

STEPWISE REFINEMENT

Block of Code Explanation



Q- What is stepwise refinement?

- The breaking down of an algorithm
- To a level of sufficient (detail)
- From which it can be programmed
 - same thing or a sub-task
- Module or a sub-routine ^{or} can be either a function or procedure.

• Module is a block of code that is refined or worked upon separately

Q - What are the uses of sub-routines in a program?

- To perform a frequently used operation within a program
- that can be called when needed
- that can be reused by another program.

Q - Why it is a good idea to construct a program using modules?

- To make more manageable / understandable solution
- Module / sub-routines may be independantly tested
- Program easier to maintain

Parameter
 $f(x) = x^2 + 4x$

$f(1) = Ans$

$f(2) = Ans$

Function Grade(A)
A = Grade
B = ---
C = ---
D = ---
End.
CALL Grade("Ans")

Q- What are the advantages of sub-routines?

- Can be called into multiple places.
- Sub-routine may be tested independantly
- If the task changes, the changes need to be made only once.

$$P(x) = x^2 + x \quad (P(n) = n^2 + n + 2)$$

✓ $P(1) = 2 \checkmark + 2$
✓ $P(2) = 6 \checkmark + 2$ $y = x^2 + x + 2$
✓ $P(3) = 12 \checkmark + 2$ $y = x^2 + x + 2$
✓ $P(4) = 20 \checkmark + 2$
 $\vdots \vdots \vdots \checkmark$

• Whenever
we want to
repeat sth,
we use functions.

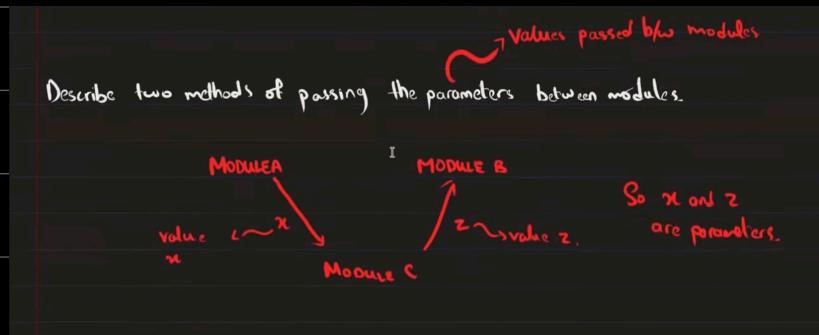
- Reduces unnecessary duplications (lines ki bachat 😊)
- Allows team to work on different parts of the solution simultaneously.

• Procedure: A sequence of steps that is given an identifier and can be called to perform a sub-task (it does not return a value).

• Function: A sequence of steps that is given an identifier and performs a sub-task (IT RETURNS A VALUE)

Q- Describe two methods of passing the parameters between the modules

Parameters: Values passed b/w modules



① BY VALUE : A copy of variable itself is passed and original value is not changed.

② BY REFERENCE : The address of the variable is passed and the original value is changed

Q- State how structured programming supports the implementation of sub-tasks ?

- Global / Local Variables
- Functions
- Procedures
- Parameters

Structured Programming: High-Level Langs.

Modular approach: - Creation of

sub-routines

- sub-tasks

PROGRAM DEVELOPMENT CYCLE

- Series of steps that are to be followed in sequence for creation of a program.
- To develop a successful program which can be used and understood by other developers.

6 stages

Problem Solving

Design

Coding

Translating

Execution

Testing

5 stages

These are in syllabus

Analysis
Design
Coding
Testing
Maintenance

EDITOR : One who produces source code in high level (Developer) language.

TRANSLATOR: Converts source code into object code

DEBUGGER: It is used to test the program / detect errors

DEVELOPMENT MODEL* — New 9618

Waterfall Model

Iterative Model

RAPID Application Development

WATERFALL MODEL

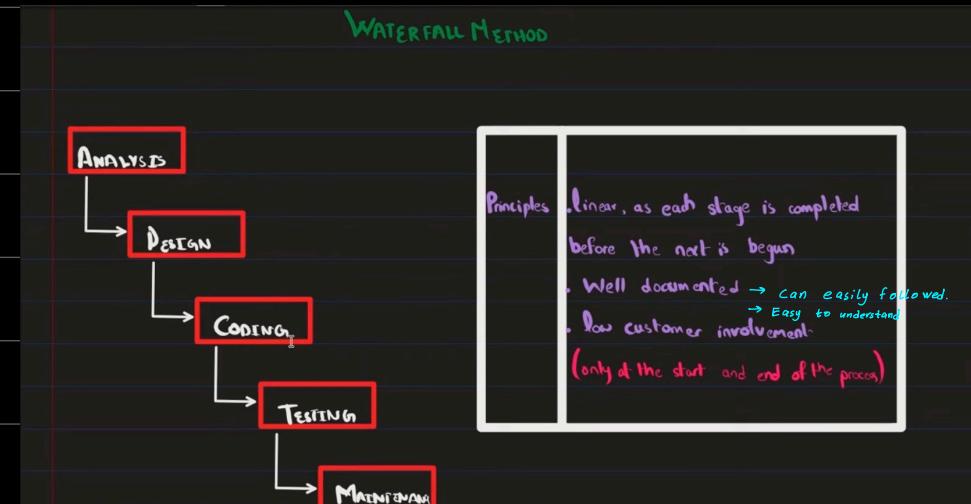
When

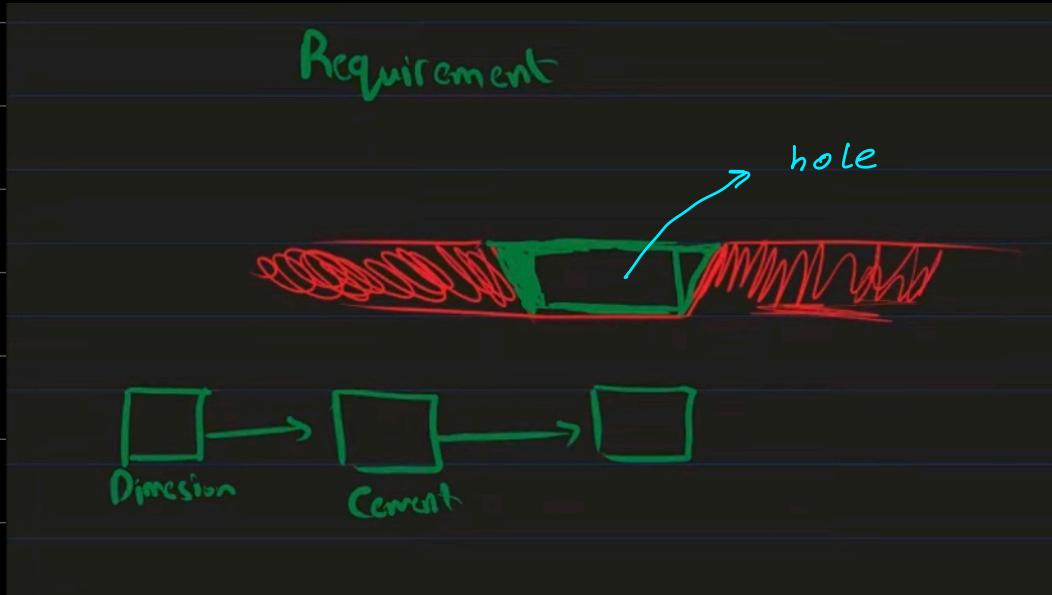
• Requirements are known, waterfall model is used

• No fixed diagram.

↳ After coding, translation can be included.

• Problem-solving





BENEFITS

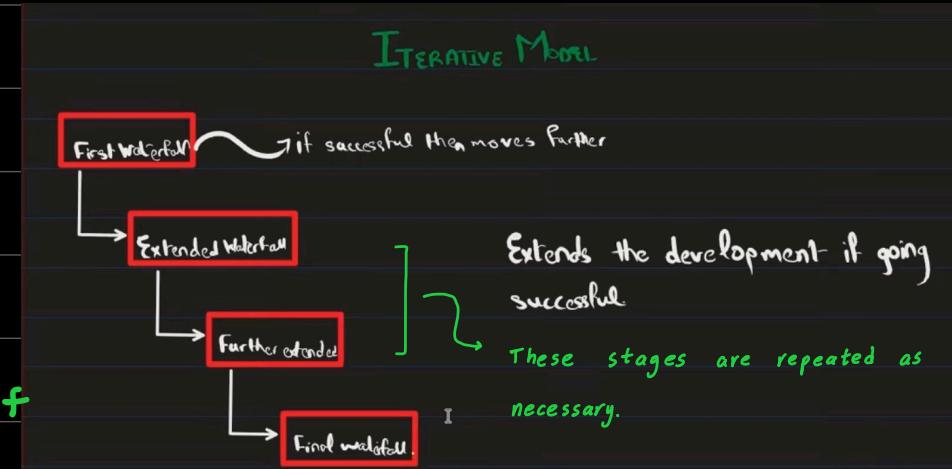
- Easy to manage, understand and use
- stages do not overlap and are completed one at a time
- Works well for smaller programs

DRAWBACKS

- Difficult to change the requirement at a later stage
- Working program is produced late in the life cycle.
- Not suitable for complex projects.
- Subject to change programs are not suitable

ITERATIVE MODEL

- Iterative means repetition.
- After every waterfall, customer is asked for feedback.
- When requirements unknown.
- The cycle develops a simple subset of the requirements and then expands the model



Benefits

- Easier to test and debug
- More Flexible as easier to alter requirements
- Customer involved at each iteration therefore no surprises when final system is delivered.

Principles	<ul style="list-style-type: none">→ Feature by featureIncremental development as the program development lifecycle is repeated.High customer involvement as part of the system can be shown to the customer after every iteration.
------------	--

Drawbacks

- Whole system needs to be defined at start
- Needs a good planning overall , at every stage.
- Not suitable for simple projects.

Rapid Application Development



- Program is divided into modules and each module has one team and its own waterfall.

Benefits

(sub-task)

- Reduced overall development time
- Rapid feedback informs the development
- very flexible as requirement evolve from feedback
- Modification is easier because each part must work independantly

Principles	<ul style="list-style-type: none">➢ Each module can be handled by its team• Minimal Planning by its team• Uses previously written code where possible• high customer involvement
------------	---

Drawbacks

- System under development needs to be modular → Task
- Need strong team of skilled developers
- Not suitable for short - simple projects
- Requires user involvement through out the life cycle.

Transferable Skills

- knowledge of one programming language that can be applied to an unknown language.

Explanation.

Job

AHMAR

✓ Python =>

✓

✓

Code => Analysis

Error => Findout.

✓ JAVA

You should be able to recognize / understand in another language :-

- . Declaration
- . Assignment
- . Sequence
- . Selection
- . Repetition / Iteration
- . Subroutines
- . Parameters passed between module
- . Input output

An algorithm is implemented in a high-level language. Changes are required and the program is given to Albert, who is an experienced programmer but he is not familiar with the language that has been used.

Explain why Albert would be able to understand the program ?

- . He would use his transferable skills to understand the new program.
- . He could recognise basic control structures in the language such as loops, selectional statements, declaration.
- . He could read the comments / meaningful variable names.

Software Testing

Black-Box Testing

$$\begin{array}{r} 5+3=8 \\ =8 \end{array}$$

- Match expected answer to actual answer
- A person does not need to know the structure of the code.
- A person chooses normal, boundary, erroneous data.

White-Box Testing

- Test every possible "logical path" through the code.
- A trace table is often used during the program testing
- A person needs to know the code.

Alpha Testing

software made in software house is tested by

- In house testing other developers rather than the actual developer.
- Testing is done by developers in-house other than the developer who actually built the software.

Beta Testing

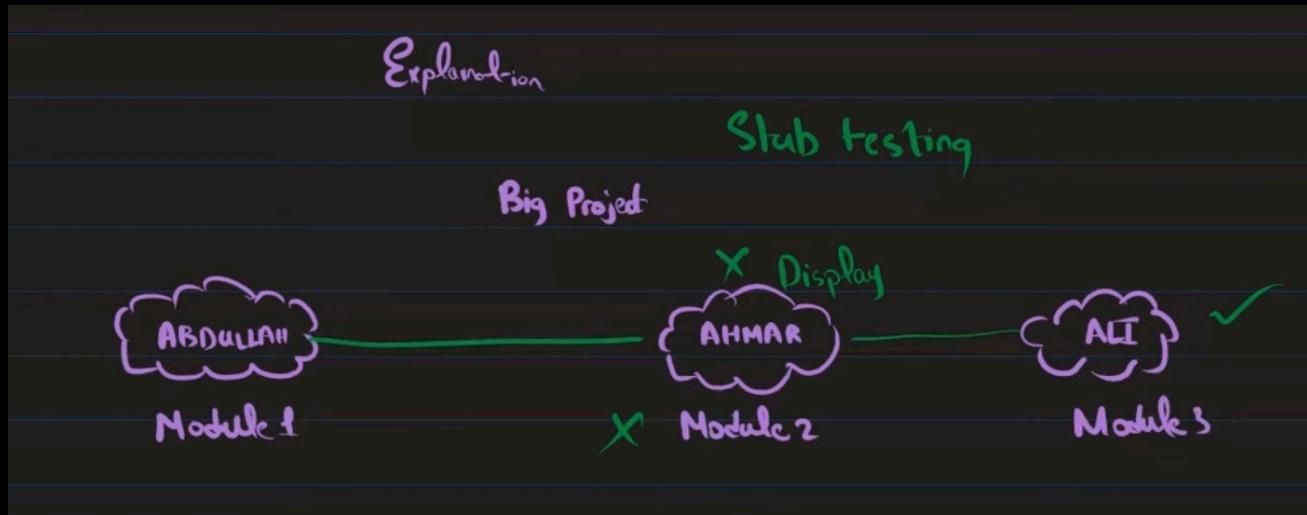
- Out of house testing
- Performed by group of people whom are assigned to find bugs (errors) in the code.

Stub Testing

- Contains an output statement
- Simple / dummy module written to simulate the module
- Test carried out on all the modules before they have been written

Integration Testing

- It is the testing of any separately written modules to ensure that they work together.



- Integration testing is combining and testing the modules

- If a module is missing then, stub testing is used

Acceptance Testing

- Is used for the completed program to prove the customer/client that the program works as required
- End-User testing

Analysis: A process of investigation, leading to the specification of what a program is required to do.

→ describes the results that a program is expected to produce

Design: Uses program specification from analysis stage to show how the program should be developed.

Coding: Writing of program in high-level languages.

Testing: Testing of the program to ensure that it works under all conditions

Maintenance: The process of making sure that the program continues to work during use.

Test strategy: An overview of the testing required to meet the program requirements

Test Plan: A detailed list showing all the stages of testing.