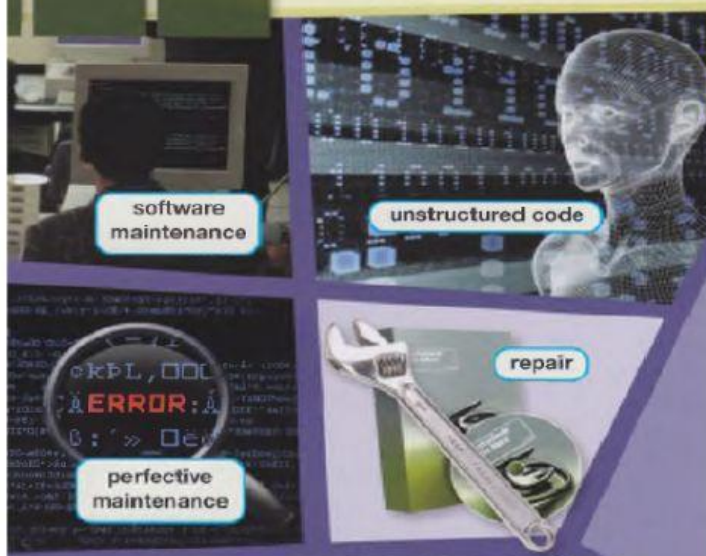


14 Software Maintenance 1



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.

- 1 The attributes of an older system may be unsuitable or not strong enough to work on updated operating systems.
_ _ s _ _ f _ _ _ _ _ t
- 2 A software engineer should fix any problems he or she finds in a code. _ _ p _ _ r
- 3 Each new updated version of existing software should come with some modifications.
_ _ l _ a _ _
- 4 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).

- 1 _ The man wants to start with corrective maintenance.
- 2 _ The engineers are adding new functionality to old software.
- 3 _ 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 _____