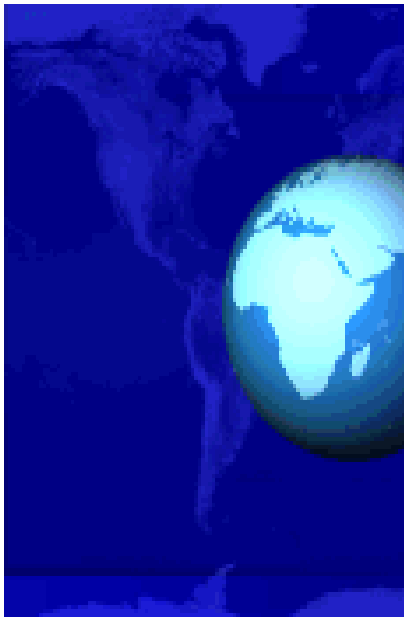

Software Maintenance

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

What is Software Maintenance?

Software Maintenance is a very broad activity that inc
corrections, enhancements of capabilities, deletion of obsolete
and optimization.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Categories of Maintenance

- Corrective maintenance

This refer to modifications initiated by defects in the software.

- Adaptive maintenance

It includes modifying the software to match changes in the e
environment.

- Perfective maintenance

It means improving processing efficiency or performance, or
the software to improve changeability. This may include enh
existing system functionality, improvement in computational eff

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Other types of maintenance

There are long term effects of corrective, adaptive and perfec
This leads to increase in the complexity of the software,
deteriorating structure. The work is required to be done to ma
reduce it, if possible. This work may be named a
maintenance.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

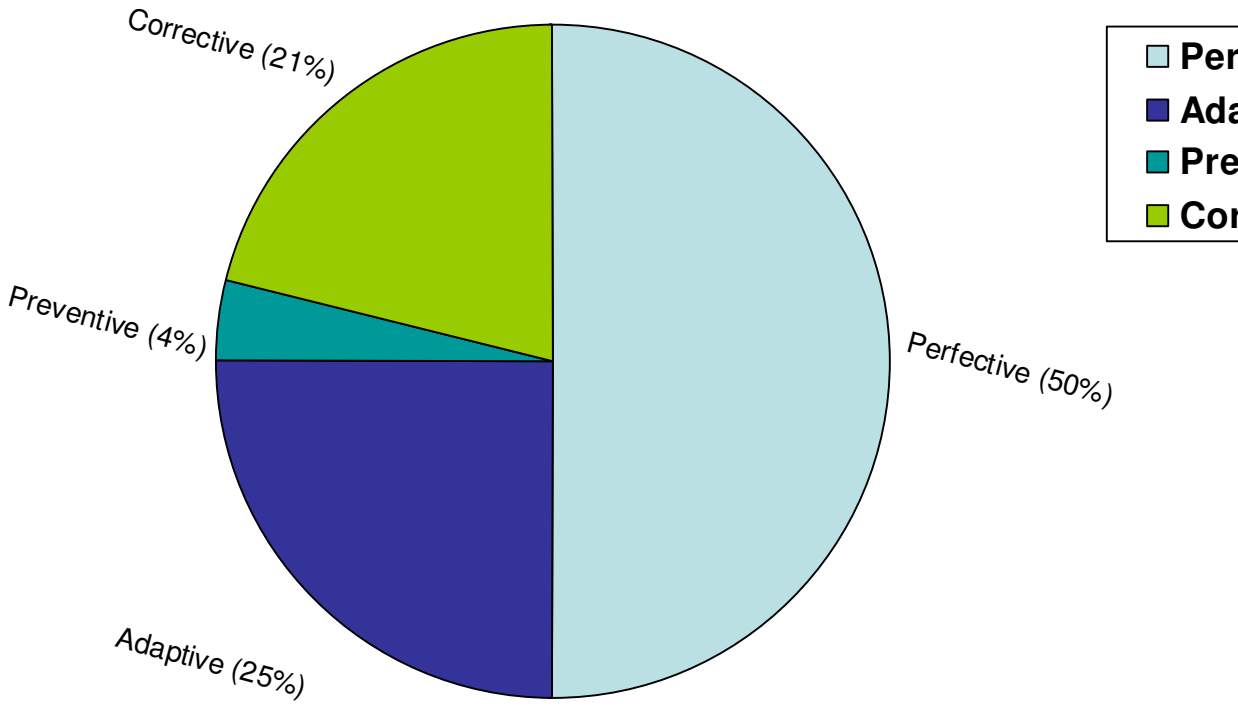

Fig. 1: Distribution of maintenance effort

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Problems During Maintenance

- Often the program is written by another person or group of
- Often the program is changed by person who did not u clearly.
- Program listings are not structured.
- High staff turnover.
- Information gap.
- Systems are not designed for change.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Maintenance is Manageable

A common misconception about maintenance is that it is not ma

Report of survey conducted by Lientz & Swanson gives some observations:

1	Emergency debugging	12.4%
2	Routine debugging	9.3%
3	Data environment adaptation	17.3%
4	Changes in hardware and OS	6.2%
5	Enhancements for users	41.8%
6	Documentation Improvement	5.5%
7	Code efficiency improvement	4.0%
8	Others	3.5%

Table 1: Distribution of maintenance effort

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Kinds of maintenance requests

1	New reports	40.8%
2	Add data in existing reports	27.1%
3	Reformed reports	10%
4	Condense reports	5.6%
5	Consolidate reports	6.4%
6	Others	10.1%

Table 2: Kinds of maintenance requests

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Potential Solutions to Maintenance Problems

- Budget and effort reallocation
- Complete replacement of the system
- Maintenance of existing system

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

The Maintenance Process

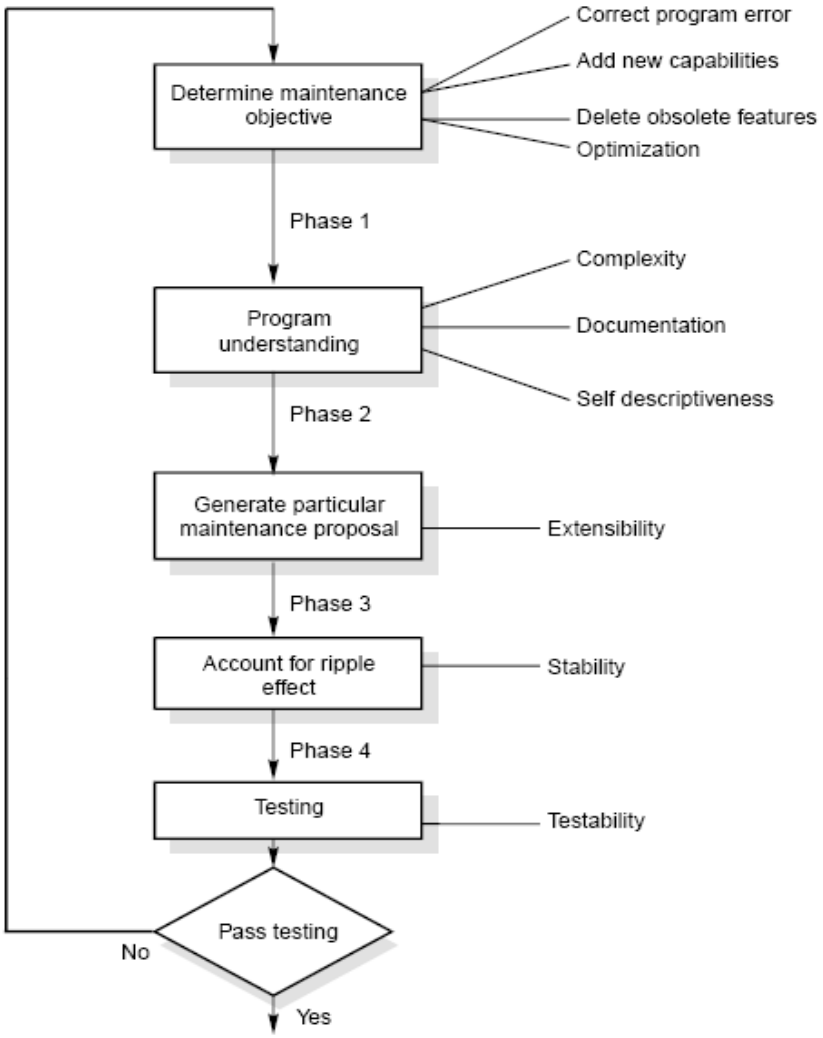

Fig. 2: T
maintena

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Program Understanding

The first phase consists of analyzing the program in order to ur

- Generating Particular Maintenance Proposal

The second phase consists of generating a particular proposal to accomplish the implementation of the maintenance

- Ripple Effect

The third phase consists of accounting for all of the ripple consequence of program modifications.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Modified Program Testing

The fourth phase consists of testing the modified program to the modified program has at least the same reliability level as b

- Maintainability

Each of these four phases and their associated software qua are critical to the maintenance process. All of these facto combined to form maintainability.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Maintenance Models

- Quick-fix Model

This is basically an adhoc approach to maintaining software fighting approach, waiting for the problem to occur and then as quickly as possible.

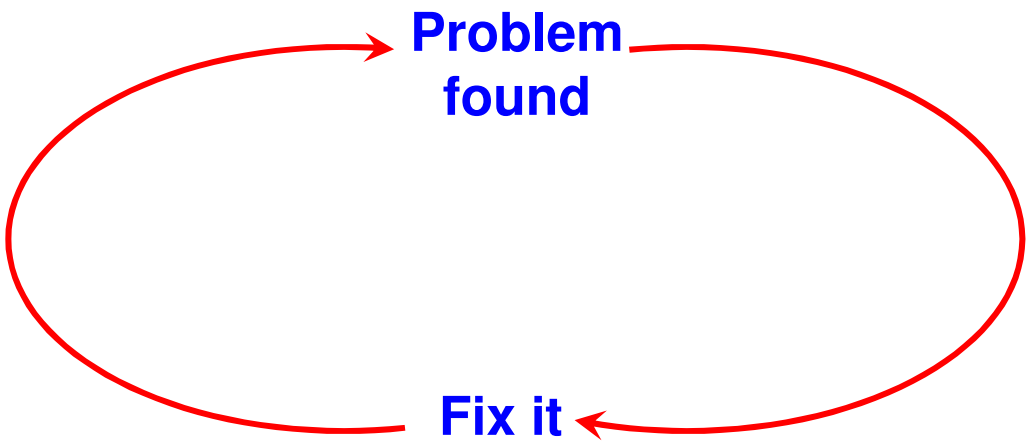

Fig. 3: The quick-fix model

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Iterative Enhancement Model
  - Analysis
  - Characterization of proposed modifications
  - Redesign and implementation

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

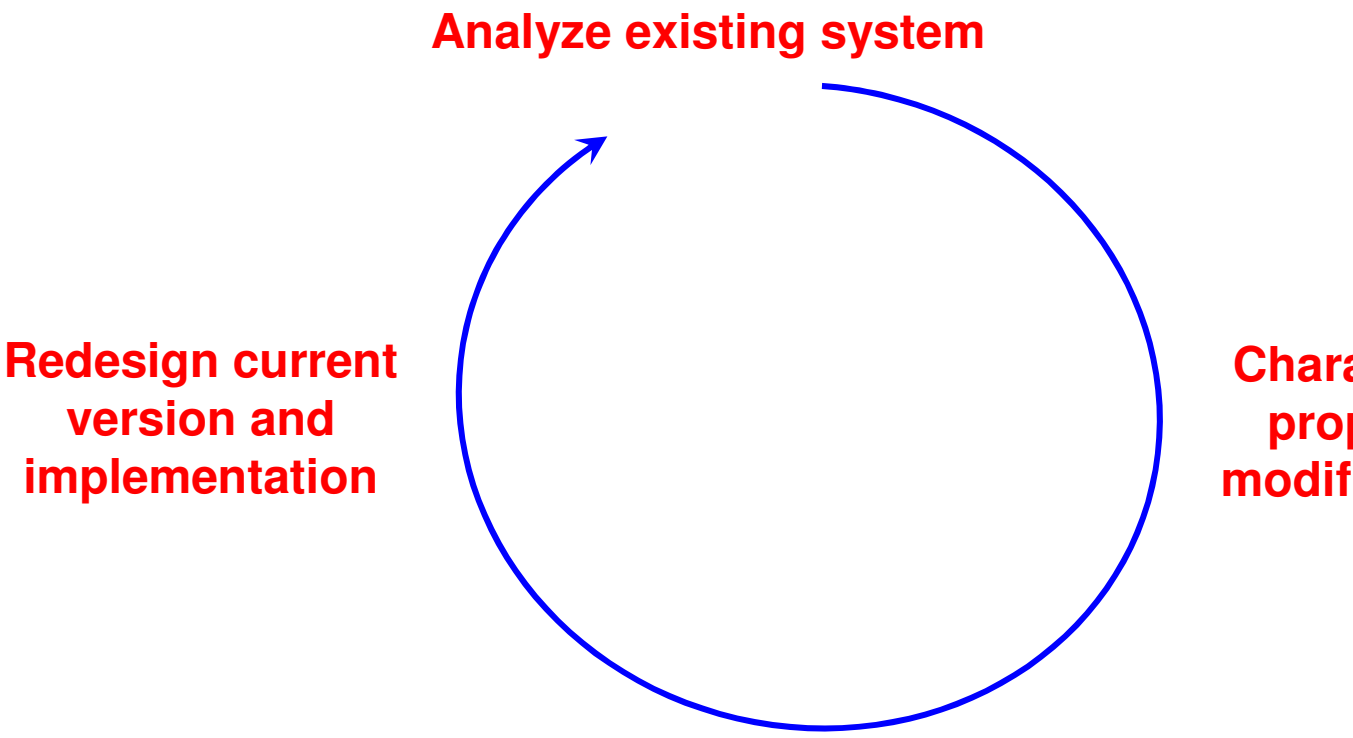

Fig. 4: The three stage cycle of iterative enhancement

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Reuse Oriented Model

The reuse model has four main steps:

1. Identification of the parts of the old system that are cand reuse.
2. Understanding these system parts.
3. Modification of the old system parts appropriate to requirements.
4. Integration of the modified parts into the new system.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

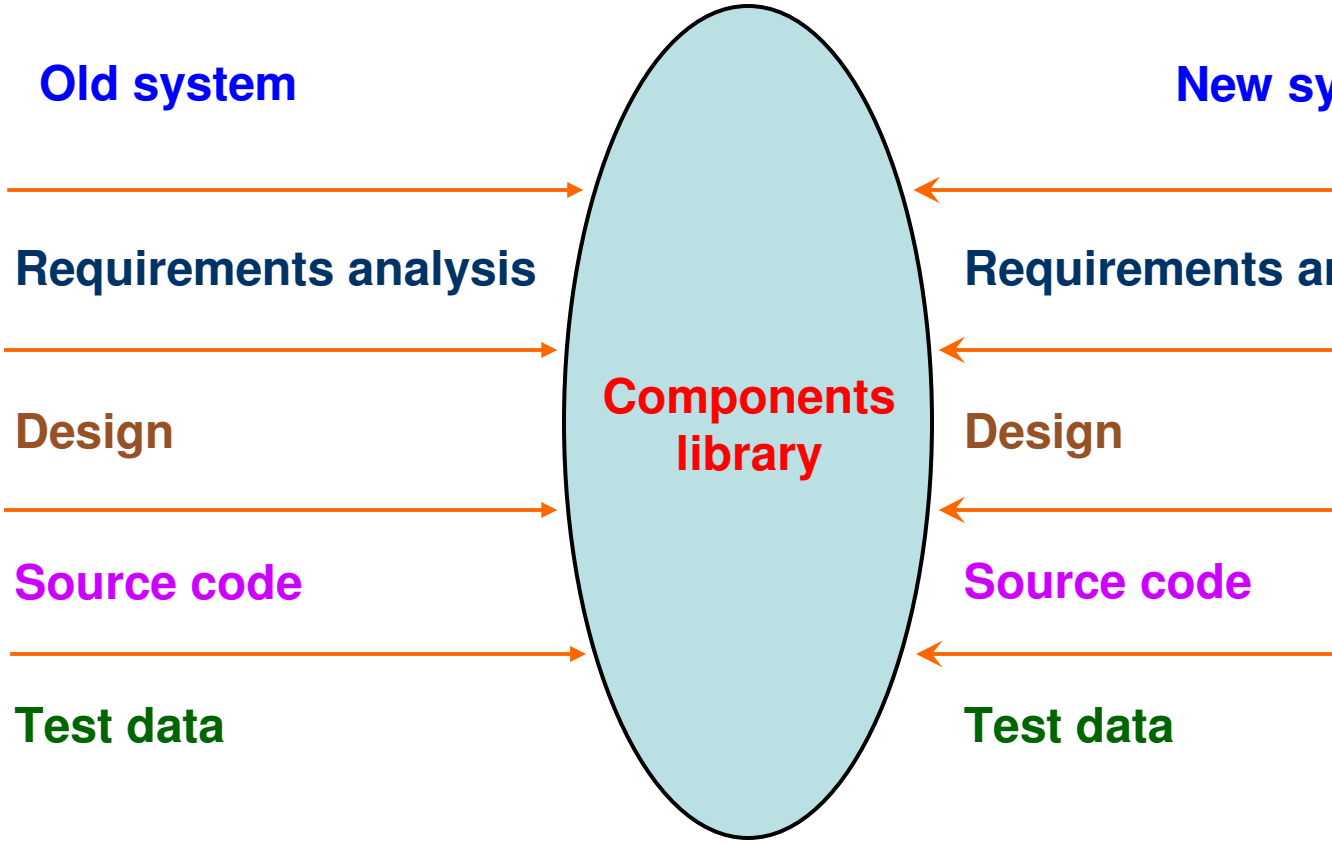

Fig. 5: The reuse model

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Boehm's Model

Boehm proposed a model for the maintenance process b
the economic models and principles.

Boehm represent the maintenance process as a closed l

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

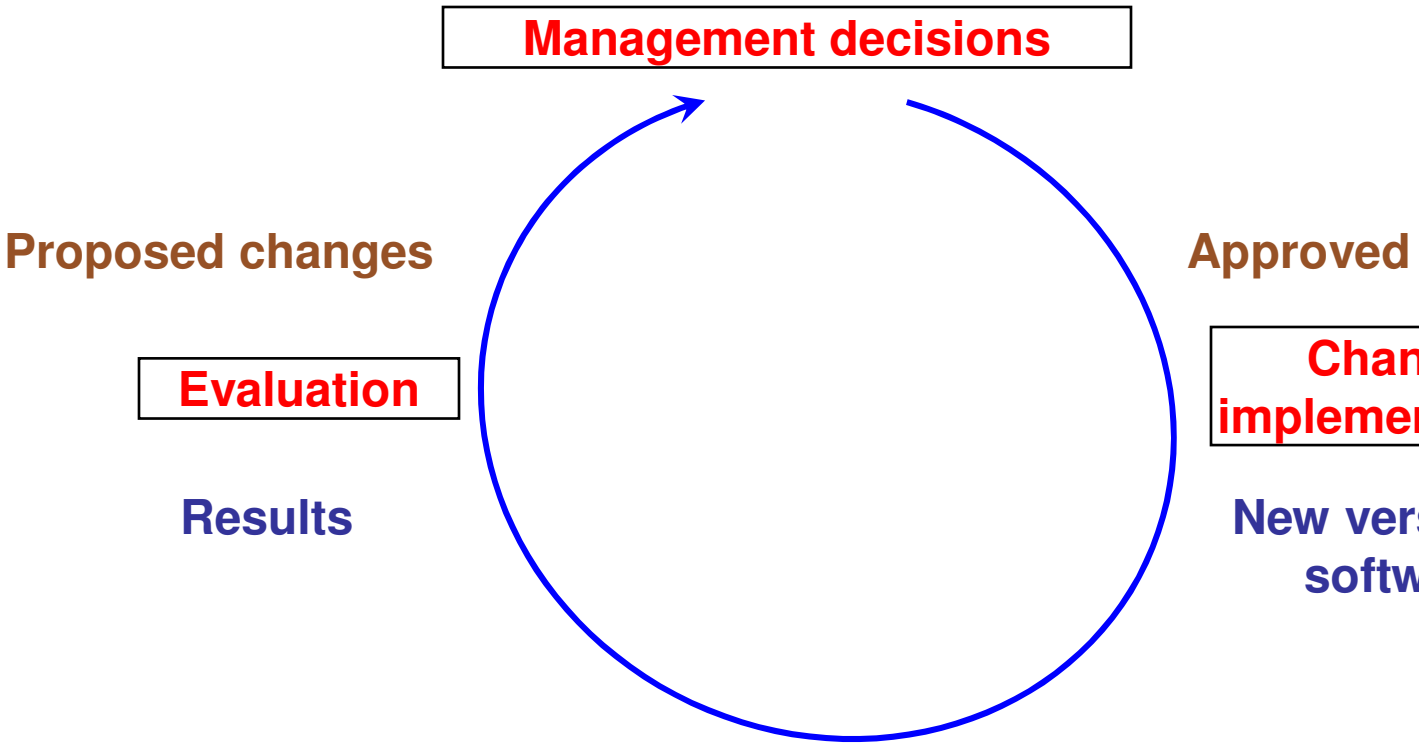

Fig. 6: Boehm's model

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

- Taute Maintenance Model

It is a typical maintenance model and has eight phases in cycle fashi
phases are shown in Fig. 7

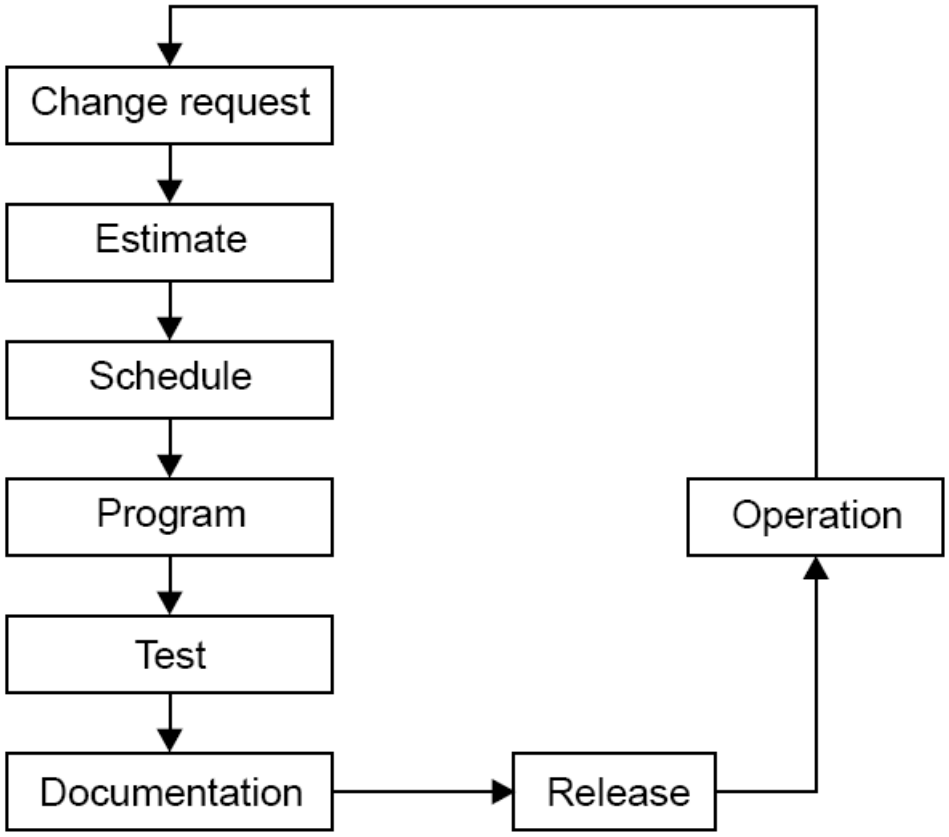

Fig. 7: Taute maintenance model

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

Phases :

1. Change request phase
2. Estimate phase
3. Schedule phase
4. Programming phase
5. Test phase
6. Documentation phase
7. Release phase
8. Operation phase

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Estimation of maintenance costs

Phase	Ratio
Analysis	1
Design	10
Implementation	100

Table 3: Defect repair ratio

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Belady and Lehman Model

$$M = P + Ke^{(c-d)}$$

where

M : Total effort expended

P : Productive effort that involves analysis, design, coding, evaluation.

K : An empirically determined constant.

c : Complexity measure due to lack of good design and docu

d : Degree to which maintenance team is familiar with the sof

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Example – 9.1

The development effort for a software project is 500 person mo
empirically determined constant (K) is 0.3. The complexity of th
quite high and is equal to 8. Calculate the total effort expended

(i) maintenance team has good level of understanding of the pr

(ii) maintenance team has poor understanding of the project (d=

Solution

Development effort (P) = 500 PM

K = 0.3

C = 8

(i) maintenance team has good level of understanding of the pro

$$M = P + Ke^{(c-d)}$$

$$= 500 + 0.3e^{(8-0.9)}$$

$$= 500 + 363.59 = 863.59 \text{ PM}$$

(ii) maintenance team has poor understanding of the project (d=

$$M = P + Ke^{(c-d)}$$

$$= 500 + 0.3e^{(8-0.1)}$$

$$= 500 + 809.18 = 1309.18 \text{ PM}$$

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Boehm Model

Boehm used a quantity called Annual Change Traffic (ACT).

"The fraction of a software product's source instructions which change during a year either through addition, deletion or modi

$$ACT = \frac{KLOC_{added} + KLOC_{deleted}}{KLOC_{total}}$$

AME = ACT x SDE

Where, **SDE :** Software development effort in person months

ACT : Annual change Traffic

EAF : Effort Adjustment Factor

AME = ACT * SDE * EAF

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Example – 9.2

Annual Change Traffic (ACT) for a software system is 15% per
development effort is 600 PMs. Compute estimate for Annual M
Effort (AME). If life time of the project is 10 years, what is the to
the project ?

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Solution

The development effort = 600 PM

Annual Change Traffic (ACT) = 15%

Total duration for which effort is to be calculated = 10 years

The maintenance effort is a fraction of development effort and is be constant.

$$AME = ACT \times SDE$$

$$= 0.15 \times 600 = 90 \text{ PM}$$

Maintenance effort for 10 years = 10 x 90 = 90 PM

Total effort = 600 + 900 = 1500

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Example – 9.3

A software project has development effort of 500 PM. It is assu
code will be modified per year. Some of the cost multipliers are

1. Required software Reliability (RELY) : high
2. Date base size (DATA) : high
3. Analyst capability (ACAP) : high
4. Application experience (AEXP) : Very high
5. Programming language experience (LEXP) : high

Other multipliers are nominal. Calculate the Annual Mainte
(AME).

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Solution

Annual change traffic (ACT) = 10%

Software development effort (SDE) = 500 Pm

Using Table 5 of COCOMO model, effort adjustment factor ca
calculated given below :

RELY = 1.15

ACAP = 0.86

AEXP = 0.82

LEXP = 0.95

DATA = 1.08

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Other values are nominal values. Hence,

EAF = 1.15 x 0.86 x 0.82 x 0.95 x 1.08 = 0.832

AME = ACT * SDE * EAF

= 0.1 * 500 * 0.832 = 41.6 PM

AME = 41.6 PM

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Regression Testing

Regression testing is the process of retesting the modified software and ensuring that no new errors have been int previously test code.

"Regression testing tests both the modified code and othe program that may be affected by the program change. It purposes :

- increase confidence in the correctness of the modified
- locate errors in the modified program
- preserve the quality and reliability of software
- ensure the software's continued operation

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Development Testing Versus Regression Testing

Sr. No.	Development testing	Regression tes
1.	We create test suites and test plans	We can make use of existing test plans
2.	We test all software components	We retest affected compon been modified by modificatio
3.	Budget gives time for testing	Budget often does not regression testing.
4.	We perform testing just once on a software product	We perform regression testi over the life of the software
5.	Performed under the pressure of release date of the software	Performed in crisis situations time constraints.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

- Regression Test Selection

Regression testing is very expensive activity and consum
amount of effort / cost. Many techniques are available to redu
cost.

1. Reuse the whole test suite

2. Reuse the existing test suite, but to apply a reg
selection technique to select an appropriate subset of
to be run.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Fragment A		Fragment B (modified form of A)	
S_1	y = (x - 1) * (x + 1)	S_1'	y = (x -1) * (x + 1)
S_2	if (y = 0)	S_2'	if (y = 0)
S_3	return (error)	S_3'	return (error)
S_4	else	S_4'	else
S_5	return $\left(\frac{1}{y}\right)$	S_5'	return $\left(\frac{1}{y-3}\right)$

Fig. 8: code fragment A and B

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Test cases		
Test number	Input	Execution Histo
t_1	x = 1	S_1, S_2, S_3
t_2	x = -1	S_1, S_2, S_3
t_3	x = 2	S_1, S_2, S_5
t_4	x = 0	S_1, S_2, S_5

Fig. 9: Test cases for code fragment A of Fig. 8

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

If we execute all test cases, we will detect this divide by zero have to minimize the test suite. From the fig. 9, it is clear that and t_4 have the same execution history i.e. S_1, S_2, S_5. If few tes the same execution history; minimization methods select only o Hence, either t_3 or t_4 will be selected. If we select t_4 then fine c not found.

Minimization methods can omit some test cases that might e the modified software and so, they are not safe.

A safe regression test selection technique is one that, u assumptions, selects every test case from the original test s expose faults in the modified program.

Software Maintenance

- Selective Retest Techniques

Selective retest techniques may be more economical than the '
technique.

Selective retest techniques are broadly classified in three categ

1. **Coverage techniques :** They are based on test cover
They locate coverable program components that have be
and select test cases that exercise these components.

2. **Minimization techniques:** They work like coverage
except that they select minimal sets of test cases.

3. **Safe techniques:** They do not focus on coverage criteria
select every test case that cause a modified program
different output than its original version.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

Rothermal identified categories in which regression te techniques can be compared and evaluated. These categories

Inclusiveness measures the extent to which a technique c cases that will cause the modified program to produce differen the original program, and thereby expose faults caused by mod

Precision measures the ability of a technique to avoid choosin that will not cause the modified program to produce different the original program.

Efficiency measures the computational cost, and thus, pra technique.

Generality measures the ability of a technique to handle diverse language constructs, arbitrarily complex modifications, testing applications.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Reverse Engineering

Reverse engineering is the process followed in order to unknown and hidden information about a software system.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Scope and Tasks

The areas there reverse engineering is applicable include (but

1. Program comprehension
2. Redocumentation and/ or document generation
3. Recovery of design approach and design details at abstraction
4. Identifying reusable components
5. Identifying components that need restructuring
6. Recovering business rules, and
7. Understanding high level system description

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Reverse Engineering encompasses a wide array of tasks related to and modifying software system. This array of tasks can be broken in classes.

- Mapping between application and program domains

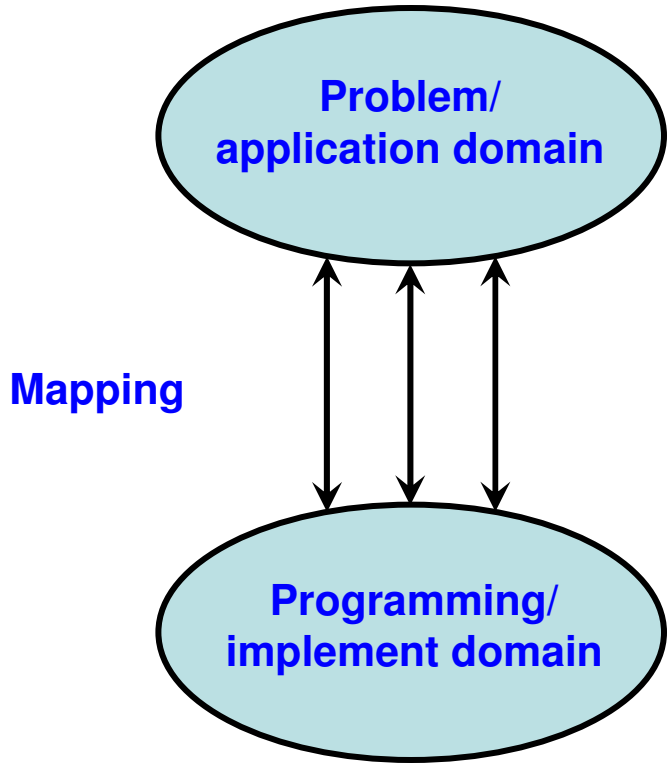

Fig. 10: Mapping between application and domains progra

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Mapping between concrete and abstract levels
- Rediscovering high level structures
- Finding missing links between program sy semantics
- To extract reusable component

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Levels of Reverse Engineering

Reverse Engineers detect low level implementation construc
them with their high level counterparts.

The process eventually results in an incremental formation
architecture of the program.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

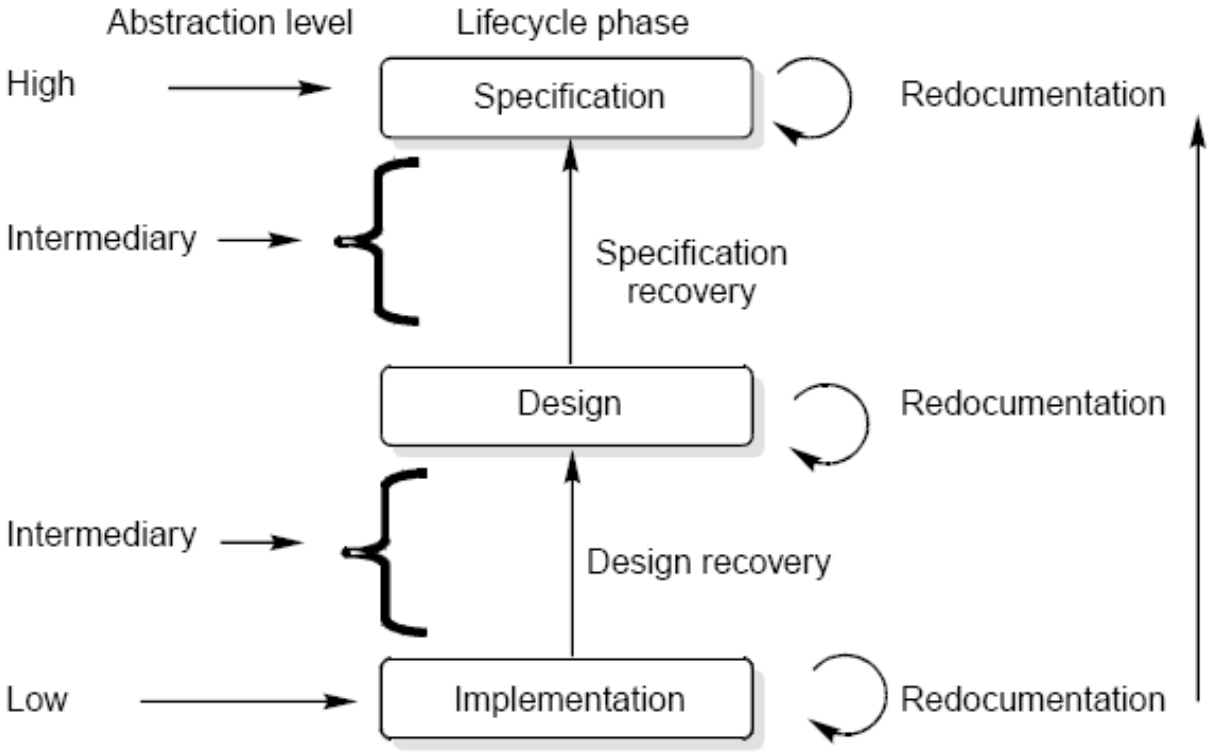

Fig. 11: Levels of abstraction

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

Redocumentation

Redocumentation is the recreation of a semanticall representation within the same relative abstraction level.

Design recovery

Design recovery entails identifying and extracting meaningfu abstractions beyond those obtained directly from examination code. This may be achieved from a combination of code, e documentation, personal experience, and knowledge of the application domains.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software RE-Engineering

Software re-engineering is concerned with taking existing le
and re-implementing them to make them more maintainable.

The critical distinction between re-engineering and n
development is the starting point for the development as show

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

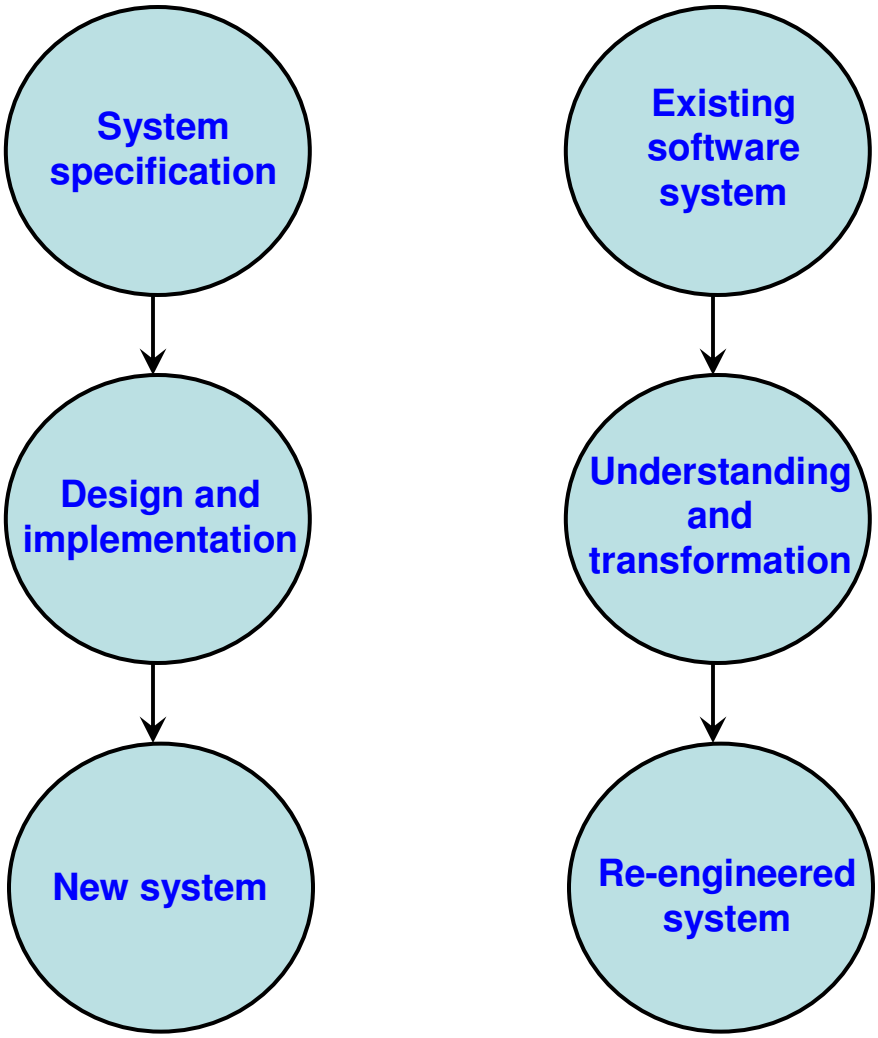

Fig. 12: Comparison of new software development with re-en

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

The following suggestions may be useful for the modification code:

- ✓ Study code well before attempting changes
- ✓ Concentrate on overall control flow and not coding
- ✓ Heavily comment internal code
- ✓ Create Cross References
- ✓ Build Symbol tables
- ✓ Use own variables, constants and declarations to localiz
- ✓ Keep detailed maintenance document
- ✓ Use modern design techniques

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Source Code Translation

1. **Hardware platform update:** The organization m change its standard hardware platform. Compilers fo language may not be available on the new platform.

2. **Staff Skill Shortages:** There may be lack maintenance staff for the original language. This is problem where programs were written in some n language that has now gone out of general use.

3. **Organizational policy changes:** An organization ma standardize on a particular language to minimize software costs. Maintaining many versions of old co be very expensive.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Program Restructuring

1. **Control flow driven restructuring:** This involves th
   of a clear control structure within the source code
   either inter modular or intra modular in nature.

2. **Efficiency driven restructuring:** This involves res
   function or algorithm to make it more efficient. A sim
   is the replacement of an IF-THEN-ELSE-IF-ELSE co
   a CASE construct.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

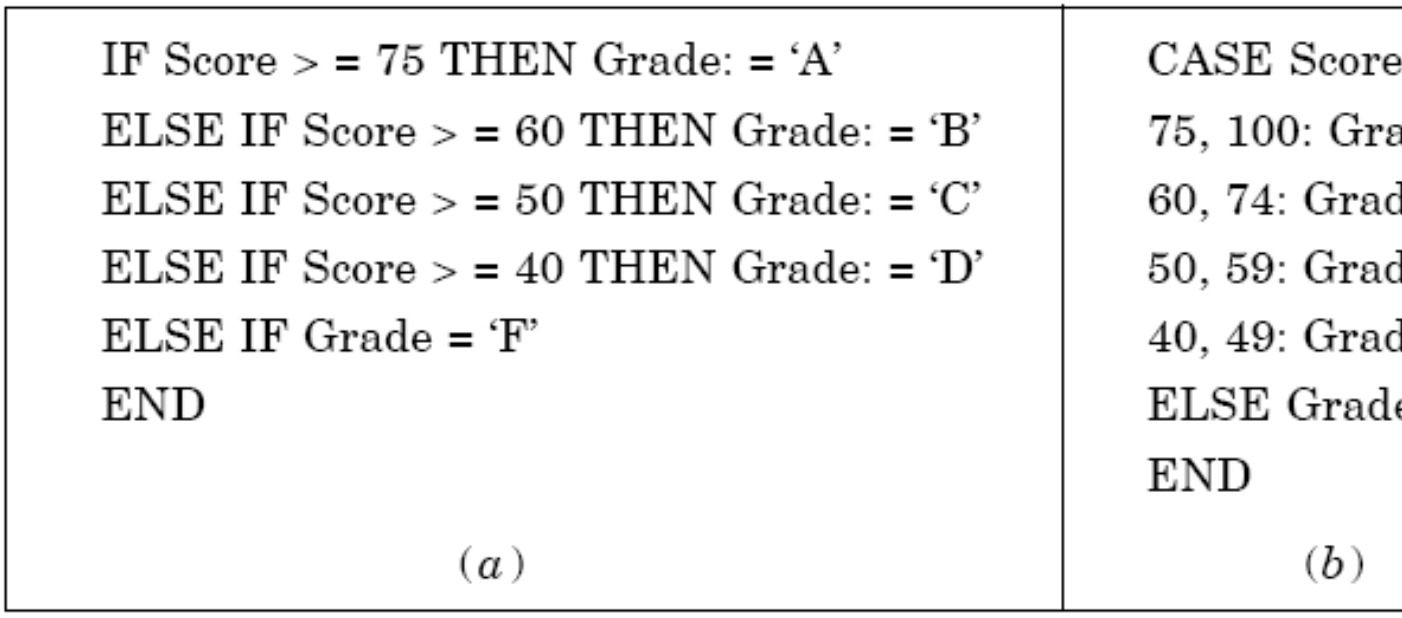

Fig. 13: Restructuring a program

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

3. **Adaption driven restructuring:** This involves chan
coding style in order to adapt the program to a new prog
language or new operating environment, for instance
an imperative program in PASCAL into a functional pr
LISP.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Configuration Management

The process of software development and maintenance is called configuration management. The configuration ma different in development and maintenance phases of life different environments.

- Configuration Management Activities

The activities are divided into four broad categories.

1. The identification of the components and changes
2. The control of the way by which the changes are made
3. Auditing the changes
4. Status accounting recording and documenting all the a that have take place

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

The following documents are required for these activities

- ✓ Project plan
- ✓ Software requirements specification document
- ✓ Software design description document
- ✓ Source code listing
- ✓ Test plans / procedures / test cases
- ✓ User manuals

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- Software Versions

Two types of versions namely revisions (replace) and variatio

Version Control :

A version control tool is the first stage towards being ab
multiple versions. Once it is in place, a detailed record of ev
the software must be kept. This comprises the

- ✓ Name of each source code component, including the va revisions
- ✓ The versions of the various compilers and linkers used
- ✓ The name of the software staff who constructed the com
- ✓ The date and the time at which it was constructed

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

- Change Control Process

Change control process comes into effect when the s
associated documentation are delivered to configuration
change request form (as shown in fig. 14), which shoul
recommendations regarding the change.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

CHANGE REQUEST FORM

Project ID:

Change Requester with date:

Requested change with date:

Change analyzer:

Components affected:

Associated components:

Estimated change costs:

Change priority:

Change assessment:

Change implementation:

Date submitted to CCA:

Date of CCA decision:

CCA decision:

Change implementer:

Date submitted to QA:

Date of implementation:

Date submitted to CM:

QA decision:

Fig. 14: Change request form

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Documentation

Software documentation is the written record of the fa
software system recorded with the intent to convey purp
and clarity.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

- User Documentation

S.No.	Document	Function
1.	System Overview	Provides general description of system
2.	Installation Guide	Describes how to set up the system, c local hardware needs and configure hardware and other software systems.
3.	Beginner's Guide	Provides simple explanations of how the system.
4.	Reference Guide	Provides in depth description of each and how it can be used.
5.	Enhancement	Booklet Contains a summary of new fe
6.	Quick reference card	Serves as a factual lookup.
7.	System administration	Provides information on services s working, security and upgrading.

Table 5: User Documentation

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- System Documentation

It refers to those documentation containing all facets of syst
analysis, specification, design, implementation, testing, s
diagnosis and recovery.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Software Maintenance

- System Documentation

S.No.	Document	Function
1.	System Rationale	Describes the objectives of the entire s
2.	SRS	Provides information on exact rec system as agreed between user and d
3.	Specification/ Design	Provides description of: (i) How system requirements are impl (ii) How the system is decomposed interacting program units. (iii) The function of each program unit.
4.	Implementation	Provides description of: (i) How the detailed system design is some formal programming languag (ii) Program actions in the form of comments.

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

S.No.	Document	Function
5.	System Test Plan	Provides description of how progra tested individually and how the whc tested after integration.
6.	Acceptance Test Plan	Describes the tests that the syster before users accept it.
7.	Data Dictionaries	Contains description of all terms that software system in question.

Table 6: System Documentation

Software Engineering (3rd ed.), By K.K Aggarwal & Yogesh Singh, Copyright © New Age International Publishers, 2007

Multiple Choice Questions

Note: Choose most appropriate answer of the following q

9.1 Process of generating analysis and design documents is called

(a) Inverse Engineering
(b) Software Engineerir
(c) Reverse Engineering
(d) Re-engineering

9.2 Regression testing is primarily related to

(a) Functional testing
(b) Data flow testing
(c) Development testing
(d) Maintenance testing

9.3 Which one is not a category of maintenance ?

(a) Corrective maintenance
(b) Effective maintenan
(c) Adaptive maintenance
(d) Perfective maintena

9.4 The maintenance initiated by defects in the software is called

(a) Corrective maintenance
(b) Adaptive maintenan
(c) Perfective maintenance
(d) Preventive maintena

9.5 Patch is known as

(a) Emergency fixes
(b) Routine fixes
(c) Critical fixes
(d) None of the above

Multiple Choice Questions

9.6 Adaptive maintenance is related to
(a) Modification in software due to failure
(b) Modification in software due to demand of new functionalities
(c) Modification in software due to increase in complexity
(d) Modification in software to match changes in the ever-changing e

9.7 Perfective maintenance refers to enhancements
(a) Making the product better
(b) Making the product faster and smaller
(c) Making the product with new functionalities
(d) All of the above

9.8 As per distribution of maintenance effort, which type of maintena consumed maximum share?
(a) Adaptive
(b) Corrective
(c) Perfective
(d) Preventive

9.9 As per distribution of maintenance effort, which type of maintena consumed minimum share?
(a) Adaptive
(b) Corrective
(c) Perfective
(d) Preventive

Multiple Choice Questions

9.10 Which one is not a maintenance model ?

(a) CMM
(b) Iterative Enhanceme
(c) Quick-fix model
(d) Reuse-Oriented mod

9.11 In which model, fixes are done without detailed analysis of the l

(a) Reuse oriented model
(b) Quick-fix model
(c) Taute maintenance model
(d) None of the above

9.12 Iterative enhancement model is a

(a) three stage model
(b) two stage model
(c) four stage model
(d) seven stage model

9.13 Taute maintenance model has

(a) Two phases
(b) six phases
(c) eight phases
(d) ten phases

9.14 In Boehm model, ACT stands for

(a) Actual change time
(b) Actual change traffi
(c) Annual change traffic
(d) Annual change time

Multiple Choice Questions

9.15 Regression testing is known as

(a) the process of retesting the modified parts of the software

(b) the process of testing the design documents

(c) the process of reviewing the SRS

(d) None of the above

9.16 The purpose of regression testing is to

(a) increase confidence in the correctness of the modified progra

(b) locate errors in the modified program

(c) preserve the quantity and reliability of software

(d) All of the above

9.17 Regression testing is related to

(a) maintenance of software

(b) development of soft

(c) both (a) and (b)

(d) none of the above.

9.18 Which one is not a selective retest technique

(a) coverage technique

(b) minimization techni

(c) safe technique

(d) maximization techni

Multiple Choice Questions

9.19 Purpose of reverse engineering is to

(a) recover information from the existing code or any other intern document

(b) redocumentation and/or document generation

(c) understand the source code and associated documents

(d) All of the above

9.20 Legacy systems are

(a) old systems

(b) new systems

(c) undeveloped systems

(d) None of the above

9.21 User documentation consists of

(a) System overview

(b) Installation guide

(c) Reference guide

(d) All of the above

9.22 Which one is not a user documentations ?

(a) Beginner's Guide

(b) Installation guide

(c) SRS

(d) System administrati

Multiple Choice Questions

9.23 System documentation may not have

(a) SRS
(b) Design document
(c) Acceptance Test Plan
(d) System administratio

9.24 The process by which existing processes and methods are replaced b techniques is:

(a) Reverse engineering
(b) Business process re-e
(c) Software configuration management
(d) Technical feasibility

9.25 The process of transforming a model into source code is

(a) Reverse Engineering
(b) Forward engineering
(c) Re-engineering
(d) Restructuring

Exercises

9.1 What is software maintenance? Describe various maintenance. Which category consumes maximum effort a

9.2 What are the implication of maintenance for a one pe production organisation?

9.3 Some people feel that "maintenance is manageable". opinion about this issue?

9.4 Discuss various problems during maintenance. Describe s to these problems.

9.5 Why do you think that the mistake is frequently made software maintenance inferior to software development?

9.6 Explain the importance of maintenance. Which categ maximum effort and why?

9.7 Explain the steps of software maintenance with help of a di

9.8 What is self descriptiveness of a program? Explain the parameter on maintenance activities.

Exercises

9.9 What is ripple effect? Discuss the various aspects of rip
how does it affect the stability of a program?

9.10 What is maintainability? What is its role during maintenan

9.11 Describe Quick-fix model. What are the advantage and di
this model?

9.12 How iterative enhancement model is helpful during
Explain the various stage cycles of this model.

9.13 Explain the Boehm's maintenance model with the help of

9.14 State the various steps of reuse oriented model. Is it a
model in object oriented design?

9.15 Describe the Taute maintenance model. What are various
model?

9.16 Write a short note on Boledy and Lehman model for the
maintenance effort.

Software Engineering, By K.K Aggarwal & Yogesh Singh, New Age International Pu

Exercises

9.17 Describe various maintenance cost estimation model.s

9.18 The development effort for a project is 600 PMs. Th determined constant (K) of Belady and Lehman mode complexity of code is quite high and is equal to 7. Calc effort expended (M) if maintenance team has reason understanding of the project (d=0.7).

9.19 Annual change traffic (ACT) in a software system is 25% initial development cost was Rs. 20 lacs. Total life time f 10 years. What is the total cost of the software system?

9.20 What is regression testing? Differentiate between re development testing?

9.21 What is the importance of regression test selection? Discus examples.

9.22 What are selective retest techniques? How are they c "retest-all" techniques?

Exercises

9.23 Explain the various categories of retest techniques. Wh
useful and why?

9.24 What are the categories to evaluate regression test selectio
Why do we use such categorisation?

9.25 What is reverse engineering? Discuss levels of reverse eng

9.26 What are the appropriate reverse engineering tools? Dis
tools in detail.

9.27 Discuss reverse engineering and re-engineering.

9.28 What is re-engineering? Differentiate between re-enginee
development.

9.29 Discuss the suggestions that may be useful for the modi
legacy code.

9.30 Explain various types of restructuring techniques
restructuring help in maintaining a program?

Exercises

9.31 Explain why single entry, single exit modules make testing
maintenance.

9.32 What are configuration management activities? Draw th
change request form.

9.33 Explain why the success of a system depends heavily on t
the documentation generated during system development.

9.34 What is an appropriate set of tools and documents require
large software product/

9.35 Explain why a high degree of coupling among modu
maintenance very difficult.

9.36 Is it feasible to specify maintainability in the SRS? If ye
we specify it?

9.37 What tools and techniques are available for software
Discuss any two of them.

Exercises

9.38 Why is maintenance programming becoming more cha
new development? What are desirable characteristics of a
programmer?

9.39 Why little attention is paid to maintainability during desig

9.40 List out system documentation and also explain their purp