Analysis

Collection of Information

Interviews

Questionnaires

Observation

Document Analysis

Who-when-where-why-what approach

Analysis of Information

How to process input to get required output

Data-flow and Object Modelling

Information Engineering – A data centred technique

Production of Specification

Full description of problem

Full description of all input, processes and outputs

Assumptions about boundaries must be described

Can represent an agreement

Design

Human Computer Interface

Consistency

Easy Navigation

On-line Help

Suited to User

Data Structure

A program must perform **operations on data supplied to it.**

Sorted array = difficult to maintain
Unsorted array = slow searching

Hardware capacity must be considered

Object oriented design: treats **data and objects together**

Main Program

Implementation

Coding

Translate **pseudo-code** (from Design stage) **into code**

Type and edit code in **Editor**

Types of Language

Imperative/Procedural or **Declarative/Non-procedural**

If **compiled**: **portable** between computers of same platform and operating system.

If **interpreted**: **not as portable**

Problems?

If the program runs too slowly:

Write faster algorithm (return to Design)

or

Revise Hardware Specification (return to Analysis)

Testing

Preparation of Test Data

Test data must be prepared **before the coding of the program** begins. If a programmer has spent months working on a program, he may be lenient when it comes to finding errors

Alpha/Integration Testing

Testing on Program:
Normal Operation
Boundary Testing
Exceptions Testing

Testing on HCI:
Normal User Activity
Unusual User Activity

Test Log:
Input; Reason; Expected Output; Actual Output

If there are bugs in program, sent back to programming team

Beta/Acceptance Testing

Product released to small group of people

Feedback considered by programming team

Documentation

User Guide

System Requirements

How to **install and use** software

Guide to HCI

Technical Guide

System Requirements

Instructions about **configuring and updating** the program

Other Documentation

Requirements Specification

Program Design Documents

Listing of Code

Test History

Internal Documentation

Evaluation

Does the program meet the requirements?

Does the software meet the design?

Was the project within budget?

Was the project completed on time?

Can the software development process be improved in the future?

Once the evaluation has been completed,
work begins on the next version of the software

Maintenance

Software creator may set up help desks and training facilities

Corrective Maintenance:
Fix errors which are **missed at testing**
(eg in the form of a patch)

Adaptive Maintenance:
Adapt programs to **different environments**
(eg. a different Operating System)

Perfective Maintenance:
Enhance program due to **changes in the requirements**