona Adler, Co-owner/Founder:** Fiona started MeshSoft six years ago. Her **dedicated** attitude and **commitment** to success built a highly profitable business. As a naturally **curious** person, she enjoys exploring new ideas. MeshSoft earned several awards last year for Fiona's **innovative** programs and systems. Nobody thinks **outside the box** like Fiona!

Fiona is everyone's favorite boss. She's a **team player** who encourages employees to share ideas. Bright, talented engineers love working at MeshSoft.

**Maxine Spencer, Co-owner:** Maxine purchased part of the business last year. Fiona's creativity is balanced by Maxine's technical **expertise**. Maxine has a remarkably **logical** mind. She enjoys **critical thinking**, and she's good at it, too. If Fiona can imagine something, Maxine can usually make it happen. They make a great team.

Maxine has the **ability** to **focus** intently for long periods. Her **goal-oriented** attitude keeps MeshSoft moving forward. Like Fiona, Maxine has great dedication and passion for her job.

## Vocabulary

3 Match the words and phrases (1-8) with the definitions (A-H).

- |              |                        |
|--------------|------------------------|
| 1 __ focus   | 5 __ dedicated         |
| 2 __ ability | 6 __ team player       |
| 3 __ logical | 7 __ outside the box   |
| 4 __ curious | 8 __ critical thinking |

- A the skill to do something
- B wanting to know more about something
- C related to unusual or creative ideas
- D the skill of drawing conclusions based on facts
- E enthusiastic about a task or cause
- F based on evidence and reason
- G someone who takes actions that benefit a group
- H to watch closely



4 Read the sentence pairs. Choose which word or phrase best fits each blank.

1 goal-oriented / innovative

A A(n) \_\_\_\_\_ engineer is always thinking about the final product.

B A(n) \_\_\_\_\_ engineer is always coming up with new ideas.

2 commitment / expertise

A The intern had no experience, but the firm was impressed with her level of \_\_\_\_\_.

B The engineer went back to school to gain more advanced \_\_\_\_\_ in his field.

5 Listen and read the webpage again. What are some ways to describe someone who thinks of unusual ideas?

## Listening

6 Listen to a conversation between an engineering firm owner and a manager. Mark the following statements as true (T) or false (F).

1 \_\_\_ The first applicant has experience at multiple engineering firms.

2 \_\_\_ The second applicant recently completed an engineering degree program.

3 \_\_\_ The man once worked on a project with the second applicant.

7 Listen again and complete the conversation.

Owner: So she has strong 1 \_\_\_\_\_. Who else is there?

Manager: I like Henry Pacheco, too. But he's 2 \_\_\_\_\_.

Owner: That might be okay. What's his experience like?

Manager: He 3 \_\_\_\_\_ the X-99 traffic analysis software.

Owner: Really? That was a fascinating project. Those engineers really had to 4 \_\_\_\_\_.

Manager: Yeah, that's what I thought. Should I set up an 5 \_\_\_\_\_?

Owner: Definitely. Experience is nice. But I'd rather go with a 6 \_\_\_\_\_.

Manager: Okay, I'll call Henry right away.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I like this one ...*

*Those engineers really had to ...*

*I'd rather go with ...*

**Student A:** You are a company owner. Talk to Student B about:

- job applications
- applicants' qualifications
- what you are looking for in a new employee

**Student B:** You are a manager. Talk to Student A about qualifications of job applicants.

## Writing

9 Use the conversation from Task 8 to complete the job interview notes.

### MeshSoft Software Engineering & Development

#### Job Interview Notes

Applicant: \_\_\_\_\_

Qualifications: \_\_\_\_\_

Do you plan to hire this applicant? Why or why not? \_\_\_\_\_

Applicant: \_\_\_\_\_

Qualifications: \_\_\_\_\_

Do you plan to hire this applicant? Why or why not? \_\_\_\_\_

# 2 Problem Solving

**From:** a.jones23@graylinesoft.net  
**To:** t.byman@graylinesoft.net  
**Subject:** Tips on Problem Solving

Hey Tim,

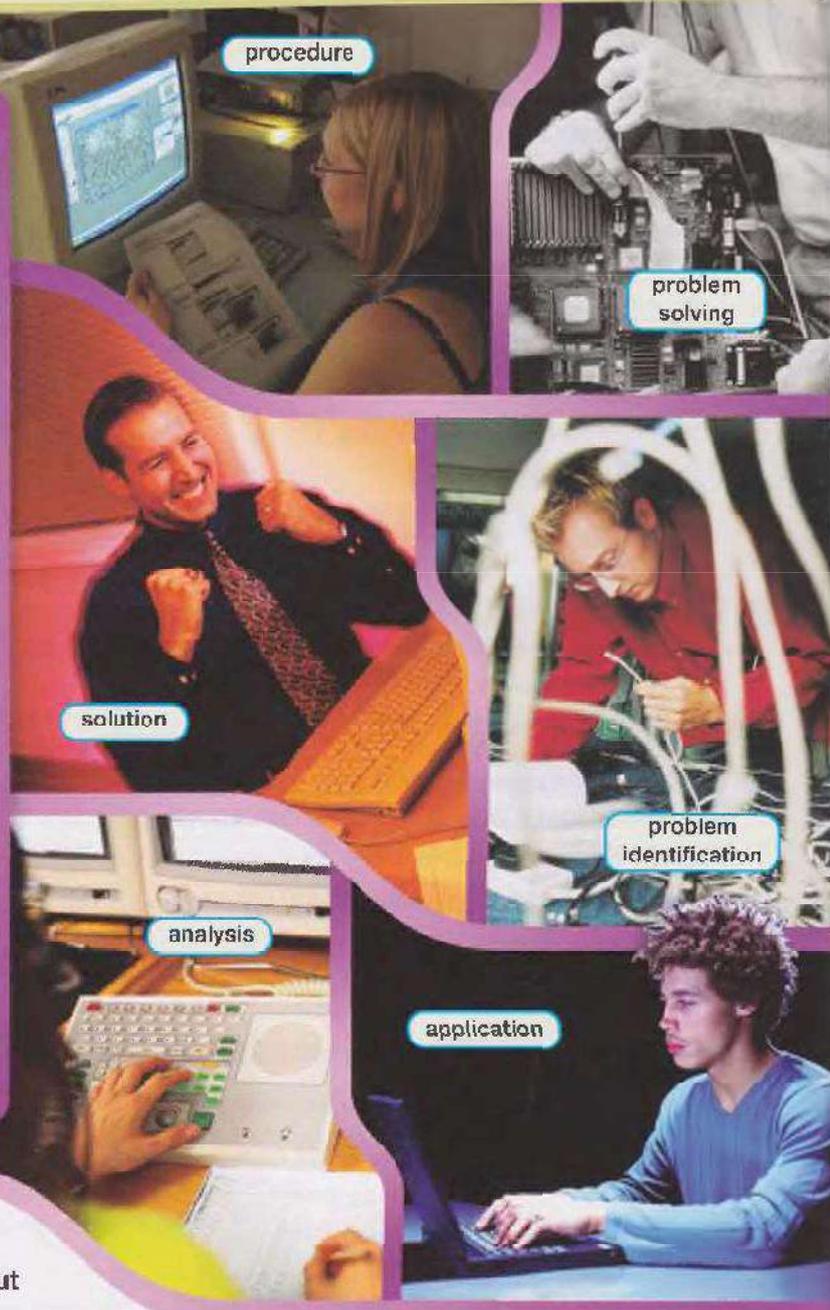
I heard you have some bugs with your latest project. I know you're new to the company. This is what works for me when I **address** problems.

The most important part of problem solving is **problem identification**. Remember to take a calm **approach** to the situation. Start with a complete **analysis** of possible causes. Then, check established **procedures** for solving common problems. Finding the problem is often the toughest step. But once you do that, **application** of a **solution** is usually easy.

With an **iterative** process like ours, things can be a little frustrating. With each **iteration**, the software is **redefined**. This means that everything we develop presents new problems. Creating a **synthesis** of two different solutions is especially difficult. Luckily, there's a solution for every problem. If you can't find it, somebody else here surely can. Don't be afraid to ask for assistance.

I hope this helps! Good luck on your project!

-Amanda



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 How are software problems identified?
- 2 What are the basic steps of problem solving?

## Reading

2 Read the email. Then, mark the following statements as true (T) or false (F).

- 1  Checking for established procedures is the first recommended step.
- 2  An iterative development process presents fewer problems than other types of processes.
- 3  According to the email, solutions are usually easy to apply.

## Vocabulary

3 Match the words and phrases (1-6) with the definitions (A-F).

- |  |   |
|--|---|
| 1 <input type="checkbox"/> application | 4 <input type="checkbox"/> problem identification |
| 2 <input type="checkbox"/> approach    | 5 <input type="checkbox"/> problem solving        |
| 3 <input type="checkbox"/> iterative   | 6 <input type="checkbox"/> synthesis              |

- A intended to be updated continually
- B a combination of multiple elements or things
- C the action of putting something into operation
- D a way of viewing or dealing with something
- E the ability to eliminate problems
- F the act of analyzing and describing problems

**4** Read the sentence pairs. Choose which word best fits each blank.

**1 address / redefine**

- A An updated version of a product can \_\_\_\_\_ the purpose of the product.
- B It's important to \_\_\_\_\_ problems with efficiency.

**2 analysis / solution**

- A Once the cause of a problem has been identified, it is easier to find a(n) \_\_\_\_\_.
- B Conducting a(n) \_\_\_\_\_ of a problem is a good first step in problem solving.

**3 iteration / procedure**

- A Each \_\_\_\_\_ of a product should feature improvements on the last one.
- B A(n) \_\_\_\_\_ gives instructions for completing a process.

**5** Listen and read the email again. What is the first step in problem solving?

**Listening**

**6** Listen to a conversation between two software engineers. Choose the correct answers.

- 1 What solution did the woman try?
  - A a quick analysis of the code
  - B consulting another engineer for assistance
  - C a synthesis of two iterations
  - D replacing the program with all new software
- 2 What will the woman likely do next?
  - A try a synthesis of her previous attempts
  - B call an administrator for help
  - C apply virus detection solutions
  - D redefine the purpose of the software

**7** Listen again and complete the conversation.

Engineer 1: Hey. How is your **1** \_\_\_\_\_ ?  
 Engineer 2: Not very well. I'm still having a **2** \_\_\_\_\_ when I try to run the software.  
 Engineer 1: Really? What have you done to fix it?  
 Engineer 2: I did **3** \_\_\_\_\_ of the code.  
 Engineer 1: Have you tried anything else?  
 Engineer 2: Yeah, I ran some standard debugging procedures. Then I tried applying some **4** \_\_\_\_\_.  
 Engineer 1: Huh. What **5** \_\_\_\_\_ next?  
 Engineer 2: Well, I just started a **6** \_\_\_\_\_ analysis of all the code.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*How is ... going?*  
*Have you tried ...*  
*I'm completing a ...*

**Student A:** You are an engineer. Talk to Student B about:

- problems with a project he or she is working on
- what solutions he or she tried already
- how he or she plans to fix the problem

**Student B:** You are an engineer. Talk to Student A about the problem solving methods you are using on a project.

**Writing**

**9** Use the conversation from Task 8 to complete the instructions for problem solving.

**Problem Solving Procedure**

- 1 \_\_\_\_\_ . A good way to do this is \_\_\_\_\_ .
- 2 Think of a solution. It's a good idea to think of a few possible solutions for the problem.  
\_\_\_\_\_ .
- 3 \_\_\_\_\_ . A good way to do this is \_\_\_\_\_ .
- 4 Complete an analysis to ensure that the problem has been fully resolved.

## Accounting

Accurate accounting is an important part of any engineering project. Engineers must carefully monitor various quantities.

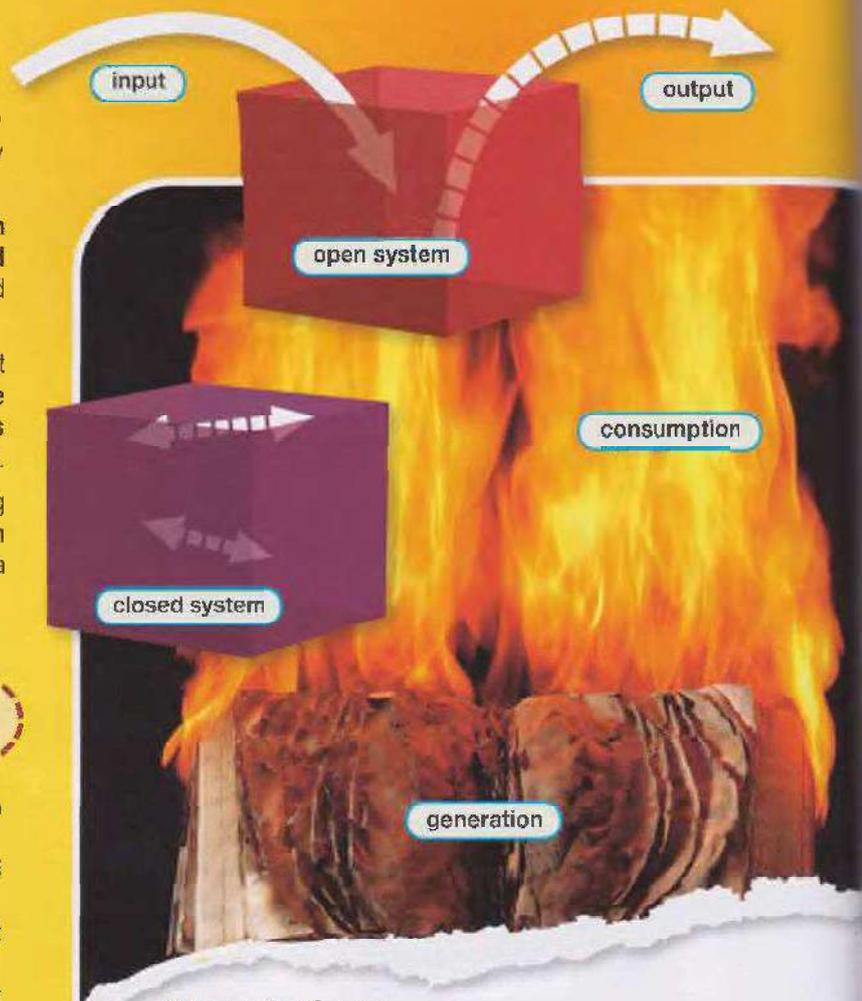
First, engineers must identify what type of **system** is monitored. Mass remains constant in a **closed system**. In an **open system**, mass passes in and out.

The nature of the quantity is also an important consideration. Some quantities, called **extensive quantities**, are countable. **Intensive quantities** are those that can be measured, but not counted.

Once these factors are determined, accounting can begin. The **universal accounting equation (UAE)** is a useful formula for engineers. It is a simple way to measure changes in amounts:

$$\text{Final amount} - \text{initial amount} = \text{input} - \text{output} + \text{generation} - \text{consumption}$$

- Step 1:** Take the amount that you started with (input).
- Step 2:** Subtract any amount that was removed (output).
- Step 3:** Add the amount of new material that was created (generation).
- Step 4:** Subtract any material that was lost or destroyed (consumption).



### Get ready!

1 Before you read the passage, talk about these questions.

- Why do engineers need accounting systems?
- What is the relationship between consumption and generation?

### Reading

2 Read the textbook excerpt. Then, mark the following statements as true (T) or false (F).

- Extensive quantities are difficult to measure.
- Engineers use the UAE to determine whether a system is open or closed.
- According to the UAE, consumption is subtracted from generation.

### Vocabulary

3 Match the words and phrases (1-8) with the definitions (A-H).

- |              |                          |
|--------------|--------------------------|
| 1 __ final   | 5 __ system              |
| 2 __ input   | 6 __ intensive quantity  |
| 3 __ initial | 7 __ extensive quantity  |
| 4 __ output  | 8 __ accounting equation |
- A a type of measurable quantity that cannot be counted
- B relating to the status of something at the beginning of a period of time
- C a type of measurable quantity that can be counted
- D a set of connected things that work together
- E the amount of a countable quantity that is added to a system
- F the amount of a countable quantity that is removed from a system
- G a way to measure changes in countable quantities
- H relating to the status of something at the end of a period of time

4 Read the sentence pairs. Choose which word or phrase best fits each blank.

1 open system / closed system

- A In a(n) \_\_\_\_\_, the amount of mass doesn't change.  
 B Mass can leave a(n) \_\_\_\_\_.

2 generation / consumption

- A The smoke that is produced by a fire is an example of \_\_\_\_\_.  
 B The wood that burns in a fire is an example of \_\_\_\_\_.

5 Listen and read the textbook excerpt again. What is the difference between open and closed systems?

## Listening

6 Listen to a conversation between two software engineering students. Choose the correct answers.

- 1 What is the main idea of the conversation?  
 A a comparison of accounting terms  
 B an error in a UAE calculation  
 C a review of a recent accounting test  
 D how accounting differs from one system to another
- 2 What does the man say about temperature?  
 A It illustrates the idea of a closed system.  
 B It is a confusing concept.  
 C It can be measured and counted.  
 D It is an intensive quantity.

7 Listen again and complete the conversation.

- Student 1: Those are easy to 1 \_\_\_\_\_.  
 Student 2: So let me see if I can remember this. If you can count something, 2 \_\_\_\_\_.  
 Student 1: That sounds right. On the other hand, intensive quantities 3 \_\_\_\_\_.  
 Student 2: Right, but you can 4 \_\_\_\_\_.  
 Student 1: I still 5 \_\_\_\_\_ how that works.  
 Student 2: Think about 6 \_\_\_\_\_. You can't count it. But you can still monitor its changes.  
 Student 1: Okay, that makes sense.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

- Do you remember the difference between ...*  
*On the other hand ...*  
*I still don't get ...*

**Student A:** You are a student. Talk to Student B about:

- differences between accounting concepts
- a concept that confuses you
- an example of the

**Student B:** You are a student. Talk to Student A about differences between accounting concepts.

## Writing

9 Use the conversation from Task 8 to complete the accounting quiz.

### SoftEng 120: Accounting Quiz #2

Please name two accounting concepts. Then, give a real-world example of each concept.

Concept	Example
1 Intensive Quantity	_____ This is a good example of this concept because _____ _____
2 _____	_____ This is a good example of this concept because _____ _____

## Get ready!

- 1 Before you read the passage, talk about these questions.
- 1 What is the purpose of requirements engineering?
  - 2 How are software specifications organized?

HOME ABOUT US OUR SERVICES CONTACT

### ShorSoft Software Developers Requirements Engineering Department

We understand that functional and **user-friendly** software comes from well-defined foundations. Therefore, we take great pride in our requirements engineering department. Our engineers perfect every **specification** for the software that we create.

We primarily create **market-driven** software for general consumers. However, we also create some **customer-driven** software for special projects. First, our team makes a detailed requirements document. This is based on **elicitation** of the requirements. Typically, the information comes from people who will likely use the software. Then, a separate team provides **validation** and **verification** for that document. This process prevents mistakes and ensures precision.

We organize requirements by the most appropriate parameters for the software. Specifications can be organized by the **mode**, **user class**, or **response**. In less-common cases, requirements documents are organized by real-life **objects**. Or they might be determined by some other **functional hierarchy**.

One way or another, we'll get the job done. For more information about requirements engineering services, contact our customer service department.

## Reading

- 2 Read the webpage. Then, choose the correct answers.

- 1 What is the purpose of the webpage?
  - A to list job qualifications for a requirements engineering position
  - B to define various requirements engineering concepts
  - C to review a company's requirements engineering process
  - D to describe the work experience of several requirements engineers
- 2 Which of the following is NOT a parameter used to organize requirements?
  - A user class
  - B objects
  - C mode
  - D verification
- 3 Why does a second team of engineers provide validation and verification?
  - A to avoid errors in the requirements document
  - B to organize requirements specifications
  - C to ensure elicitation of important information
  - D to determine the most appropriate functional hierarchy

## Vocabulary

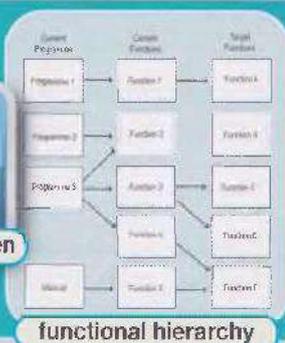
- 3 Match the words and phrases (1-8) with the definitions (A-H).

- |                               |                    |
|-------------------------------|--------------------|
| 1 customer-driven             | 5 elicitation      |
| 2 market-driven               | 6 validation       |
| 3 specification               | 7 verification     |
| 4 __ requirements engineering | 8 __ user-friendly |

- A designed in response to specific needs of potential users
- B a precise definition of a problem
- C designed for broad purposes
- D the process of becoming apparent or realized
- E the act of checking that requirements are correct
- F the act of checking that requirements are stated correctly
- G easy for most people to understand or use
- H the practice of specifying the necessary features and functions of software



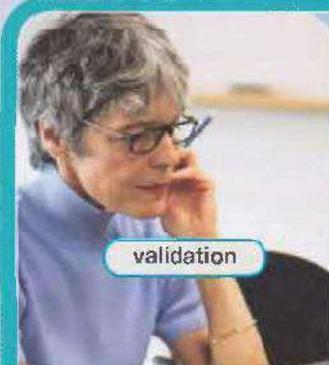
customer-driven



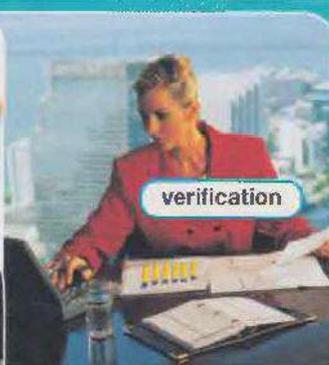
functional hierarchy



user class



validation



verification

**4 Write a word or phrase that is similar in meaning to the underlined part.**

- Some software is defined in terms of its relationship to real life things that can be touched or seen.      j c s
- Software specifications can change depending on the status of the person who is using the software.      \_ s \_ r \_ c \_ \_ s \_
- If no traditional specifications are appropriate, a requirements document can be organized by any undefined system.      n   i   a \_ \_ \_ e \_ \_ r \_ h \_
- Software that changes according to the way it is used can be defined according to its changeable system of operation.      \_ \_ d e
- The type of information provided by software upon request is sometimes a specification in requirements documents.      \_ \_ s \_ o \_ s \_

**5 Listen and read the webpage again. How do engineers identify errors in a requirements document?**

**Listening**

**5 Listen to a conversation between two engineers. Mark the following statements as true (T) or false (F).**

- The engineers recently assessed requirements for a software update.
- The specifications were made with the wrong data.
- The engineers missed an important deadline.

**7 Listen again and complete the conversation.**

**Engineer 1:** Can you give me an update 1 \_\_\_\_\_ for the library catalog application?

**Engineer 2:** It's going slowly. It was good at first, but we found some major problems when we did 2 \_\_\_\_\_.

**Engineer 1:** I don't like the sound of that. 3 \_\_\_\_\_?

**Engineer 2:** 4 \_\_\_\_\_ were based on old data. The requirements didn't include the library's new DVD catalog.

**Engineer 1:** Oh, that's right. Originally, they only included books. Will we able to fix the problem?

**Engineer 2:** We're 5 \_\_\_\_\_ now. Then we'll just plug it into the existing requirements document.

**Engineer 1:** How long will that take?

**Engineer 2:** We should have it done by the 6 \_\_\_\_\_.

**Speaking**

**8 With a partner, act out the roles below based on Task 7. Then, switch roles.**

**USE LANGUAGE SUCH AS:**

- Can you give ...
- How long ...
- I was worried ...

**Student A:** You are an engineer. Talk to Student B about:

- a project your company is working on
- problems with the requirements document
- when the project will be completed

**Student B:** You are an engineer. Talk to Student A about a project your company is working on.

**Writing**

**9 Use the conversation from Task 8 to complete the project update.**

**ShorSoft Software Developers**

From the desk of: Allison Baxter

Hi Greg,

Here is an update on the project.

Current stage: \_\_\_\_\_

Problems encountered: \_\_\_\_\_

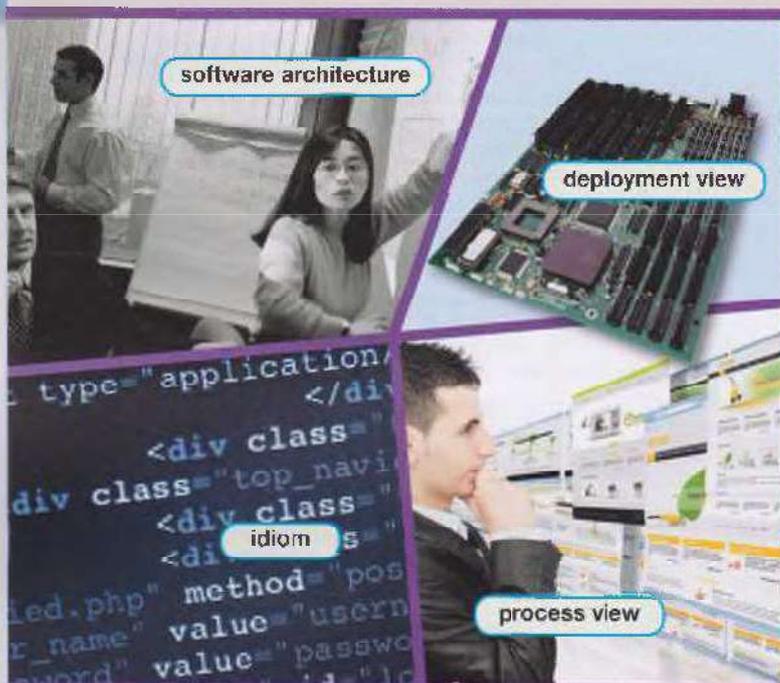
Next steps: \_\_\_\_\_

Let me know if you have any questions.  
-Allison

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What is the purpose of software architecture?
- 2 What are some different views in software architecture?

Meeting Agenda – July 22<sup>nd</sup>

## Presentation Information:

Sheryl will discuss the updated **application framework** for our new accounting software. The **process view** and **implementation view** of the project are already complete. Sheryl will share those views with us. She will also talk about developing the **deployment view**. Finally, she will share plans for the **conceptual view** in the next phase.

Then, Scott will share some of his new **programming plans**. These general plans will be helpful for various current and future projects.

Erik will present on his team's new word processing **modules**. These modules will be the basis for several software upgrades. He will explain how simple **idioms** can create complex modules.

Finally, Maria will give a presentation on utilizing **design patterns**. These can be used to solve common problems in **software architecture**. Her presentation will focus on mistakes during our last project. This should give us an idea of which areas need the most improvement.

## Reading

2 Read the agenda. Then, choose the correct answers.

- 1 What is the purpose of the meeting?
  - A to give progress updates on different projects
  - B to assign tasks to each employee
  - C to explain new techniques in software architecture
  - D to brainstorm new applications to develop
- 2 Which of the following is NOT a topic that will be covered at the meeting?
  - A plans for developing the deployment view of accounting software
  - B idioms that can be used to create modules
  - C problems caused by design patterns
  - D new programming plans for future projects
- 3 What will the problem solving presentation focus on?
  - A examinations of application frameworks
  - B the progress of the process and implementation views
  - C new programming plans that have been developed
  - D examples of previous mistakes

## Vocabulary

3 Match the words and phrases (1-4) with the definitions (A-D).

- 1 \_\_\_ conceptual view
- 2 \_\_\_ implementation view
- 3 \_\_\_ process view
- 4 \_\_\_ deployment view

- A a view that shows modules of packages and layers
- B a view that shows the way tasks are assigned to physical nodes
- C a view that shows major design elements and interactions between elements
- D a view that shows the interaction of tasks and processes of a system

- 4 Fill in the blanks with the correct words and phrases from the word bank.

**Word BANK**

application framework    design pattern    idiom  
module    programming plan    software architecture

- 1 \_\_\_\_\_ divides the components of software according to their functions.
  - 2 A(n) \_\_\_\_\_ is a commonly used fragment of code.
  - 3 Similar software functions may be bundled into a(n) \_\_\_\_\_.
  - 4 Families of systems are connected with a(n) \_\_\_\_\_.
  - 5 A(n) \_\_\_\_\_ is used for common functions or actions.
  - 6 A(n) \_\_\_\_\_ can be used to fix common problems.
- 5 Listen and read the agenda again. What information will Sheryl present?

**Listening**

- 6 Listen to a conversation between a manager and a software engineer. Mark the following statements as true (T) or false (F).
- 1 \_\_\_ The woman will give the same presentation to a different client.
  - 2 \_\_\_ The man gives the woman advice on improving his presentation.
  - 3 \_\_\_ The implementation view for the new project is complete.

- 7 Listen again and complete the conversation.

**Manager:** No. Actually, quite the opposite. I really enjoyed 1 \_\_\_\_\_ last week on the design pattern for our current project.

**Engineer:** Really? Thank you. It's hard to 2 \_\_\_\_\_ to clients, but I did my best.

**Manager:** You did a great job. Would you be interested in putting together 3 \_\_\_\_\_ presentation?

**Engineer:** Sure. I'd be happy to.

**Manager:** Great. I need to give a presentation to some other clients. It's an overview of our progress on the 4 \_\_\_\_\_.

**Engineer:** Sure. Just send me the information. I'm sure I can 5 \_\_\_\_\_ together.

**Manager:** Great. We already have 6 \_\_\_\_\_. I'll send it to you this afternoon.

**Speaking**

- 9 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*If you ... I can ...  
Would you be interested in ...  
I need to ...*

**Student A:** You are a company manager. Talk to Student B about:

- a presentation that he or she gave
- what you enjoyed about the presentation
- what you would like him or her to talk about in a future presentation

**Student B:** You are an engineer. Talk to Student A about a presentation that you recently gave.

**Writing**

- 9 Use the conversation from Task 8 to complete the meeting agenda.

Linda will present on \_\_\_\_\_

Terry will present on \_\_\_\_\_

Alan will present on \_\_\_\_\_

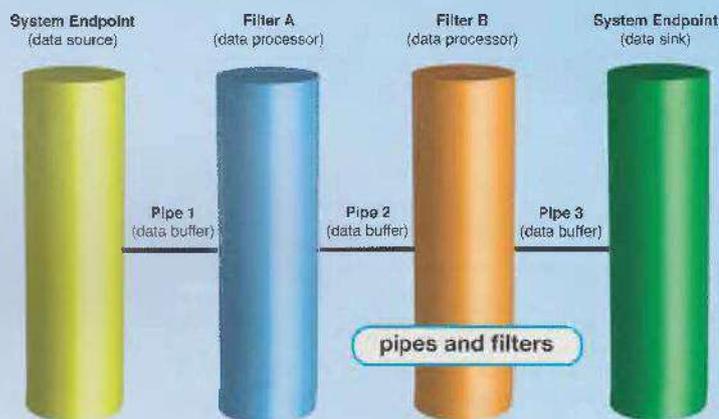
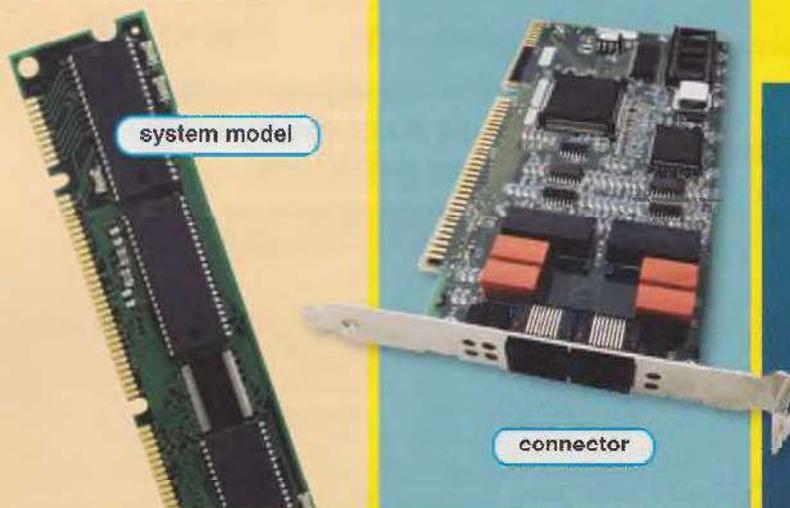
## Chapter 5 Architectural Styles

Software architecture provides a framework for engineers to create software systems. Use of these frameworks is called Domain-Specific Software Architecture, or **DSSA**. Structures in software architecture are loosely defined. They provide a standard plan and **control structure** for engineers to use.

The engineer must decide what type of framework will work best for a project. The engineer should consider the type of data that he or she is working with. It is also important to consider the functions that the software will perform. This process determines the system requirements. After the system is planned, the engineer adds specific **components** and **connectors** to it. The result is a complete **system model**. This is a plan for how the software will operate.

Below are some common structures in software architecture. These structures will be described in detail later in this chapter:

Abstract Data Type  
Implicit Invocation  
Layered  
Main Program with Subroutines  
Pipes and Filters  
Repository



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What is the purpose of domain-specific software architecture?
- 2 What are some different architectural styles for designing software?

## Reading

2 Read the textbook excerpt. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ Structures in software architecture are defined by strict limitations.
- 2 \_\_\_ An engineer adds components and connectors after determining system requirements.
- 3 \_\_\_ A system model is a common structure in software architecture.

## Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- |                           |                                     |
|---------------------------|-------------------------------------|
| 1 ___ abstract data type  | 4 ___ main program with subroutines |
| 2 ___ implicit invocation | 5 ___ pipes and filters             |
| 3 ___ layered             | 6 ___ repository                    |

- A a style in which computations occur because of internal events in the system
- B a hierarchical system in which a top-level module invokes other modules in a given order
- C a style that is designed for systems which manage a body of data with an inherent structure
- D a style that relies on input streams and system operations to process ordered data
- E a style that is designed to match the structure of the original data
- F organized by ascending functionality

**4** Write a word or phrase that is similar in meaning to the underlined part.

- A component that dictates the execution of other components maintains the proper order of operations within a system.  
c \_ \_ t \_ \_ \_ \_ \_ r \_ c \_ \_ \_ \_
- A computation element or data store is one of the basic parts of a software structure. \_ \_ \_ p \_ n \_ \_ t
- A description of the characterization of a system explains a system's components and their interactions.  
\_ \_ s \_ \_ m \_ m \_ \_ e \_
- An element that determines how components interact is an important part of a software architect's plan. \_ \_ n \_ \_ c \_ o \_
- Any style of system architecture must include a reference architecture, component library, and application configuration method. D \_ \_ A

**5** Listen and read the textbook excerpt again. How does a software engineer use an established structure?

**Listening**

**5** Listen to a conversation between two software engineers. Choose the correct answers.

- What is the purpose of the conversation?  
A to determine the best plan for the components and connectors  
B to decide what elements to include in a client presentation  
C to identify problems in a system model  
D to decide what type of structure to use
- According to the woman, what is a disadvantage of a pipes-and-filters system?  
A It is unnecessarily complicated.  
B It is not compatible with the data.  
C It is not what the client requested.  
D It functions on an ordered hierarchy.

**7** Listen again and complete the conversation.

Engineer 1: We need to start on this 1 \_\_\_\_\_. Do you have any ideas?

Engineer 2: Well, first, we need to decide on an 2 \_\_\_\_\_.

Engineer 1: I agree. But I'm not sure which one we should use.

Engineer 2: Hmm. We're going to bring in a lot of data from outside sources. Maybe we should think 3 \_\_\_\_\_.

Engineer 1: I thought about that. But I don't think 4 \_\_\_\_\_ work with a repository system.

Engineer 2: Okay. What 5 \_\_\_\_\_ then?

Engineer 1: I think 6 \_\_\_\_\_ work better. It has better mechanisms for ordering data.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I agree. But ...*

*I thought about ...*

*I don't really like ...*

**Student A:** You are an engineer. Talk to Student B about:

- the best architectural system for a software project
- the advantages and disadvantages of different systems

**Student B:** You are an engineer. Talk to Student A about the best architectural system for a software project.

**Writing**

**9** Use the conversation from Task 8 to complete the email about architectural styles for a project.

**Architectural Styles**

Hi Ken,

I looked over your ideas for the new project. You suggested using the \_\_\_\_\_ architectural style. I don't think this is a good idea because \_\_\_\_\_.

Instead, why don't you use the \_\_\_\_\_ architectural style? It's better because \_\_\_\_\_. Or you could use the \_\_\_\_\_ architectural style. The advantage of that is \_\_\_\_\_.

Let me know what you decide.  
-Sylvia

**Date:** 14 June

**Subject:** A Little Planning Leads to a Big Payoff in Software Design

Every engineer worries about **wicked problems**. Without **stopping rules**, engineers often don't know whether problems are fixed or not. But I'd like to point out that these problems can be easily avoided. Programmers just need to use **call graphs** to depict **system structures**. These let the programmer eliminate problems before they even begin.

Some programmers are geniuses with code, but they're unfamiliar with call graphs. This is unfortunate, because these systems greatly **simplify** the design process. Software planning provides programmers with an **abstraction** of the final product. Abstract systems are described in terms of their **modularity**, **cohesion**, and **coupling**. Programmers can even make allowances for **information hiding**.

When the plan is finished, the programmer can examine the **inter-modular attributes** and the **intra-modular attributes**. Errors can be eliminated while the software's **complexity** is low. Then, engineers are less likely to encounter complex problems later on.

The screenshot shows a software interface with a 'Select Attributes' dropdown menu containing 'total', 'Op', 'total', 'Opnd', 'branchCount', and 'defects'. Below the menu, there are several lines of code or log output. A blue callout box labeled 'stopping rule' points to a line of code. To the right, a hierarchical diagram labeled 'system structure' shows a tree of nodes representing system components.

The diagram is a call graph with a central node 'parse()' and several other nodes connected to it. The nodes include 'Element\_ptr()', 'token\_to\_comp()', 'add\_list()', 'add\_graph()', 'add\_section()', 'stop\_completed\_blocks()', 'raise\_graphs()', and 'tokenize()'. A blue callout box labeled 'call graph' points to the central 'parse()' node.

The image shows a man in a white shirt and glasses sitting at a desk with a computer monitor, looking thoughtful with his hand on his head. A blue callout box labeled 'wicked problem' points to him. To the right, a diagram shows a network of nodes representing modules. A legend indicates 'MODULE 1' (yellow), 'MODULE 2' (pink), and 'MODULE 3' (blue). The diagram is labeled with 'intra-modular attributes' and 'inter-modular attributes'. A legend at the bottom explains 'Usual links' (solid line) and 'Links assigned small random value' (dashed line).

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What does a call graph show?
- 2 How do call graphs help programmers avoid problems?

## Reading

2 Read the blog. Then, choose the correct answers.

- 1 What is the purpose of the blog?
  - A to compare different software planning methods
  - B to explain the value of call graphs
  - C to give solutions for common wicked problems
  - D to describe the inter modular attributes of a system
- 2 Which of the following is NOT a reason to use call graphs?
  - A to avoid wicked problems
  - B to examine intra-modular attributes
  - C to eliminate problems at an early stage
  - D to create stopping rules
- 3 Which is a part of abstraction?
  - A modularity
  - B minute details
  - C increased complexity
  - D stopping points

## Vocabulary

3 Fill in the blanks with the correct words and phrases from the word bank.

### WORD BANK

call graph    information hiding    simplify  
stopping rule    wicked problem

- 1 A(n) \_\_\_\_\_ can have multiple causes and may be difficult to solve.
- 2 A(n) \_\_\_\_\_ shows the basic structure of how a system will work.
- 3 Modules conceal information from each other in a process called \_\_\_\_\_.
- 4 A problem without a(n) \_\_\_\_\_ may be difficult or impossible to solve.
- 5 Use of systems and procedures can \_\_\_\_\_ complicated processes.

**4** Read the sentence pairs. Choose which word or phrase best fits each blank.

**1 cohesion / coupling**

**A** \_\_\_\_\_ describes the strength of connections between modules.

**B** \_\_\_\_\_ is the connection between modules in a system.

**2 inter-modular attributes / intra-modular attributes**

**A** Characteristics of individual modules are \_\_\_\_\_.

**B** \_\_\_\_\_ describe the characteristics of an entire system.

**3 system structure / abstraction**

**A** A(n) \_\_\_\_\_ is the network of connections between modules.

**B** A(n) \_\_\_\_\_ ignores details.

**4 complexity / modularity**

**A** \_\_\_\_\_ is judged by the amount of time it would take to change something in a system.

**B** \_\_\_\_\_ indicates that a system is made up of smaller interconnected systems.

**5** Listen and read the blog again. How can programmers avoid wicked problems in software designs?

**Listening**

**5** Listen to a conversation between an engineer and an intern. Mark the following statements as true (T) or false (F).

- 1  The woman is having difficulty reading a call graph.
- 2  The woman suggests removing information from a design.
- 3  The man explains the importance of excluding details from a design.

**7** Listen again and complete the conversation.

**Intern:** Actually there is. I'm 1 \_\_\_\_\_ about a few things.

**Engineer:** Let's see if we can get that 2 \_\_\_\_\_. What is confusing you?

**Intern:** Why do we need to 3 \_\_\_\_\_ before adding the details?

**Engineer:** What do you mean?

**Intern:** Wouldn't it save time to 4 \_\_\_\_\_ as we create the design?

**Engineer:** No, you need to consider the purpose of an 5 \_\_\_\_\_.

**Intern:** An abstraction?

**Engineer:** Yes. Creating a design without details lets us find problems early. That way we can fix them 6 \_\_\_\_\_. \_\_\_\_\_ is too complex. Does that make sense?

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Is there anything ...*

*I don't see why ...*

*What you're not considering is ...*

**Student A:** You are an engineer. Talk to Student B about:

- the software design process
- reasons for a particular process
- the value of particular design tools or elements

**Student B:** You are an intern. Talk to Student A about reasons for using a particular software design process.

**Writing**

**9** Use the conversation from Task 8 to complete the intern's notes.

Design	Benefits
_____	provides a plan of what a final product will look like
abstraction	_____
_____	_____
_____	_____

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some different types of design methods?
- 2 How can programmers manage problems in software designs?

## CMSC 1500

### Introduction to Design Methods

This course is an introduction to basic software design methods. It will teach students about **design methods** in both **idealistic** and realistic settings. Students will learn the **philosophy** and practical applications of design.

Students will become familiar with both **top-down designs** and **bottom-up designs**. This involves the advantages and challenges associated with each type of design. Differentiating main user functions and base functions is a major component of this section.

In hands-on workshops, students will learn how effective designs are created using each method. The course also covers problem solving in software design. This section will focus on **functional decomposition**. Students will learn to **decompose** software down to its **subfunctions**. As a final project, students will create **primitive** software with **rational** design plans.

This course is a prerequisite for many advanced software courses. It prepares students for higher-level concepts like **stepwise refinement**.

## Reading

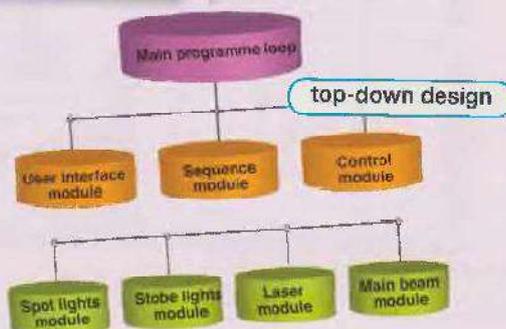
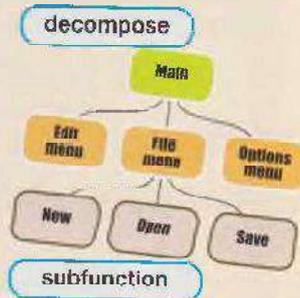
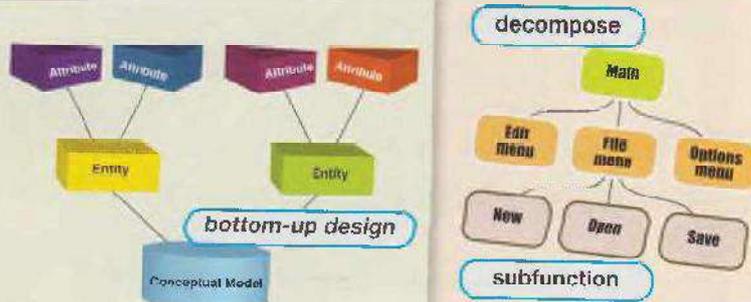
2 Read the course description. Then, choose the correct answers.

- 1 What is true of the design course?
  - A It requires a prerequisite.
  - B It focuses on the history of different design methods.
  - C It teaches advanced-level computing concepts.
  - D It covers the philosophy of software design.
- 2 Which of the following is NOT a topic covered in the course?
  - A top-down designs
  - B stepwise refinement
  - C decomposing software
  - D functional decomposition
- 3 What will students do for their final projects?
  - A design very basic software functions
  - B make a diagram of bottom-up designs
  - C write about the philosophy of design methods
  - D make a plan for problem solving

## Vocabulary

3 Choose the sentence that uses the underlined part correctly.

- 1 A Engineers use a design method when designing new software systems.  
B To decompose software is to make it more complicated.
- 2 A Functional decomposition can be used to reveal and eliminate problems in software.  
B Stepwise refinement is a method of creating a plan for new types of software.
- 3 A Engineers can use a process called stepwise refinement to solve problems in existing software.  
B In an ideal world, software designs will never follow a rational procedure.
- 4 A A subfunction is a higher-order function.  
B When an engineer decomposes a function, he or she is left with smaller subfunctions.
- 5 A It is important that engineers follow a rational procedure.  
B A philosophy is primarily concerned with the real-world application of ideas.



4 Read the sentence pairs. Choose which word or phrase best fits each blank.

- top-down design / bottom-up design
  - A A \_\_\_\_\_ begins with the main user functions.
  - B A \_\_\_\_\_ begins with the most primitive functions.
- primitive / idealistic
  - A A(n) \_\_\_\_\_ person ignores possible problems that may arise.
  - B If a software is \_\_\_\_\_, it is very simple.
- philosophy / subfunction
  - A A \_\_\_\_\_ is combined with others to make a function.
  - B A \_\_\_\_\_ provides a means of viewing the world.

5 Listen and read the course description again. What topics will be covered in the course?

## Listening

6 Listen to a conversation between an instructor and a student. Mark the following statements as true (T) or false (F).

- The woman correctly identifies the limitations of a bottom-up design.
- A top-down design is less likely to have problems.
- The man encourages the woman to do more reading.

7 Listen again and complete the conversation.

**Instructor:** Very good. Can you tell me about some of the benefits of 1 \_\_\_\_\_ ?

**Student:** Um, I'm not sure.

**Instructor:** What about in regards to 2 \_\_\_\_\_ in a design?

**Student:** Oh, well, a bottom-up design is less likely to have problems. But, if it does, it's more difficult 3 \_\_\_\_\_.

**Instructor:** Great. And 4 \_\_\_\_\_ designs?

**Student:** Well, top-down designs tend to have more problems. But 5 \_\_\_\_\_ eliminate them.

**Instructor:** And how would one do that?

**Student:** I'm not sure. Maybe 6 \_\_\_\_\_ the software?

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

Can you tell me about ... / What about ...  
It's easier to ...

**Student A:** You are an instructor. Talk to Student B about:

- different types of software designs
- the pros and cons of each design
- how to handle problems with different software designs

**Student B:** You are a student. Talk to Student A about the pros and cons of different types of software designs.

## Writing

6 Use the conversation from Task 8 to complete the list of pros and cons about top-down and bottom-up designs.

### Top-Down

#### Pros

- allows engineer to define user functions first
- \_\_\_\_\_

#### Cons

- \_\_\_\_\_
- \_\_\_\_\_

### Bottom-Up

#### Pros

- \_\_\_\_\_

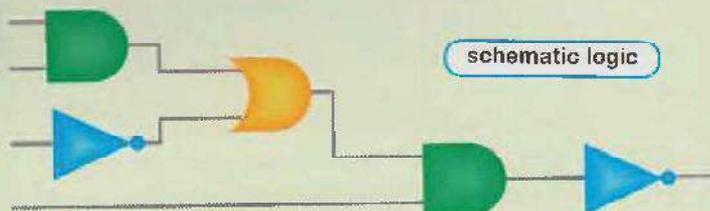
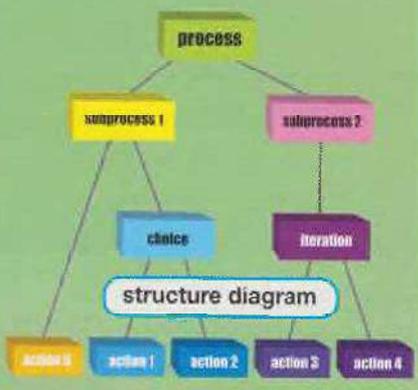
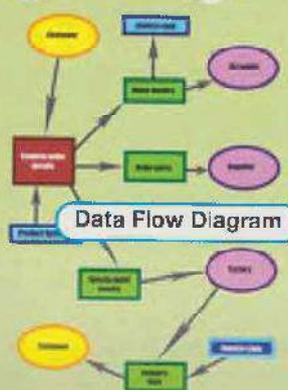
#### Cons

- does not allow for functional decomposition until the end of the process
- \_\_\_\_\_

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 How do software engineers visualize software structures?
- 2 What are the stages of Jackson System Development?



## Chapter 8 Design Methods

Modern approaches to software design focus heavily on a few standard practices. The first is the practice of creating **data flow designs**. These designs are represented graphically as Data Flow Diagrams (DFDs). By using them, software developers visualize the way their software will operate. This allows them to create the most efficient software possible.

A **structure chart** works in a similar way. These ensure that **Structured Designs** are optimized for efficiency. Structure charts and **structure diagrams** are expressed with **schematic logic**. These methods are part of Structured Analysis (SA). SA is the process of creating software that solves real-world problems.

Developers also use Jackson System Development (JSD) for efficient programming-in-the-large. In JSD, the development of software is split into three phases. These are the **modeling stage**, the **network stage**, and the **implementation stage**. JSD is based on an earlier model: Jackson Structured Programming (JSP). JSP is primarily used on a smaller scale.

## Reading

2 Read the textbook excerpt. Then, choose the correct answers.

- 1 What is the purpose of data flow designs?
  - A to solve real-world problems
  - B to generate efficient software
  - C to determine standard practices
  - D to divide development phases
- 2 Which of these is NOT part of structured analysis?
  - A use of structure charts
  - B creation of software that addresses real problems
  - C development of software in three distinct phases
  - D implementation of Structured Designs
- 3 What is true of structure charts?
  - A They are expressed with schematic logic.
  - B They begin with a modeling stage.
  - C They appear in data flow designs.
  - D They are part of Jackson Structured Programming.

## Vocabulary

3 Match the words and phrases (1-5) with the definitions (A-E).

- |                           |         |
|---------------------------|---------|
| 1 __ implementation stage | 4 __ SA |
| 2 __ modeling stage       | 5 __ SD |
| 3 __ network stage        |         |

- A the development of modules and module hierarchies
- B the identification of the problems that a software needs to solve
- C the point when a design becomes a working piece of software
- D a method for ensuring that software fulfills real-life requirements
- E the expression of a design as a set of communicating processes

**4** Write a word or phrase that is similar in meaning to the underlined part.

- 1 A chart that shows the functions of a system is used by engineers to guide software designs.  
s t \_ \_ t \_ \_ \_ h \_ \_ t
- 2 The modeling stage is the first step in the method of software development with three distinct phases. \_ S \_
- 3 To understand structure diagrams, engineers must first understand the code used in structure diagrams.  
\_ c \_ \_ \_ \_ i c \_ l \_ g \_ \_
- 4 A diagram representing compound components in a structure shows all of the functions and modules that a completed structure will have.  
\_ r \_ t \_ r \_ \_ \_ a \_ r \_ \_
- 5 The practice of mapping the movement of data through a system is represented graphically in DFD.  
\_ \_ t \_ \_ f \_ o \_ \_ \_ s \_ g v
- 6 A method of system development is based on data flow and program structure. J \_ \_
- 7 A graphical representation of data flow is used to optimize the functions of a system. \_ \_ D

**5** Listen and read the textbook excerpt again. What are DFDs used for?

**Listening**

**6** Listen to a conversation between an intern and an engineer. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The woman finds structured stages difficult to work with.
- 2 \_\_\_ The woman is using JSD for her project.
- 3 \_\_\_ The project is currently in the implementation stage.

**7** Listen again and complete the conversation.

**Intern:** I'm interested in different software design methods. Are you **1** \_\_\_\_\_?

**Engineer:** No, but you're close. **2** \_\_\_\_\_, I'm using the JSD system.

**Intern:** Oh, that's interesting. What made you decide to plan it that way?

**Engineer:** Well, I prefer the structured stages of the JSD system. It's **3** \_\_\_\_\_ how far along you are with the project.

**Intern:** That makes sense. **4** \_\_\_\_\_ of the project are you in now?

**Engineer:** Right now, I'm working on **5** \_\_\_\_\_. But I'm having some problems.

**Intern:** What kind of problems?

**Engineer:** **6** \_\_\_\_\_ isn't working like I hoped. I might have to go back to my DFDs before I continue.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Do you mind if ... / What made you decide ...  
I prefer ...*

**Student A:** You are an intern. Talk to Student B about:

- what design method he or she is using for a project
- how the project is going
- how you can help with the project

**Student B:** You are an engineer. Talk to Student A about the design method of a project you are working on.

**Writing**

**9** Use the conversation from Task 8 to complete the intern's notes on JSD.

1. Begin with the modeling stage. This involves \_\_\_\_\_
2. Continue to the networking stage. In this stage, \_\_\_\_\_
3. Finally, execute the implementation stage. This is when \_\_\_\_\_

# 10 Design Methods 3

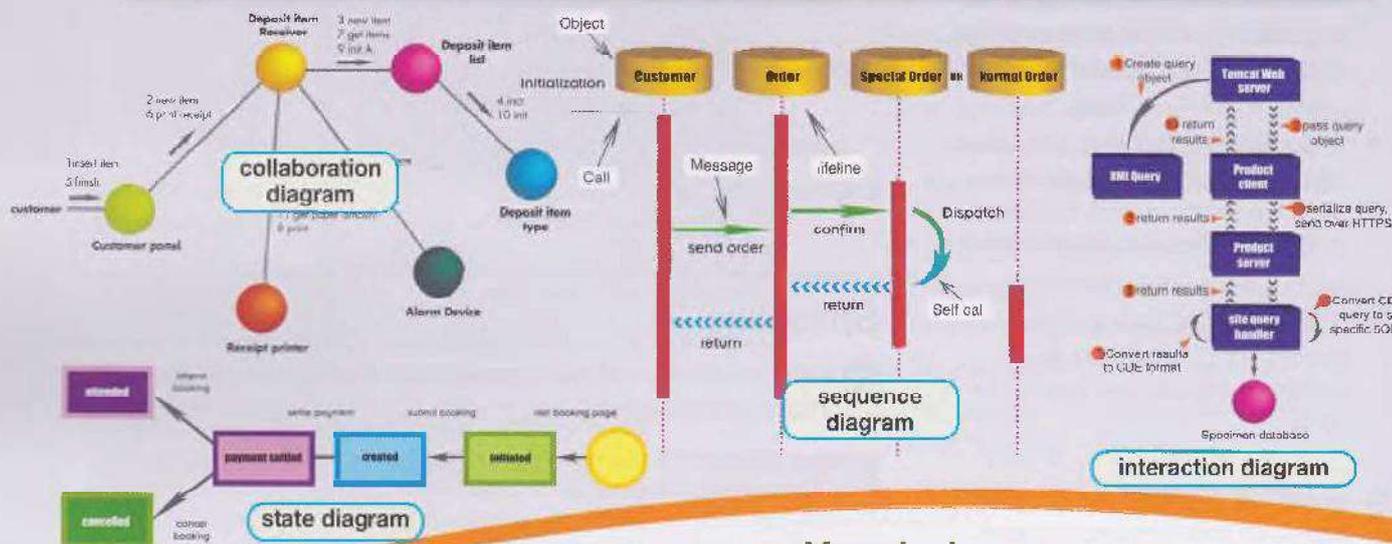
## Object-Oriented Software Design

**Object-oriented** design is a recent trend in software development. It begins when an engineer maps out different parts of the design. This includes the **attributes, properties, and relationships** of objects. These characteristics of objects make up an object's **state**. Different objects with similar states are in the same **class**.

There are a few popular models of object-oriented design. One is the **Booch method**. This method uses its own descriptive modeling language. Another is the **Fusion method**, which structures the design process into three distinct phases. Both of these methods are considered Object Modeling Techniques, or **OMTs**.

Different kinds of diagrams illustrate various aspects of a system. Developers can use **class diagrams, state diagrams, or collaboration diagrams** to depict the objects in systems. They can use **interaction diagrams or sequence diagrams** to depict the functions of a system.

Are you an expert in this subject? You can help by [expanding this article](#).



## Vocabulary

3 Match the words and phrases (1-8) with the definitions (A-H).

- |                             |                     |
|-----------------------------|---------------------|
| 1 ___ class diagram         | 5 ___ state diagram |
| 2 ___ collaboration diagram | 6 ___ Booch method  |
| 3 ___ interaction diagram   | 7 ___ Fusion method |
| 4 ___ sequence diagram      | 8 ___ class         |

- A a software modeling language and process  
 B an image that depicts objects in relation to a particular interaction  
 C an image that measures the time ordering of events within an interaction  
 D an image that depicts sequences of messages  
 E an image that models relationships between nodes and depicts decomposition of a system  
 F an image that depicts the dynamic behavior of single objects  
 G a system that structures the development process into three phases  
 H a group of objects that share similar attributes

### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What is the purpose of object-oriented design?
- 2 What are some different types of diagrams used in object-oriented design?

### Reading

1 Read the online encyclopedia article. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ An object's state is composed of its attitudes, properties, and relationships.
- 2 \_\_\_ The Booch method structures the design process into three phases.
- 3 \_\_\_ Developers can use interaction diagrams to depict objects in a system.

**4** Fill in the blanks with the correct words: *attribute, object-oriented, OMTs, property, relationship, state.*

- 1 A(n) \_\_\_\_\_ can be used to identify an object.
- 2 An object's \_\_\_\_\_ refers to all of its fundamental qualities.
- 3 The interaction between two entities is their \_\_\_\_\_.
- 4 When software developers design their systems according to the objects contained, they are using \_\_\_\_\_.
- 5 An unchanging quality of one entity is a(n) \_\_\_\_\_.
- 6 The Booch method and the Fusion method are examples of \_\_\_\_\_ design.

**5** Listen and read the article again. What are some popular models of object-oriented design?

## Listening

**6** Listen to a conversation between an engineer and an intern. Choose the correct answers.

- 1 What is the purpose of the conversation?
  - A to establish what diagrams need to be made
  - B to discuss the general goals of a software design project
  - C a debate on the merits of different types of diagrams
  - D to plan an OMT presentation
- 2 What will the man work on first?
  - A mapping object interactions
  - B compiling the interaction data
  - C creating the collaboration diagram
  - D correcting the interaction diagram

**7** Listen again and complete the conversation.

**Engineer:** I need you to 1 \_\_\_\_\_. They're for our new accounting software.

**Intern:** 2 \_\_\_\_\_. I can definitely do that. What kind of diagrams do you need?

**Engineer:** We'll need a collaboration diagram, a class diagram, and 3 \_\_\_\_\_.

**Intern:** Okay. Did the engineers finish 4 \_\_\_\_\_ yet?

**Engineer:** No, so you'll have to work on the interaction diagram last. In the meantime, you can get started on the collaboration diagram.

**Intern:** Okay. 5 \_\_\_\_\_ the interaction data when it's available?

**Engineer:** Of course. Now listen, these diagrams need to be done by the 6 \_\_\_\_\_. They're for a client.

## Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I need you to ...*  
*In the meantime ...*  
*So they need to ... right?*

**Student A:** You are an intern. Talk to Student B about:

- diagrams he or she needs you to create
- the information needed to create the diagrams
- when the diagrams should be completed

**Student B:** You are an engineer. Talk to Student A about diagrams you need him or her to create.

## Writing

**9** Use the conversation from Task 8 to complete the intern's notes.

Type of Diagram	Description
1 class diagram	depicts the decomposition of a system
2 _____	_____
3 _____	_____
4 _____	_____

HOME

ABOUT US

SERVICES

CONTACT

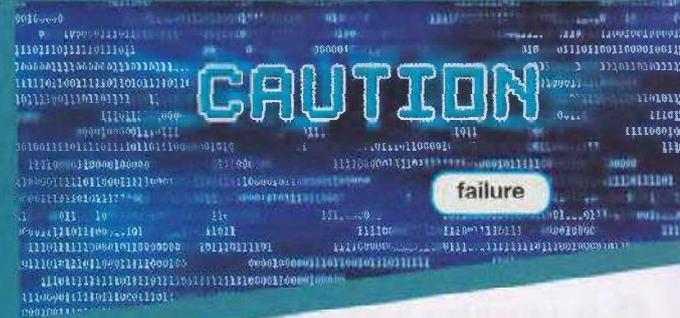
## ProGoTools Software Testing &gt; Services &gt; Overview



We offer third-party **fault detection** and **fault prevention**. Our services are available for software at all phases of development. We use **oracles** to generate projections of your software. Our team of engineers **compares** those projections to test results. Our testers make sure that your code is free of **errors**. This means that we catch problems before they become **faults** or **failures**.

Our engineers develop specific **test criteria** for each project. We work closely with clients to ensure that we understand the software's requirements. We test carefully to make sure we **satisfy** all **expected** qualities.

Our engineers are experts in all models of fault prevention. We also offer **prevention model** and **evaluation model** testing. We recommend these services for software in earlier stages of development. We offer **demonstration model** and **destruction model** testing. These are recommended for software in later stages of development.



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What causes software failures to occur?
- 2 What are some different models of fault detection?

## Reading

2 Read the webpage. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ Oracles are compared to the test criteria.
- 2 \_\_\_ The company changes test criteria for each project.
- 3 \_\_\_ Evaluation model testing is recommended for software in the later stages of development.

## Vocabulary

3 Fill in the blanks with the correct words and phrases from the word bank.

## word BANK

error    evaluation model    expected  
oracle    prevention model    test criteria

- 1 Faults that are predicted are \_\_\_\_\_.
- 2 If an engineer makes a mistake in a software's code, this is a(n) \_\_\_\_\_.
- 3 The \_\_\_\_\_ attempts to stop faults before they occur.
- 4 The \_\_\_\_\_ detects design and implementation faults.
- 5 Software must meet all \_\_\_\_\_ before it passes a test.
- 6 Engineers can use a(n) \_\_\_\_\_ as a comparison tool.

4 Read the sentence pairs. Choose which word or phrase best fits each blank.

1 failure / fault

- A A \_\_\_\_\_ is caused by a human error in coding or input.
- B A \_\_\_\_\_ is an observable occurrence.

2 fault detection / fault prevention

- A \_\_\_\_\_ stops problems from occurring.
- B \_\_\_\_\_ identifies problems.

3 satisfy / compare

- A Engineers use programs to \_\_\_\_\_ test results with expectations.
- B Software must \_\_\_\_\_ certain requirements set by engineers.

4 demonstration model / destruction model

- A The \_\_\_\_\_ ensures that software completes required tasks.
- B The \_\_\_\_\_ detects implementation faults.

5 Listen and read the webpage again. How does the company detect and prevent faults?

## Listening

6 Listen to a conversation between a software engineer and a project manager. Choose the correct answers.

- 1 Which type of test results does the man want to see?
  - A prevention model
  - B demonstration model
  - C destruction model
  - D evaluation model
- 2 What is the woman likely to do next?
  - A perform the same test again to verify results
  - B consult another team of engineers
  - C reevaluate the test criteria
  - D compare the test results to an oracle

7 Listen again and complete the conversation.

Manager: I need an update 1 \_\_\_\_\_ for the latest project.

Engineer: I was just 2 \_\_\_\_\_ the latest test results.

Manager: What do they say?

Engineer: It looks like the software did pretty well. Only 3 \_\_\_\_\_ were detected.

Manager: What testing model did you use?

Engineer: For this test we used 4 \_\_\_\_\_.

Manager: Good. 5 \_\_\_\_\_ to determine where the failures are coming from. Then have an engineer to fix them.

Engineer: Okay. We will probably 6 \_\_\_\_\_ tests while we do that.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I need an update on ...*  
*It looks like ... / I'd like to see ...*

**Student A:** You are a project manager. Talk to Student B about:

- the test results on a software project
- how faults will be fixed
- what testing models to use

**Student B:** You are an engineer. Talk to Student A about the test results on a software project.

## Writing

9 Use the conversation from Task 8 to complete the testing summary.

### Testing Summary Report

Project: TX907

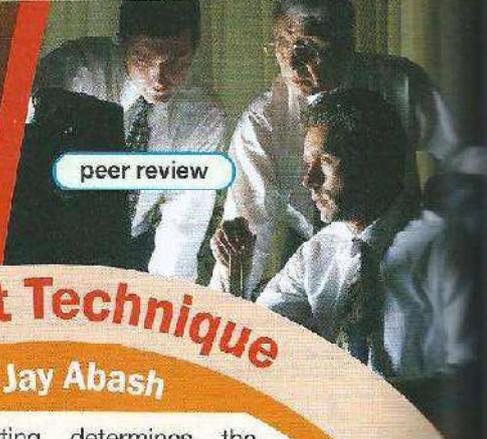
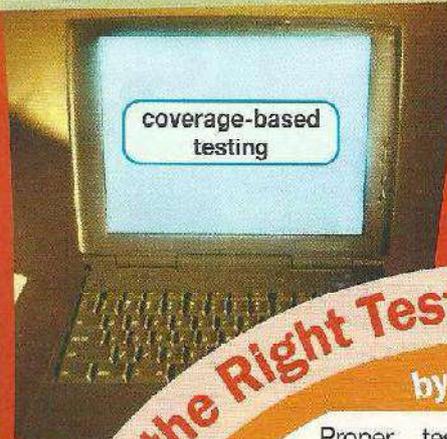
Tests completed: \_\_\_\_\_

Results: \_\_\_\_\_

Recommendation: \_\_\_\_\_

Tests to Complete: \_\_\_\_\_

# 12 Software Testing Techniques



## Choosing the Right Test Technique

by Jay Abash

Proper testing determines the strength of a software program. If faults are not detected, they cannot be fixed. Testing techniques can be divided generally into two categories: **static analysis** and **dynamic analysis**. The type of analysis used will depend on what stage of development the software is in.

The function of software often dictates what type of testing is best. **Error-based testing** and **scenario-based evaluation** work well for interactive software. **Black-box testing** is used for software that is more standard. For non-standard types of software, **white-box testing** and **fault-based testing** are appropriate.

The size of the software is also important. Large software projects will do well in **Fagan inspection** but not **stepwise abstraction**. Smaller projects may only require **peer review** from one other engineer.

Software developed by companies will need **proof of correctness**. The standard for personal-use software does not need to be as strict. **Coverage-based testing** should always be a consideration, regardless of software size.



### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some different types of software testing?
- 2 How do engineers work together on software testing?

### Reading

2 Read the journal article. Then, choose the correct answers.

- 1 What is the purpose of the article?  
A to discuss which testing methods are best for what projects  
B to compare the costs of different types of software testing  
C to explain how software testing has changed over time  
D to encourage engineers to try a particular type of testing
- 2 Which type of testing is best for software that is interactive?  
A white-box testing  
B peer review  
C error-based testing  
D stepwise abstraction
- 3 According to the article, what kind of testing is inappropriate for large projects?  
A stepwise abstraction  
B Fagan inspection  
C proof of correctness  
D fault-based testing

### Vocabulary

3 Match the phrases (1-8) with the definitions (A-H).

- |                              |                                 |
|------------------------------|---------------------------------|
| 1 ___ black-box testing      | 5 ___ fault-based testing       |
| 2 ___ coverage-based testing | 6 ___ proof of correctness      |
| 3 ___ dynamic analysis       | 7 ___ scenario-based evaluation |
| 4 ___ error-based testing    | 8 ___ white-box testing         |
- 
- A testing conducted while a program is running
  - B a process that formally states a program's specifications
  - C testing with specifically developed criteria
  - D testing that focuses primarily on identifying problems
  - E testing that is measured by the amount of code it tests
  - F testing done according to pre-set standards
  - G testing that assesses common situations
  - H testing that looks specifically for common human mistakes

**4 Write a phrase that is similar in meaning to the underlined part.**

- 1 A(n) testing process involving a team of engineers is a good way to test software projects that are very large.  
\_ g \_ n \_ \_ s \_ \_ c \_ \_ \_ \_
- 2 Testing done on code that is not being executed can be used to test software in the early stages of development.  
\_ a \_ c \_ a \_ \_ y \_ \_ \_
- 3 Testing done starting with the most primitive parts of a code is a technique that analyzes all parts of a piece of software.  
\_ t \_ \_ \_ s \_ \_ b \_ r \_ \_ t \_ \_ \_
- 4 Software engineers should feel comfortable asking each other for manual review of software done by an engineer who did not write the code.  
\_ e \_ \_ \_ e v \_ \_ w

**5 Listen and read the journal article again. How can engineers decide what type of testing to use?**

**Listening**

**6 Listen to a conversation between a company owner and a software engineer. Mark the following statements as true (T) or false (F).**

- 1  The woman found faults during white-box testing.
- 2  The man was disappointed with the results of a Fagan inspection.
- 3  The man recommends coverage-based testing to the woman.

**7 Listen again and complete the conversation.**

**Owner:** How is the testing going on the accounting software?  
**Engineer:** Actually, I have some bad news about that. 1 \_\_\_\_\_ we did find a lot of faults.  
**Owner:** That's not good. What are you going to do?  
**Engineer:** We're going to do some black-box testing before we start 2 \_\_\_\_\_.  
**Owner:** Good. Are you using both 3 \_\_\_\_\_ analysis?  
**Engineer:** For now we're just doing static analysis.  
**Owner:** Okay. Try to get this taken care of quickly. I want this software 4 \_\_\_\_\_ by next month.  
**Engineer:** I understand. I've got the 5 \_\_\_\_\_ on it.  
**Owner:** Maybe you should get another team leader to 6 \_\_\_\_\_.

**Speaking**

**8 With a partner, act out the roles below based on Task 7. Then, switch roles.**

**USE LANGUAGE SUCH AS:**

- I've got some bad news about ...*
- We're going to do ...*
- Maybe you should ...*

**Student A:** You are a company owner. Talk to Student B about:

- the results from recent software tests
- what the engineer is doing to fix problems
- what types of testing you recommend

**Student B:** You are an engineer. Talk to Student A about the results from recent software tests.

**Writing**

**9 Use the conversation from Task 8 to complete the chart about software testing methods.**

Test Method	Advantages	Disadvantages
black-box testing		may not be best for all types of software

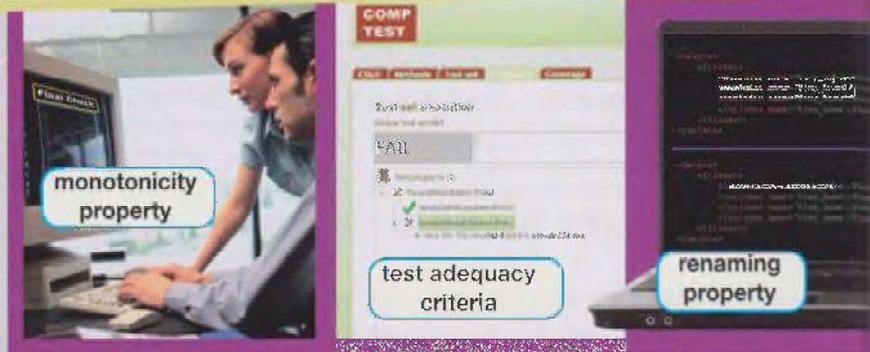
# 13 Test Adequacy Criteria

## LaditSoft – Employee Handbook

### Software Test Set Criteria

All software produced at LaditSoft must be adequately tested. Therefore, we ask that all engineers refer to these tips on creating **test adequacy criteria** when evaluating their programs.

- Remember the **complexity property**. Testing is easier at the earlier stages when programs are simpler.
- An **inadequate empty set** is never acceptable. Your test sets should be large enough to satisfy the **statement coverage property**.
- Know which similarities the **antiextensionality property** applies to. It's easy to confuse



- those with similarities under the **renaming property**. Follow the **general multiple change property** and create test sets for each program.
- If you get frustrated, remember the **applicability property** and the **non-exhausting applicability property**. The test you need can be created.
- Follow the **antidecomposition property** and the **anticomposition property**. Test program components together and separately.
- Always follow the **monotonicity property**. It never hurts to check again.

### Get ready!

- 1 Before you read the passage, talk about these questions.

- 1 What is the purpose of test adequacy criteria?
- 2 What are some different properties of test adequacy?

### Reading

- 2 Read the handbook. Then, choose the correct answers.

- 1 According to the handbook, which property is likely to be confused with the antiextensionality property?  
A the applicability property  
B the renaming property  
C the statement coverage property  
D the monotonicity property
- 2 What will employees do when they follow the monotonicity property?  
A eliminate mistakes made along the way  
B create large test sets  
C check their software multiple times  
D create simulated environments
- 3 Why are inadequate empty sets unacceptable?  
A Test sets must satisfy the statement coverage property.  
B Programs need to be tested in isolation.  
C It is important to check programs often.  
D They do not fit into the antiextensionality property.

### Vocabulary

- 3 Match the phrases (1-8) with the definitions (A-H).

- 1 \_\_\_ antidecomposition property
  - 2 \_\_\_ applicability property
  - 3 \_\_\_ general multiple change property
  - 4 \_\_\_ inadequate empty set property
  - 5 \_\_\_ non-exhausting applicability property
  - 6 \_\_\_ renaming property
  - 7 \_\_\_ statement coverage property
  - 8 \_\_\_ monotonicity property
- A a property that states that programs with the same structure and dataflow characteristics should still be tested on different criteria
- B a property that states that an adequate test set exists for every program
- C a property that states that a criterion should not require exhaustive testing in all circumstances
- D a property that states that two programs that differ only in unimportant ways can be tested with the same test sets
- E a property that states that an empty set is not an adequate test set for any program
- F a property that states that components that have been tested in isolation should still be tested as a whole
- G a property that states that every possible action should be executed by its test sets
- H a property that states that software can always be tested further

4 Read the sentence pairs. Choose which phrase best fits each blank.

1 complexity property / test adequacy criteria

- A Many different rules are used as \_\_\_\_\_ .  
 B The \_\_\_\_\_ reminds engineers that programs with more components require more testing.

2 anticomposition property / antiextensionality property

- A The \_\_\_\_\_ refers to programs that have already been tested in isolation.  
 B The \_\_\_\_\_ refers to the relationship between the testing of two programs that have similar names.

5 Listen and read the handbook again. What are some test properties that employees should use?

## Listening

6 Listen to a conversation between a software engineer and an intern. Mark the following statements as true (T) or false (F).

- The woman followed the statement coverage property.
- The man reminds the woman to follow the renaming property.
- The woman wrote more tests than she needed.

7 Listen again and complete the conversation.

Engineer: Your test sets were 1 \_\_\_\_\_ . But I have some feedback for you.

Intern: Okay. I'd love to hear it.

Engineer: You covered 2 \_\_\_\_\_ really well, so that was great.

Intern: I'm glad you noticed. I worked really hard on that.

Engineer: But there 3 \_\_\_\_\_ too.

Intern: Like what?

Engineer: It seems like you relied too much on the 4 \_\_\_\_\_ .

Intern: Well, I used the renaming property.

Engineer: I see. I'd advise you to be careful about that. Remember the general multiple change property.

Intern: So I should try to 5 \_\_\_\_\_ ?

Engineer: Yes, 6 \_\_\_\_\_ . Even if programs are similar, they often need different tests.

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*You covered ...*

*I'd advise you to ...*

*So I should ...*

**Student A:** You are an engineer. Talk to Student B about:

- a test set that he or she used
- the strengths and weaknesses of the test set
- what advice you have about writing test sets

**Student B:** You are an intern. Talk to Student A about a test set that you wrote.

## Writing

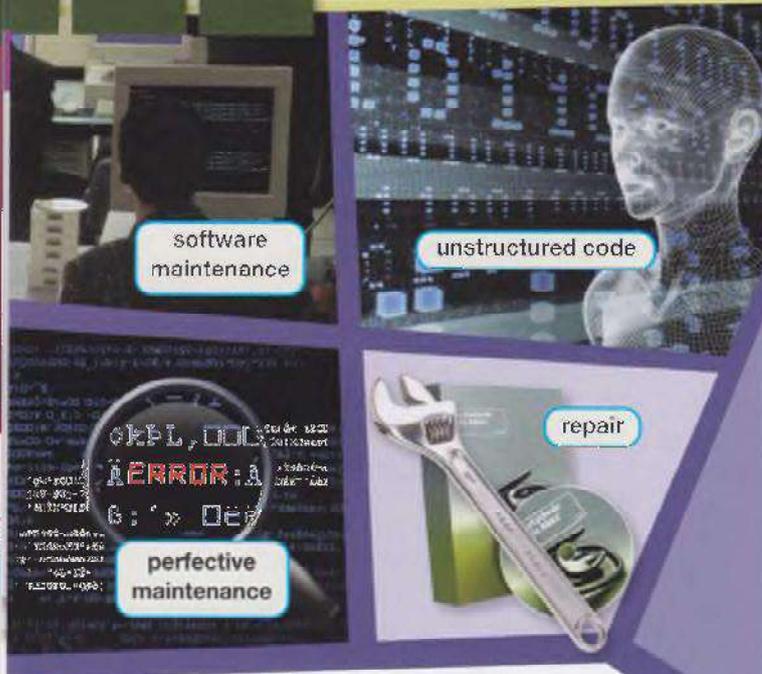
9 Use the conversation from Task 8 to complete the test set criteria guidelines.

All employees should observe the following guidelines in regards to test sets.

Antidecomposition property- \_\_\_\_\_

\_\_\_\_\_ - Employees should be cautious with this property. Only a few systems are similar enough to fit the requirements for similar test sets.

# 14 Software Maintenance 1



To: All Employees  
From: c.bellman@shorsoft.net  
Subject: Updated Policies for **Software Maintenance**

Good Morning Employees,  
I understand that most departments are practicing **corrective maintenance**. This practice should continue, but I think it is **insufficient**. I believe we need to focus on **adaptive maintenance** as well. According to the **law of continuing change**, this will allow us to grow more rapidly.

Developing new software is important. But to stay competitive, we must **enhance** our existing software, too. New **releases** are the best way to keep customers interested in our products. This will require engineers to practice **perfective maintenance**. Always **repair** problems as soon as they are identified.

However, engineers must also remember the **law of increasing complexity**. If software becomes too complex, it becomes difficult to maintain. Engineers should know when to update and when to write a new program.

**Unstructured code** will no longer be tolerated. It causes confusion and makes further updates more difficult. Remember, **preventive maintenance** is the strongest software maintenance practice.

-Clinton Bellman  
CEO, ShorSoft Corporation

## Get ready!

- 1 Before you read the passage, talk about these questions.
  - 1 What are the benefits of software maintenance?
  - 2 What are some different types of software maintenance?

## Reading

- 2 Read the memo. Then, choose the correct answers.
  - 1 What is the purpose of the memo?
    - A to warn employees about maintenance risks
    - B to reprimand employees who are not following maintenance procedures
    - C to show techniques for software maintenance
    - D to inform employees about new policies
  - 2 Which of the following is NOT something that the CEO wants engineers to do?
    - A Avoid unstructured code.
    - B Focus more on corrective maintenance.
    - C Get existing software ready for new releases.
    - D Increase preventive maintenance measures.
  - 3 According to the email, how can engineers enhance existing software?
    - A creating less unstructured code
    - B using templates from other software programs
    - C repairing problems in the software
    - D practicing corrective maintenance

## Vocabulary

- 3 Match the phrases (1-8) with the definitions (A-H).
  - 1 \_\_\_ adaptive maintenance
  - 2 \_\_\_ corrective maintenance
  - 3 \_\_\_ law of continuing change
  - 4 \_\_\_ law of increasing complexity
  - 5 \_\_\_ perfective maintenance
  - 6 \_\_\_ preventive maintenance
  - 7 \_\_\_ software maintenance
  - 8 \_\_\_ unstructured code
  - A the practice of accommodating new user requirements
  - B the practice of repairing software faults
  - C the format of a system with no clear order
  - D the practice of making systems easier to maintain
  - E the process of fixing faults and making improvements in software
  - F states that a system should undergo modification until it is no longer cost-effective
  - G the practice of updating software according to changes in environment
  - H states that a structure becomes more complex with every change

**4** Write a word that is similar in meaning to the underlined part.

- The attributes of an older system may be unsuitable or not strong enough to work on updated operating systems.  
\_ \_ s \_ \_ f \_ \_ \_ \_ \_ l
- A software engineer should fix any problems he or she finds in a code.    \_ \_ p \_ \_ r
- Each new updated version of existing software should come with some modifications.  
\_ \_ / \_ a \_ \_
- Consumers of software are happy when engineers improve existing components in new versions of software products.  
\_ n \_ \_ n c e

**5** Listen and read the memo again. Why does the CEO want engineers to perform adaptive maintenance?

**Listening**

**6** Listen to a conversation between two engineers. Mark the following statements as true (T) or false (F).

- The man wants to start with corrective maintenance.
- The engineers are adding new functionality to old software.
- The woman discovered unstructured code in the software.

**7** Listen again and complete the conversation.

**Engineer 1:** We have 1 \_\_\_\_\_ to do on that accounting software.

**Engineer 2:** Yeah, I'm really 2 \_\_\_\_\_ all of that work.

**Engineer 1:** Neither am I. But I think if we make a plan, we can save ourselves a lot of time.

**Engineer 2:** That's a good idea. 3 \_\_\_\_\_ fix all of the problems with the software.

**Engineer 1:** Okay. We can start with a round 4 \_\_\_\_\_, then.

**Engineer 2:** Exactly. Next, we need to 5 \_\_\_\_\_ all of the code.

**Engineer 1:** Why do we need to do that?

**Engineer 2:** So that it can 6 \_\_\_\_\_

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

We're going to be ...  
First, we need to ...  
Next ...

**Student A:** You are an engineer. Talk to Student B about:

- software maintenance that is needed
- reasons for particular types of maintenance

**Student B:** You are an engineer. Talk to Student A about software maintenance.

**Writing**

**9** Use the conversation from Task 8 to complete the memo from a project manager to an engineer.

Hi Karen,

I'm pleased with the new programs your team is developing. However, I want you to shift your focus to software maintenance for a few weeks.

On the GrayWhite 9.7 software, we need \_\_\_\_\_.  
This is because \_\_\_\_\_.

On the SP008 software, we need \_\_\_\_\_.  
This is because \_\_\_\_\_.

On the FANFARE II software, we need \_\_\_\_\_.  
This is because \_\_\_\_\_.

-Arnold

## Reverse Engineering

by Chad Greenman

Most of the programs we use every day are **legacy systems**. Software companies often **modernize** these programs for contemporary use. This process is called **reverse engineering**. Some are changed from computer applications to **web-based** applications. In other cases, the only changes are in the appearance of the user interface. This is known as **revamping**.

Generally, reverse engineering involves the **redocumentation** and **design recovery** of software. Programs typically maintain high **functional equivalence** with earlier versions. Engineers use **reengineering**, or **renovation**, to create new programs quickly. This practice is efficient because the basic program already exists. Engineers just need to update it with newer technologies. In other cases, engineers practice **restructuring** of poorly written programs. This creates functionally equivalent programs that can be edited and updated easily.

All of these practices save time and resources. Starting with an existing program is more efficient than writing a whole new program.

### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some different ways to update a legacy system?
- 2 What is the purpose of redocumentation and design recovery?

### Reading

2 Read the journal article. Then, mark the following statements as true (T) or false (F).

- 1  Revamping creates a functionally similar program.
- 2  Reverse engineering is rarely applied to legacy systems.
- 3  Renovation primarily fixes problems with poorly-written programs.

### Vocabulary

2 Fill in the blanks with the correct words and phrases from the word bank.

#### Word BANK

design recovery    functional equivalence  
redocumentation    reverse engineering  
legacy system    web-based    modernize

- 1 \_\_\_\_\_ creates functionally equivalent versions of disorganized programs.
- 2 A \_\_\_\_\_ is an old system that has been updated multiple times.
- 3 \_\_\_\_\_ creates programs that exist at the same level of abstraction.
- 4 \_\_\_\_\_ means that two programs serve the same purpose in the same way.
- 5 Engineers may need to edit a program so that it is \_\_\_\_\_ instead of disc-based.
- 6 \_\_\_\_\_ creates new versions of software at higher levels of abstraction.
- 7 The engineers decided to \_\_\_\_\_ old software instead of creating entirely new software.



4 Read the sentence pairs. Choose which word best fits each blank.

1 revamping / reengineering

A \_\_\_\_\_ creates a new user interface for software.

B \_\_\_\_\_ makes substantial changes to a software system.

2 restructuring / renovation

A In \_\_\_\_\_, engineers make functional changes to a software system.

B \_\_\_\_\_ transforms systems from one representation to another.

5 Listen and read the journal article again. What is the benefit of restructuring?

## Listening

Listen to a conversation between two software engineers. Choose the correct answers.

1 Why do the engineers need to perform redocumentation?

- A in order to change the user interface
- B to remove errors in the program
- C because the software is unstructured
- D to prove that the software has no faults

2 What will the engineers do first?

- A revamp of the user interface
- B redocument the code
- C modernize the software functions
- D assess the legacy system

Listen again and complete the conversation.

Engineer 1: The database software needs some 1 \_\_\_\_\_ . What are we doing about that?

Engineer 2: We'll definitely need to 2 \_\_\_\_\_ .

Engineer 1: Really? Won't that take a long time?

Engineer 2: It will, but the 3 \_\_\_\_\_ to edit right now.

Engineer 1: Okay. Do you think we'll 4 \_\_\_\_\_ the software after redocumentation?

Engineer 2: Definitely. It's going to take a lot of 5 \_\_\_\_\_ this software.

Engineer 1: Should we 6 \_\_\_\_\_ also?

## Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

The software is too ...

Do you think ...

Should we ...

**Student A:** You are an engineer. Talk to Student B about:

- the software maintenance that is needed on a project
- how long the maintenance will take
- why the maintenance is necessary

**Student B:** You are an engineer. Talk to Student A about software maintenance on a project.

## Writing

9 Use the conversation from Task 8 to complete the note from a project manager to an engineer.

### Software Maintenance Progress Report

Software ID: X920

Maintenance needed: \_\_\_\_\_

We are performing this maintenance because \_\_\_\_\_

Current Progress: \_\_\_\_\_

# Glossary

- ability** [N-COUNT-U1] An **ability** is the skill to do something.
- abstract data type** [N-COUNT-U6] An **abstract data type** is a software structure that matches the structure of the original data, and whose system components are designed to maintain themselves.
- abstraction** [N-COUNT-U7] An **abstraction** is a general system plan that ignores details that are irrelevant at a broad level.
- adaptive maintenance** [N-UNCOUNT-U14] **Adaptive maintenance** is the practice of updating software according to changes in environment, such as upgrades to hardware or operating systems, without changing the functionality of the software.
- address** [V-T-U2] To **address** something is to give it appropriate action or attention.
- analysis** [N-COUNT-U2] An **analysis** is an examination and report on the structure or functionality of something.
- anticomposition property** [N-COUNT-U13] The **anticomposition property** is a property of test adequacy that states that components that were tested alone should still be tested when assembled as a whole.
- antidecomposition property** [N-COUNT-U13] The **antidecomposition property** is a property of test adequacy that states that components that were tested as part of a whole should still be tested alone.
- antiextensionality property** [N-COUNT-U13] The **antiextensionality property** is a property of test adequacy that states that programs need different types of testing even if they have similar names or functions.
- applicability property** [N-COUNT-U13] The **applicability property** is a property of test adequacy that states that an adequate test set exists for every program.
- application** [N-COUNT-U2] An **application** is the action of putting something into operation or practice.
- application framework** [N-COUNT-U5] An **application framework** is a semi-finished system that describes the relations between a family of similar systems.
- approach** [N-COUNT-U2] An **approach** is a way of dealing with or looking at something.
- attribute** [N-COUNT-U10] An **attribute** is a fundamental quality of something.
- black-box testing** [N-UNCOUNT-U12] **Black-box testing** is a software testing technique in which testing criteria are developed specifically for a particular piece of software.
- Booch method** [N-COUNT-U10] The **Booch method** is a software modeling language and process that is used in object oriented software development.
- bottom-up design** [N-COUNT-U8] A **bottom-up design** is a method of designing software in which the engineer begins with the software's most basic functions and proceeds to more complicated functions, until the higher-order functions of the software have been created.
- call graph** [N-COUNT-U7] A **call graph** is a graph depicting the outcome of a software design process.
- class** [N-COUNT-U10] A **class** is a group of objects that have the same set of attributes.
- class diagram** [N-COUNT-U10] A **class diagram** is a graph that models the relationships between nodes and depicts the decomposition of a system.
- closed system** [N-COUNT-U3] A **closed system** is a system that does not gain or lose mass.
- cohesion** [N-UNCOUNT-U7] **Cohesion** is the connection between the modules of a system.
- collaboration diagram** [N-COUNT-U10] A **collaboration diagram** is a graph that depicts objects in relation to a particular interaction.
- commitment** [N-UNCOUNT-U1] **Commitment** is an attitude of strong support of or loyalty to something.
- compare** [V-T-U11] To **compare** two things is to note the similarities or differences between them.
- complexity** [N-UNCOUNT-U7] **Complexity** is the measure of the amount of time and other resources required to construct or change a system.

- complexity property** [N-COUNT-U13] The **complexity property** is a property of test adequacy that states that the complexity of a program directly relates to the complexity required of its test sets.
- component** [N-COUNT-U6] A **component** is a computational element or data store used in software architecture structures.
- conceptual view** [N-COUNT-U5] A **conceptual view** is a way to describe a software system in terms of major design elements and the interactions between those elements.
- connector** [N-COUNT-U6] A **connector** is a computational element that determines how components interact.
- consumption** [N-UNCOUNT-U3] **Consumption** is the amount of an extensive quantity that is destroyed during a particular period of time.
- control structure** [N-COUNT-U6] A **control structure** is a component that shows and dictates the order of execution of components.
- corrective maintenance** [N-UNCOUNT-U14] **Corrective maintenance** is the practice of repairing faults in software systems.
- coupling** [N-UNCOUNT-U7] **Coupling** is the measure of the strength of connections between modules in a system.
- coverage-based testing** [N-UNCOUNT-U12] **Coverage-based testing** is a model of software testing in which the adequacy of a test is determined by the percentage of the software that is examined.
- critical thinking** [N-UNCOUNT-U1] **Critical thinking** is the ability to draw logical conclusions based on facts and evidence.
- curious** [ADJ-U1] If someone is **curious**, he or she wants to know more about something.
- customer-driven** [ADJ-U4] If software is **customer-driven**, it is developed in response to a clear, specific need of potential customers.
- data flow design** [N-COUNT-U9] A **data flow design** is a plan that shows the way data will move through a system.
- decompose** [V-T-U8] To **decompose** a function is to split it into the subfunctions that make it up.
- dedicated** [ADJ-U1] If someone is **dedicated**, he or she is enthusiastic about a task or cause.
- demonstration model** [N-COUNT-U11] The **demonstration model** is a type of software testing which ensures that software satisfies its intended purpose.
- deployment view** [N-COUNT-U5] A **deployment view** is a way to describe a software system in terms of the way software assigns tasks to physical nodes.
- design method** [N-COUNT-U8] A **design method** is a set of guidelines and procedures for designing a software system.
- design pattern** [N-COUNT-U5] A **design pattern** is a reusable solution that can be applied to commonly occurring software design problems.
- design recovery** [N-UNCOUNT-U15] **Design recovery** is the process of creating a program that is identical to an existing program in function but is better organized in abstraction.
- destruction model** [N-COUNT-U11] The **destruction model** is a type of software testing intended to detect implementation faults in a new piece of software.
- DFD** [N-COUNT-U9] A **DFD** (Data Flow Diagram) is a graphical representation of the route that data takes as it moves through a system.
- DSSA** [N-UNCOUNT-U6] **DSSA** (Domain Specific Software Architecture) is any style of system architecture which includes a reference architecture, component library, and application configuration method.
- dynamic analysis** [N-UNCOUNT-U12] **Dynamic analysis** is a type of software testing in which a program is executed and the results of this execution are examined.
- elicitation** [N-UNCOUNT-U4] **Elicitation** is the process of causing something to become apparent or realized.
- enhance** [V-T-U14] To **enhance** something is to improve its function.

# Glossary

- error** [N-COUNT-U11] An **error** is a human action that produces an incorrect result in software.
- error-based testing** [N-UNCOUNT-U12] **Error-based testing** is a software testing technique that detects common errors made by humans.
- evaluation model** [N-COUNT-U11] The **evaluation model** is a type of software testing intended to detect requirement, design, and implementation faults.
- expected** [ADJ-U11] If something is **expected**, it is considered likely to happen.
- expertise** [N-UNCOUNT-U1] **Expertise** is extensive or advanced knowledge in a particular subject or area.
- extensive quantity** [N-COUNT-U3] An **extensive quantity** is an amount that changes based on the size of a system and has distinct, countable units.
- Fagan inspection** [N-COUNT-U12] A **Fagan inspection** is a process in which a team of engineers manually inspects the code of a piece of software.
- failure** [N-COUNT-U11] A **failure** is the observable results of a fault in software.
- fault** [N-COUNT-U11] A **fault** is the result of an error made by an engineer.
- fault detection** [N-UNCOUNT-U11] **Fault detection** is the process of finding faults and exposing failures in software.
- fault prevention** [N-UNCOUNT-U11] **Fault prevention** is the process of anticipating and stopping problems by testing software multiple times during the development phase.
- fault-based testing** [N-UNCOUNT-U12] **Fault-based testing** is a software testing technique that focuses primarily on testing for faults.
- final** [ADJ-U3] If something is **final**, it is related to the status of something at the end of a process or period of time.
- focus** [V-I-U1] To **focus** on something is to watch it closely or give full attention to it.
- functional decomposition** [N-UNCOUNT-U8] **Functional decomposition** is a design philosophy in which a function is decomposed into a number of subfunctions.
- functional equivalence** [N-UNCOUNT-U15] **Functional equivalence** is a measure of how similar two programs are in purpose and function, even though they may be coded differently.
- functional hierarchy** [N-COUNT-U4] A **functional hierarchy** is an undefined system used to organize specifications in a requirements document.
- Fusion method** [N-COUNT-U10] The **Fusion method** is an object-oriented software development process that structures the process into analysis, design, and implementation phases.
- general multiple change property** [N-COUNT-U13] The **general multiple change property** is a property of test adequacy that states that programs with the same structure and dataflow characteristics should still be tested on different criteria.
- generation** [N-UNCOUNT-U3] **Generation** is the amount of an extensive quantity that is created during a particular period of time.
- goal-oriented** [ADJ-U1] If someone is **goal-oriented**, he or she works or acts towards a particular purpose.
- idealistic** [ADJ-U8] If a something is **idealistic**, it assumes the best possible conditions and situations.
- idiom** [N-COUNT-U5] An **idiom** is a low level pattern that is specific to a programming language and can be used to perform a basic function.
- implementation stage** [N-COUNT-U9] The **implementation stage** is a stage in JSD in which a system is transformed from a network of processes to a working design.
- implementation view** [N-COUNT-U5] An **implementation view** is a way to describe a software system in terms of modules of packages and layers.
- implicit invocation** [N-COUNT-U6] An **implicit invocation** is a system in which computations are invoked by certain events rather than by interaction with the user.

- inadequate empty set property** [N-COUNT-U13] The **inadequate empty set property** is a property of test adequacy that states that an empty set is not an adequate test set for any program.
- information hiding** [N-UNCOUNT-U7] **Information hiding** is a system in which modules contain information that is not likely to change, thereby protecting parts of a program from extensive modifications.
- initial** [ADJ-U3] If something is **initial**, it is related to the status of something at the beginning of a process or period of time.
- innovative** [ADJ-U1] If something is **innovative**, it is new, creative, and advanced.
- input** [N-COUNT-U3] An **input** is the amount of an existing extensive quantity that is added to a system during a particular period of time.
- insufficient** [ADJ-U14] If something is **insufficient**, it is not suitable or strong enough for a particular purpose.
- intensive quantity** [N-COUNT-U3] An **intensive quantity** is an amount that does not change based on the size of a system, which can be measured, but cannot be separated into distinct, countable units.
- interaction diagram** [N-COUNT-U10] An **interaction diagram** is a graph that depicts the sequence of messages of which a typical graph is composed.
- inter-modular attribute** [N-COUNT-U7] An **inter-modular attribute** is a feature or characteristic of an entire system of modules.
- intra-modular attribute** [N-COUNT-U7] An **intra-modular attribute** is a feature or characteristic of an individual module.
- iteration** [N-UNCOUNT-U2] An **iteration** is a new or updated version of a piece of hardware or software.
- iterative** [ADJ-U2] If something is **iterative**, it is intended to be updated in order to improve or perfect it.
- JSD** [N-UNCOUNT-U9] **JSD** (Jackson System Development) is a method of system development which contains three distinct phases in the development process.
- JSP** [N-UNCOUNT-U9] **JSP** (Jackson Structured Programming) is a method of system development that is based on data flow and program structure.
- law of continuing change** [N-COUNT-U14] The **law of continuing change** is a principle that states that a system in use should undergo continuing change until it becomes more cost effective to restructure the system.
- law of increasing complexity** [N-COUNT-U14] The **law of increasing complexity** is a principle that states that a structure becomes more complex with every change that is made to it.
- layered** [ADJ-U6] If an architectural style is **layered**, it is organized by ascending functionality.
- legacy system** [N-COUNT-U15] A **legacy system** is an old software system that continues to be updated and used.
- logical** [ADJ-U1] If something is **logical**, it is based on evidence and reason.
- main program with subroutines** [N-COUNT-U6] A **main program with subroutines** is a hierarchical system in which a top level module invokes other modules in a given order.
- market-driven** [ADJ-U4] If software is **market-driven**, it is developed for a broad purpose rather than a specific need.
- mode** [N-COUNT-U4] A **mode** is a changeable system of operation that dictates how software behaves.
- modeling stage** [N-COUNT-U9] The **modeling stage** is a stage in JSD in which a description is made of the problem that the software needs to solve.
- modernize** [V-T-U15] To **modernize** something is to make it compatible with new technology or update its appearance and functionality.
- modularity** [N-UNCOUNT-U7] **Modularity** is a way of viewing a system as a series of smaller interconnected systems.
- module** [N-COUNT-U5] A **module** is a group of software functions that are bundled together.
- monotonicity property** [N-COUNT-U13] The **monotonicity property** is a property of test adequacy criteria that states that additional testing can be performed even after a program has been adequately tested.

# Glossary

- network stage** [N-COUNT-U9] The **network stage** is a stage in JSD in which a system is shown as a network of communicating processes.
- non-exhausting applicability property** [N-COUNT-U13] The **non-exhausting applicability property** is a property of test adequacy criteria that states that a criterion does not require exhaustive testing in all circumstances.
- object** [N-COUNT-U4] An **object** is a physical thing that can be touched and seen.
- object-oriented** [ADJ-U10] If a design is **object-oriented**, it uses objects, or data structures, as a basis for designing software.
- OMT** [N-UNCOUNT-U10] The **OMT** (Object Modeling Technique) is an object-oriented approach to software development.
- open system** [N-COUNT-U3] An **open system** is a system that allows mass to enter and leave it.
- oracle** [N-COUNT-U11] An **oracle** is a mechanism used to compare predicted results with the actual results of a software test.
- output** [N-COUNT-U3] An **output** is the amount of an extensive quantity that is removed from a system, but not destroyed, during a particular period of time.
- outside the box** [ADV PHRASE-U1] If someone thinks **outside the box**, he or she has ideas that are creative or unusual for a particular situation.
- peer review** [N-UNCOUNT-U12] **Peer review** is a practice in which engineers read the programs of other engineers to identify faults or inadequacies in programs.
- perfective maintenance** [N-UNCOUNT-U14] **Perfective maintenance** is the practice of updating software in order to accommodate new user requirements.
- philosophy** [N-COUNT-U8] A **philosophy** is a way of understanding or viewing something.
- pipes and filters** [N-UNCOUNT-U6] **Pipes and filters** is a style that relies on input streams and system operations to process ordered data.
- prevention model** [N-COUNT-U11] The **prevention model** is a type of software testing intended to prevent faults in design, requirements, and implementation.
- preventive maintenance** [N-UNCOUNT-U14] **Preventive maintenance** is the practice of improving the structure of a system in order to make it easier to maintain.
- primitive** [ADJ-U8] If something is **primitive**, it is simple or basic.
- problem identification** [N-UNCOUNT-U2] **Problem identification** is the act of describing and analyzing problems at the first stage of the problem solving process.
- problem solving** [N-UNCOUNT-U2] **Problem solving** is the ability to identify problems, think of solutions, and enact those solutions.
- procedure** [N-UNCOUNT-U2] A **procedure** is an established series of actions that dictates how to do something.
- process view** [N-COUNT-U5] A **process view** is a way to describe a software system in terms of the tasks and processes a system performs and the way those tasks and processes interact.
- programming plan** [N-COUNT-U5] A **programming plan** is a program fragment that is used to describe a common action.
- proof of correctness** [N-UNCOUNT-U12] **Proof of correctness** is a process which formally states a program's specification and proves that the program meets that specification.
- property** [N-COUNT-U10] A **property** is an identifying and descriptive characteristic or attribute, and may apply to a single entity or a relationship between entities.
- rational** [ADJ-U8] If a design process is **rational**, it works according to a logical system.
- redefine** [V-T-U2] To **redefine** something is to change its function or meaning.

- redocumentation** [N-COUNT-U15] **Redocumentation** is the process of improving or simplifying a program's code without changing its function or level of abstraction.
- reengineering** [N-UNCOUNT-U15] **Reengineering**, also called renovation, is the process of making functional changes to a system.
- relationship** [N-COUNT-U10] A **relationship** is a property that depends on the way two entities interact.
- release** [N-COUNT-U14] A **release** is an updated version of an existing software program.
- renaming property** [N-COUNT-U13] The **renaming property** is a property of test adequacy that states that two programs that differ only in unimportant ways can be tested with the same test sets.
- renovation** [N-UNCOUNT-U15] **Renovation**, also called reengineering, is the process of making functional changes to a system.
- repair** [V-T-U14] To **repair** something is to fix parts of it that are not functioning correctly.
- repository** [N-COUNT-U6] A **repository** is an architectural style designed for systems which manage a body of data with an inherent structure.
- requirements engineering** [N-UNCOUNT-U4] **Requirements engineering** is the practice of creating and documenting requirements for software and other computer systems.
- response** [N-COUNT-U4] A **response** is information provided by software upon search or request.
- restructuring** [N-UNCOUNT-U15] **Restructuring** is the process of updating a system while keeping the same functionality and level of abstraction.
- revamping** [N-UNCOUNT-U15] **Revamping** is the process of updating the user interface of a program without changing the program's structure.
- reverse engineering** [N-UNCOUNT-U15] **Reverse engineering** is the process of analyzing an existing software system and creating a new version of the system at a higher level of abstraction.
- SA** [N-COUNT-U9] An **SA** (Structured Analysis) is a method for converting real-life requirements into software that will fulfill a specific need.
- satisfy** [V-T-U11] To **satisfy** a requirement is to complete the necessary tasks or meet all of the expectations involved in the requirement.
- scenario-based evaluation** [N-UNCOUNT-U12] **Scenario-based evaluation** is a model of software testing which is guided by simulations of common use scenarios.
- schematic logic** [N-UNCOUNT-U9] **Schematic logic** is a code that is used in a structure diagram.
- SD** [N-COUNT-U9] A **SD** (Structured Design) is the development of modules and module hierarchies with the goal of *creating an optimal module structure*.
- sequence diagram** [N-COUNT-U10] A **sequence diagram** is a graph that depicts the time ordering of events within an interaction.
- simplify** [V-T-U7] To **simplify** something is to eliminate unnecessary elements.
- software architecture** [N-UNCOUNT-U5] **Software architecture** is the practice of viewing systems in terms of their major components and characterizing the interaction between those components.
- software maintenance** [N-UNCOUNT-U14] **Software maintenance** is the process of adapting or modifying a software system to correct faults or improve performance.
- solution** [N-COUNT-U2] A **solution** is a way of solving or fixing a problem.
- specification** [N-COUNT-U4] A **specification** is the precise definition or description of a problem.
- state** [N-COUNT-U10] A **state** is the set of attributes of a particular object.
- state diagram** [N-COUNT-U10] A **state diagram** is a graph which depicts the dynamic behavior of single objects.

# Glossary

- statement coverage property** [N-COUNT-U13] The **statement coverage property** is a property of test adequacy that states that every possible action of a program should be executed by its test sets.
- static analysis** [N-UNCOUNT-U12] **Static analysis** is a type of software testing in which a program's structure and components are examined without being executed.
- stepwise abstraction** [N-UNCOUNT-U12] **Stepwise abstraction** is a technique for analyzing software in which all of the code is examined, from the most primitive to the most abstract.
- stepwise refinement** [N-UNCOUNT-U8] **Stepwise refinement** is a problem-solving approach in software design in which a problem is divided into smaller, more manageable sections.
- stopping rule** [N-COUNT-U7] A **stopping rule** is an indication that the solution to a problem has been reached.
- structure chart** [N-COUNT-U9] A **structure chart** is a chart that shows the functions of a system from the most complex to the most primitive.
- structure diagram** [N-COUNT-U9] A **structure diagram** is a diagram representing compound components in a structure.
- subfunction** [N-COUNT-U8] A **subfunction** is a component which combines with other subfunctions to make up a function.
- synthesis** [N-COUNT-U2] A **synthesis** is a combination of multiple items or elements.
- system** [N-COUNT-U3] A **system** is a set of connected things that work together to produce a result.
- system model** [N-COUNT-U6] A **system model** is a description of the characterization of a system as defined by its components and connectors.
- system structure** [N-COUNT-U7] A **system structure** is the makeup of a system's modules and how they are connected.
- team player** [N-COUNT-U1] A **team player** is someone who takes actions that benefit a larger group rather than only his or her own interests.
- test adequacy criteria** [N-COUNT-U13] **Test adequacy criteria** are sets of requirements that measure the effectiveness of a software testing process.
- test criterion** [N-COUNT-U11] A **test criterion** is a set standard or qualification by which something is tested.
- top-down design** [N-COUNT-U8] A **top-down design** is a method of designing software in which the engineer begins by defining the main user functions and decomposes those functions into subfunctions, until the basic operations of the software are defined.
- universal accounting equation** [N-COUNT-U3] The **universal accounting equation** is an equation that is used to measure changes in extensive quantities over particular periods of time.
- unstructured code** [N-UNCOUNT-U14] **Unstructured code** is the code for a system that is designed poorly or coded without a clear structure or order.
- user class** [N-COUNT-U4] A **user class** is a distinction that changes the function of software according to the particular user of the software.
- user-friendly** [ADJ-U4] If something is **user-friendly**, it is easy for most people to understand or use.
- validation** [N-UNCOUNT-U4] **Validation** is the process of determining that the requirements of a problem are correct.
- verification** [N-UNCOUNT-U4] **Verification** is the process of determining that a problem's requirements are expressed correctly.
- web-based** [ADJ-U15] If something is **web-based**, it is used on the Internet.
- white-box testing** [N-UNCOUNT-U12] **White-box testing** is a software testing technique which examines the internal logical structure of software.
- wicked problem** [N-COUNT-U7] A **wicked problem** is a problem encountered in software design that has both a complicated cause and complicated solution, and may be the result of another problem.

**CAREER  
PATHS**

# SOFTWARE ENGINEERING

Book

**3**

Virginia Evans

Jenny Dooley

Enrico Pontelli



**Express Publishing**

## Scope and Sequence

Unit	Topic	Reading context	Vocabulary	Function
1	User Interface Design 1	Textbook Excerpt	apparatus level, cognitive view, CLG, communication component, conceptual model, conceptual component, design view, keystroke level, linguistic view, material component, mental model, semantic level, spatial layout level, syntax level, task level, user interface	Assigning tasks
2	User Interface Design 2	Job Listing	artistic design, dialog, end user, ergonomics, functionality, groupware, HCI, humanities, layer, MVC paradigm, presentation, Seeheim model, task analysis, UVM	Rating importance
3	Software Reuse 1	Journal Article	ad hoc, approach, black-box reuse, compositional, COTS, generative, product, scope, substance, software crisis, software reuse, source code, technique, usage, white-box reuse	Making a recommendation
4	Software Reuse 2	Textbook Excerpt	ADL, application generator, code scavenging, domain analysis, instantiate, intermediate product, middleware, MLL, program library, skeleton, template, transformation system, VHLL	Describing ability
5	Software Reliability	Handbook	BM, defensive programming, exception domain, expected exception domain, fault-tolerant, LPM, N-version programming, probability, recovery block, redundancy, reliability, robust programming, software reliability model, standard domain, threshold	Stating a preference
6	Software Tools 1	Webpage	CASE, city, environment, family, individual, integrated environment, language-centered environment, process-centered environment, process scale, state, tool, toolkit, user scale, workbench	Estimating time
7	Software Tools 2	Review	AWB, back-end, IPSE, MWB, PCTE, programming environment, PWB, reserved checkout, SSCS, UNIX, unreserved checkout, visual programming environment	Describing necessity
8	Configuration Management	Email	approve, baseline, COB, change-oriented, change request, configuration item, configuration management, corresponding, delta, flaw, incorporate, parallel development, retrace, version-oriented, workflow	Describing a process
9	Programming Teams	Letter	commitment style, chief programmer team, hierarchical organization, integration style, matrix organization, open structured team, relation style, relation directness unit, separation style, specialize in, SWAT team, task directedness, unit	Expressing concerns
10	Software Quality Control	Report	CMM, common feature, conform, improve, IEEE Standard for Quality Assurance Plans, ISO 9001, key practice, key process area, maturity level, quality control, quality criteria, quality factor, FQM	Making a realization
11	Development and Cost	Memo	algorithmic model, budget, base formula, COCOMO, comparison method, Delphi-method, development time, estimate, KLOC, learning effect, man-month, optimistic, Putnam model, Walston-Felix	Delivering good news
12	Project Management	Advertisement	allocation problem, critical path, degree of certainty, design problem, exploration problem, Gantt chart, PERT chart, process certainty, product certainty, realization problem, resource certainty, risk factor, risk management, WBS	Summarizing a point
13	Ethics	Poster	adequate, best interests, deceptive, ensure, ethics, health, integrity, principle, professional judgment, public interest, safety, standard, unethical, welfare	Emphasizing a point
14	Cloud Computing: SaaS & PaaS	Journal Article	bandwidth, browser, cloud computing, computing platform, distribution, metered fee, online, PaaS, pay as you go, SaaS, software as a product, software license, software on demand	Giving an opinion
15	Career Options	Webpage	advance, ACM, analyst, architect, contractor, developer, development, educator, freelancer, IEEE, manager, membership, professional development, researcher, software life cycle, technical support, tester	Asking for advice

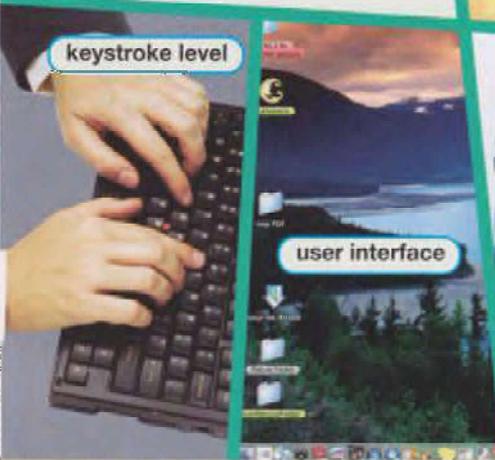
# Table of Contents

<b>Unit 1 – User Interface Design 1</b> .....	4
<b>Unit 2 – User Interface Design 2</b> .....	6
<b>Unit 3 – Software Reuse 1</b> .....	8
<b>Unit 4 – Software Reuse 2</b> .....	10
<b>Unit 5 – Software Reliability</b> .....	12
<b>Unit 6 – Software Tools 1</b> .....	14
<b>Unit 7 – Software Tools 2</b> .....	16
<b>Unit 8 – Configuration Management</b> .....	18
<b>Unit 9 – Programming Teams</b> .....	20
<b>Unit 10 – Software Quality Control</b> .....	22
<b>Unit 11 – Development and Cost</b> .....	24
<b>Unit 12 – Project Management</b> .....	26
<b>Unit 13 – Ethics</b> .....	28
<b>Unit 14 – Cloud Computing: SaaS and PaaS</b> .....	30
<b>Unit 15 – Career Options</b> .....	32
<b>Glossary</b> .....	34

```
def add5 x
  return x+5

def dotwrite out
  nodename getNodename
  label symbol sym name get out out
  print "% label %s %s %s" % label %s %s %s
  if out[0] == '1'
    print "%s %s %s" % out %s %s
  else
    print "%s" % out
  end
end

print
print children
for n child in enumerate out
  children append dotwrite child
end
for name in children
  print "%s %s" % name
```



## Chapter 10

# User Interface Design

Engineers use **conceptual models** to optimize their systems for ease of use. In order to do this, they study the **user interface** carefully. Various views help engineers see different aspects of a design.

Engineers begin by considering the user's **mental model** of a system. This is closely tied to the **cognitive view** of a system. This view considers what information a user will need to know in order to operate a system. **Linguistic view** helps engineers imagine the ways a user will interact with a system. **Design view** lets them examine screen layouts and other elements of user interface design.

**CLG** (Command Language Grammar) divides a system into even more specific views. These views encompass all elements of interaction between the system and the user. CLG views a system according to three components: the **conceptual component**, the **communication component**, and the **material component**.

Each of these components is made up of two levels, each focusing on different information. The conceptual component includes the **semantic level** and the **task level**. The material component includes the **spatial layout level** and the **apparatus level**. The communications component includes the **keystroke level** and the **syntax level**.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 How do engineers optimize user interface designs?
- 2 What are the levels of CLG?

## Reading

2 Read the textbook excerpt. Then, choose the correct answers.

- 1 Why do engineers use conceptual models?
  - A to make software easier to use
  - B to eliminate mistakes in software
  - C to provide information for software users
  - D to update a system's software
- 2 Which of the following is NOT a division of CLG?
  - A conceptual      C linguistic
  - B material        D communication
- 3 What do engineers use design view for?
  - A to divide components into levels
  - B to consider what information a user needs
  - C to consider the mental model of systems
  - D to examine elements of the system's user interface

## Vocabulary

3 Place the correct phrases from the word bank under the correct headings.

### Word BANK

- apparatus level    keystroke level
- semantic level    spatial layout level
- syntax level      task level

Conceptual Component Views	Communication Component Views	Material Component Views

**4** Read the sentence pairs. Choose where the words best fit the blanks.

- 1 **communication component / conceptual component**
  - A The \_\_\_\_\_ concerns the dialog between systems and users.
  - B The \_\_\_\_\_ concerns the functions the systems will perform for users.
- 2 **mental model / conceptual model**
  - A A \_\_\_\_\_ is rendered in terms of system's reactions to user actions.
  - B A \_\_\_\_\_ concerns the way a user understands a computer system.
- 3 **linguistic view / cognitive view**
  - A \_\_\_\_\_ considers what a user needs to understand about a system in order to operate it.
  - B \_\_\_\_\_ considers the interactions between a human and a system.
- 4 **material component / CLG**
  - A A \_\_\_\_\_ considers both the graphics of the user interface and the system hardware.
  - B \_\_\_\_\_ describes the user interface of all aspects of a system.
- 5 **design view / user interface**
  - A The \_\_\_\_\_ describes the attributes of a system that are relevant to the user.
  - B The \_\_\_\_\_ is a conceptual model that focuses on icons and screen layouts.

**5** Listen and read the textbook excerpt again. What are some different views that engineers use when considering user interfaces?

**Listening**

**6** Listen to a conversation between two software engineers. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The material component was already finished.
- 2 \_\_\_ The woman confuses the task level and the semantic level.
- 3 \_\_\_ The man will work on the semantic level.

**7** Listen again and complete the conversation.

**Engineer 1:** We need to talk 1 \_\_\_\_\_ on the library software.

**Engineer 2:** Okay. Well, the material component 2 \_\_\_\_\_.

**Engineer 1:** When will it be done?

**Engineer 2:** Probably by the end of this week.

**Engineer 1:** That's good. What about 3 \_\_\_\_\_?

**Engineer 2:** We didn't start that yet.

**Engineer 1:** Oh, I think we should get to work on that as soon as possible.

**Engineer 2:** I agree. Do you want to 4 \_\_\_\_\_ and then work on them separately?

**Engineer 1:** That's a good idea. I'll take the semantic level. You get 5 \_\_\_\_\_.

**Engineer 2:** What goes into the task level again?

**Engineer 1:** You define the tasks performed 6 \_\_\_\_\_ and the tasks performed by the software.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I think we should ... / Do you want to ...  
I'll take ...*

**Student A:** You are an engineer. Talk to Student B about:

- the status of the software components you are developing
- when certain components will be completed
- who will work on each component

**Student B:** You are an engineer. Talk to Student A about the software components you are developing.

**Writing**

**9** Use the textbook excerpt and conversation from Task 8 to complete a guide to components. Include: a list of software components, the levels within each component, and the purpose of each level.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 How do engineers use user interface designs to improve the user experience?
- 2 What are the components of the Seeheim model and the model-view-controller paradigm?

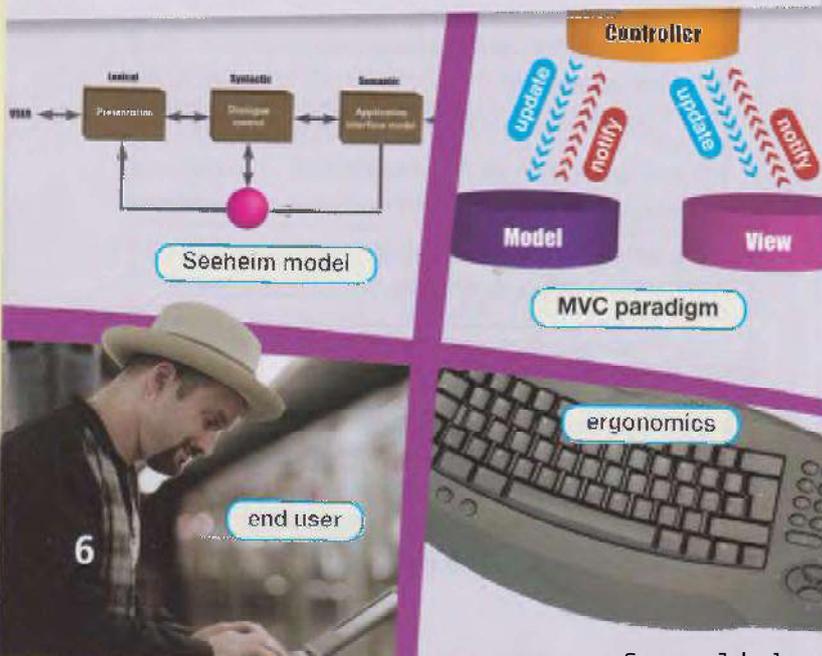
## Seeking a Software Engineer

ShorSoft Designs needs a software engineer. This position is in our user interface design department. It involves the creation of functional, attractive software interfaces. The right candidate should have experience with **artistic design** and **ergonomics**. Candidates should be familiar with both the **Seeheim model** and the **MVC** (model-view-controller) **paradigm**. They should also be able to work on all **layers** of a system. A degree in software engineering or a related field is required. A specialization in **HCI** or **task analysis** is preferred. Candidates should also have 3+ years experience in software engineering and design.

The position involves perfecting the **presentation** of software systems for our **end user**. We hope to streamline **dialog** and simplify interactions. The right candidate should be concerned with **functionality** as well as **humanities**. In other words, we need someone who understands both our systems and our users. We place strong emphasis on **user-centered design**.

New hires for this position will begin with **groupware** projects. If the fit is good, the employee will be assigned to a **UVM** project. We offer competitive salaries and an innovative work environment.

To apply, please send a résumé and cover letter to hr@shorsoft.org. Résumé should include references with contact information.



## Reading

2 Read the job listing. Then, choose the correct answers.

- 1 What will the employee be responsible for?
  - A teaching new engineers about new software developments
  - B reviewing the references of intern applicants
  - C perfecting the presentation of software
  - D developing new user interface models
- 2 Which of the following is NOT a requirement for candidates?
  - A degree in software engineering
  - B three or more years of experience
  - C familiarity with the Seeheim model
  - D experience working on groupware projects
- 3 What is a benefit of working for the company?
  - A an innovative work environment
  - B financial assistance towards an engineering degree
  - C opportunity to develop a specialization
  - D access to perfect presentation software

## Vocabulary

2 Match the words and phrases (1-8) with the definitions (A-H).

- |                      |                    |
|----------------------|--------------------|
| 1 __ artistic design | 5 __ layer         |
| 2 __ end user        | 6 __ presentation  |
| 3 __ groupware       | 7 __ task analysis |
| 4 __ HCI             | 8 __ MVC paradigm  |

- A software that assists groups in working towards a common goal
- B all aspects of a system that are perceptible to a user
- C the practice of using graphic design in user interfaces
- D the consumer who will ultimately use a product
- E a level of system operation
- F a way of evaluating a complex system
- G a model that splits user interface design into three distinct parts
- H the study and design of interactions between computers and users

4 Fill in the blanks with the correct words and phrases from the word bank.

**WORD BANK**

- dialog    Seeheim model    ergonomics    functionality
- humanities    UVM    user-centered design

- 1 Designers study \_\_\_\_\_ so their physical components are comfortable for users.
- 2 It is important that \_\_\_\_\_ between users and computers is easy.
- 3 The \_\_\_\_\_ divides an application from its user interface.
- 4 The interface is very attractive, but unfortunately it has limited \_\_\_\_\_.
- 5 \_\_\_\_\_ comprises a system's hardware and software.
- 6 \_\_\_\_\_ includes the way people think and react to various features and events.
- 7 In \_\_\_\_\_, engineers focus extensively on the experience of people who will use the software.

5 Listen and read the job listing again. What qualifications should applicants have?

**Listening**

6 Listen to a conversation between a software engineer and a hiring manager. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The man designed mice for a hardware production company.
- 2 \_\_\_ The man has a degree in computer science.
- 3 \_\_\_ The woman expresses some doubt about the man's skills.

7 Listen again and complete the conversation.

**Manager:** Good. Let's talk about your experience. It says here that you 1 \_\_\_\_\_ before?

**Engineer:** Yes. I 2 \_\_\_\_\_ for a hardware production company.

**Manager:** How would you incorporate that experience 3 \_\_\_\_\_?

**Engineer:** I think it's really important that software engineers understand the 4 \_\_\_\_\_.

**Manager:** How do you think they can do that?

**Engineer:** They need to think about the physical way a user interacts with a computer.

**Manager:** I agree. Do you have experience 5 \_\_\_\_\_?

**Engineer:** Yes. I worked on 6 \_\_\_\_\_ at my previous firm.

**Speaking**

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Let's talk about ...*

*Can you tell me ...*

*It says here ...*

**Student A:** You are an engineer. Talk to Student B about Student A:

- your experience and education
- your views on different aspects of user interface design

**Student B:** You are a hiring manager. Talk to Student A about a program that you are developing.

**Writing**

9 Use the job listing and conversation from Task 8 to write an application letter for a job in user interface design. Include: the applicant's experience, educational background, and views on different aspects of user interface design.





## Get ready!

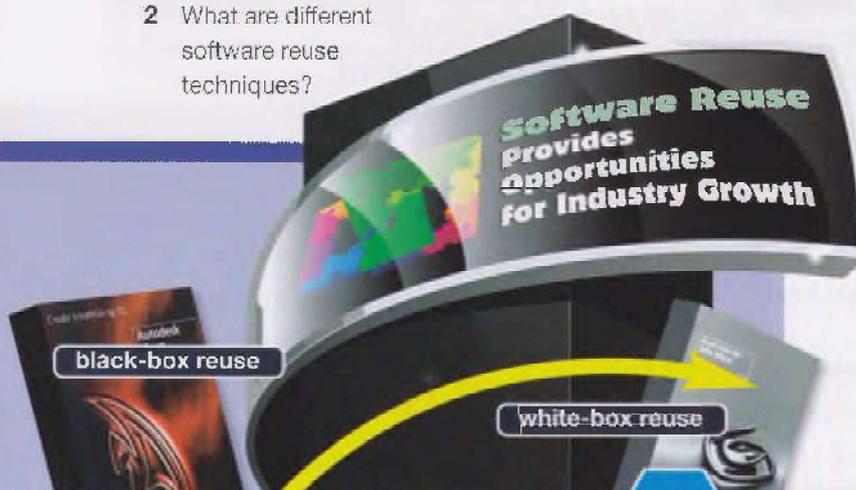
1 Before you read the passage, talk about these questions.

- 1 How can software reuse help with the software crisis?
- 2 What are different software reuse techniques?

## Reading

2 Read the journal article. Then, choose the correct answers.

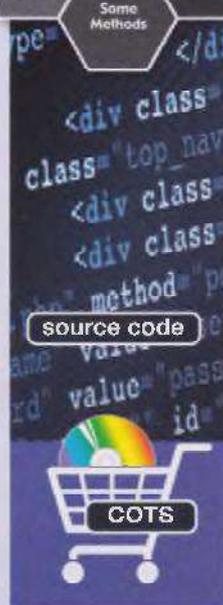
- 1 What is the purpose of the article?
  - A to explain the historical development of software reuse
  - B to suggest that engineers practice software reuse
  - C to list different types of software that can be reused
  - D to discuss the legal issues involved in software reuse
- 2 Which of the following is NOT a benefit of software reuse?
  - A elimination of usage problems
  - B increased productivity for software companies
  - C stimulation of software companies
  - D improved development of compositional software
- 3 What is a problem with white-box reuse?
  - A It uses unreliable source codes.
  - B It leads to mistakes in software development.
  - C It does not support mass development of new software.
  - D It is limited in scope.



Today's demand for new software exceeds the industry's ability to produce it. If software companies want to overcome the **software crisis**, they must practice **software reuse**. Many developers do practice software reuse. However, the **scope** of some software is much wider than its current usage. Engineers can avoid extra work by using existing components to begin projects.

Software reuse is a common practice, but it is often handled ineffectively. One problem is that it typically occurs in an **ad hoc** manner. This kind of **white-box reuse** does not support mass development of new software. With new software networks, engineers can improve their reuse **techniques** and increase their overall productivity. Particularly smart companies can even market their **source codes** as **products**. These **COTS** codes would be available for **black-box reuse**. These would not only stimulate the software market, but also provide **substance** that is tailored for reuse.

Of course, widespread reuse of software components would require a general change in **approaches** to development. Instead of creating software that is **compositional**, they must create **generative** software. This could eliminate problems with **usage**. It could also open up new opportunities for extensive software development.



## Vocabulary

3 Match the words (1-7) with the definitions (A-G).

- |               |                   |
|---------------|-------------------|
| 1 ___ ad hoc  | 5 ___ source code |
| 2 ___ COTS    | 6 ___ substance   |
| 3 ___ product | 7 ___ technique   |
| 4 ___ scope   |                   |

- A unmodified from its original state
- B a skill or specific method
- C the extent or area that something covers
- D the components, concepts, and procedures of something
- E done in a particular instance only
- F something that is available for purchase
- G a list of commands to be executed

**4** Read the sentence pairs. Choose where the words best fit the blanks.

**1 software reuse / software crisis**

- A** The \_\_\_\_\_ is created by a rising demand for new applications.
- B** \_\_\_\_\_ is the practice of recycling pieces of code into new applications.

**2 approach / usage**

- A** A(n) \_\_\_\_\_ is a way of dealing with something.
- B** Something's \_\_\_\_\_ is how it's utilized.

**3 black-box reuse / white-box reuse**

- A** \_\_\_\_\_ requires engineers to modify software.
- B** \_\_\_\_\_ does not require engineers to modify software.

**4 compositional / generative**

- A** \_\_\_\_\_ software can easily be reused in new systems.
- B** \_\_\_\_\_ software can be used to create new programs.

**5** Listen and read the journal article again. What are the benefits of black-box reuse?

## Listening

**6** Listen to a conversation between two software engineers. Mark the following statements as true (T) or false (F).

- \_\_\_ The man was unhappy with the software reuse on a previous project.
- \_\_\_ The woman practiced white-box reuse on her last project.
- \_\_\_ The woman recommends that the man try black-box reuse.

**7** Listen again and complete the conversation.

**Engineer 1:** I'm thinking about 1 \_\_\_\_\_ on a new project. But I never tried that before.

**Engineer 2:** I did 2 \_\_\_\_\_ on my last application.

**Engineer 1:** Did it work well for you?

**Engineer 2:** Yeah, it was great. It really saved me a lot of 3 \_\_\_\_\_.

**Engineer 1:** Huh. Would you recommend black-box reuse?

**Engineer 2:** Yeah, it was okay. I used 4 \_\_\_\_\_ were for sale. They were more suited to the project I was working on.

**Engineer 1:** Is it possible to get source codes for free?

**Engineer 2:** I think so, but they 5 \_\_\_\_\_ as specific as the kind I used.

**Engineer 1:** I could just do some 6 \_\_\_\_\_.

## Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Do you know ... / Is it possible to ... / I could just ...*

**Student A:** You are an engineer. Talk to Student B about:

- his or her opinion on software reuse
- different types of software reuse
- what he or she recommends

**Student B:** You are an engineer. Talk to Student A about software reuse.

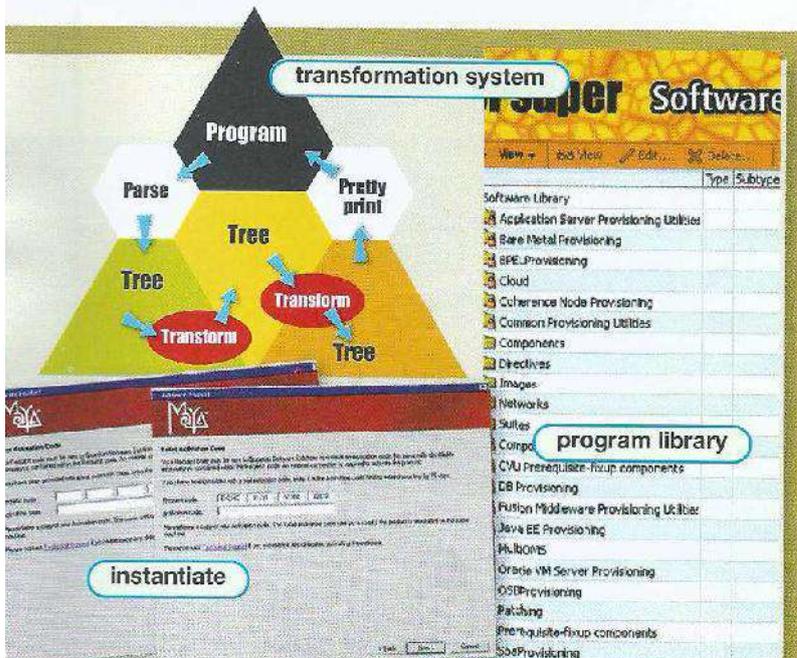
## Writing

**9** Use the journal article and conversation from Task 8 to write an email from one software engineer to another. Include: the kind of software reuse you would recommend, the advantages of your recommendation, and the disadvantages of another kind of software reuse.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What tools do engineers use for software reuse?
- 2 How does a transformation system work?



## Chapter 24

## Software Reuse

Engineers use various tools for software reuse. Through **code scavenging**, they use pieces of old code as basis for new code. They also have access to **program libraries** that are full of **intermediate products**. Engineers can easily **instantiate** these **templates** during system development. The templates are often used to form the basis for whole new systems. **Application generators** provide engineers with support tools for developing these systems.

Engineers who choose to reuse code are aided by **domain analysis** programs. These programs search for usable code and information within existing pieces of code. This means that engineers do not need to waste time scanning code manually for useful parts.

**Transformation systems** are another resource for engineers. At first, programs are written in languages such as **ADLs** and **MILs**. Transformation systems turn these **skeleton** plans into working systems. They translate lower level code into higher order processes. They also help developers create **middleware**. This is software that connects applications to computer operating systems. All of these systems that aid engineers are written in **VHLL**. This type of programming language does not always provide a user interface. However, its high level of abstraction makes it easy for programmers to use.

## Reading

2 Read the textbook excerpt. Then, choose the correct answers.

- 1 What is true of intermediate products?
  - A They transform skeleton plans into systems.
  - B They can be the foundation of new systems.
  - C They are only used in unusual situations.
  - D They cost less than traditional engineering methods.
- 2 Which of the following is NOT a tool for software reuse?
  - A transformation systems
  - B domain analysis programs
  - C ADLs
  - D program libraries
- 3 What makes a VHLL useful for programmers?
  - A It has a high level of abstraction.
  - B It searches for usable code and information.
  - C It provides support tools for developing code.
  - D It contains various intermediate products.

## Vocabulary

3 Fill in the blanks with the correct words and phrases: *ADL, MIL, domain analysis, VHLL, application generator, intermediate product, transformation system, code scavenging*.

- 1 A(n) \_\_\_\_\_ provides a plan for the architecture of a new piece of software.
- 2 A(n) \_\_\_\_\_ is ready to be used in the development of new software.
- 3 The practice of finding and reusing code is generally known as \_\_\_\_\_.
- 4 \_\_\_\_\_ has a very high level of abstraction.
- 5 \_\_\_\_\_ is a process that identifies reusable code and information.
- 6 A(n) \_\_\_\_\_ provides various support tools for developers.
- 7 A(n) \_\_\_\_\_ helps turn vague plans into detailed systems.
- 8 A(n) \_\_\_\_\_ provides plan for the overall structure of a software system.

**4** Write a word or phrase that is similar in meaning to the underlined part.

- Engineers must complete software templates before they are considered complete programs.       s   t   a
- Application generators often provide a(n) unfinished component for engineers to use.       m   l   e
- Programs in the early stages of development are often unfinished programs.    s   k   t
- A(n) collection of available codes allows engineers to search for usable codes and insert them into new software.       o   r          b   r
- Software that connects applications to operating systems ensures that applications can run simultaneously without causing problems.       d   e   r

**5** Listen and read the textbook excerpt again. How do domain analysis programs work?

**Listening**

**6** Listen to a conversation between a student and an instructor. Mark the following statements as true (T) or false (F).

- \_\_\_ Engineers use transformation systems to save time.
- \_\_\_ According to the woman, transformation systems are unreliable.
- \_\_\_ Middleware is used to translate code into VHLL.

**7** Listen again and complete the conversation.

**Student:** I don't really understand the benefits 1 \_\_\_\_\_. Why can't engineers do that work themselves?

**Instructor:** Transformation systems save time. While the system 2 \_\_\_\_\_, engineers can work on other projects.

**Student:** Aren't they worried that the system 3 \_\_\_\_\_?

**Instructor:** Not really. These systems are usually pretty sophisticated. The 4 \_\_\_\_\_ is more basic than it might sound.

**Student:** Okay. I guess I understand. Can you 5 \_\_\_\_\_ to me?

**Instructor:** Sure. Middleware is go-between software. It 6 \_\_\_\_\_ to an operating system.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

- I don't really understand ...*
- The benefit of ...*
- It seems like ...*

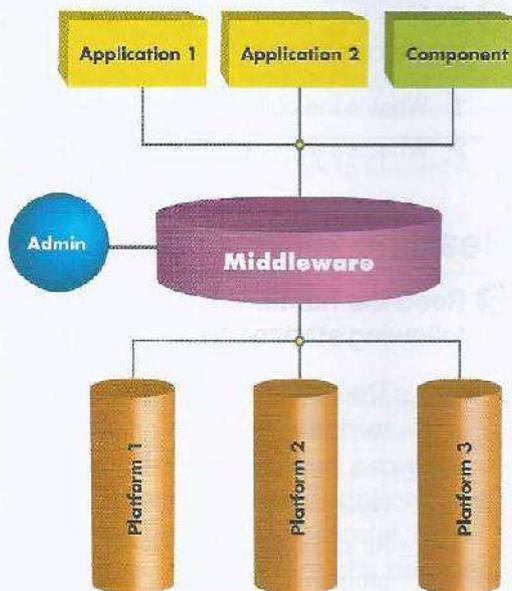
**Student A:** You are a student. Talk to Student B about:

- your questions about software reuse
- the benefits of different types of tools for software reuse

**Student B:** You are an instructor. Talk to Student A about the benefits of tools for software reuse.

**Writing**

**9** Use the textbook excerpt and conversation from Task 8 to write the student's notes on software reuse. Include: two tools that engineers use for software reuse, a description of each, and their benefits.





### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What is the purpose of defensive programming?
- 2 What are some software reliability models?

### Reading

2 Read the handbook. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ The company uses multiple software reliability models.
- 2 \_\_\_ The company attempts to decrease the redundancy of its software.
- 3 \_\_\_ All applications come with built-in N-version programming.

## Defensive Programming

At KernSoft, we are proud of our safe, reliable software. Malfunctioning software can be frustrating and even dangerous for our customers. That's why we use a few **software reliability models** to ensure excellent **reliability**.

**Robust programming** begins in the development stage. In addition to the **standard domain**, we do our best to predict the **exception domain**. We include as many errors in the **expected exception domains** as we can. Our **threshold** for tolerating them is very low.

In case of expected exception domains, we make all of our software **fault-tolerant**. Depending on the type of software, this could mean using a few types of systems. All of our applications come with built-in **recovery blocks**. We sometimes use **N-version programming** to create strong systems, where appropriate. Depending on the software, either **BM** (basic execution time model) or **LPM** (logarithmic Poisson execution time model) is recommended to programmers.

Our techniques increase the **redundancy** of our software. This decreases the **probability** of negative effects from system errors. We require our engineers to craft the strongest software they are capable of. We understand that increasing reliability often means decreasing software productivity somewhat. Although we are committed to creating productive software, we never sacrifice reliability.

### Vocabulary

3 Match the words (1-9) with the definitions (A-I).

- |                             |                       |
|-----------------------------|-----------------------|
| 1 ___ BM                    | 5 ___ recovery block  |
| 2 ___ exception domain      | 6 ___ reliability     |
| 3 ___ LPM                   | 7 ___ standard domain |
| 4 ___ N-version programming | 8 ___ threshold       |
|                             | 9 ___ probability     |

- A a record that can be accessed in case of error
- B a limit that must be exceeded for certain reactions to take place
- C user input that is incorrect
- D a system in which the decrease in failure intensity is constant
- E the quality of being consistent
- F a system in which multiple programs are generated from one specification
- G a system in which the decrease in failure intensity is exponential
- H user input that is correct
- I the likelihood of software errors

**4** Choose the sentence that uses the underlined part correctly.

- 1 A Anticipated errors are in the expected exception domain.  
 B The expected exception domain refers to user input.
- 2 A A common problem in software reliability is redundancy.  
 B Fault-tolerant software minimizes malfunctions.
- 3 A A software reliability model is used to demonstrate the speed of a system.  
 B Defensive programming reduces errors and limits the damage.
- 4 A Redundancy helps to prevent the loss of important information.  
 B Probability indicates the number of errors that have occurred in an application.
- 5 A Defensive programming protects a new system.  
 B Robust programming helps ensure general correctness.
- 6 A A software reliability model intends to limit malfunctions.  
 B Robust programming is in the final stage.

**5** Listen and read the handbook again. What happens when a program is not fault-tolerant?

**Listening**

**6** Listen to a conversation between two software engineers. Choose the correct answers.

- 1 Why does the man prefer BM and LPM?
  - A They have higher exception domain thresholds.
  - B They are easier to use.
  - C They give more accurate error probabilities.
  - D They take less time to run.
- 2 What problem did the woman have with a previous application?
  - A It was not fault-tolerant enough.
  - B It did not give the right error probabilities.
  - C It had a very limited expected exception domain.
  - D It had an inaccurate standard domain.

**7** Listen again and complete the conversation.

**Engineer 2:** I haven't even finished doing the standard and 1 \_\_\_\_\_.

**Engineer 1:** Yeah, but you still need to think about 2 \_\_\_\_\_.

**Engineer 2:** I guess so. What would you recommend I do?

**Engineer 1:** When I 3 \_\_\_\_\_, I like to use BM and LPM.

**Engineer 2:** Why do you prefer those?

**Engineer 1:** I find that they give more 4 \_\_\_\_\_ than other methods.

**Engineer 2:** I didn't know that. I'll keep it in mind when I get 5 \_\_\_\_\_.

**Engineer 1:** You definitely should. Accurate error checks will save you a lot of trouble 6 \_\_\_\_\_.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*You still need to think about ...*  
*Why do you prefer ... / I find ...*

**Student A:** You are an engineer. Talk to Student B about:

- an application that he or she is designing
- how to check the reliability of a program
- what software reliability models you prefer

**Student B:** You are an engineer. Talk to Student A about software reliability models for a new application.

**Writing**

**9** Use the handbook and conversation from Task 8 to write an email to a software engineer about the reliability of a new program. Include: a recommendation for a particular software reliability model, how the model works, and the model's benefits.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are the different parts of CASE?
- 2 What are some different levels of user scale?

## The Development Phase

At ShorSoft, we take great pride in the quality of our **CASE** software. The development phase affects the quality of the final product. That's why we seek to support our engineers in every way possible. We make a wide range of development **tools** available to our engineers. This means that we are equipped to work with any client on any program.

Our engineers use **language-centered environments**, **integrated environments**, and **process-centered environments**. Some projects require tools from just one environment. Most, however, include elements from multiple types of systems. Each engineer has the freedom to choose the best environment for a particular project. We also allow engineers to define their own **process scales** for each project. In every **environment**, engineers find extensive **toolkits** and **workbenches**.

Because our resources are varied, we are able to manage projects of any **user scale**. We also have the network capabilities to support projects on **family**, **city**, and **state** levels. But many of our clients just need resources for small-scale projects. For those, we offer **individual** environments.

The result of this array of options is evident in our reliable software designs. Using the correct tools on a software project is important to us. It ensures that every product we create is the best that it can possibly be.

## Reading

2 Read the webpage. Then, choose the correct answers.

- 1 What is the purpose of the webpage?
  - A to describe the types of environments used at a software company
  - B to compare environments used at two different software companies
  - C to explain the differences between two software environments
  - D to discuss changes in software environments over time
- 2 According to the passage, what is NOT true about the company?
  - A It offers individual environments for small projects.
  - B It allows engineers to define their own process scales.
  - C It includes toolkits and workbenches in every environment.
  - D It assigns each engineer to a particular environment.
- 3 What kind of environments does the company use for small products?
 

A state	C individual
B toolkit	D process centered

## Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- |                                     |                     |
|-------------------------------------|---------------------|
| 1 ___ CASE                          | 4 ___ process scale |
| 2 ___ environment                   | 5 ___ tool          |
| 3 ___ language-centered environment | 6 ___ user scale    |

- A the application of support resources in software development
- B a feature that specifies what a product supports
- C a product that supports a specific task in software development
- D a system that measures the number of users a product can support
- E an application that focuses on the whole process of software development
- F an application that is designed for a specific programming language

**4** Read the sentence pairs. Choose where the words best fit the blanks.

**1 city / family**

- A A \_\_\_\_\_ environment could be a small corporation or development team.
- B A \_\_\_\_\_ environment is a value that encourages large-scale cooperation.

**2 individual / state**

- A A(n) \_\_\_\_\_ environment indicates that a system is very large.
- B A(n) \_\_\_\_\_ environment indicates that a product assists in independent software construction.

**3 integrated environment / process-centered environment**

- A A(n) \_\_\_\_\_ contains data on the final product.
- B A(n) \_\_\_\_\_ shares a description of the development steps.

**4 toolkit / workbench**

- A A \_\_\_\_\_ contains a group of tools that work in limited scope.
- B A \_\_\_\_\_ contains tools that are not well integrated.

**5** Listen and read the webpage again. What types of environments does the company use?

**Listening**

**6** Listen to a conversation between two engineers. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The man is likely to select a process-centered environment.
- 2 \_\_\_ The man will use city level for the project.
- 3 \_\_\_ In a few weeks, the woman will decide on the environment.

**7** Listen again and complete the conversation.

**Engineer 1:** Great! I need to talk to you about **1** \_\_\_\_\_.

**Engineer 2:** I think it's **2** \_\_\_\_\_ pretty big.

**Engineer 1:** Do you think it will be **3** \_\_\_\_\_?

**Engineer 2:** Actually, I'm looking at city level. If we're going to finish on time, we'll **4** \_\_\_\_\_.

**Engineer 1:** Wow, I guess it's even bigger than I thought it would be.

**Engineer 2:** Yeah. Anyway, you might want to start **5** \_\_\_\_\_.

**Engineer 1:** Great. I'll do that right away.

**Engineer 2:** What **6** \_\_\_\_\_ do you think is best?

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I need to talk to you about ...*  
*I'll do that ...*  
*How soon do you think ...*

**Student A:** You are an engineer. Talk to Student B about:

- the user scale for a project
- the type of environment that is right for a project
- when certain tasks will be completed

**Student B:** You are an engineer. Talk to Student A about the user scale and environments for a project.

**Writing**

**9** Use the webpage and conversation from Task 8 to write a memo to a software development team. Include: what type of CASE software to use, the features of the software, and why those features will be useful for the particular project.

## ShorSoft misses the mark with new CASE software

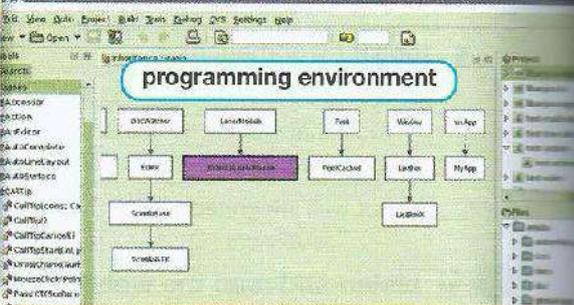
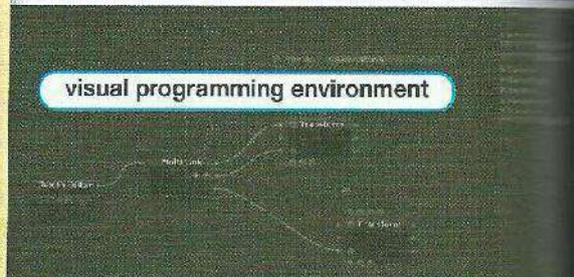
by Gregory Fairchild

Earlier this month, ShorSoft put out a new iteration of their popular line of CASE software. Unfortunately, the improvements are disappointing. When they heard about the upgrade, users expected a number of changes. However, very few of these changes were actually included.

The biggest problem is that the **IPSE** (integrated project support environment) still forces engineers to use **reserved checkout**. The issues with this system are obvious. It slows development and frustrates users. Why doesn't ShorSoft implement simple **unreserved checkout**? Even the most basic **UNIX** software comes standard with that option.

Apart from that drawback, the new version does have a few nice upgrades. An **MWB** (management workbench) environment was added. ShorSoft kept the standard **AWB** (analyst workbench) and **PWB** (programmer workbench) environments. However, their graphical designs were improved significantly. This makes them much easier to use now. The **PCTE** (Portable Common Tool Environment) and **SCCS** (Source Code Control System) saw some improvements. Still, the overall **visual programming environment** of the software should be much easier to navigate.

It seems that this software may not be able to keep up with the more sophisticated **programming environments**. A number of new programs are appearing on the market. Many of these are more user-friendly than ShorSoft's CASE software. If ShorSoft doesn't improve its **back-end** development, it will lose its customers.



### Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some common features of programming environments?
- 2 What are some different types of workbenches found in integrated environments?

### Reading

2 Read the review. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ The new IPSE features a new type of checkout.
- 2 \_\_\_ Basic UNIX software comes standard with unreserved checkout.
- 3 \_\_\_ The company implemented new AWBs and PWBs.

### Vocabulary

3 Match the words (1-6) with the definitions (A-F).

- |           |            |            |
|-----------|------------|------------|
| 1 ___ AWB | 3 ___ IPSE | 5 ___ PWB  |
| 2 ___ MWB | 4 ___ PCTE | 6 ___ SCCS |

- A an environment that assists the manager during planning and control stages of development
- B an environment that supports activities in the early stages of software development
- C an environment that supports the implementation and test phases of software development
- D an environment that supports all phases of the development process
- E an environment that supports engineers in the development phase
- F an environment that allows the user to keep track of changes in files

- 4 Fill in the blanks with the correct words and phrases from the word bank.

**word BANK**

back-end programming environment  
reserved checkout unreserved checkout  
visual programming environment UNIX

- 1 \_\_\_\_\_ allows only one person at a time to edit a file.
- 2 \_\_\_\_\_ activities may include testing and implementation.
- 3 A(n) \_\_\_\_\_ helps engineers develop, change, and test programs.
- 4 \_\_\_\_\_ is a general support environment.
- 5 \_\_\_\_\_ allows multiple users to edit the same file at the same time.
- 6 A(n) \_\_\_\_\_ provides graphical or iconic elements which can be manipulated by the user in an interactive way.

- 5 Listen and read the review again. What improvements were made to the IPSE?

**Listening**

- 6 Listen to a conversation between a software engineer and the owner of a software company. Then, choose the correct answers.

- 1 Why is a new UNIX program necessary?
  - A It will allow the company to hire fewer engineers.
  - B It will eliminate errors in existing programs.
  - C It will allow the company to produce software at a competitive rate.
  - D It will reduce design costs on future projects.
- 2 What is the owner most likely to do next?
  - A place an order for the new software
  - B tell the engineering staff that the new software is not available
  - C determine if there is enough money for the new software
  - D stop the current project until the company upgrades the software

- 7 Listen again and complete the conversation.

**Engineer:** Can I talk to you for a minute? I have a request from 1 \_\_\_\_\_.

**Owner:** Sure. What's up?

**Engineer:** It involves 2 \_\_\_\_\_ that we are currently using.

**Owner:** Is it malfunctioning?

**Engineer:** Not exactly. It is kind of outdated though. We'd like to get 3 \_\_\_\_\_.

**Owner:** How is the UNIX program going to help?

**Engineer:** Well, one of our problems is 4 \_\_\_\_\_. Right now, only one engineer at a time can work 5 \_\_\_\_\_. With unreserved checkout, several engineers could work at once.

**Owner:** I see. Are there other benefits?

**Engineer:** Yes. The UNIX program comes with an MWB. This would help out 6 \_\_\_\_\_.

**Speaking**

- 8 With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I have a request ... / We'd like to ...  
Is ... really necessary?*

**Student A:** You are an engineer. Talk to Student B about:

- a proposed upgrade to your company's CASE
- the problems with the current system
- the benefits of a new system

**Student B:** You are the owner of a software company. Talk to Student A about upgrading your CASE software.

**Writing**

- 9 Use the review and conversation from Task 8 to write a review of a new CASE software. Include: the features of the new software, how it compares to other CASE software that is available.

**From:** Lhawkins@shorsoft.net  
**To:** <all staff>  
**Subject:** Flaw Discovery and Management

Good Morning Team Members,

We found a pretty major **flaw** in the NSR project. Our analysis identified the extent of the problem and we're ready to attack it. Please read this email carefully and follow the **workflow**.

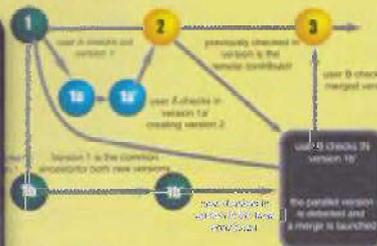
Allen, I need you to **retrace** the **deltas** over the past week. Once you find the flawed pieces, make note of the **configuration items**. These will need to be changed in the **baseline**. Submit a **change request** to the **CCB**. It's important that we send this as soon as possible. We can't eliminate the problem until they **approve** all of the changes.

In the meantime, Mary and Robert will continue **parallel development** for the next phases. I don't see any reason to stop progress in those areas. Later, we will **incorporate** the **corresponding** fixes to your branch.

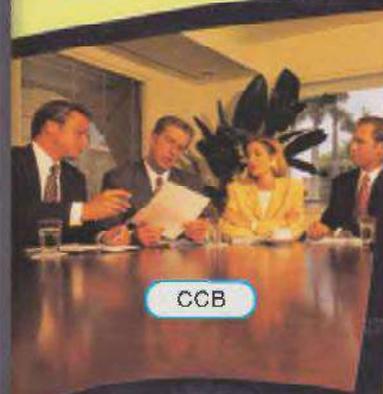
As a reminder to all employees, don't forget about the changes to our **configuration management system**. We are no longer using **version-oriented** models. Instead, please only use **change-oriented** configurations from now on. This makes it much easier to identify the difference between particular software versions.

Thanks for your cooperation. With everyone's help, I know we can eliminate these problems quickly and efficiently.

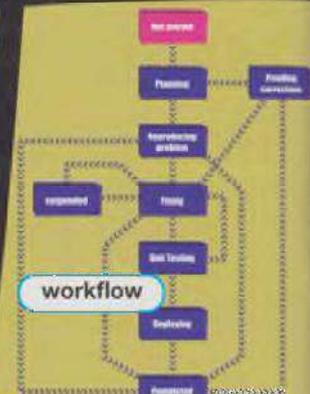
-Tyler Hawkins  
 Project Manager



parallel development



CCB



workflow

## Vocabulary

3 Write a word or phrase that is similar in meaning to the underlined part.

- The baseline should contain a complete list of hardware or software that aids in configuration management.  
 \_ \_ n \_ g \_ a \_ \_ \_ \_ i \_ e \_ \_
- The staff that manages changes made to the baseline can be made up of investors, clients, or project managers. \_ \_ \_ \_
- If a change is made to a piece of software, it should also be made to pieces that are similar or identical in character.  
 \_ \_ r \_ e \_ o \_ i \_ \_
- The practice of creating different branches of development makes it easier for many engineers to work on the same project.  
 p \_ \_ \_ \_ \_ e \_ \_ \_ \_ m \_ \_ \_
- Engineers should wait for the CCB to officially accept proposed changes before implementing them. \_ \_ \_ \_ r \_ v \_ \_
- A proposed adjustment or correction to the baseline should only be made if absolutely necessary. \_ \_ h \_ \_ g \_ \_ \_ q \_ \_ s \_
- Systematic monitoring of changes and updates aims to eliminate software flaws and make it easier for many engineers to work on one project.  
 \_ \_ n f \_ \_ \_ \_ a t \_ \_ \_ \_ \_ \_ \_ \_ g \_ \_ e \_ t

## Get ready!

1 Before you read the passage, talk about these questions.

- How does configuration management protect against flaws?
- What are some different types of configuration management?

## Reading

2 Read the email. Then, mark the following statements as true (T) or false (F).

- The team is waiting for CCB to approve parallel development.
- Work on the next phases will continue while the flaw is being identified.
- The company is no longer using version-oriented models.

**4** Read the sentence pairs. Choose where the words best fit the blanks.

**1** delta / flaw

- A If a \_\_\_\_\_ is detected, it should be edited or eliminated.  
 B A change to a piece of software is saved as a \_\_\_\_\_.

**2** incorporate / retrace

- A Engineers sometimes must \_\_\_\_\_ new items into a baseline.  
 B Engineers can \_\_\_\_\_ their steps to find the origin of problems.

**3** change-oriented / version-oriented

- A \_\_\_\_\_ models use descriptions of changes made to the baseline.  
 B \_\_\_\_\_ models use numbers to keep track of changes.

**4** baseline / workflow

- A The project's \_\_\_\_\_ needs to follow an established order.  
 B Changes to a \_\_\_\_\_ must be approved by the CCB.

**5** Listen and read the email again. What step is each team member instructed to take?

**Listening**

**6** Listen to a conversation between a project manager and an engineer. Then, choose the correct answers.

- 1** According to the woman, what step is time-consuming?  
 A retracing her steps  
 B waiting for approval to change the baseline  
 C double checking her work  
 D merging the fixed version with the updated version
- 2** What does the woman need before she can continue working?  
 A information on the location of the flaw  
 B a list of all the program deltas  
 C the parallel development documents  
 D a report on the results of retracing

**7** Listen again and complete the conversation.

**Manager:** Great. Did you send a request to the CCB about 1 \_\_\_\_\_ ?  
**Engineer:** The request is already sent. I'm just waiting for them 2 \_\_\_\_\_ .  
**Manager:** Great. You're really on top of things!  
**Engineer:** Well, there are still a few steps 3 \_\_\_\_\_ is fixed.  
**Manager:** Like what?  
**Engineer:** First, I need to get the latest copy 4 \_\_\_\_\_ documents.  
**Manager:** That shouldn't take too long. What else do you have to do?  
**Engineer:** Well, then I need to 5 \_\_\_\_\_ with the updated version.  
**Manager:** Oh. That sounds like it could be time-consuming.  
**Engineer:** It will be. But I should be able to finish it by the 6 \_\_\_\_\_ .

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

Can you give me an update on ...  
 First, I need to ... / Then, I ...

**Student A:** You are a project manager. Talk to Student B about:

- a flaw in a software development project
- what he or she has done to fix it
- what he or she will do next to fix it

**Student B:** You are a software engineer. Talk to Student A about what you are doing to fix a flaw in a software development project.

**Writing**

**9** Use the email and conversation from Task 8 to write a memo to a software development team. Include: a flaw in a development project, what steps are required to fix the flaw, and who should perform each task.

Dear Mr. Sherman,

I have some concerns about our team organization at StorSoft. We've had an **open structured team** since this company's inception. However, the company is growing, and I think we need a new organization method. Our current system focuses too heavily on **relation directedness**. I think we need to choose a structure that is more focused on **task directedness**. This will increase our productivity and minimize personal conflicts in the office.

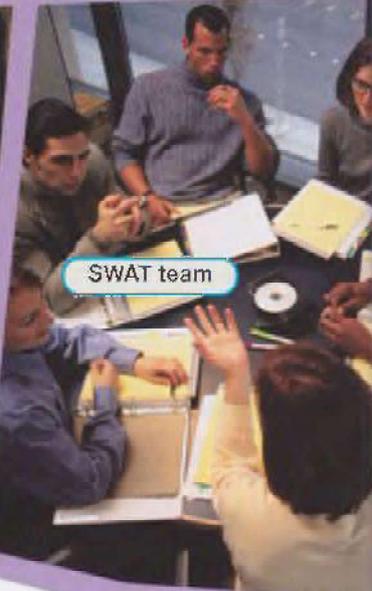
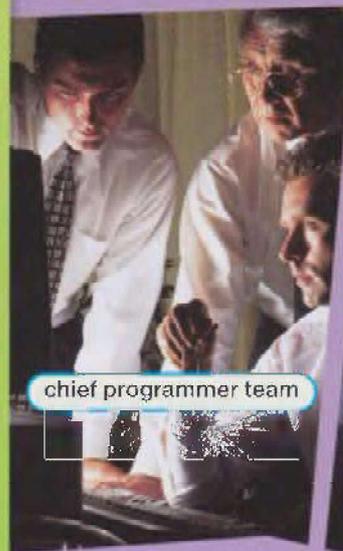
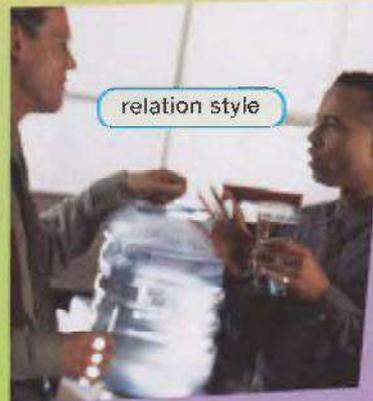
A **hierarchical organization** could add the extra structure that we need. We need managers to hold employees accountable for doing their jobs. I don't think that our current **matrix organization** offers this possibility.

I previously worked at another corporation that organized employees into **SWAT teams**. These were **units** of employees who **specialized** in similar areas. Units like this can fit into any number of organizational structures. At my last job, we used **commitment style**. But **integration style** or **relation style** would also work for this company.

Another option is **chief programmer teams**. This would also divide employees into units. However, these units would each work on separate projects. I like this system because it provides a clear leadership structure. This would work well with **separation style** organization.

Please consider these management structures. I think a change would really improve our efficiency.

Sincerely,  
Linda Dunn  
Software Engineering Manager



## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What are some different management styles?
- 2 What are some different team organization styles?

## Reading

2 Read the letter. Then, mark the following statements as true (T) or false (F).

- 1 \_\_\_ The company currently has an open structured team.
- 2 \_\_\_ The writer of the letter previously worked at a corporation that used relation style.
- 3 \_\_\_ The letter recommends choosing either separation style or chief programmer teams.

## Vocabulary

2 Match the phrases (1-8) with the definitions (A-H).

- 1 \_\_\_ commitment style
- 2 \_\_\_ integration style
- 3 \_\_\_ relation style
- 4 \_\_\_ separation style
- 5 \_\_\_ task directedness
- 6 \_\_\_ hierarchical organization
- 7 \_\_\_ matrix organization
- 8 \_\_\_ relation directedness

- A a style that focuses on efficiency
- B a style that relies on motivation
- C a style that designates different levels of management
- D a style in which project goals guide decisions
- E a style that organizes employees into units
- F a style that focuses on strategies for achieving tasks
- G a style that focuses on individual employees and relationships
- H a style that features informal decision making

**4** Fill in the blanks with the correct words and phrases: *SWAT Team, chief programmer team, open structured team, specialize in, unit.*

- 1 A(n) \_\_\_\_\_ is made up of three people.
- 2 A(n) \_\_\_\_\_ focuses on task and relation directedness.
- 3 Some programmers \_\_\_\_\_ specific parts of development.
- 4 A(n) \_\_\_\_\_ combines open management with a decision making process.
- 5 It is a manager's job to decide which \_\_\_\_\_ each employee will be in.

**5** Listen and read the letter again. What management styles does the woman recommend?

### Listening

**6** Listen to a conversation between a company owner and a manager. Choose the correct answers.

- 1 What is the purpose of the conversation?
  - A to assign employees to particular teams for a software project
  - B to discuss employees' complaints about an organizational structure
  - C to explain which project each team should work on
  - D to compare different ways of dividing the staff for a new project
- 2 According to the woman, what is the benefit of SWAT teams?
  - A They provide employees with a clear manager.
  - B They give employees more time to complete tasks.
  - C They assign tasks to distinct units of employees.
  - D They encourage employee communication.

**7** Listen again and complete the conversation.

**Manager:** I did. I think it will be most efficient to 1 \_\_\_\_\_ into SWAT teams.

**Owner:** What benefit will that offer us?

**Manager:** With SWAT teams, we can assign specific tasks to 2 \_\_\_\_\_. Each unit works independently on a particular task.

**Owner:** That sounds good. But I'm worried that the units won't communicate with each other enough.

**Manager:** Hmm, that's a good point. We could 3 \_\_\_\_\_ structure instead.

**Owner:** I think that might be a better idea.

**Manager:** Maybe. But I'm not sure if it would be 4 \_\_\_\_\_.

**Owner:** You're right. We need a 5 \_\_\_\_\_ units. But we also need strong management.

**Manager:** We could 6 \_\_\_\_\_ teams.

### Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

#### USE LANGUAGE SUCH AS:

*Do you have any ideas for ...*  
*What benefit would that ...*  
*We could use ...*

**Student A:** You are a company owner. Talk to Student B about:

- the team organization for a new project
- the advantages and disadvantages of different organization styles
- your concerns.

**Student B:** You are a manager. Talk to Student A about the team organization for a new project.

### Writing

**9** Use the letter and conversation from Task 8 to write a recommendation for a particular management style. Include: the structure of the style, the advantages of the style, and possible disadvantages of the style.

# 10 Software Quality Control

## Quality Control Report

**Company:** ShorSoft Software Engineering and Development

**Assessor:** Jason Edwards, Quality Control Analyst

I completed an audit of ShorSoft's **quality control** during the week of April 10. The company attempts to **conform to the IEEE Standard for Quality Assurance Plans**. However, I identified many areas in which their system is lacking.

Employees at the company are following the directions in the **CMM**. Nonetheless, there are still some problems with the implementation of the system. In most cases, the **key process areas** are well defined. However, the **key practices** aren't as useful as they should be. Employees need to assess the efficiency of their actions carefully.

Another way to **improve** the system is to focus less on **quality factors**. The emphasis should be on **quality criteria** instead. This will allow employees to measure their successes more easily. I also recommend that employees review **common features**. The audit found that the **maturity levels** of the projects were not clearly defined. A better understanding of common features would prevent this problem.

Currently, ShorSoft meets almost none of the standards listed in the **ISO 9001**. However, the company should meet those standards fairly easily. I recommend restructuring quality control practices to support renewed commitment to **TQM**.

### Get ready!

1 Before you read the passage, talk about these questions.

- How do software companies maintain quality control?
- What are some different sets of guidelines for software quality control?

### Reading

2 Read the report. Then, mark the following statements as true (T) or false (F).

- The company's key process areas were clearly defined.
- The review recommends that the company focus on quality factors.
- The company already meets most of the industry standards for quality.

### Vocabulary

3 Write a word that is similar in meaning to the underlined part.

1 The set of procedures for maintaining quality in software systems ensures that software is all up to the same standard.

l \_ \_ \_ \_ a \_ \_ \_ \_  
\_ u \_ \_ l \_ \_ \_ s \_ r \_ \_ c \_ \_ \_ a \_ \_

2 A system of maintaining standards is important for every stage of the development process.

\_ \_ \_ \_ i \_ \_ y \_ o \_ \_ t \_ \_ \_

3 It is important that engineers pay attention to each activity that aids the implementation of a key process area in order to create quality software.

\_ \_ y \_ \_ a \_ \_ i \_ \_ c \_

4 Managers should encourage their employees to practice the pursuit of excellence in every step of a process.

\_ \_ Q \_

5 Engineers must be aware of the list of issues that must be addressed at each maturity level.

k \_ \_ \_ r \_ \_ o \_ \_ a \_ \_ o \_

**4** Read the sentence pairs. Choose where the words best fit the blanks.

**1 Quality factors / Quality criteria**

- A \_\_\_\_\_ cannot be measured directly.  
 B \_\_\_\_\_ can be measured subjectively or objectively.

**2 CMM / ISO 9001**

- A The \_\_\_\_\_ states the general requirements for a system.  
 B The \_\_\_\_\_ is aimed at improving the development process.

**3 conform to / improve**

- A Companies need to \_\_\_\_\_ regulations to assure customers of software quality.  
 B Managers should encourage their engineers to continually \_\_\_\_\_ their development techniques.

**4 common feature / maturity level**

- A A \_\_\_\_\_ makes up the key practices.  
 B A \_\_\_\_\_ measures whether a software process achieves a particular standard.

**5** Listen and read the review again. How can the company improve its quality control?

**Listening**

**6** Listen to a conversation between two engineers. Choose the correct answers.

- 1 What problem have the engineers noticed?  
 A inconsistencies creating mature software  
 B violations of the ISO 9001 standards  
 C problems meeting the key practices  
 D misunderstandings about the CMM
- 2 What does the woman think is the benefit of quality criteria?  
 A They allow engineers to measure their improvement.  
 B They are easier to understand than quality factors.  
 C They are more likely to produce mature software.  
 D They are more cost effective to analyze than other features.

**7** Listen again and complete the conversation.

**Engineer 1:** I know. I was really surprised! I felt we did a good job 1 \_\_\_\_\_.

**Engineer 2:** Really? I expected them to 2 \_\_\_\_\_. My department's having trouble meeting key practices.

**Engineer 1:** Now that you mention it, I guess that's a problem in my department too. I think we'll be okay 3 \_\_\_\_\_ the key process areas.

**Engineer 1:** But I didn't 4 \_\_\_\_\_ problems. Did you?

**Engineer 2:** A few. I think the company needs to stop focusing so much on 5 \_\_\_\_\_.

**Engineer 1:** Really? Why?

**Engineer 2:** Because it's hard for a lot of engineers to work that way. It's better to think 6 \_\_\_\_\_.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*I was surprised because ...*  
*Now that you mention it ...*  
*It's better to ...*

**Student A:** You are an engineer. Talk to Student B about:

- the result of a quality control audit
- changes you think the company should make
- changes you expect management to make

**Student B:** You are an engineer. Talk to Student A about quality control at your company.

**Writing**

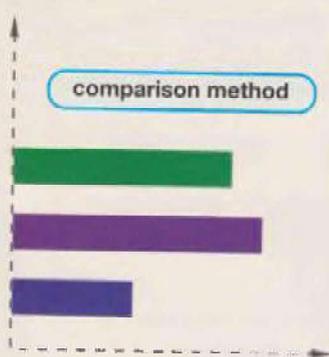
**9** Use the report and conversation from Task 8 to write suggestions for improving a company's quality control. Include: tasks employees should perform, methods the company can use, and guidelines the company should follow.

WEEKLY TIMESHEET		
	Mon	Tu
Start Time	9 AM	9
End Time	5 PM	6
Reg Time	8-	
Over Time		

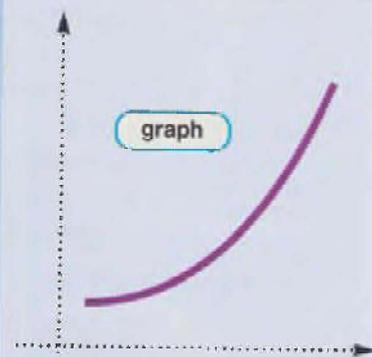
man-month



budget



comparison method



graph

To: Greg Ferguson, Senior Manager  
 From: Leslie Alvarez, Project Manager  
 Re: Budget Report

Greg,

We significantly exceeded our **budget** on our last project. Our analysts are currently investigating the reasons for this. So far, we believe that the primary problem was in our estimation method.

We used a **comparison method** for deciding our budget for this project. Though this method is sometimes valid, we applied it too hastily. The **KLOC** of this project was much greater than the KLOC of the previous project. As a result, our estimates were far too low for a project of this size. Also, we overestimated the **learning effect** for the project. This caused us to exceed our allotted **man-months** and subsequently cost us more money.

For our next project, we intend to use a different budgeting system. An **algorithmic model** will offer us more precision in our calculations. We will not use the **Watson-Felix** method. Its **base formula** is known to be unreliable. We have also ruled out **COCOMO** because it is more complex than our project requires. We are considering either the **Delphi-method** or the **Putnam model**. We will also be less **optimistic** when we **estimate** the total **development time** needed.

With these precautions, we hope to create more accurate budgets on all future projects.

Leslie

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What factors must be taken into account in a budget for a software project?
- 2 What are some different methods for calculating a budget?

## Reading

2 Read the memo. Then, mark the following statements as true (T) or false (F).

- 1  The company got its best results from the comparison method.
- 2  The company used more man-months than the budget allowed.
- 3  According to the memo, an algorithmic model will be more effective.

## Vocabulary

3 Match the words (1-10) with the definitions (A-J).

- |  |  |
|--|--|
| 1 <input type="checkbox"/> comparison method | 6 <input type="checkbox"/> algorithmic model |
| 2 <input type="checkbox"/> Delphi-method     | 7 <input type="checkbox"/> COCOMO            |
| 3 <input type="checkbox"/> Putnam model      | 8 <input type="checkbox"/> development time  |
| 4 <input type="checkbox"/> Watson-Felix      | 9 <input type="checkbox"/> learning effect   |
| 5 <input type="checkbox"/> estimate          | 10 <input type="checkbox"/> optimistic       |

- A a system of cost estimation that involves a panel of experts
- B a system of cost estimation that uses previous projects to determine costs for new projects
- C a system of cost estimation that uses a specific formula
- D a system of cost estimation that divides projects into three classes
- E a system of cost estimation that considers 29 variables
- F the period between the beginning and end of a software project
- G a theory about increasing rates of productivity
- H a system of cost estimation that considers problems and solutions
- I assuming the best possible situation
- J to determine the likely amount or cost of something

**4** Read the sentence pairs. Choose where the words best fit the blanks.

- 1 **base formula / man-month**
- A A \_\_\_\_\_ calculates the effort required to create new software.
- B A \_\_\_\_\_ determines how much work can be done in a particular amount of time.
- 2 **budget / KLOC**
- A A source code is measured in \_\_\_\_\_.
- B A \_\_\_\_\_ is an estimation of expenditures that should be as accurate as possible.

**5** Listen and read the memo again. Why was the budget for the last project inaccurate?

## Listening

**6** Listen to a conversation between a company owner and a project manager. Choose the correct answers.

- 1 Why is the project slightly behind schedule?
- A the manager was too optimistic about the learning effect
- B incorrect information was used to estimate man-months
- C the project's KLOC was not calculated accurately
- D the managers did not use an appropriate estimate model
- 2 According to the man, what is true about the overall budget estimate?
- A It is not too optimistic.
- B It is based on previous projects.
- C It was approved by the owner.
- D It adequately accounted for the learning effect.

**7** Listen again and complete the conversation.

**Manager:** I have the latest 1 \_\_\_\_\_ for the accounting software project.

**Owner:** Great. I'd love to hear them. How do they look?

**Manager:** It's hard to say. The reports are a little different depending 2 \_\_\_\_\_ you look at.

**Owner:** Give me the 3 \_\_\_\_\_.

**Manager:** Okay. Fortunately, 4 \_\_\_\_\_ we are right on schedule.

**Owner:** That's great! What about if you look at the Delphi-method?

**Manager:** According to the Delphi-method we are just 5 \_\_\_\_\_.

**Owner:** That's not too bad. Why are we behind schedule?

**Manager:** We estimated some software sizes incorrectly. Some 6 \_\_\_\_\_ than we expected.

## Speaking

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*The good news is ...*  
*What about ...*  
*I can't afford ...*

**Student A:** You are a project manager. Talk to Student B about:

- the progress of a software project
- how well the project matches its budget
- changes needed on the project

**Student B:** You are a company owner. Talk to Student A about the budget for a software project.

## Writing

**9** Use the memo and conversation from Task 8 to write a memo about a budget method. Include: how the method works, the advantages of the method, and the disadvantages of the method.

# 12 Project Management

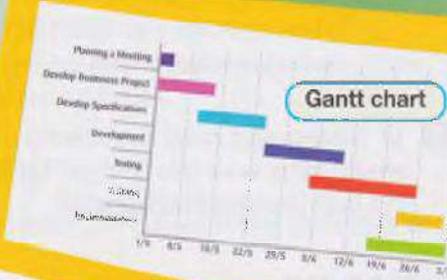
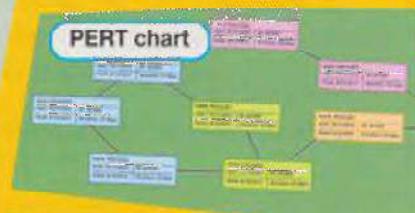
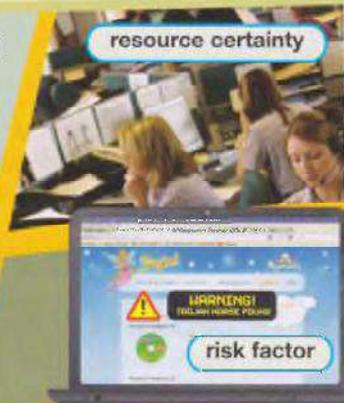
**Sign up TODAY** for the **Project Management Workshop!**

This three-part workshop will address different areas of project management. The first part of the workshop is from 10:00 to 11:00. It covers everything you need to know about **degree of certainty**, **Product certainty**, **process certainty**, and **resource certainty** will be presented as ways to measure this important metric.

Part two of the workshop is from 11:00 to 11:30. This section focuses on a crucial part of any project: **risk management**. Participants will learn how to identify and address **risk factors**.

Finally, part three of the workshop is from 11:30 to 12:00. This is a general overview of the various methods of organizing project tasks. For example, a **WBS** is a helpful tool for any project manager. It provides a view of the entire scope of a project. A **PERT chart** and its **critical paths** are used to coordinate tasks. Finally, the class will look at the value of maintaining schedules with **Gantt charts**.

The course also covers the four control situations that are affected by certainty. Students will learn how to identify simple situations like **realization problems** and **allocation problems**. Then, they will examine more challenging situations, such as **design problems** and **exploration problems**.



## Get ready!

- Before you read the passage, talk about these questions.
  - What role do risks play in project management?
  - What tools help software engineers organize projects?

## Reading

- Read the advertisement. Then, choose the correct answers.
  - What is the flyer mostly about?
    - different types of development projects
    - ways to approach projects
    - methods for minimizing risk factors on projects
    - problems that are likely to arise on a project
  - What will students learn in part one of the workshop?
    - how to gain resources
    - how to handle risk factors
    - how to organize projects
    - how to measure degrees of certainty
  - Which of the following is NOT used to organize project tasks?
 

A WBS	C critical paths
B PERT chart	D risk management chart

## Vocabulary

- Match the words (1-8) with the definitions (A-H).
 

1 ___ critical path	5 ___ degree of certainty
2 ___ risk factor	6 ___ resource certainty
3 ___ WBS	7 ___ exploration problem
4 ___ design problem	8 ___ risk management

  - a scale that measures the dependability of user requirements and resources
  - a metric that is determined by the availability of supplies
  - a situation in which the steps to completing a project are unknown
  - a situation in which a project's overall degree of certainty is low
  - a process that identifies potential problems and prevents them from becoming setbacks
  - a characteristic that increases the possibility of problems
  - a decomposition of a project into smaller groups to view the overall project
  - a part of a PERT chart that identifies when tasks must be completed

4 Read the sentence pairs. Choose where the words best fit the blanks.

1 Gantt chart / PERT chart

- A A \_\_\_\_\_ uses bars to indicate the timing of a project.
- B Have you organized the tasks into a \_\_\_\_\_ yet?

2 product certainty / process certainty

- A The functionality and quality of user requirements influences \_\_\_\_\_.
- B \_\_\_\_\_ measures the stage a project is in.

3 realization problem / allocation problem

- A Since the company does not have enough employees, it has a(n) \_\_\_\_\_.
- B All the user requirements are stable, so we need to focus on the \_\_\_\_\_.

5 Listen and read the advertisement again. What are some ways that the degree of certainty is measured?

Listening

6 Listen to a conversation between two software engineers. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The man and the woman attended the workshop together.
- 2 \_\_\_ The man used risk management methods on a recent project.
- 3 \_\_\_ Gantt charts were not covered in the workshop.

7 Listen again and complete the conversation.

Engineer 2: I was at the 1 \_\_\_\_\_ . It was interesting.

Engineer 1: I didn't sign up for it. Was there any good information?

Engineer 2: Overall, it was really informative. They discussed three main topics in project management. I learned a lot about 2 \_\_\_\_\_ .

Engineer 1: Interesting. What else did they discuss?

Engineer 2: They also talked about risk management. I didn't realize how important it actually is to 3 \_\_\_\_\_ .

Engineer 1: I hadn't really thought about it either. 4 \_\_\_\_\_ any tips for organization?

Engineer 2: Yeah, that was the 5 \_\_\_\_\_ they addressed. I already knew about a lot of them, though.

Engineer 1: Which methods did they discuss? I'm about to run my first project, and I'm wondering what might be useful.

Engineer 2: Well, they talked about WBS, which I use all the time. They also 6 \_\_\_\_\_ .

Speaking

8 With a partner, act out the roles below based on Task 7. Then, switch roles.

USE LANGUAGE SUCH AS:

*What else did they ... / They discussed ...*  
*Overall, I thought ...*

**Student A:** You are an engineer. Talk to Student B about:

- a workshop on project management
- the topics discussed
- what you thought about it

**Student B:** You are an engineer. Talk to Student A about a workshop on project management.

Writing

9 Use the advertisement and conversation from Task 8 to write an email to a coworker about a workshop on project management. Include: what the workshop covered, what you learned, and what you already knew.

## Get ready!

1 Before you read the passage, talk about these questions.

- 1 What is a code of ethics?
- 2 Why is it important for software engineers to maintain integrity in their work?

## QRC Software Company Policy

The employees at QRC Software are expected to abide by a code of **ethics**. Our code includes several important **principles**. These rules guide interactions within the company and with the public. Following these principles improves the **welfare** of the company, the employees, and the clients.

The company strives to deliver excellent products to clients. Software engineers **ensure** that products meet the highest **standards** of excellence. Software should not be released to the public until it has been through **adequate** testing.

Employees maintain the value of **integrity** by being honest in their work. As experts, engineers are expected to use reasonable **professional judgment**. They must NEVER be **deceptive** about software performance to meet a project deadline.

Engineers should act according to the **best interests** of the client whenever possible. However, they must not violate the **public interest** in the process.

The **health** and **safety** of QRC employees and the public are important. We will only approve software that is deemed safe and useful.

Dishonest behavior and actions contrary to these principles are considered **unethical**. Employees in violation of this code are subject to disciplinary action.



## Reading

2 Read the poster. Then, choose the correct answers.

- 1 What is the flyer mostly about?
  - A consequences of ethics violations at a company
  - B changes to a company's ethics policy
  - C an inquiry into a company's ethical performance
  - D the principles of a company's ethics code
- 2 According to the company, what is true about the public interest?
  - A It is directly related to the health and safety of the clients.
  - B It is more important than the client's best interests.
  - C It is sometimes unethical to act according to public interest.
  - D It is often difficult to establish.
- 3 What is NOT a requirement for releasing software?
  - A It meets standards of excellence.
  - B It goes through adequate testing.
  - C It serves employees' best interests.
  - D It is deemed safe and useful.

## Vocabulary

3 Match the words (1-8) with the definitions (A-H).

- |                 |                             |
|-----------------|-----------------------------|
| 1 ___ ethics    | 5 ___ adequate              |
| 2 ___ principle | 6 ___ professional judgment |
| 3 ___ welfare   | 7 ___ public interest       |
| 4 ___ ensure    | 8 ___ best interests        |

- A the overall state of a person or group's health and happiness
- B being enough or acceptable
- C the ability to make good decisions based upon work experience
- D to be certain that something will happen
- E circumstances that are advantageous for a specific person
- F the well-being of a large group of people
- G a rule that guides the process of decision-making
- H a set of morals that governs the actions of an individual or a group

4 Choose the sentence that uses the underlined part correctly.

- 1 A The health of a person includes physical, mental, and emotional aspects.  
B Employees are expected to prevent the safety of clients.
- 2 A Deceptive behavior is unacceptable.  
B Engineers must always be unethical.
- 3 A Employees displaying integrity will be subject to disciplinary actions.  
B The company we hired has very high standards for their employees.

5 Listen and read the poster again. What behaviors might be considered ethical?

### Listening

6 Listen to a conversation between a job interviewer and an engineer. Mark the following statements as true (T) or false (F).

- 1  The woman outlines the company's policy on unethical behavior.
- 2  The man refused to release new software early.
- 3  The man lost his previous job over an ethical disagreement.

7 Listen again and complete the conversation.

**Interviewer:** I'm glad to hear you say that. Can you tell me about a time when you had to 1 \_\_\_\_\_ ?

**Engineer:** At my last job, my manager really pushed me to 2 \_\_\_\_\_. But the software wasn't ready for public use yet.

**Interviewer:** What was wrong with it?

**Engineer:** It was designed to protect private records. But it 3 \_\_\_\_\_ to ensure security.

**Interviewer:** So what did you do?

**Engineer:** Well, I wanted to 4 \_\_\_\_\_ of the public. I refused to deliver the software without adequate testing.

**Interviewer:** That was very bold. Why did you 5 \_\_\_\_\_ ?

**Engineer:** For one thing, I have 6 \_\_\_\_\_ that I produce.

### Speaking

4 With a partner, act out the roles below based on Task 7. Then, switch roles.

USE LANGUAGE SUCH AS:

- Can you tell me ...
- One time ...
- I want to emphasize ...

**Student A:** You are a job interviewer. Talk to Student B about:

- his or her personal code of ethics
- a time he or she defended ethical principles
- why he or she made a particular decision

**Student B:** You are an engineer. Talk to Student A about a time you defended your ethical principles.

### Writing

5 Use the poster and conversation from Task 8 to write a cover letter to a potential employer. Include: your personal code of ethics, why they are important, and the result of not following ethical principles.



# 14 Cloud Computing: SaaS and PaaS

## Get ready!

- 1 Before you read the passage, talk about these questions.
  - 1 What are some ways that software is distributed?
  - 2 How do users pay for cloud computing services?

## Reading

- 2 Read the journal article. Then, choose the correct answers.
  - 1 What is the main idea of the article?
    - A the challenges of developing cloud computing software
    - B recent changes in cloud computing technology
    - C the history of the development of cloud computing
    - D an overview of cloud computing models
  - 2 What is true of SaaS users?
    - A They access software that is provided by a network host.
    - B They rent software from a cloud computing provider.
    - C They purchase software from the developer.
    - D They need a specialized web browser.
  - 3 Which of the following is NOT required for users to access a cloud?
    - A web browser
    - B software license
    - C a fee
    - D a large bandwidth



### Computer World:

## Specialized Software

Cloud computing is changing the distribution of computing and storage services. Cloud computing is a network of companies and users sharing resources. Software engineers must be ready to meet demands for this specialized software.

One model of cloud computing is **SaaS**, or Software as a Service. In this model, software is hosted by a provider and accessed over a network. Another example is **PaaS**, or Platform as a Service. In this model, computer hardware and software are rented. **Software on demand** falls under the PaaS category.

To provide their services, companies purchase **software licenses**. Then they charge users for access to the software. Some companies charge users by **metered fees**. This means that users pay afterwards for whatever they used. Others support **pay-as-you-go** systems. In these cases, **software as a product** is purchased in advance, when it is needed.

Several requirements of the **computer platform** are necessary to access cloud technology. The computer must be **online**. The **bandwidth** of the system needs to be large in order to quickly send and receive information. But if the specifications are right, cloud computing is a tremendously useful tool. Users can easily access data from web **browsers**, among other methods. A tip to software engineers: get into this expanding industry!

## Vocabulary

- 3 Match the words (1-8) with the definitions (A-H).

1 ___ cloud computing	6 ___ online
2 ___ distribution	7 ___ bandwidth
3 ___ software license	8 ___ software on demand
4 ___ computer platform	
5 ___ browser	

  - A a software system that is used to gain access to information on the internet
  - B being connected to the internet
  - C a measure of a computer system's capacity to send and receive information
  - D a combination of hardware and system software that allows an application to run
  - E a model in which software is rented from a provider
  - F a legal agreement which grants the buyer of a program the right to use it
  - G the action of supplying a product or service
  - H a model in which computing is delivered as a service rather than as a product

4 Choose the sentence that uses the underlined part correctly.

- 1 A In PaaS, computer software can be rented.  
B A computer must be equipped with SaaS to access a web browser.
- 2 A The company uses a pay-as-you-go system, requiring a set monthly fee.  
B The customer pays a metered fee, so he only pays for the services he uses.
- 3 A To access data storage, a computer must have software on demand.  
B Some software developers only supply software as a product.

5 Listen and read the journal article again. What must a computer have in order to access the cloud?

## Listening

6 Listen to a conversation between a manager and an engineer. Mark the following statements as true (T) or false (F).

- 1  The man had a negative experience with cloud computing.
- 2  The company recently requested a new software license.
- 3  The woman recommends SaaS.

7 Listen again and complete the conversation.

**Manager:** Hey, Tonya. I'm thinking about upgrading the company's network, and I was wondering what 1 \_\_\_\_\_.

**Engineer:** SaaS is a networking model.

**Manager:** I don't know much about computers. What does that mean?

**Engineer:** Let me explain the basics. SaaS stands for Software as a Service. It is a model that 2 \_\_\_\_\_.

**Manager:** What's cloud computing?

**Engineer:** It's a 3 \_\_\_\_\_ services.

**Manager:** So it's a way to access information?

**Engineer:** Yes, partly. Clients purchase the right to a 4 \_\_\_\_\_ . Then they share that software with their users.

**Manager:** Okay. How 5 \_\_\_\_\_ for that, though?

**Engineer:** Clients typically pay regular fees. Some companies charge users metered fees, while others 6 \_\_\_\_\_ . It just depends on your needs.

## Speaking

7 With a partner, act out the roles below based on Task 7. Then, switch roles.

### USE LANGUAGE SUCH AS:

*I'm thinking of ...*  
*What are your thoughts on ...*  
*Some prefer ...*

**Student A:** You are a manager.

Talk to Student B about:

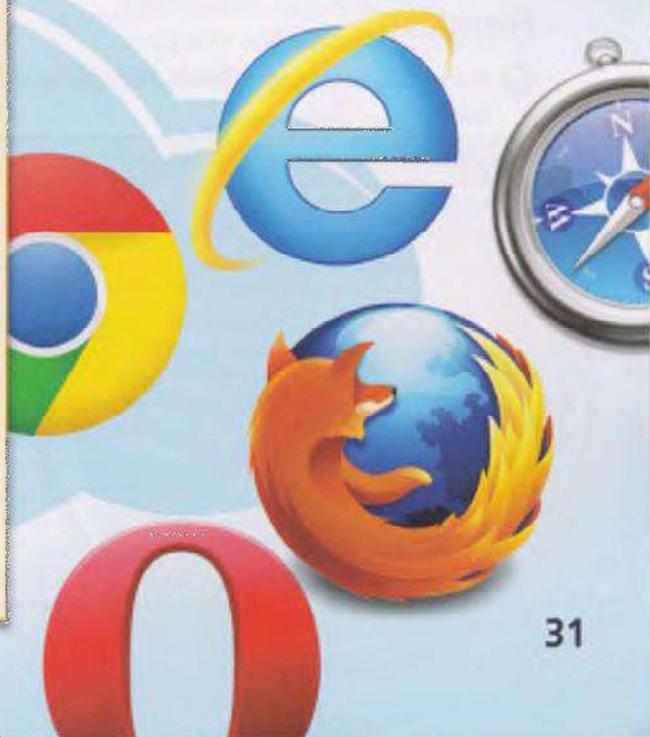
- elements of cloud computing
- how the services work
- his or her recommendation

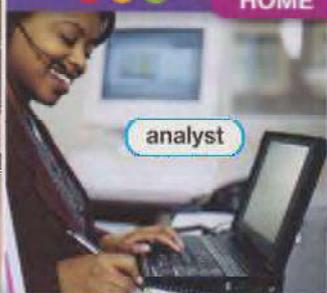
**Student B:** You are an engineer.

Talk to Student A about elements of cloud computing.

## Writing

9 Use the journal article and conversation from Task 8 to write a review of SaaS and PaaS. Include: user options for accessing information, how users can pay for services, and what computer requirements users need to access the cloud.





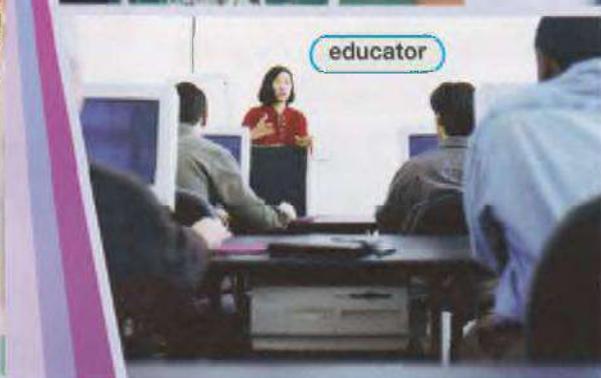
analyst



technical support



architect



educator

## Techcareer: High-Tech Jobs > What Can I Do With It? > Software Engineering

Software engineering is a rapidly growing industry in today's high-tech economy. The **software life cycle** is quickening. This means that companies must develop technology faster and faster.

So what are your options? Many engineers start in **technical support**. This is a good place to become familiar with different technologies. Some engineers also learn about products as **testers** and **analysts**. These jobs promote critical thinking and problem-solving skills.

If you enjoy concepts and theories, check out the educational field. Universities need well-trained **educators**, especially those with skills to be **researchers**. Even if education isn't your long term goal, it's a great opportunity for **professional development**.

More experience and education will

help you **advance** your career. Nowadays, almost every professional industry has some need for software development. Some companies hire full-time **developers**, while others take on **freelancers** and **contractors**. Many developers are owners and **managers** of their own small businesses.

Are you looking for something a little different? If you enjoy general computer engineering, consider becoming an **architect**. Functional hardware is an important part of reliable software. Its development is another expanding industry that needs bright, talented engineers.

Whatever your goals, consider joining the **IEEE** and **ACM**. **Memberships** in these professional organizations come with opportunities for networking and further career development.

### Get ready!

- Before you read the passage, talk about these questions.
  - What are some different career options for software engineers?
  - How does the software life cycle affect jobs in the software industry?

### Reading

- Read the webpage. Then, complete the table.

Action	Benefit
Starting as a tester or analyst	1 _____
2 _____	is an opportunity for professional development.
3 _____	Most industries have software development needs.
Becoming an architect	4 _____
Joining a professional organization	5 _____

### Vocabulary

- Match the words and phrases (1-10) with the definitions (A-J).

- |                |                                |
|----------------|--------------------------------|
| 1 __ tester    | 6 __ freelancer                |
| 2 __ ACM       | 7 __ researcher                |
| 3 __ advance   | 8 __ membership                |
| 4 __ manager   | 9 __ technical support         |
| 5 __ architect | 10 __ professional development |

- a worker who is hired for temporary jobs
- an official status indicating that someone is part of a group
- a professional who studies and analyzes something
- the process of gaining knowledge that furthers one's career
- a professional who runs a business
- a professional organization that supports the study of computers
- the process of assisting people with hardware or software problems
- a professional who designs and creates hardware
- to cause something to achieve a higher status
- a professional who uses products to determine how well they function

**4** Read the sentences and choose the correct words or phrases.

- 1 The company's software **testers / developers** design all the new programs
- 2 The **IEEE / professional development** is a group that supports technological innovation.
- 3 An **analyst's / architect's** job is to examine existing systems and identify opportunities for improvement.
- 4 The engineer is a **manager / contractor**, so she works on projects for different companies.
- 5 **Educators / Freelancers** are most commonly found in classrooms.
- 6 The **membership / software life cycle** explains why there are so many jobs in software development.

**5** Listen and read the webpage again. What is a benefit of joining a professional organization?

**Listening**

**6** Listen to a conversation between an intern and an engineer. Mark the following statements as true (T) or false (F).

- 1 \_\_\_ The man recently completed an engineering degree program.
- 2 \_\_\_ The woman recommends starting an engineering career as a researcher.
- 3 \_\_\_ The man applied to be a member of the IEEE.

**7** Listen again and complete the conversation.

**Intern:** Eventually, I want to **1** \_\_\_\_\_ .  
But the academic side interests me, too.

**Engineer:** So you **2** \_\_\_\_\_ be a researcher?

**Intern:** I'm considering it. What **3** \_\_\_\_\_ of that idea?

**Engineer:** I think that's a smart move. You have to be able to **4** \_\_\_\_\_ before you can develop it.

**Intern:** That's what I thought. Then maybe I could be a freelancer **5** \_\_\_\_\_ .

**Engineer:** Oh, sure. That's a great way to advance your career.

**Intern:** Right. Wow, it really seems like there are **6** \_\_\_\_\_ in this field.

**Speaking**

**8** With a partner, act out the roles below based on Task 7. Then, switch roles.

**USE LANGUAGE SUCH AS:**

*Have you decided what ...*  
*Eventually, I want to ...*  
*What do you think of ...*

**Student A:** You are an intern. Talk to Student B about:

- your engineering career goals
- his or her opinion about your plans
- other resources for career information

**Student B:** You are an engineer. Talk to Student A about his or her engineering career goals.

**Writing**

**9** Use the webpage and conversation from Task 8 to write an application letter for an engineering degree program. Include: the applicant's career goals, his or her plans immediately after graduation, and how the program will help meet his or her goals.



# Glossary

- ACM** [N-COUNT-U15] The **ACM** (Association for Computing Machinery) is a professional organization that supports the study and development of computers.
- ad hoc** [ADJ-U3] If something is **ad hoc**, it is unsystematic and done for one particular instance only.
- adequate** [ADJ-U13] If something is **adequate**, it is enough or acceptable.
- ADL** [N-COUNT-U4] An **ADL** (architecture description language) is a system that formally describes the architectural configuration of a software system.
- advance** [V-T-U15] To **advance** something is to cause something to achieve a higher status.
- algorithmic model** [N-COUNT-U11] An **algorithmic model** is a system that uses a specific formula in order to estimate time, effort, or cost.
- allocation problem** [N-COUNT-U12] An **allocation problem** is a situation in which resources are unstable, and the focus is on finishing the project with the available resources.
- analyst** [N-COUNT-U15] An **analyst** is a professional who monitors or examines something, usually for the purpose of identifying ways to improve it.
- apparatus level** [N-UNCOUNT-U1] **Apparatus level** is a view of the material component that specifies the shape and feel of buttons, keys, and other hardware that the user will interact with.
- application generator** [N-COUNT-U4] An **application generator** is a tool that helps engineers write programs on a large scale.
- approach** [N-COUNT-U3] An **approach** is a way or strategy for doing or creating something.
- approve** [V-T-L8] To **approve** something is to officially accept it as satisfactory.
- architect** [N-COUNT-U15] An **architect** is a professional who designs and creates hardware.
- artistic design** [N-UNCOUNT-U2] **Artistic design** is the practice of using graphic design to draw a user's attention to important parts of an interface.
- AWB** [N-COUNT-U7] An **AWB** (analyst workbench) is an integrated environment that supports early software development stages of a project.
- back-end** [ADJ-U7] If an activity is **back-end**, it is only accessed by developers and not by users.
- bandwidth** [N-UNCOUNT-U14] **Bandwidth** is a measure of a computer system's capacity to send and receive information over a network.
- base formula** [N-COUNT-U11] A **base formula** is a formula for calculating the relation between software size and effort.
- baseline** [N-COUNT-U8] A **baseline** of established specifications is set and serves as the basis for the development of something.
- best interests** [N-COUNT-U13] **Best interests** are circumstances that are most advantageous for a specific person or a group of people.
- black-box reuse** [N-UNCOUNT-U3] **Black-box reuse** is a method of software reuse in which software elements are reused without modification.
- BM** [N-COUNT-U5] A **BM** (basic execution time model) is a software reliability model in which the decrease in failure intensity is constant.
- browser** [N-COUNT-U14] A **browser** is a software system that is used to gain access to information on the Internet.
- budget** [N-COUNT-U11] A **budget** is an estimation and allotted cost of a project.
- CASE** [N-UNCOUNT-U6] **CASE** (Computer Aided Software Engineering) is the application of various support systems in the software development process.
- CCB** [N-COUNT-U8] A **CCB** (configuration control board) is a staff that ensures that any changes made to the baseline of a software development project are approved and completed correctly.
- change request** [N-COUNT-U8] A **change request** is a proposed adjustment or correction to the baseline of a project.
- change-oriented** [ADJ-U8] If a development model is **change-oriented**, it identifies new configurations by describing the changes made to the baseline.

- chief programmer team** [N-COUNT-U9] A **chief programmer team** is a team of three people in which the person with the most responsibility is designated as the chief programmer of a project.
- city** [ADJ-U6] If a value on the user scale is **city**, it indicates that a product supports the development of a system larger than a family.
- CLG** [N-COUNT-U1] The **CLG** (command language grammar) is a specific grammatical structure that describes the user interface aspects of a computer system.
- cloud computing** [N-UNCOUNT-U14] **Cloud computing** is a mode in which computing is delivered as a service rather than as a product, with resources shared over a network rather than used locally.
- CMM** [N-COUNT-U10] The **CMM** (capability maturity model) is a set of directions aimed at improving the development process.
- COCOMO** [N-UNCOUNT-U11] **COCOMO** is a model of cost estimation that distinguishes between three classes of projects.
- code scavenging** [N-UNCOUNT-U4] **Code scavenging** is the process of reusing code that has been previously written if it happens to solve current problems.
- commitment style** [N-UNCOUNT-U9] **Commitment style** is a management style in which project decisions are guided by the goals of the project.
- common feature** [N-COUNT-U10] A **common feature** is one of the five sections of the CMM's key practices.
- communication component** [N-COUNT-U1] A **communication component** is a view of a system that concerns the dialog between systems and users.
- comparison method** [N-COUNT-U11] The **comparison method** is a system of estimating costs for a project by comparing it to a similar completed project.
- compositional** [ADJ-U3] If technology is **compositional**, its existing components can easily be reused in new systems.
- computing platform** [N-COUNT-U14] A **computing platform** is a combination of hardware and system software that allows a particular kind of application to run.
- conceptual component** [N-COUNT-U1] A **conceptual component** is a view of a system that concerns the functions that the system will perform for users.
- conceptual model** [N-COUNT-U1] A **conceptual model** is a technically-accurate model of a computer system that is rendered in terms of a system's reactions to user actions.
- configuration item** [N-COUNT-U8] A **configuration item** is a piece of hardware or software that aids in configuration management and is contained in the baseline.
- configuration management** [N-UNCOUNT-U8] **Configuration management** is the practice of systematically monitoring the creation and updating of elements during the software development process.
- conform to** [V-T-U10] To **conform to** something is to follow its standards or rules.
- contractor** [N-COUNT-U15] A **contractor** is an independent worker who is hired to perform particular work under contract, which may be long- or short-term.
- corresponding** [ADJ-U8] If two things are **corresponding**, they are related to each other or contain references to the same subjects.
- COTS** [ADJ-U3] If a software is **COTS** (commercial, off-the-shelf), it is unmodified from its original state and the contents of the software are generally unknown.
- critical path** [N-COUNT-U12] A **critical path** is a part of a PERT chart that identifies which tasks must be completed on time for the entire project to be successful.
- deceptive** [ADJ-U13] If something is **deceptive**, it is misleading or dishonest.
- defensive programming** [N-UNCOUNT-U5] **Defensive programming** is the practice of creating reliable software by ensuring that components can function properly in a number of contexts.
- degree of certainty** [N-COUNT-U12] The **degree of certainty** is a scale that measures the dependability of software user requirements and development resources.

# Glossary

- Delphi-method** [N-COUNT-U11] The **Delphi-method** is a method of estimating costs in which a panel of experts estimate costs separately and then discuss their estimations until they reach an agreement.
- delta** [N-COUNT-U8] A **delta** is a difference between one version of software and the next version of the same software.
- design problem** [N-COUNT-U12] A **design problem** is a situation in which the steps to carrying out the project are unknown, and the focus is on assigning responsibilities and accomplishing individual milestones.
- design view** [N-COUNT-U1] A **design view** is a conceptual model that focuses on the user interface design.
- developer** [N-COUNT-U15] A **developer** is a professional who designs and creates software.
- development time** [N-UNCOUNT-U11] **Development time** is the time between the beginning of the requirements engineering phase and the moment when the software is delivered to a customer.
- dialog** [N-UNCOUNT-U2] **Dialog** is a reciprocal communication between a computer and a user.
- distribution** [N-UNCOUNT-U14] **Distribution** is the action of supplying a product or service.
- domain analysis** [N-UNCOUNT-U4] **Domain analysis** is a process which identifies, captures, structures, and reorganizes information for software development.
- educator** [N-COUNT-U15] An **educator** is a professional who teaches other people about something.
- end user** [N-COUNT-U2] An **end user** is a consumer who becomes the intended or primary user of a product.
- ensure** [V-T-U13] To **ensure** something is to be certain that something will happen.
- environment** [N-COUNT-U6] An **environment** is an application that supports the complete software development process.
- ergonomics** [N-UNCOUNT-U2] **Ergonomics** is the study of designing hardware that is intended to be operated physically by users.
- estimate** [V-I-U11] To **estimate** something is to attempt to determine the likely amount or cost of it before it is finished.
- ethics** [N-COUNT-U13] **Ethics** are a set of morals that govern the actions of an individual or a group.
- exception domain** [N-COUNT-U5] An **exception domain** is the set of incorrect inputs to a software component.
- expected exception domain** [N-COUNT-U5] An **expected exception domain** is the incorrect input that is anticipated and recognized by software.
- exploration problem** [N-COUNT-U12] An **exploration problem** is a challenging situation in which a project's degree of certainty is low, and the focus is on achieving unspecified goals.
- family** [ADJ-U6] If a value on the user scale is **family**, it indicates that a product is designed to facilitate interactions between developers.
- fault-tolerant** [ADJ-U5] If a disk is **fault-tolerant**, it contains backup data in case of software failure.
- flaw** [N-COUNT-U8] A **flaw** is a fault or weakness.
- freelancer** [N-COUNT-U15] A **freelancer** is a worker who is hired as temporary staff or on a job-by-job basis.
- functionality** [N-UNCOUNT-U2] **Functionality** is the range of operations that a computer or software system can perform.
- Gantt chart** [N-COUNT-U12] A **Gantt chart** is a type of graph that uses bars to detail the project's schedule.
- generative** [ADJ-U3] If technology is **generative**, its components are used to create programs that generate new programs.
- groupware** [N-UNCOUNT-U2] **Groupware** is software designed to assist a group of people achieve a common goal or complete a collaborative task.
- HCI** [N-UNCOUNT-U2] **HCI** (human-computer interaction) is the study and design of interactions between users and computers.
- health** [N-UNCOUNT-U13] **Health** is a state of physical, mental, and social well-being.
- hierarchical organization** [N-UNCOUNT-U9] **Hierarchical organization** is a team organization style in which different levels of management are distinguished.
- humanities** [N-UNCOUNT-U2] **Humanities** is the study of or focus on how people perceive, learn, think, and feel.

- IEEE** [N-COUNT-U15] The **IEEE** (Institute of Electrical and Electronics Engineers) is a professional organization that supports technological development and sets widely-accepted standards for technological product specifications.
- IEEE Standard for Quality Assurance Plans** [N-UNCOUNT-U10] The **IEEE Standard for Quality Assurance Plans** is a set of procedures aimed specifically at testing and verifying quality in software systems.
- improve** [V-T-U10] To **improve** something is to make it better.
- incorporate** [V-T-U8] To **incorporate** something is to include it as part of a whole.
- individual** [ADJ-U6] If a value on the user scale is **individual**, it indicates that a product is designed to assist in software construction by individual developers.
- instantiate** [V-T-U4] To **instantiate** something is to complete it or give it substance.
- integrated environment** [N-COUNT-U6] An **integrated environment** is a development environment that contains the specifications of a final product.
- integration style** [N-UNCOUNT-U9] **Integration style** is a management style that features informal decision-making and promotes creativity from employees.
- integrity** [N-UNCOUNT-U13] **Integrity** is the value of being honest.
- intermediate product** [N-COUNT-U4] An **intermediate product** is a piece of code that is ready to be used in the development of a more complicated application.
- IPSE** [N-COUNT-U7] An **IPSE** (Integrated Project Support Environment) is an application that contains tools to support all phases of the software development process.
- ISO 9001** [N-UNCOUNT-U10] The **ISO 9001** is a set of standards that states general requirements for the quality of a software system.
- key practice** [N-COUNT-U10] A **key practice** is an activity that implements the CMM's key process areas.
- key process area** [N-COUNT-U10] A **key process area** is an indication of issues that must be addressed in order to reach a given maturity level, as determined by the CMM.
- keystroke level** [N-COUNT-U1] A **keystroke level** is a view of the communication component that describes the physical actions of a user, such as keystrokes or mouse clicks.
- KLOC** [N-COUNT-U11] **KLOC** (kilo lines of code) is a measurement of a computer program's size as determined by the number of lines of source code that it has.
- language-centered environment** [N-COUNT-U6] A **language-centered environment**, also called a programming environment, is an interactive development environment that contains tools for development in a particular programming language.
- layer** [N-COUNT-U2] A **layer** is a level of operation of a system.
- learning effect** [N-COUNT-U11] The **learning effect** is the theory that the rate of productivity increases as a project continues.
- linguistic view** [N-COUNT-U1] A **linguistic view** is a conceptual model that describes the interactions between a human and a system.
- LPM** [N-COUNT-U5] An **LPM** (logarithmic Poisson execution time model) is a software reliability model in which the decrease in failure intensity is exponential.
- manager** [N-COUNT-U15] A **manager** is a professional who runs a business or supervises some part of a business.
- man-month** [N-COUNT-U11] A **man-month** is an estimate of the amount of work performed by an employee in a given month.
- material component** [N-COUNT-U1] A **material component** is a view of a system that concerns the graphics of the user interface and the hardware that the user will interact with.
- matrix organization** [N-UNCOUNT-U9] **Matrix organization** is a team organization style in which employees are organized into units according to their specialization.
- maturity level** [N-COUNT-U10] A **maturity level** is a measure of the progress of a software process towards achieving a particular standard of quality.

# Glossary

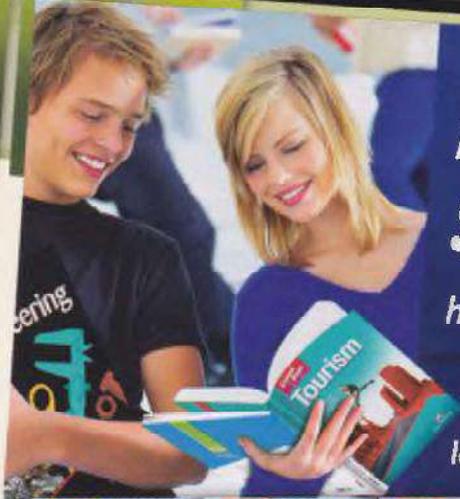
- membership** [N-COUNT-U15] A **membership** is an official status indicating that someone is part of a group or organization.
- mental model** [N-COUNT-U1] A **mental model** is a user's understanding of a computer system.
- metered fee** [N-COUNT-U14] A **metered fee** is a payment for only the services which a customer actually uses from a potentially unlimited resource.
- middleware** [N-UNCOUNT-U4] **Middleware** is software that connects a computer's operating system to individual applications and ensures that programs can run together smoothly.
- MIL** [N-COUNT-U4] A **MIL** (Module Interconnection Language) is a formal description of the overall structure of a software system.
- MVC paradigm** [N-COUNT-U2] The **MVC** (model-view-controller) **paradigm** is a design pattern for user interfaces that splits the application into three areas: the model, the view, and the controller.
- MWB** [N-COUNT-U7] An **MWB** (management workbench) is a programming environment that contains tools for planning and control of a software development project.
- N-version programming** [N-UNCOUNT-U5] **N-version programming** is a technique for software fault tolerance in which multiple functionally-equivalent programs are generated from the same initial specifications.
- online** [ADJ-U14] If something is **online**, it is connected to or available through the Internet.
- open structured team** [N-COUNT-U9] An **open structured team** is a team organization style that combines an open management style with clear guidelines for decision making.
- optimistic** [ADJ-U11] If something is **optimistic**, it assumes the best possible scenarios and the lowest possible costs.
- PaaS** [N-UNCOUNT-U14] **PaaS** (Platform as a Service) is a model in which computer hardware and software is rented as a service rather than purchased.
- parallel development** [N-UNCOUNT-U8] **Parallel development** is the practice of creating different branches of revision from the same original baseline or software version.
- pay-as-you-go** [ADJ-U14] If software is **pay-as-you-go**, it is purchased as it is needed by the customer.
- PCTE** [N-UNCOUNT-U7] **PCTE** (Portable Common Tool Environment) is a tool interface that supports engineers in the development of environments.
- PERT chart** [N-COUNT-U12] A **PERT** (program evaluation review technique) **chart** is a tool used to coordinate tasks within a project.
- presentation** [N-COUNT-U2] A **presentation** is the collective aspects of a system that are perceptible to the user, such as the screen layout or the keyboard layout.
- principle** [N-COUNT-U13] A **principle** is a rule that guides the process of decision-making.
- probability** [N-UNCOUNT-U5] **Probability** is the extent to which something is likely to happen.
- process certainty** [N-UNCOUNT-U12] **Process certainty** is a metric that is determined by the stage of development of software and whether it can or must be changed.
- process scale** [N-COUNT-U6] A **process scale** is a software development feature that specifies whether a product supports code development or general human activities.
- process-centered environment** [N-COUNT-U6] A **process-centered environment** is a development environment that focuses on the process of software development.
- product** [N-COUNT-U3] A **product** is something that is available for purchase.
- product certainty** [N-UNCOUNT-U12] **Product certainty** is a metric that is determined by the functionality and quality of user requirements.
- professional development** [N-UNCOUNT-U15] **Professional development** is the process of gaining knowledge, skills, and experiences that make someone able or better qualified to perform a job.
- professional judgment** [N-UNCOUNT-U13] **Professional judgment** is the ability to make good decisions based upon professional experience.
- program library** [N-COUNT-U4] A **program library** is a collection of ready to use pieces of code.

- programming environment** [N-COUNT-U7] A **programming environment**, also called a language-centered environment, is an interactive development environment that contains tools for development in a particular programming language.
- cognitive view** [N-COUNT-U1] A **cognitive view** is a means of understanding a system that considers what a user needs to understand about a system in order to operate it.
- public interest** [N-UNCOUNT-U13] **Public interest** is the well-being of the public as a whole.
- Putnam model** [N-UNCOUNT-U11] **Putnam model** is a cost estimation model that considers problems and their solutions to estimate the effort and budget required for a project.
- PWB** [N-COUNT-U7] A **PWB** (programmer workbench) is an integrated environment that supports the software development stages of testing and implementation.
- quality control** [N-UNCOUNT-U10] **Quality control** is a system of maintaining particular standards across a development process.
- quality criteria** [N-COUNT-U10] **Quality criteria** are sets of quality attributes which can be measured directly or indirectly.
- quality factor** [N-COUNT-U10] A **quality factor** is a quality attribute that can be measured only indirectly.
- realization problem** [N-COUNT-U12] A **realization problem** is a situation in which the software requirements are stable and the focus is on how to reach the goals of the project under ideal circumstances.
- recovery block** [N-COUNT-U5] A **recovery block** is an automatically saved file of data that is used as backup in case an operation causes a software failure.
- redundancy** [N-UNCOUNT-U5] **Redundancy** is the inclusion of components that are not necessary or are copies of existing components to ensure proper function of software in case of error or failure.
- relation directedness** [N-UNCOUNT-U9] **Relation directedness** is a management style which focuses on individual employees and their relationships with other employees.
- relation style** [N-UNCOUNT-U9] **Relation style** is a management style that relies heavily on motivation and training.
- reliability** [N-UNCOUNT-U5] **Reliability** is the quality of being consistent and free of errors.
- researcher** [N-COUNT-U15] A **researcher** is a professional who studies and analyzes something to get more information about it.
- reserved checkout** [N-UNCOUNT-U7] **Reserved checkout** is a system that allows only one person at a time to edit a file.
- resource certainty** [N-UNCOUNT-U12] **Resource certainty** is a metric that is determined by the availability of resources, such as qualified people, to work on a project.
- retrace** [V-T-U8] To **retrace** something is to review steps that have already been completed.
- risk factor** [N-COUNT-U12] A **risk factor** is a condition/characteristic that increases the likelihood of problems.
- risk management** [N-UNCOUNT-U12] **Risk management** is a process that identifies risks and prevents them from becoming setbacks.
- robust programming** [N-UNCOUNT-U5] **Robust programming** is the practice of ensuring that software components function correctly regardless of their context.
- SaaS** [N-UNCOUNT-U14] **SaaS** (Software as a Service) is a model in which software and the associated data are hosted by a provider and accessed over a network.
- safety** [N-UNCOUNT-U13] **Safety** is the condition of being shielded against danger.
- scope** [N-UNCOUNT-U3] **Scope** is the extent of something or the area that it includes.
- Seeheim model** [N-COUNT-U2] The **Seeheim model** is a model of software design that separates the application from the user interface.
- semantic level** [N-COUNT-U1] A **semantic level** is a view of a conceptual component that describes system objects and general task delegation.
- separation style** [N-UNCOUNT-U9] **Separation style** is a management style in which the main goal is efficiency.
- skeleton** [ADJ-U4] If a component is **skeleton**, not all of its details have been filled in.

# Glossary

- software as a product** [N-UNCOUNT-U14] **Software as a product** is a distribution model in which software is sold as a packaged commodity to consumers.
- software crisis** [N-COUNT-U3] The **software crisis** is a problem in the software industry caused by the fact that the demand for new software applications is higher than what software developers can fulfill.
- software license** [N-COUNT-U14] A **software license** is a legal agreement which grants the buyer of a program the right to use it.
- software life cycle** [N-COUNT-U15] The **software life cycle** is an ongoing process for creating, developing, and improving software.
- software on demand** [N-UNCOUNT-U14] **Software on demand** is a model in which software is rented from a provider at the time that it is needed.
- software reliability model** [N-COUNT-U5] A **software reliability model** is a statistical model that aims to predict and prevent software failures.
- software reuse** [N-UNCOUNT-U3] **Software reuse** is the practice of incorporating modified or unmodified pieces of source code from existing software into the creation of new software.
- source code** [N-COUNT-U3] A **source code** is a listing of commands to be executed in a computer program.
- spatial layout level** [N-COUNT-U1] A **spatial layout level** is a view of a material component which specifies the graphic elements that are displayed on screen.
- specialize** [V-I-U9] To **specialize** in something is to focus primarily on one specific task or area.
- SSCS** [N-COUNT-U7] A **SSCS** (source code control system) is a system for configuration control that allows the user to keep track of changes in files and generate any version of the system.
- standard** [N-COUNT-U13] A **standard** is a commonly accepted level of accomplishment by which actual accomplishments are judged.
- standard domain** [N-COUNT-U5] A **standard domain** is the set of correct inputs to a software component.
- state** [ADJ-U6] If a value on the user scale is **state**, it indicates that a product focuses on commonality and standardization across a very large system.
- substance** [N-UNCOUNT-U3] **Substance** is the components, concepts, and procedures of something.
- SWAT team** [N-COUNT-U9] A **SWAT team** is a relatively small team that focuses on task and relation directedness.
- syntax level** [N-COUNT-U1] A **syntax level** is a view of a communication component that describes the dialog style by specifying all user and system interactions.
- task analysis** [N-UNCOUNT-U2] **Task analysis** is the act of evaluating a complex system in terms of its users, tasks, hardware, social environment, and physical environment.
- task directedness** [N-UNCOUNT-U9] **Task directedness** is a management style which focuses on the tasks that need to be achieved and the methods of achieving those tasks.
- task level** [N-COUNT-U1] A **task level** is a view of the conceptual component that concerns the tasks performed both by the machine and by the user.
- technical support** [N-UNCOUNT-U15] **Technical support** is the process of assisting people with computer and software problems.
- technique** [N-COUNT-U3] A **technique** is a skill or specific method of doing or creating something.
- template** [N-COUNT-U4] A **template** is a skeleton component that does not have all of the details of a complete program.
- tester** [N-COUNT-U15] A **tester** is a professional who uses products in order to determine how well they function.
- threshold** [N-COUNT-U5] A **threshold** is a limit that must be exceeded for a certain reaction to take place.
- tool** [N-COUNT-U6] A **tool** is a product that performs a particular task in the software development process.
- toolkit** [N-COUNT-U6] A **toolkit** is a development environment in which tools are independent of each other, and are not well integrated.

- TQM** [N-UNCOUNT-U10] **TQM** (total quality management) is the pursuit of excellence in every step of a process.
- transformation system** [N-COUNT-U4] A **transformation system** is an application that assists engineers in transforming systems from sets of specifications to executable programs.
- unethical** [ADJ-U13] If something is **unethical**, it is not morally right.
- unit** [N-COUNT-U9] A **unit** is a small, specialized group of people.
- UNIX** [N-UNCOUNT-U7] **UNIX** is a general support environment for software development.
- unreserved checkout** [N-UNCOUNT-U7] **Unreserved checkout** is a system in which files can be edited by multiple developers simultaneously.
- usage** [N-UNCOUNT-U3] **Usage** is the way that something is utilized.
- user-centered design** [N-UNCOUNT-U2] **User-centered design** is a design process that places great emphasis on the experience of end users.
- user interface** [N-COUNT-U1] A **user interface** is a collection of attributes that governs the way a user interacts with a system.
- user scale** [N-COUNT-U6] A **user scale** is a system that measures the number of users a product is capable of supporting.
- UVM** [N-COUNT-U2] A **UVM** (user virtual machine) is the hardware and software of a given system.
- version-oriented** [ADJ-U8] If a development model is **version-oriented**, it identifies new configurations with a linear numbering system.
- VHLL** [N-COUNT-U4] A **VHLL** (very high level language) is a programming language with a high level of abstraction that is used primarily by programmers for assistance in creating new programs.
- visual programming environment** [N-COUNT-U7] A **visual programming environment** is a programming environment that is typically used to highlight the graphic capabilities of the environment.
- Walston-Felix** [N-UNCOUNT-U11] **Walston-Felix** is a model of calculating software cost and effort that identifies 29 variables influencing productivity.
- WBS** [N-COUNT-U12] A **WBS** (work breakdown structure) is the decomposition of a project into smaller groups in a way that displays the overall project.
- welfare** [N-UNCOUNT-U13] **Welfare** is the overall state of a person or group's health and happiness.
- white-box reuse** [N-UNCOUNT-U3] **White-box reuse** is a method of software reuse in which software elements are modified before they are incorporated into new software.
- workbench** [N-COUNT-U6] A **workbench** is a set of related tools that support the software development process in a limited scope.
- workflow** [N-UNCOUNT-U8] **Workflow** is the sequences of processes through which something must pass to reach completion.

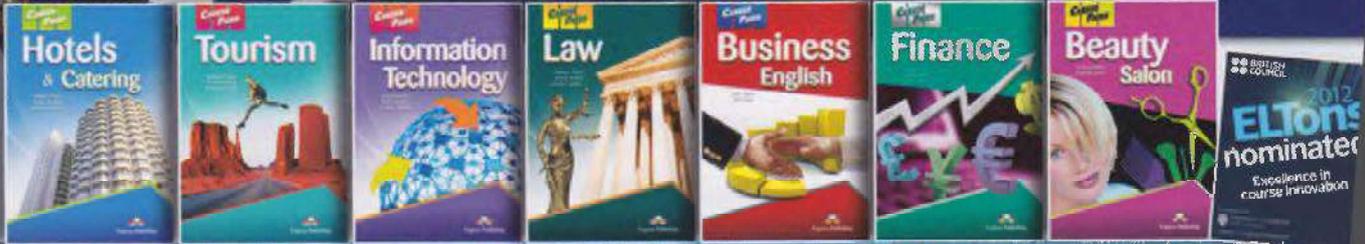


# English for Specific Purposes

<http://www.expresspublishing.co.uk>

## CAREER PATHS

The ideal series to help professionals and students develop the language skills they need to succeed in a professional work situation



Express Publishing



**CAREER  
PATHS**



# SOFTWARE ENGINEERING

*Career Paths: Software Engineering* is a new educational resource for software engineering professionals who want to improve their English communication in a work environment. Incorporating career-specific vocabulary and contexts, each unit offers step-by-step instruction that immerses students in the four key language components: reading, listening, speaking, and writing. *Career Paths: Software Engineering* addresses topics including software development, software testing, the user interface, modeling, and career options.

The series is organized into three levels of difficulty and offers a minimum of 400 vocabulary terms and phrases. Every unit includes a test of reading comprehension, vocabulary, and listening skills, and leads students through written and oral production.

**Included Features:**

- A variety of realistic reading passages
- Career-specific dialogues
- 45 reading and listening comprehension checks
- Over 400 vocabulary terms and phrases
- Guided speaking and writing exercises
- Complete glossary of terms and phrases

Books 1-3 of *Career Paths: Software Engineering* are rated for the Common European Framework of Reference for Languages at A1, A2 and B1 respectively.



**Express Publishing**

ISBN 978-1-4715-1300-7



9 781471 519307